

Полтавський національний педагогічний університет
імені В.Г. Короленка
Фізико-математичний факультет
Кафедра математичного аналізу та інформатики

Ю.Г. Подошвелев

Система \LaTeX

Полтава – 2016



Сайт ПНПУ

Головна

Зміст



Стор. 1 із 189

Назад

Перегляд

Закрити

Вихід

УДК 004.738.5:655.41

ББК 76.2

П 44

Система \LaTeX

Ю.Г. Подошвелев

mailto: optimist1618@mail.ru

26 грудня 2016 р.

Анотація

Файли системи \LaTeX зазвичай мають текстове розширення. Вони складаються з простого тексту з вкрапленням деяких команд \LaTeX . Не лякайтеся слова команда. Для створення першого \LaTeX документу, не потрібно вивчати нову мову програмування або сотні команд. Розпочинають із простого шаблону обираючи за зразок уже готові розробки. Потім користувач додає власний текст всередині розділів. Основний текст абзаців – це просто звичайний текст. В кінці кінців необхідно буде познайомитися з деякими командами для більш продуктивної праці. Саме для цього і стане в пригоді дана книга. Зміст значної частини команд в ній буде розкрито, а ефекти їх застосування будуть візуалізовані.

Рецензенти: кандидат фізико-математичних наук, доцент кафедри загальної фізики і математики ПНПУ ім. В.Г. Короленка **Марченко В.О.**
доктор фізико-математичних наук, професор кафедри вищої математики ПНТУ імені Ю. Кондратюка **Сєров М.І.**

Ухвалено Вченою радою Полтавського національного педагогічного університету імені В.Г. Короленка. Протокол № 8 від 28 грудня 2016 року



Сайт ПНПУ

Головна

Зміст



Стор. 1 із 189

Назад

Перегляд

Закрити

Вихід

Зміст

Вступ	7
1 Вихідний файл	30
2 Символи, що вводяться безпосередньо	31
3 Спецсимволи	32
4 Команди та їхнє задання в тексті	34
5 Структура вихідного тексту	36
6 Розбиття вихідного файлу на частини	39
7 Групи	41
8 Параметри, команди з аргументами, оточення. Зірочка після ім'я команди	42
8.1 Оточення	43
8.2 Одиниці довжини	44
9 Автоматична генерація посилань	48
10 Основні принципи набору формул у системі \LaTeX	50
11 Степені й індекси	52



[Сайт ПНПУ](#)

[Головна](#)

[Зміст](#)



Стор. 2 із 189

[Назад](#)

[Перегляд](#)

[Закрити](#)

[Вихід](#)

12 Дужки	54
13 Штрихи й три крапки	56
14 Таблиці спецзнаків	58
14.1 Операції, відношення й просто значки	58
14.2 Імена функцій, що верстаються прямим шрифтом	60
14.3 Різне	63
15 Нумерація формул	64
16 Переноси у формулах	66
17 Включення тексту у формули	67
18 Надрядкові знаки, звичайні дроби	68
19 Набір матриць	75
20 Проміжки в математичній моді	76
21 Додаткові проміжки навколо формул	79
22 Обробка помилок	80
23 Спеціальні та діактричні знаки	82
24 Проміжки між словами та реченнями	85



[Сайт ПНПУ](#)

[Головна](#)

[Зміст](#)



[Стор. 3 із 189](#)

[Назад](#)

[Перегляд](#)

[Закрити](#)

[Вихід](#)

25 Шрифти та їх зміна в тексті	86
25.1 Класифікація шрифтів	90
25.2 Найпростіші команди вибору шрифту й окремого символу	93
25.3 Збільшення шрифтів	96
26 Посилання	98
27 Абзаци	99
28 Переноси	101
29 Команда \sloppy і параметр \emergencystretch	103
30 Ручне керування розривами рядків	104
31 Поняття про режими	106
32 Вилучення абзацного відступу	107
33 Керування розривами сторінок	108
34 Вертикальні проміжки	109
35 Інтерліньяж	111
36 Набір у дві колонки	112
37 Стили та класи документів, їх стильові опції, пакети	113



[Сайт ПНПУ](#)

[Головна](#)

[Зміст](#)

[◀◀](#) [▶▶](#)

[◀](#) [▶](#)

Стор. 4 із 189

[Назад](#)

[Перегляд](#)

[Закрити](#)

[Вихід](#)

38	Стиль оформлення сторінки	136
39	Поля, розмір сторінки	137
40	Розділи документа	139
41	Зміна стандартних заголовків	141
42	Анотація, додаток	142
43	Титульний аркуш, зміст, список літератури	143
44	Використання баз даних при складанні бібліографії	147
44.1	Команди роботи з бібліографією, що зберігається в базі даних	147
44.2	Процедура одержання списку літератури з бібліографічних баз даних	149
44.3	Формат файлу бібліографічних даних	150
44.3.1	Типи записів	151
44.3.2	Преамбула	155
44.3.3	Символьні послідовності	156
44.3.4	Уведення інформації про авторів і редакторів	156
44.4	Кілька зауважень	157
45	Пакет pdfscreen та його завантаження	165
45.1	Опції пакету pdfscreen	165
45.2	Інші параметри	167
45.3	Типова преамбула	168
45.4	Пакети, що потрібні для запуску pdfscreen	169



[Сайт ПНПУ](#)

[Головна](#)

[Зміст](#)



Стор. 5 із 189

[Назад](#)

[Перегляд](#)

[Закрити](#)

[Вихід](#)

46 Панель навігації	171
47 Інші можливості пакету pdfscreen	175
47.1 Фон	175
47.2 Нижні та верхні кнопки	175
47.3 Зміст в панелі	175
47.4 Файл конфігурації	176
47.5 Створення форм	176
48 Слайди	178
48.1 Шрифти	178
48.2 Подальша обробка	178
48.3 Вбудовування інтерактивних Flash- і 3D-об'єктів, аудіо- та відеофайлів . .	179
48.4 Вкладення інтерактивних 3D-об'єктів в PDF-документи	181
49 Перехід між сторінками	183
50 Використання пакету Exerquiz в Pdfscreen	185
Список використаних джерел	186

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 6 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

Вступ

Даний посібник орієнтований на читача, котрому по роду своєї діяльності необхідно готувати видання високої якості, тексти яких містять формули, графіки, кольорові діаграми, ілюстрації, так і на фахівця з комп'ютерної верстки.

На даний час чи не єдиною потужною видавничою системою, реалізованою практично для всіх платформ та операційних систем і здатною створювати на їх основі ідентичні результати з повною сумісністю файлів, є T_EX (вимовляється «тех», назва складена з букв давньогрецького слова «мистецтво» – тау-епсілон-хі). Незважаючи на досить солідний вік і на появу нових, надзвичайно потужних видавничих систем, популярність T_EX'а в науковому середовищі усього світу продовжує зростати. Його прийнято за стандарт більшістю відомих науково-технічних видавництв світу, зокрема: Addison Wesley Longman, Springer-Verlag, John Wiley & Sons, *AMS*, SIAM, Kluwer, Мир, ТВП, Факторіал.

Ця система дозволяє:

- одержувати при друкуванні тексти високої поліграфічної якості робіт із математики, фізики, хімії та інших природничих наук;
- виконувати математичні розрахунки, включати в текст як завгодно складні формули (при друкуванні не мають спотворень та естетично виглядають), конструювати абзаци, знаходити точки переносу;
- не прив'язувати до якогось одного типу комп'ютерів підготовлений у вигляді файлу текст;
- працювати вітчизняному користувачеві з легальним програмним забезпеченням, так як реалізація цієї програми для IBM PC поширюється вільно.

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 7 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

Завдяки цим якостям, а також тому, що це вільно розповсюджуваний продукт, T_EX став стандартом у багатьох академічних відділеннях і дослідницьких лабораторіях. Разом із тим він замінив у професійному видавничому світі процеси набору.

Програмованість T_EX'a у взаємозв'язку з графічною гнучкістю та можливостями маніпулювання шрифтами PostScript'a¹ (однієї з кращих мов опису друкованої сторінки) значно розширює можливості як автора, так і друкаря.

Історичний аспект. Видавнича система T_EX була створена професором Стенфордського (Стенфордського) університету (місто Станфорд, шт. Каліфорнія, США, заснований у 1885 році) Дональдом Кнотом (Donald E. Knuth) і призначалася для створення книг, дисертацій, статей із великою кількістю математичних формул. Автор визначає T_EX як *«інструмент для перетворення манускрипту отриманого в результаті комп'ютерного набору в документ із високою поліграфічною якістю, що дають найсучасніші друкарські пристрої»*.

До Стенфордського університету перейшов у 1968 році. Із 1968 року по 1969 він працював математиком у відділенні телекомунікацій Інституту оборонних досліджень. У 1972–1973 роках як запрошений професор читав лекції з математики в університеті Осло, Норвегія.

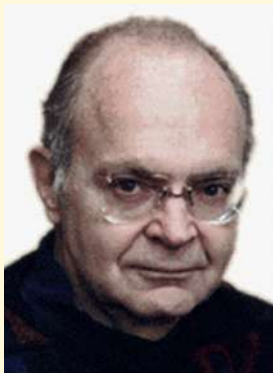
Одержавши титул почесного професора Стенфордського університету в області програмування (1993 рік), Дональд Кнут пішов у відставку. У Стенфорді під його керівництвом захистили докторські дисертації 28 здобувачів.

Ним опубліковано більше 20 книг і більше 150 праць по самих різних питаннях комп'ютерної науки: від дослідження цифрових феноменів у біблійних текстах та історії походження й зміни накреслення букви «S» до абстрактних математичних проблем. Він є:

- засновником теорії компіляції мов програмування, теорії формальних граматик, методу

¹PostScript – мова програмування для пристроїв друку (розробка фірми Adobe), що використовується на багатьох лазерних пристроях. Дозволяє малювати фігури за допомогою ліній будь-якої товщини, зафарбовувати їх будь-яким кольором. Дає змогу створювати бібліотеки кольорів, шрифтів, форм, візерунків. Текст повністю суміщається з графікою, а текстові символи трактуються як графічні об'єкти.

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 8 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)



Дональд Ервін Кнут (Donald Ervin Knuth) народився 10 січня 1938 року в Мілуокі (Milwaukee), штат Вісконсін (Wisconsin). У 1960 році Кнут із відзнакою закінчив інститут і одержав ступінь бакалавра. За значні досягнення в програмуванні він був визнаний гідним ступеня майстра. Докторський ступінь в області математики Кнут отримав у Каліфорнійському технологічному інституті (1963 рік). Перебуваючи у CalTech на посаді професора математики, він був консультантом із проблем розробки програмного забезпечення в Burroughs Corporation.

аналізу алгоритмів;

- автором всесвітньо відомих видавничих систем T_EX і METAFONT, описаних ним у п'ятитомнику Computers&Typesetting. У 2003 р. Вільямс видано російською мовою книги [5, 6];
- автором багатотомної Біблії програмістів «The Art of Computer Programming». Російський переклад: [4, 7, 8, 9] У нову серію книг «Мистецтво програмування» за задумом автора повинно вийти сім томів. Перші три томи цієї серії вийшли в 1998 році у видавництві Addison Wesley Longman. Журнал American Scientist, в останньому номері за 1999 рік, включив The Art of Computer Programming"у список 12-ти кращих наукових монографій що йде (тепер уже минулого) століття. У цей список включені роботи таких видатних представників різних наук, як Альберт Ейнштейн, Лайнос Полинг, Норберт Вінер і ін. Планується видати російською мовою п'ятий том: «Синтаксичний і лексичний аналіз». Автор сподівається, «якщо дозволить здоров'я», випустити ще два томи, том 6 буде присвячений контекстно-незалежним мовам програмування, а том 7 – техніці компіляції.

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 9 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

Серед численних премій і нагород, якими нагороджений Дональд Кнут, найзначніші такі: Премія Тьюринга (1974), Національна медаль за заслуги в науці (1979), Премія Кіото (1996). Він є дійсним членом Американської академії мистецтв і наук, Національної академії наук, Національної академії інженерних наук. Більше двох десятків університетів шести країн світу визнали Кнута почесним доктором.

Він – одна з деяких фігур, оточених міфами ще при житті.

Учений почав розробляти T_EX у 1977 році у зв'язку з конфліктом, що виник між ним та Американським Математичним Товариством, так як останнє спотворювало його статті в процесі їхньої публікації. Десь у 1974 році він навіть припинив надсилати роботи: «Просто мені було занадто боляче дивитися на кінцевий результат».

Ядро системи – процесор T_EX – являє собою машинно-незалежна мова форматування поліграфічних документів. Дональд Кнут, створюючи пакет для видання власних робіт, розробив систему «літературного програмування» і вихідний «літературний» (WEB) код T_EX'у. Він запропонував T_EX для всезагального користування при умові збереження незмінним ретельно відлагодженого ним коду ядра.

Вигляд T_EX'у, який використовують сьогодні, був створений у 1982 році. Отримавши невелике вдосконалення, останні кілька років T_EX став світовим лідером серед систем типографічного набору. Дональд Кнут стверджує, що в ньому практично немає помилок. Нещодавно він офіційно оголосив про те, що в інтересах стабільності T_EX не буде піддаватися подальшому вдосконаленню.

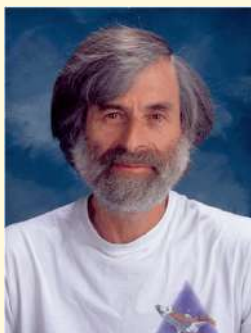
T_EX містить більше 900 команд і примітивів. За своєю суттю є макропроцесор, який надає потужні можливості, але при цьому вимагає певного рівня програмування. Тому для практичних цілей Кнут розробив макропакет plain T_EX, який має мінімальний набір макросів і команд високого рівня (моделі, шаблони) для форматування тексту й не тільки. Фахівці, які працюють у plain T_EX'ові, користуються особливою повагою колег як досить гарні знавці премудростей T_EX'а.

Однією з популярних вільних версій, що відрізняється високою якістю програм і документа-

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 10 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

ції, є пакет em-TeX німецького програміста Еберхарда Маттеса. Кирилична версія CyrTUG-em-TeX, поширювана Асоціацією CyrTUG (див. с. 16), зроблена на основі em-TeX'a, є продуктом public domain.

Пакет P_STeX – це популярна комерційна версія, створена компанією Personal TeX, Inc. До її складу також входять макророзширення, оболонки, прев'юери, драйвери, документація, але всі ці тексти й програми є як комерційні, так і вільні.



Леслі Лампорт (Leslie Lamport, Нью-Йорк; 7 лютого 1941) – американський дослідник в галузі інформатики. Випускник з Вищої школи в Бронксі, отримав ступінь бакалавра математики в MIT 1960 р., та ступінь магістра 1963 р. і Ph.D. 1972 р. з математики в Університеті Брандейса. Його дисертація присвячена сингулярності в аналітичних диференціальних рівняннях з частинними похідними. Лампорт відомий, в першу чергу, за дослідження розподілених систем.

Узявши за основу TeX, на початку вісімдесятих років Леслі Лампорт (Leslie Lamport) написав потужний макропакет L^ATeX (від слів **L**amport **T**EX, вимовляється «латех» або «лейтех»). Створена ним видавнича система L^ATeX (у якості механізму для верстки він використав TeX) дозволяє користувачеві більше зосередитися над структурою документа, ніж на форматуванні. L^ATeX містить близько 600 макрокоманд. Ці команди визначають такі параметри тексту, як типи й розміри шрифтів основного тексту, заголовків, висоту й ширину сторінок, проміжки між рядками, абзацами, відступи й багато чого іншого. Лампорт говорив: «Використовувати L^ATeX дуже легко, якщо ви належите до двох відсотків населення, які здатні мислити логічно й можуть прочитати посібник користувача. Інші дев'яносто вісім відсотків знайдуть його надзвичайно складним або неможливим для використання». Паралельно з L^ATeX'ом ученим розроблений формат Sl_ITeX, орієнтований на підготовку слайдів, які друкуються на аркушах

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 11 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

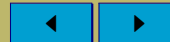
із прозорої плівки й використовуються для проекції на екран при виступах перед великою аудиторією.

У 1989 році команда розробників L^AT_EX3 на чолі з відомими програмістами Франком Миттельбахом (Frank Mittelbach), Крісом Роулі (Chris Rowley) та Райнер Шопф (Schopf, Rainer) почала роботу над удосконаленням і розширенням L^AT_EX'a. Одним із варіантів їхньої роботи є версія L^AT_EX 2_ε (проект L^AT_EX3). У ній були зроблені деякі давно очікувані поліпшення (уведено зручну схему вибору шрифтів – NFSS), та знову об'єднані всі варіанти L^AT_EX'a після розгалужень, що виникли багато років тому з випуском L^AT_EX версії 2.09. На сьогодні використовується версія L^AT_EX 2_ε, що помітно відрізняється від оригінальної версії Леслі Лампорта. L^AT_EX 2_ε дозволяє легко описати структуру документа, яка при використанні відповідних клавіш документів і додаткових пакетів, може давати на виході по-різному оформлений результат.

Рис. 1, взятий із `wots.tex` (автор Kees van der Laan), відображає спільну роботу T_EX і L^AT_EX 2_ε.

Існує багато інших макропакетів ($\mathcal{A}\mathcal{M}\mathcal{S}$ -T_EX, $\mathcal{A}\mathcal{M}\mathcal{S}$ -L^AT_EX, T_EXinfo, ChemT_EX ...), що поширюються в текстовому (ASCII) форматі та можуть функціонувати самостійно або поверх L^AT_EX. Хоча останній і надає гарні можливості для набору математичних формул, однак, якщо виникає потреба неодноразового введення складних рівнянь та інших математичних об'єктів, це призводить до певних незручностей у роботі.

Із огляду на цю обставину, Майкл Співак (M.Spivak) у 1982 році за замовленням Американського Математичного Товариства (A M S) розробив формат, відомий нині як $\mathcal{A}\mathcal{M}\mathcal{S}$ -T_EX. Він має у своєму розпорядженні чудову колекцію шрифтів, у тому числі кириличних зі своєрідними лігатурами. Прикрасою цієї колекції є математичні шрифти, унікальні кількістю й різноманітністю представлених у них символів. Основне достоїнство $\mathcal{A}\mathcal{M}\mathcal{S}$ -T_EX'a полягає в тому, що він спрощує набір математичних формул завдяки створенню нових команд, даючи на виході результат, який задовольняє високим стандартам, прийнятим у математичних видавництвах. Функції $\mathcal{A}\mathcal{M}\mathcal{S}$ -T_EX'a дозволяють не обтяжувати користувача тонкощами поліграфічного оформлення позаштатних ситуацій, таких, наприклад, як набір матриць усередині матриць або

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 12 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

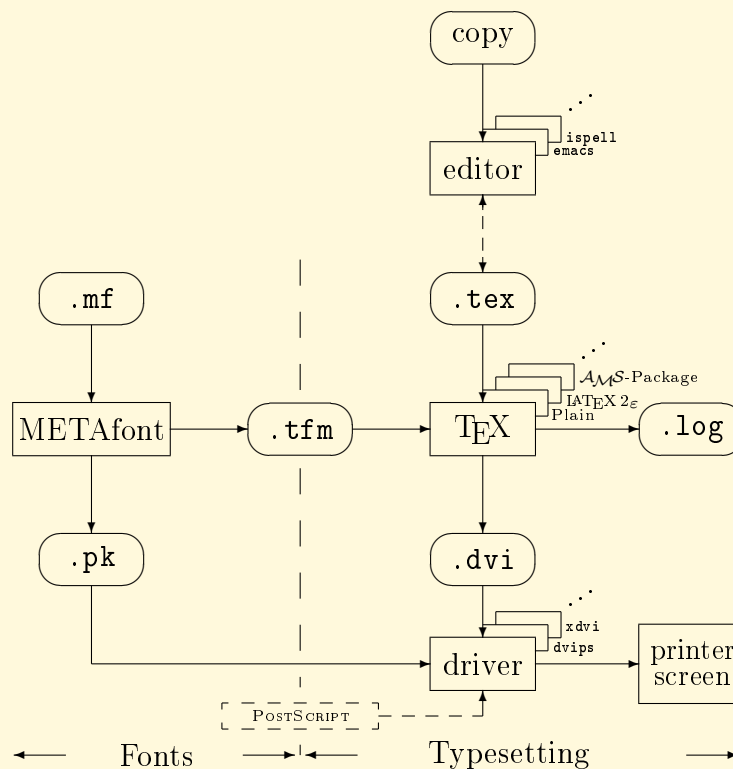


Рис. 1. Складові T_EX системи



Сайт ПНПУ

Головна

Зміст



Стор. 13 із 189

Назад

Перегляд

Закрити

Вихід

тексту як нижнього індексу. Але в $\mathcal{A}\mathcal{M}\mathcal{S}$ -T_EX'ові, на відміну від L^AT_EX'a, були відсутні деякі досить зручні для авторів можливості – достатні засоби керування версткою тексту (автоматична нумерація та перехресні посилання). Завдяки відзначеним зручностям, до середини 80-их років L^AT_EX придбав широку популярність, і в Американське Математичне Товариство стали надходити пропозиції від авторів приймати електронні варіанти їхніх статей в L^AT_EX'ові. Так, у 1987 році виник проект створення $\mathcal{A}\mathcal{M}\mathcal{S}$ -L^AT_EX'a і через 3 роки була випущена його версія 1.0.

Щодо макропакета T_EXinfo, то його створено для одночасного подання документа в поліграфічному вигляді й у вигляді ASCII-роздруку), а ChemT_EX – для оформлення хімічних текстів.

Бінарні реалізації T_EX процесора (tex, tex386, initex, virtex, ...), звичайно є різні для різних платформ і операційних систем і є складовими частинами різних дистрибутивів, наприклад:

- teT_EX для UNIX;
- em-T_EX² для MS-DOS, OS/2 і Windows 3.xx;
- MiK_TE_X для Windows 9x/Me/NT/XP;
- ozT_EX для Macintosh.

Тому, перш ніж починати шукати T_EX для своїх потреб, варто засвоїти деякі базові поняття.

Практично в усі версії T_EX'a входять вільно розповсюджені plain T_EX, $\mathcal{A}\mathcal{M}\mathcal{S}$ -T_EX і L^AT_EX, але прев'юери, оболонки й шрифти звичайно є комерційними. Досить широко відомі такі комерційні версії, як TurboT_EX, Y&YT_EX, VT_EX, а серед кириличних – ViT_EX (М.Виноградова) і ProT_EX (Протвинська).

MiK_TE_X – безкоштовний і загальнодоступний продукт, створений Крістіаном Ченком (Christian Schenk), — являє собою дистрибутив для платформи Win32, призначений для роботи з

²Програма розроблена Еберхардом Маттесом (Eberhard Mattes) отримала велику популярність у колишньому СРСР.

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 14 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

T_EX'ом. МікT_EX містить у собі, крім трансляторів T_EX'а й L^AT_EX'а з повним набором стандартних пакетів, ще й безліч додаткових програм, зокрема:

- T_EX 3.14159 – класичний компілятор з мови T_EX;
- e-T_EX – розширена й удосконалена версія T_EX;
- Yip – зручний прев'юер для перегляду створених DVI-файлів;
- PDFT_EX і PDFL^AT_EX – для одержання вихідних документів у форматі *.pdf (замість DVI);
- METAFONT – для генерації шрифтів високої поліграфічної якості, розроблена одночасно з T_EX'ом Д. Кнотом. T_EX і METAFONT призначаються для автоматизації процесу виготовлення оригіналів макетів складних науково-технічних видань. Слід зауважити, що T_EX правильно розташовує букви або символи на сторінці, а METAFONT визначає їх форми і співвідношення між ними, тобто програма METAFONT уможливорює дизайн шрифтів, використовуваних при друкуванні готових сторінок. За допомогою METAFONT можна не тільки змінити форму вже існуючих букв і символів, а також з нуля створити практично будь-яку гарнітуру. Особливо легко в METAFONT створюються логотипи або спеціальні символи;
- METAPost – потужна система, автор якої Джон Хоббі (John D. Hobby), призначена для створення векторних малюнків у форматі PostScript. Спочатку створюється текстовий файл із описом малюнка на скриптовій мові, дуже близькій по синтаксису до METAFONT, а потім запускається сама програма для одержання PostScript файлу;
- dviPS³ – для конвертації DVI-файлу в PostScript-файл;

³Написана Томасом Рокіккі (Tomas Rokicki) і реалізована для основних операційних систем

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 15 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

- MakeIndex⁴ – для створення різного роду алфавітних покажчиків;
- BibT_EX – для створення списку посилань із бібліографічної бази даних, написана Ореном Паташником (Oren Patashnik)⁵;
- TtH – програма перекладу документів T_EX в HTML-документи;
- $\mathcal{A}\mathcal{M}\mathcal{S}$ L_TE_X, Babel, PSNFSS, ... – стандартні пакети L^AT_EX;
- T_EXware, METAFONTware, Psutils, ... – службового призначення.

Ресурси L^AT_EX у мережі INTERNET. Існує міжнародний файловий архів, що називається STAN – Comprehensive T_EX Archive Network (Повна T_EX-овська архівна мережа). Так як T_EX є продуктом, що вільно поширюється, то будь-які його складові частини можна отримати безкоштовно на ftp-серверах. Основні сайти, що входять до складу STAN наведені у табл. 1. Кожен сервер STAN має потужні засоби пошуку, а також можливість копіювати цілі директорії в заархівованому вигляді, ввівши команду `get name_of_directory.ext`, де `.ext` – розширення, яке може бути `.zip`, `.tar`, `.tar.gz`, `.tgz`, `.zoo` і т.п. На даний час важливим є те, що всі ці сервери мають каталог `systems/win32/miktex/`, де MikT_EX – реалізація T_EX для Windows-платформ.

Користувачі T_EX усього світу об'єднані в Групу користувачів – TUG (T_EX Users Group – TUG). Перша така група утворена в 1979 році в США. Крім світового об'єднання TUG у різних країнах є свої національні об'єднання, а саме, Асоціація CyrTUG⁶ і аналогічні національні групи інших країн – Dante (Німеччина), NTG (Нідерланди), GUTenberg (Франція), UK

⁴Програма Пеонга Чена (Pehong Chen) і Майкла Харрисона (Michael Harrison).

⁵Співробітник Дослідницького центру засобів зв'язку, Ла-Холья.

⁶CyrTUG (Cyrillic T_EX Users Group) або Асоціація користувачів кириличного T_EX'а була заснована в травні 1991 р. Настановча конференція, на якій були присутні 23 T_EXuser'и з Москви, Санкт-Петербургу, Новосибірська й Протвіно (Інститут Високих Енергій), проходила в Москві, у видавництві «Мир». Вона співпала за часом зі святкуванням днів Кирила й Мефодія. CyrTUG (вимовляється «сиртаг») – суспільне об'єднання

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 16 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

Таблиця 1. Основні сайти CTAN

Країна	Сайти
Великобританія	ftp://ftp.tex.ac.uk/texarchive/
Німеччина	ftp://ftp.dante.de/tex-archive/
США	ftp://ctan.tug.org/tex-archive/
Схід США	ftp://tug2.cs.umb.edu/tex-archive/
Захід США	ftp://ftp.cdrom.com/pub/tex/ctan/
Австралія	ftp://unsw.edu.au/tex-archive/
Японія	ftp://ftp.riken.go.jp/pub/tex-archive/

TUG (Великобританія), GUST (Польща) та інші. Ці об'єднання працюють під керівництвом міжнародної групи TUG (США). Щорічно представники національних груп зустрічаються на міжнародних конференціях TUG і EuroT_EX. Більшість груп має свій бюлетень.

Асоціація CysTUG ставить перед собою наступні завдання:

- координацію роботи з удосконалювання кириличної версії T_EX;
- обмін інформацією й досвідом між користувачами T_EX'a, включаючи видання інформаційного бюлетеня, найпопулярніший із яких TUGBoat (видається 4 номери за рік), проведення конференцій, семінарів, курсів та ін.;
- забезпечення методичними й навчальними матеріалами з системи T_EX.

У рамках Асоціації функціонують електронний архів і бібліотечний абонемент, до яких члени Асоціації мають доступ. Усе програмне забезпечення, поширюване Асоціацією, є public

користувачів видавничої системи T_EX, що застосовують у своїй роботі російську та інші мови на кириличній основі.



Сайт ПНПУ

Головна

Зміст



Стор. 17 із 189

Назад

Перегляд

Закрити

Вихід

domain, що означає відсутність захисту й можливість вільного копіювання (користувач — користувачеві) у навчальних і просвітницьких цілях.

Асоціація закупає для своїх членів навчальну й методичну літературу по видавничій системі T_EX, що надається членам Асоціації через бібліотечний абонемент.

На сервері TUG можна знайти багато корисної інформації про T_EX, його складові компоненти, існуючі макропакети й дистрибутиви.

Україна на даний час немає свого національного об'єднання, а в Росії (Москва) існує Асоціація користувачів кириличного T_EX'у — CyrT_EX (утворена в 2000 році з CyrTUG, що діяла при видавництві «Мир»).

Слід звернути увагу на web-вузли:

- <http://www.tug.org/> – T_EX Users Group (TUG);
- <http://www.cemi.rssi.ru/cyrtug/> – Cyrillic T_EX Users Group (CyrTUG);
- <http://cyrtext.da.ru/> – CyrT_EX – Асоціація користувачів кириличного T_EX'у;
- <http://www.tex.ac.uk/cgi-bin/texfaq2html> – T_EX Frequently Asked Questions (FAQ);
- <http://www.library.lviv.ua/tex/> – Українське T_EX-видавництво;
- <http://www.word2tex.com/> – Chikrii SoftLab (фільтр-конвертор для експортно-імпортних операцій між файлами форматів T_EX/L^AT_EX та MS Word);
- <http://www.loria.fr/tex/> – T_EX/L^AT_EXnavigator.

Існують телеконференції по T_EX'у та його технологіях:

- FIDO: основна – ru.tex;
- додаткова – pvt.tex;

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 18 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

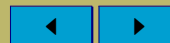
- usenet: основна – comp.text.tex.

Переваги й недоліки. Усі видавничі системи на базі T_EX'а мають переваги, що закладені в ньому самому. Ось вони:

- висока якість результату, недоступна іншим засобам поліграфічної підготовки (верстки) текстів. Друк із високою якістю виконується як на матричному принтері, так і на фото-складальному автоматі;
- простота підготовки дуже складних документів самим недосвідченим користувачем;
- велика кількість макропакетів, що дозволяють задовольнити всі вимоги користувача;
- на відміну від інших текстових процесорів, невибагливий до обчислювальної техніки;
- T_EX'овські файли мають високий ступінь стійкості: підготовлений у системі L^AT_EX вихідний текст, можна переслати у видавництво з впевненістю, що текст буде правильно оброблений, і на друці вийде в точності те, що вийшло при пробному друкуванні на принтері. Завдяки цьому T_EX дістав популярність як мова міжнародного обміну статтями з математики та фізики. Доречі, документи Word є нестійкими у відношенні коректного відображення деяких символів при подібних перенесеннях. Проблеми повної нечитабельності формул виникають навіть при використанні різних версій текстового процесора Word, а що вже говорити про різні типи комп'ютерів і операційних систем. Через закритість формату Word у жодному з альтернативних офісних пакетів немає дійсно якісного конвертора, у всякому разі поки що. Більше того, навіть якщо документ і вдасться перетворити (наприклад, у кросплатформному StarOffice), формули будуть у найкращому разі сприйняті як графічні об'єкти і правити їх можна тільки за допомогою MathType, що працює на платформах Windows і Mac;
- підтримка будь-яких мов у рамках одного документу;

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 19 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

- оскільки T_EX'овський файл – це звичайний ASCII файл, він має досить малий розмір. Книгу в 500 сторінок із сотнями й тисячами формул можна запросто вмістити на дискеті або надіслати поштою. Враховуючи те, що текстові файли добре архівуються, навіть незважаючи на наявність формул (до речі, кожний математичний символ займає в середньому три–п'ять байтів), розмір файлу стане разів у три меншим. Для порівняння: одна сторінка тексту, насиченого формулами, займає при наборі в Microsoft Office Word 46 Кб; ця ж сторінка в системі L^AT_EX займає 4 Кб, а в підготовленому для друку *.dvi-файлі – 6 Кб (файл типу .dvi – device independent, тобто незалежний від пристрою);
- L^AT_EX надає можливість вибору стандартних шаблонів оформлення документів: книга, стаття, доповідь, діловий лист;
- висока якість і гнучкість верстки абзаців і математичних формул (у цьому відношенні T_EX дотепер неперевершений). У підтвердження сказаного, слід відмітити, що можливостей загальновідомого редактора математичних формул Equation Editor і, тим більше, MathType з головою вистачає для підготовки щодо невеликих текстів: рефератів, проєктів і т.п. Тим більше, 64 МВ оперативної пам'яті на даний час є нормою для більшості комп'ютерів, а її вистачає для відносно комфортної роботи в Word 2000 з досить об'ємними документами. Втім, у цьому випадку авторів чекає суттєва проблема: формули зберігаються у вигляді об'єктів, і правити їх доводиться по черзі. Наприклад, неможливо виконати глобальну заміну фрагментів, як це робиться зі звичайним текстом;
- указавши логічну структуру тексту за допомогою простих засобів, автор може не вникати в деталі оформлення, які при необхідності неважко змінити (для зміни шрифту, яким друкуються заголовки, не треба нищпорити по всьому тексту, міняючи кожен, а досить замінити один рядок в «стильовому файлі»). До того ж, нумерація розділів, посилання, зміст і т.п. утворюються майже автоматично. У такий спосіб структурований підхід системи L^AT_EX дозволив найкращим чином реалізувати можливості T_EX'a;

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 20 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

- більш складні елементи тексту, такі як виноски, бібліографія, зміст, список таблиць і т.п., а також прості малюнки можуть бути виконані без особливих труднощів.

Більше 10-ти років тому в передмові до книги «T_EX і METAFONT. Нові напрямки в наборі» Гордон Белл написав: «T_EX Дональда Кнута — великий винахід нашого століття у видавничій справі. Він вводить стандартну мову комп'ютерної типографії й по своїй значимості може стояти поряд із винаходом друкарського верстата Гутенберга». Слід зауважити, що така оцінка цілком справедлива!

Втім, технології не стоять на місці, і сьогодні цілком можливо об'єднати універсальність і гнучкість T_EX системи із простотою й інтуїтивністю сучасних текстових процесорів. Створена чудова програма Scientific Word. По комфортності роботи й достатку візуальних засобів це, мабуть, абсолютний чемпіон. На жаль, далеко не кожний студент і навіть учений (і не тільки в нашій країні) може викласти за неї більше 500 долрів США. Розробники поширюють повнофункціональні пробні версії даної програми, з якими варто познайомитися.

До недоліків T_EX'а слід віднести:

- неможливість одночасного навчання й використання, навички здобуваються лише в процесі усвідомленого навчання. Знання обмінюються на особистий час;
- відсутня наочність при створенні таблиць;
- потрібно володіти навичками роботи в текстовому редакторі;
- орієнтований на багатфункціональне середовище;
- вимагає знання елементарних основ поліграфії;
- використання значної кількості пам'яті, складність друку отриманого результату на дешевому принтері;

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 21 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

- створення нових стилів оформлення – справа складна й під силу лише професіоналам. Звичайний користувач, як правило, з таким завданням не впорається. Утім, T_EX надає досить широкі можливості для модифікації стандартних стилів;
- більшість систем, заснованих на T_EX, не є системами WYSIWYG⁷ (L^AT_EX у тому числі). Робота з вихідним текстом і перегляд результату, який буде отриманий при друкуванні, – різні операції. До речі, завдяки цій особливості час на підготовку тексту типографської якості істотно скорочується. Слід відмітити, що на базі T_EX існують й WYSIWYG-системи підготовки публікацій. Деякі з них поширюються вільно (відповідно до ідеології Лупих, для якої вони написані), а деякі являють собою повноваговий комерційний продукт. Зокрема, безкоштовна WYSIWYG-система – текстовий редактор LuX. Самою, напевно, популярною системою підготовки текстів, заснованою на T_EX є розповсюджуваний компанією MacKichan Software науковий редактор Scientific Word та Scientific WorkPlace. Основним недоліком цих продуктів (для українського та російського користувача) є його ціна – близько 500 доларів. Їх можна рекомендувати тим, у кого є нагальна потреба в наборі математичних текстів і немає часу й бажання вивчати послідовності команд оригінального L^AT_EX'a.

Схема отримання документу за допомогою системи L^AT_EX. Спочатку автор повинен підготувати, за допомогою будь-якого текстового редактора, файл із текстом, що має розширення *.tex (наприклад, Disertac.tex). Текст цього файлу повинен починатися структурно-організованою послідовністю спеціальних команд L^AT_EX'a. Підбір команд та їх упорядкованість суттєво впливають на кінцевий результат видання. Отже, для реалізації задумів по створенню документів, листів, доповідей, звітів, статей, дисертацій та книг необхідно чітко знати призначення кожної команди, що використовується при написанні вихідних текстових файлів.

⁷WYSIWYG – це акронім для "what you see is what you get" ("що ви бачите, то й одержуєте"), іноді це записується так: "what you see is all you've got".

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 22 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

Наприклад, текстовий фрагмент набору книги наведено нижче⁸.

```
\documentclass[11pt,fleqn,draft]{book}
\usepackage[cp1251]{inputenc}
\usepackage[T2A]{fontenc}
\usepackage[ukrainian]{babel}
\usepackage{latexsym,amssfonts,amssymb}
```

```
\setlength{\textwidth}{170.0mm}
\setlength{\textheight}{240.0mm}
```

```
\pagestyle{myheadings}
```

```
\begin{document}
\tableofcontents
\input{introdaction}
\input{paragr_11}
```

Формули всередині тексту оточуються з обох боків знаками $\$$. Відокремлені формули оточуються парами знаків долара $\$\$$ і $\$\$$ по обидва боки.

```
\end{document}
```

Відмітимо, що для створення даного типу файлів існують спеціальні редактори-оболонки:

⁸Для прикладів по створенню вихідного файлу використовується шрифт друкарської машинки, який інколи застосовується при вказівках на команди операційної системи або на зовнішні імена файлів

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 23 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

TeXShell⁹, TeXnicCenter, TeXtelnExtel¹⁰, WinShell, WinEdt та інші.

Зручною й розвинутою оболонкою для MikTeX є текстовий редактор TeXnicCenter – безкоштовна програма, доступна всім бажаючим. Можливості редактора дуже широкі: він підтримує одночасну роботу з файлами практично будь-яких розмірів, має розвинений інтерфейс, багато в чому спеціально орієнтований на L^AT_EX, включає синтаксичний аналіз текстів, засоби макропрограмування й ін. Слід відзначити, що інтерфейс оболонки дуже нагадує середовище розробки Developer Studio 6.0.

Одним із кращих редакторів під Windows для набору в TeX'ові є WinEdt. Він автоматично інтегрується з уже проінстальованим MikTeX'ом, здійснює підсвічування синтаксису, дозволяє підключати словники, надає можливість писати макрокоманди, зручний при наборі формул. Взагалі кажучи, залишає позаду всі інші існуючі середовища. Робоче вікно WinEdt зображене на рис. 2.

Коротко про настройки WinEdt'a. Основні з них можна здійснити в меню:

- «**Options** → **Preferences**» – містить 15 закладок, дає можливість змінювати шрифт редактора, колір фону або підсвічування синтаксису;
- «**Options** → **Setting**» і «**Options** → **Appearance**» – дозволяє здійснювати більш тонкі настройки;
- «**Options** → **Menu Setup...**» – дуже важливий розділ, у якому можна реалізовувати вимоги користувача: доповнювати **Toolbar** новими розділами та кнопками, змінювати наявні.

Підключення українських та російських словників у WinEdt. Словники для WinEdt – це файли з розширенням *.dic, які знаходяться в каталозі «%B/Dict», де «%B» – це

⁹TeXShell Юргена Шлегельмільха (Juergen Schlegelmilch) вважається однією із кращих оболонк для операційної системи MS DOS.

¹⁰Оболонка для Windows розроблена Андреасом Кребсом (Andreas Krebs)

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 24 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

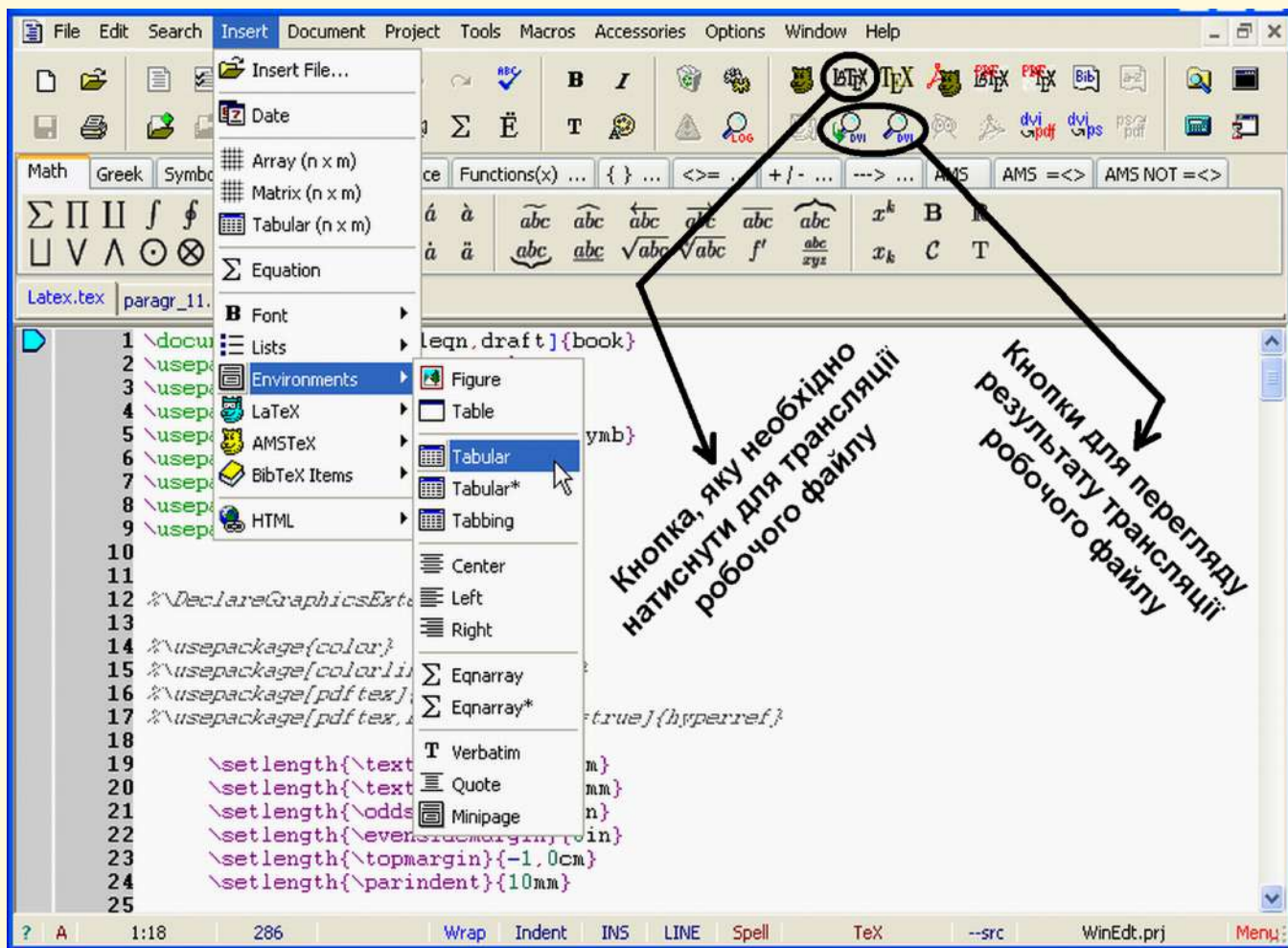


Рис. 2. Вікно WinEdt



Сайт ПНПУ

Головна

Зміст



Стор. 25 із 189

Назад

Перегляд

Закрити

Вихід

каталог, у якому встановлений WinEdt. Після інсталяції WinEdt'а є тільки англійські словники, а українські та російські – потрібно встановити самостійно: створюється в каталозі «%B/Dict» додатковий каталог, наприклад, «Rus» і поміщаються туди файли словників. Для активізації словників слід здійснити їх підключення в меню «**Options** → **Dictionary**». У закладці «**Dictionaries**» додається словник за аналогією з уже підключеними. Зміна підсвічування помилкових слів відбувається в меню «**Options** → **Dictionary** → **Errors**».

Подальша робота відбувається в два етапи. Перший із них – це обробка вихідного файлу за допомогою програми-транслятора. Якщо транслятор L^AT_EX, то його запуск здійснюється командою

```
LATEX ім'я вихідного файлу
```

У випадку, коли вихідний файл має розширення `tex`, досить указати коротке ім'я файлу.

При підготовці вихідного файлу за допомогою однієї з вище перерахованих спеціалізованих оболонок, трансляція проводиться, як правило, натисканням призначеної комбінації клавіш, або кнопки панелі інструментів. На рис. 2 вказано відповідну кнопку для редактора WinEdt, що дозволяє транслювати робочий файл.

На початку роботи L^AT_EX читає файли, що формують стиль документа. Якщо який-небудь зі стильових файлів недоступний, L^AT_EX зупиниться з діагностикою, що вказує відсутній файл, і запропонує ввести ім'я файлу, дозволяючи в такий спосіб користувачеві оперативно виправити допущену помилку.

Аналогічно L^AT_EX діятиме у випадку, коли виявиться помилковим фрагментом тексту зазначений у вихідному файлі.

У процесі роботи L^AT_EX, як і T_EX, використовує так званий формат.

Формат – це файл, що містить систему команд у відтрансльованій формі. Формат-файл має розширення `fmt`.

Для форматування тексту L^AT_EX'ові необхідно знати метрику шрифтів.

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 26 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

Метрика шрифту – це набір чисел, що визначає для кожного графічного подання символу – гліфу – необхідний йому простір на сторінці: розмір боксу, що містить у себе гліф, і деякі граничні умови, наприклад, відносні відстані між парами гліфів.

Метрика кожного шрифту зберігається у файлі з ім'ям шрифту й розширенням `tfm`.

У результаті роботи L^AT_EX'a створюється кілька файлів. Усі вони мають ім'я вихідного файлу, але різні розширення.

Основний вихідний файл L^AT_EX'a має розширення `dvi` (device-independent – незалежний від пристрою) і містить сформатований текст документу в машинно-незалежному вигляді.

Якщо документ містить зміст, список літератури й посилання на нього, список рисунків і(або) список таблиць, алфавітний покажчик, то потрібно здійснити дві трансляції. У перший раз будуть створюватися відповідні робочі файли з розширеннями:

`toc` – для змісту;

`aux` – для посилань;

`lof` – для списку рисунків;

`lot` – для списку таблиць;

`idx` – для списку термінів, включених в алфавітний покажчик.

За другим разом L^AT_EX вставить у текст документу всі посилання й доповнить його необхідними списками літератури, таблицями і т.д.

На наступному етапі отриманий `dvi`-файл за допомогою відповідних DVI драйверів переглядається на екрані, створюється PostScript файл, та здійснюється друк на принтері. Ці драйвери є різні для різних платформ за виключенням `dvips` (драйвер друку в PostScript форматі), який існує для всіх операційних систем і вважається штатним. Для перегляду екрані `dvi`-файла потрібно натиснути функціональні кнопки, що виділені на рис. 2. Незадоволений результатом, автор вносить зміни у вихідний файл – і цикл повторюється. Насправді повторень циклу буде більше, тому що при синтаксичних помилках у вихідному тексті транслятор буде видавати повідомлення про помилки, які потрібно виправляти. Графічна інтерпретація отримання документу за допомогою L^AT_EX представлена на рис. 3.

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 27 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

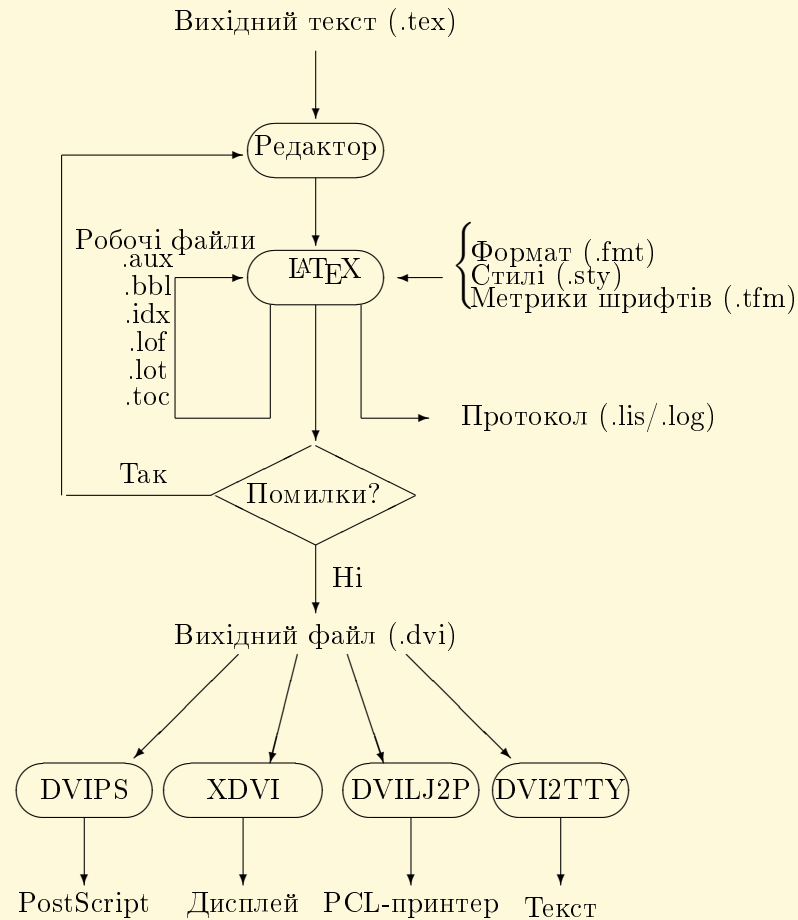


Рис. 3. Схема отримання документа за допомогою L^AT_EX.



- Сайт ПНПУ
- Головна
- Зміст
- ◀
▶
- ◀
▶
- Стор. 28 із 189
- Назад
- Перегляд
- Закрити
- Вихід

Отже, перед тим як почати працювати в системі L^AT_EX, необхідно знати:

- як створювати текстові файли з розширенням *.tex;
- набір команд та їх зміст для створення вихідного файлу;
- що потрібно зробити, щоб відтранслювати вихідний текст (тобто створити з нього dvi-файл);
- що необхідно зробити, щоб переглянути dvi-файл на екрані;
- що слід зробити, щоб надрукувати dvi-файл;
- принципи роботи хоча б із одним текстовим редактором типу WinEdt.

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 29 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

1. Вихідний файл

Вихідний файл для системи L^AT_EX являє собою текст документа разом зі спецсимволами й командами, за допомогою яких передаються вказівки відносно розміщення тексту. Цей файл можна створити в будь-якому текстовому редакторі (рекомендується використовувати спеціалізований редактор WinEdt, див. с. 24), але при цьому необхідно, щоб у результаті був одержаний так званий «чистий» текстовий файл (ASCII-файл). Це означає, що текст не повинен містити шрифтових виділень, розбивки на сторінки й т.п.

Вихідний текст документа не повинен містити переносів (T_EX зробить їх сам). Слова відокремлюються одне від одного проміжками, при цьому T_EX не розрізняє, скільки саме проміжків залишено між словами (щоб власноруч керувати проміжками між словами, є спеціальні команди). Кінець рядка також сприймається як проміжок. Окремі абзаци повинні бути відокремлені одне від одного порожніми рядками (немає різниці, скільки порожніх рядків стоїть між абзацами, важливо, щоб був хоча б один). У правому стовпчику наведений фрагмент вихідного тексту, а в лівому – те, як він буде виглядати при друку після обробки системою L^AT_EX.

Слова розділяються проміжками, а абзаци — порожніми рядками.

Абзацний відступ у вихідному тексті залишати не потрібно: він утворюється автоматично.

Слова розділяються проміжками, а абзаци - порожніми рядками.

Абзацний відступ у вихідному тексті залишати не потрібно: він утворюється автоматично.

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 30 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

2. Символи, що вводяться безпосередньо

Як зазначалося, вихідний файл документу, що готується до публікації засобами L^AT_EX'a, містить інформацію двох типів: текст і розмітку (команди форматування). Текст в основному вводиться безпосередньо, хоча існують символи, вводити які необхідно певним чином, тому що: вони можуть бути відсутні на клавіатурі, або L^AT_EX використовує їх як керуючі символи (див. розділ 3, [10, 12, 11]).

Шляхом простого натискання відповідної клавіші до дисплея вводяться проміжки, табулятори й ознака кінця рядка, а також:

Малі літери

a b c d e f g h i j k l m n o p q r s t u v w x y z
 а б в г г д е ж з и і ї й к л м н о п р с т у ф х ц ч ш щ ь ю я

Прописні букви

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
 А Б У Г Г Д Е Є Ж З И І Ї Й К Л М Н О П Р С Т У Ф Х Ц Ч Ш Щ Ь Ю Я

Арабські цифри

1 2 3 4 5 6 7 8 9 0

Знаки пунктуації, дужки й інші неbukвенні символи, наявні на клавіатурі будь-якого дисплею

. , ? ! : ; () [] ' ' - * / + = @


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)


Стор. 31 із 189

[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

3. Спецсимволи

До спеціальної групи відносяться символи, які при безпосередньому введенні інтерпретуються як керуючі (мають особливий статус), а також спеціальні та діакричні знаки (див. розділ 23). Наступні 10 символів є керуючими:

```
{ } $ & # % _ ^ ~ \
```

Друковане зображення знаків, що відповідають першим семи з них, можна одержати, якщо у вихідному тексті поставити перед відповідним символом без проміжку знак \ (англійська назва «backslash»):

Курс долара підвищився на 2%. Тепер \$1 коштує 5 гривен 33 копійки.

Курс долара підвищився на 2\%.
Тепер \\$1 коштує 25 гривен 33 копійки.

Знаки # (ознака параметра в T_EX-визначенні макрокоманди) і & (табулятор, за допомогою якого виконується розмітка стовпців у таблицях або масивах даних) використовуються в складних конструкціях T_EX'a.

У L^AT_EX дещо робиться інакше, ніж в простому редакторі. Так, не можна ставити лапки через Shift+2, а ставити їх слід або “текст в лапках”, або так «текст в лапках». Згідно вітчизняних поліграфічних традицій в якості основних лапок використовуються лапки-ялинки («»), а інші, що набираються поелементно (‘ ’) – є допоміжними.

Якщо символ % ужитий у тексті не в складі комбінації \%, то він є «символом коментарю»: усі символи, розташовані на рядку після нього, T_EX ігнорує (у тому числі й сам %). За допомогою символу % у вихідний текст можна вносити позначки «для себе»:


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)


Стор. 32 із 189

[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

Це приклад.

Рідна мати моя, ти ночей не доспала. Ти водила мене у поля край села.

Це % повчальний
приклад. %Слова з пісні.

Рідна мати моя, ти ночей
не доспала. Ти водила ме%
не у поля край села.

Необхідно звернути увагу на передостанній рядок: після знака відсотка ігнорується весь рядок, включаючи його кінець, що в нормальних умовах відіграє роль проміжку; з іншого боку, початкові проміжки в рядку ігноруються завжди. Тому T_EX не бачить проміжку між словами *ме й не*, тому вони складаються в слово «мене».

Фігурні дужки обмежують групи у вихідному файлі, а також обов'язкові параметри команд. T_EX і L^AT_EX мають широкі можливості групувати матеріал, що у деякому розумінні повинен трактуватися однаково. Наприклад, фрагмент тексту треба виділити, надрукувавши його іншим шрифтом або центрованим, оформити його у вигляді посилання та інше. Такий фрагмент оформляється в групу за допомогою фігурних дужок, які вказують область дії поточної команди. Групування може бути поширене на будь-яку глибину (дужки усередині дужок усередині дужок...) з однією тільки умовою: дужки повинні бути збалансовані.

Знак долара обмежує математичні формули. При наборі математичних формул застосовуються знаки $_$ і $^$ («знак підкреслення» і «кришка»), що є ознаками нижнього та верхнього індексів.

Знак \sim позначає «нерозривний проміжок» між словами. Слова, зв'язані таким проміжком, не можуть бути рознесені на окремі рядки.

Зі знака \backslash починаються всі T_EX'овські команди.



Сайт ПНПУ

Головна

Зміст



Стор. 33 із 189

Назад

Перегляд

Закрити

Вихід

4. Команди та їхнє задання в тексті

Задання друкованого знака відсотка за допомогою послідовності символів `\%`, – приклад важливого поняття T_EX'а, – названого командою.

Команди L^AT_EX'а починаються керуючим символом `\` і мають наступний формат:

```
\ім'я[необов'язкові параметри]{обов'язкові параметри}
```

Параметри задаються або числом, або послідовністю символів. Число може бути виражено розмірною або безрозмірною величиною, а також командою, якій уже привласнено деяке значення. Параметром команди як правило є фрагмент тексту.

В іменах команд не використовуються цифри. Необов'язкові параметри варто поміщати у квадратні дужки, обов'язкові – у фігурні. Порядок розміщення параметрів команд суттєвий.

Відносно запису у вихідному тексті, команди поділяться на два типи:

- команди, що складаються зі знака `\` та одного символу після нього, що не є буквою. Саме до цього типу відносяться команди `\{`, `\}`, ..., `\%`;
- команди, що складаються з `\` і послідовності букв – *ім'я* команди (*ім'я* може складатися й з однієї букви).

Наприклад, команди `\TeX` і `\LaTeX` генерують емблеми систем T_EX і L^AT_EX відповідно. В імені команди, а також між `\` та ім'ям, не повинно бути проміжків; ім'я команди не можна розривати при переносі на інший рядок.

Прописні й малі літери в іменах команд розрізняються. Наприклад, `\large`, `\Large` і `\LARGE` — це три різні команди (вони задають різні розміри шрифту).

Після команди першого типу проміжок у вихідному тексті ставиться або не ставиться залежно від того, що необхідно одержати при друкуванні:

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 34 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

У чому різниця між \$1 і \$ 1?

У чому різниця між \ \$1 і \\$ 1?

Після команд другого типу, що не містять параметрів повинен обов'язково стояти, або проміжок, або символ, що не є буквою (це необхідно, щоб T_EX зміг визначити, де кінчається ім'я команди й починається подальший текст). Приклади з використанням команди \sl (перемикає шрифт на похилий):

24 роки ми вільні.
Слава Україні!

\sl24 ми вільні. \sl Слава Україні!

Написавши \slСлава Україні!, при трансляції T_EX фіксує помилку й повідомляє про те, що команда \slСлава не визначена.

Якщо після команди з \ і букв у вихідному тексті йдуть проміжки, то при трансляції вони ігноруються. При необхідності отримання проміжку після команди у вихідному тексті (наприклад, щоб генероване за допомогою команди слово не зливалося з наступним текстом), треба цей проміжок створити. Один із можливих способів — поставити після команди відкриваючу та закриваючу фігурні дужки {}, і вже після них зробити проміжок, якщо потрібно. Іноді можна також поставити команду \ (backslash із проміжком після нього), що генерує проміжок. Приклад.

Увага! Після L^AT_EX є \. При його відсутності чи фігурних дужок {} після команди T_EX відбудеться злиття емблеми з наступним словом.

Увага! Після \LaTeX\ є \\$\backslash.\$
При його відсутності чи фігурних дужок \{\} після команди \TeX відбудеться злиття емблеми з наступним словом.

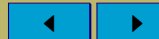
В останньому рядку цього прикладу свідомо не створено проміжку після команди T_EX, щоб продемонструвати злиття емблеми T_EX'a з наступним текстом.



Сайт ПНПУ

Головна

Зміст



Стор. 35 із 189

Назад

Перегляд

Закрити

Вихід

5. Структура вихідного тексту

Початок L^AT_EX-файлу – команда, що задає стиль оформлення документа. Приклад:

```
\documentstyle{book}
```

Слово `book` у фігурних дужках вказує, що документ буде оформлений, як книга: усі розділи починатимуться з непарної сторінки, у тексті будуть певного типу колонтитули й т.п. Крім стилю `book`, у стандартний комплект L^AT_EX'a входять базові стилі `article` (для оформлення статей), `report` (щось середнє між `article` і `book`) та `letter` (для оформлення ділових листів так, як це прийнято в США). Слід зауважити, що форма ділового листа, прийнята на Україні й на Заході (розташування адрес, дати, підписів), відрізняється настільки, що робить останній базовий стиль у його оригінальній формі маловживаним. Щоб задати оформлення документа за допомогою одного із цих стилів, треба у фігурних дужках після команди `\documentstyle` вказати замість `book` назву необхідного стилю.

У L^AT_EX 2_ε замість команди `\documentstyle` застосовується команда `\documentclass`. Те, що в L^AT_EX'ові 2.09 називалося «стилем документа», називається класом. Крім класів `article`, `report`, `book` і `letter`, L^AT_EX 2_ε підтримує й деякі інші, наприклад, `slides` – для створення слайдів. Якщо L^AT_EX 2_ε буде обробляти файл, написаний для L^AT_EX 2.09 (розпочинається з команди `\documentstyle`), то він перейде в «режим сумісності» і буде імітувати роботу L^AT_EX'a 2.09.

Отже, перше, що L^AT_EX повинен знати при обробці початкового файлу, це тип створюваного автором документа, що задається командою:

```
\documentclass[опції]{клас}
```

Параметр `опції` змінює поводження класу документа. Опції повинні розділятися комами. У табл. 2 перераховані самі вживані опції стандартних класів документів.

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 36 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

Таблиця 2. Опції класів документів

<code>10pt, 11pt, 12pt</code>	Установлює розмір основного шрифту документа. Якщо жодна із цих опцій не зазначена, виконуватиметься <code>10pt</code> .
<code>a4paper, legalpaper, a5paper, executivepaper, b5paper, letterpaper.</code>	Визначає розмір аркуша. За замовчуванням використовується <code>letterpaper</code> .
<code>fleqn</code>	Виділені формули будуть вирівняні по лівому краю.
<code>leqno</code>	Формули нумеруються ліворуч, а не праворуч.
<code>titlepage, notitlepage</code>	Указує на те, чи повинна друкуватися титульна сторінка в документі чи ні. За замовчуванням клас <code>article</code> не має титульної сторінки, а <code>report</code> і <code>book</code> – мають.
<code>twoside</code>	Задає друк з різними полями на непарних і парних сторінках (як у книгах). Її можна вказувати тільки для стилів <code>article</code> й <code>report</code> ; у стилі <code>book</code> такий друк виходить автоматично і вказувати цю опцію непотрібно.
<code>twocolumn</code>	Змушує L ^A T _E X набирати документ у два стовпці.
<code>openright, openany</code>	Створює глави, що починаються, або тільки на правій сторінці, або на першій доступній. Це не працює із класом <code>article</code> , тому що він нічого не знає про глави. Клас <code>report</code> за замовчуванням починає глави на наступній сторінці, а клас <code>book</code> – на правій.



Сайт ПНПУ

Головна

Зміст



Стор. 37 із 189

Назад

Перегляд

Закрити

Вихід

Стандартні стилі можна змінювати, а також створювати нові. Наступні команди, що йдуть після `\documentclass` відносяться до всього документу, і встановлюють різні параметри його оформлення. Наприклад:

```
\documentclass[12pt,fleqn]{report}
\usepackage[cp1251]{inputenc}      %% №1
\usepackage[T2A]{fontenc}         %% №2
\usepackage[ukrainian]{babel}     %% №3
\usepackage{latexsym,amsmath}     %% №4
.....
\setlength{\textwidth}{170.0mm}   %% №5
\setlength{\textheight}{240.0mm}
.....
\begin{document}
.....
\end{document}
```

Щодо пронумерованих команд у наведеному прикладі: №1 – готує L^AT_EX для сприйняття кодування Windows; №2 – говорить L^AT_EX’у шрифт із яким внутрішнім кодуванням варто його використати в самому документі; №3 – потрібна для підтримки переносів в обраній мові; №4 – підключає додаткові пакети `latexsym`, `amsmath`, що розширюють можливості набору формул; №5 – встановлюють розміри області друку.

Далі слідує команда `\begin{document}`, після якої повинен розміщуватися авторський текст. Якщо текст або команди, що генерують текст розташовані до `\begin{document}`, L^AT_EX повідомляє про помилку. Частина файлу між командами `\documentclass` і `\begin{document}`, називається преамбулою. Закінчувати файл необхідно командою `\end{document}`. Усе написане у файлі після `\end{document}` L^AT_EX проігнорує.


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)

[Стор. 38 із 189](#)
[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

6. Розбиття вихідного файлу на частини

Часто буває зручно розбити великий вихідний текст на кілька частин, що зберігаються в різних файлах. Щоб можна було об'єднати їх у єдине ціле, передбачена команда `\input`. Якщо в тексті написати

```
\input ім'я файлу,
```

то T_EX буде працювати так, ніби замість рядка з командою `\input` знаходився текст файлу зі вказаним ім'ям. Також можна написати

```
\input{ім'я файлу},
```

взявши ім'я файлу в фігурні дужки. Звичайно, коли готують текст великого об'єму, то створюють невеликий файл, у якому між `\begin{document}` і `\end{document}` розміщені рядки з командами `\input`, що дають змогу вставляти файли з записом основної частини тексту. Наприклад, книжку з двох розділів, записаних у файлах `ch1.tex`, `ch2.tex`, можна представити у вигляді файлу з одинадцяти рядків. Саме його, а не файли з окремими главами, треба буде передати для обробки L^AT_EX'у:

```
\documentclass[11pt]{book}
\usepackage[cp1251]{inputenc}
\usepackage[T2A]{fontenc}
\usepackage[ukrainian]{babel}
\begin{document}
\input ch1.tex
\input ch2.tex
\end{document}
```

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 39 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

Кожну команду `\input` варто розташовувати на окремому рядку, як у наведеному вище прикладі. Якщо розширення файлу, що є аргументом команди `\input`, не зазначено, то T_EX за замовчуванням вважає, що це розширення має вигляд `.tex`. Якщо в тексті присутні команди `\input`, то в процесі трансляції при початку зчитування відповідного файлу на екран видається його ім'я, щоб було зрозуміло, до якого з файлів у разі наявності будуть належати подальші повідомлення T_EX'а. Для того щоб T_EX прочитав тільки частину файлу, можна скористатися командою `\endinput`. При її наявності у файлі, що читається T_EX'ом за допомогою команди `\input`, файл буде прочитаний тільки до рядка з написом `\endinput`.

Для включення тексту можна застосувати й інший спосіб. За допомогою декларації `\include` зручно вставляти файли з окремими главами документа. При цьому розширення `.tex` потрібно опустити. Крім того, `\include`, на відміну від `\input`, не можна використовувати рекурсивно, тобто не можна її поміщати в уже вкладені файли.

Виділяти потрібний фрагмент документа за допомогою коментарів не зовсім зручно так, як порушується нумерація сторінок, переробляється зміст документа, порушуються посилання на літературу і т. п. Щоб цього не відбувалося, в преамбулі документа використовується команда `\includeonly` із зазначенням імен файлів, які необхідно включити в документ. Відмітимо, що `\includeonly` використовується тільки разом з командами `\include`, на `\input` вона не діє.

Припустимо, що на певному етапі підготовки документа нас цікавить тільки матеріал, що міститься у файлах `file2.tex`, `file3.tex`. Тоді необхідно прописати так:

```
\documentclass[a4paper]{article}

\includeonly{file2,file3} % пробілу після коми бути не повинно
\begin{document}
\include{file1} \include{file2} \include{file3}
\end{document}
```

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 40 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

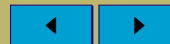
7. Групи

Частина тексту вихідного файлу, що знаходиться між фігурними дужками визначає поняття групи. Самі по собі фігурні дужки не генерують ніякого тексту й не впливають на шрифт; єдине, що вони роблять – це обмежують групу всередині файлу. Команди T_EX'а, що змінюють різні параметри, діють у межах тієї групи, усередині якої вони були встановлені. По закінченні групи відновлюється той режим, що був до її початку. Розглянемо приклад.

Спочатку перемкнемо шрифт на напівжирний, потім на *курсив*; тимчасово перейдемо знову на **напівжирний**; слід звернути увагу на те, що шрифт відновиться після кінця групи. **Це також напівжирний шрифт.** *А це вже курсив.*

Спочатку {перемкнемо шрифт на напівжирний, потім на `\it` курсив; тимчасово перейдемо знову на `{\bf напівжирний}`; слід звернути увагу на те, що шрифт відновиться після кінця групи.
`{`
`\bfseries` Це також напівжирний шрифт. `}`
`\textit`
`{А це вже курсив.}`

Бачимо, що пробіли ігноруються після імені команди (між ім'ям команди і відкритою фігурною дужкою першого аргументу), але не після відкритої або перед закритою фігурними дужками. Групи можуть бути вкладені одна в одну. Усередині зовнішньої групи напівжирний шрифт почався не з того місця, де було відкрито дужку, а тільки після команди `\bf` (сама команда, а не дужка, перемикає шрифт). Остання група на друк ніяк не вплинула: адже всередині дужок ніяких команд немає. Ставити пару дужок `{}` після ім'я команди – теж приклад використання груп. У цьому випадку дужки обмежують «порожню» групу. Фігурні дужки у вихідному тексті повинні бути збалансовані: кожній відкриваючій дужці повинна відповідати закриваюча. Якщо не дотримуватися цієї умови, при трансляції буде повідомлено про помилку.


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)


Стор. 41 із 189

[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

8. Параметри, команди з аргументами, оточення. Зірочка після ім'я команди

Поряд із поточним шрифтом, T_EX при обробці вихідного тексту враховує значення різних параметрів: величина абзацного відступу, ширина й висота сторінки, вертикальна відстань між сусідніми абзацами, а також безліч інших. Параметри T_EX'a позначаються аналогічно командам: за допомогою символу \ («backslash»), за яким слідує або послідовність букв, або одна не-буква. Наприклад, команда `\parindent` вказує T_EX'ові величину абзацного відступу; якщо необхідно, щоб абзацний відступ дорівнював двом сантиметрам, можна написати так:

```
\parindent=2cm
```

Аналогічно роблять і в інших випадках: для зміни параметру необхідно написати його позначення, потім знак рівності, а потім значення, що «присвоюється» цьому параметру. У залежності від параметру, це може бути просто ціле число чи довжина, або щось інше.

Команди, наприклад, `\LaTeX` чи `\bf` діють «самі по собі», але багатьом командам, необхідно задати аргументи. Одним із прикладів є команда `\documentstyle` (L^AT_EX 2_ε – `\documentclass`): слово у фігурних дужках – її аргумент; не вказування його, призводить до помилки. У L^AT_EX'ові аргументи команд бувають обов'язкові й необов'язкові. Обов'язкові аргументи задаються у фігурних дужках. Якщо для команди передбачено їх наявність, то вона без них правильно працювати не буде. Необов'язкові аргументи, впливають на роботу команди, але при цьому нормальна робота команди не порушиться й при їхній відсутності. Необов'язкові аргументи задаються у квадратних дужках. Зокрема, у команди `\documentclass` передбачені принаймні один обов'язковий та один необов'язковий аргументи. У квадратних дужках перед обов'язковим аргументом можна вказати список (через кому) необов'язкових, так званих стильових опцій, що дозволяє отримати додаткові можливості оформлення. Наприклад, для набору книги у два стовпчики шрифтом кегля 12 замість кегля 10, прийнятого за замовчуванням, потрібно

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 42 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

розпочати файл командою

```
\documentclass[12pt,twocolumn]{book}
```

Якщо перші два рядки файлу є такими:

```
\documentclass[12pt,twocolumn]{book}
```

```
\usepackage{amsmaths,longtable}
```

то це свідчить про те, що у даному випадку підключено пакет `amsmaths`, щоб використовувати в математичних формулах додаткові шрифти, та пакет `longtable`, який потрібний для набору громіздких таблиць.

Слід відмітити, що необов'язкові аргументи іноді повинні розташовуватися до обов'язкових, іноді – після. Їх упорядкованості необхідно строго дотримуватися. Між дужками з обов'язковими аргументами, можуть бути проміжки, але не повинно бути порожніх рядків.

8.1. Оточення

Ще одна важлива конструкція L^AT_EX'a – це оточення (`environment`).

Оточення – це фрагмент файлу, що починається з тексту

```
\begin{Ім'я оточення}
```

де *Ім'я оточення* являє собою перший обов'язковий (і, можливо, не єдиний) аргумент команди `\begin`. Закінчується оточення командою

```
\end{Ім'я оточення}
```

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 43 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

(команда `\end` має тільки один аргумент – ім'я оточення, яке вона завершує). Наприклад:

Всі рядки наведеного абзацу будуть розташовані по центру.

```
\begin{center}
Всі рядки наведеного абзацу будуть
розташовані по центру.
\end{center}
```

Кожній команді `\begin`, що відкриває оточення, повинна відповідати команда, що його закриває `\end` (зрозуміло, з тим же ім'ям оточення в якості аргументу). Властивість оточення: частина файлу, що перебуває всередині оточення, утворює групу.

У L^AT_EX'ові деякі команди й оточення мають варіанти, у яких безпосередньо після ім'я команди або оточення ставиться зірочка *. Наприклад, команда `\chapter` означає «почати новий розділ документа», а команда `\chapter*` означає «почати новий розділ документа, не нумеруючи його». Після ім'я команди із зірочкою проміжки не ігноруються; якщо команда із зірочкою має аргументи, проміжку між зірочкою й аргументами не повинно бути.

8.2. Одиниці довжини

У табл. 3 зібрані одиниці довжини, які можна використовувати в T_EX'ові для задання розмірів.

Можна задавати розміри за допомогою будь-яких із цих одиниць. При запису дробового числа можна використовувати як десяткову кому, так і десяткову крапку. Прописні й малі літери в позначеннях одиниць довжини не розрізняються. Навіть якщо довжина, що вказується T_EX'у, дорівнює нулю, однак після нього необхідно вказати будь-яку з використовуваних T_EX'ом одиниць довжини. Крім перерахованих, у T_EX'ові використовуються ще дві «відносні» одиниці довжини, розмір яких залежить від поточного шрифту: `em` – приблизно рівна ширині букви `M`; `ex`, наближено дорівнює висоті букви `x`. Ці одиниці зручно використовувати в коман-

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 44 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

Таблиця 3. T_EX'овські одиниці довжини

pt	пункт ≈ 0.35 міліметра
pc	піка = 12pt mm
mm	міліметр
cm	сантиметр = 10 mm
in	дюйм = 10 mm
in	дюймнкт Дідо $\approx 1,07$ pt
cc	цицero = 12 dd

дах, що повинні однаково працювати для шрифтів різних розмірів. Зокрема, відстань у 1em зазвичай візуально сприймається як «один проміжок».

Параметр довжини – це команда довжини, що в L^AT_EX'ові впливає на зовнішній вигляд вихідного результату. Параметри довжини бувають *жорсткими* й *еластичними*. Параметр еластичної довжини – це параметр, реальне значення якого залежить від контексту, у якому він з'являється, у той час як значення параметра жорсткої довжини фіксоване й ніколи не розтягується й не стискується. Наприклад, значення параметра жорсткої довжини `\textheight` дорівнює висоті тіла сторінки вихідного результату, отриманого L^AT_EX'ом.

Параметри жорсткої довжини

Приведемо список параметрів жорсткої довжини L^AT_EX'а:


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)

[Стор. 45 із 189](#)
[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

<code>\arraycolsep</code>	<code>\headsep</code>	<code>\marginparpush</code>
<code>\arrayrulewidth</code>	<code>\itemindent</code>	<code>\marginparsep</code>
<code>\bibindent</code>	<code>\labelsep</code>	<code>\marginparwidth</code>
<code>\columnsep</code>	<code>\labelwidth</code>	<code>\mathindent</code>
<code>\columnseprule</code>	<code>\leftmargin</code>	<code>\oddsidemargin</code>
<code>\doublerulesep</code>	<code>\leftmargini</code>	<code>\parindent</code>
<code>\evensidemargin</code>	<code>\leftmarginii</code>	<code>\rightmargin</code>
<code>\fboxrule</code>	<code>\leftmarginiii</code>	<code>\tabbingsep</code>
<code>\fboxsep</code>	<code>\leftmarginiv</code>	<code>\tabcolsep</code>
<code>\footheight</code>	<code>\leftmarginv</code>	<code>\textheight</code>
<code>\footnotesep</code>	<code>\leftmarginvi</code>	<code>\textwidth</code>
<code>\footskip</code>	<code>\linewidth</code>	<code>\topmargin</code>
<code>\headheight</code>	<code>\listparindent</code>	<code>\unitlength</code>

Щоб змінити будь-який із них, треба включити у вхідний або стильовий файл присвоювання типу `\arraycolsep=5pt` або використати команду `\setlength`, яка еквівалентна оператору присвоювання, наприклад, так:

```
\setlength{\unitlength}{1mm}
```

або так:

```
\setlength{\rightmargin}{\leftmargin}.
```

Параметри еластичної довжини

Довжина в L^AT_EX це не просто якийсь певний розмір це більш складна структура із зазначенням меж можливого стиснення і розтягування. Межі розтягування визначаються з допомогою


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)

[Стор. 46 із 189](#)
[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

інструкції `plus`, а стиснення `minus`. При формуванні абзаців T_EX використовує цю інформацію для максимально гарного заповнення.

Приведемо список параметрів еластичної довжини L^AT_EX'a:

<code>\abovedisplayshortskip</code>	<code>\doubletextfloatsep</code>	<code>\partopsep</code>
<code>\abovedisplayskip</code>	<code>\floatsep</code>	<code>\textfloatsep</code>
<code>\baselineskip</code>	<code>\intertextskip</code>	<code>\topsep</code>
<code>\belowdisplayshortskip</code>	<code>\itemsep</code>	<code>\topskip</code>
<code>\belowdisplayskip</code>	<code>\parsep</code>	
<code>\dblfloatsep</code>	<code>\parskip</code>	

Любий із цих параметрів можна змінити присвоюванням наступного виду:

```
\parskip=12pt plus 4pt minus 2pt
```

Це означає, що реальне значення еластичного параметра довжини дорівнює 12 пунктам – у T_EXнології воно називається *природним проміжком* – але воно може розтягуватися (або розширюватися) на 4 пункти й стискуватися (або скорочуватися) на 2 пункти. У присвоюваннях значень параметрам еластичної довжини ключові слова `plus` та `minus` і пов'язані з ними довжини є обов'язковими; якщо яке-небудь із них опущене, T_EX вважає його рівним нулю.

Докладніше про деякі команди. Перша `\abovedisplayskip` – це додатковий вертикальний інтервал над довгими виокремленими формулами, які починаються лівіше кінця попереднього рядка. Якщо виокремлення формули вирівнюються по лівому краю сторінки, замість команди `\abovedisplayskip` використовується `\topsep`.

Команда `\belowdisplayskip` додатковий пробіл під довгими виокремленими формулами.

Додатковий пробіл над короткими виокремленими формулами, що розпочинаються праворуч від місця закінчення попереднього рядка, ставиться командою `\abovedisplayshortskip`.

Аналогічно, `\belowdisplayshortskip` – пробіл під короткими виокремленими формулами. Команди не працюють із опцією `fleqn` при використанні `\topsep` і є розтяжними довжинами.


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)

[Стор. 47 із 189](#)
[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

9. Автоматична генерація посилань

L^AT_EX надає можливість організувати посилання на окремі сторінки або розділи документа таким чином, щоб програма сама визначала номери сторінок, або розділів у цих посиланнях. Для цього слід виконати наступне:

- позначити місце в тексті, на яке потрібно посилатися надалі (або перед цим);
- у тому місці тексту, де необхідно розмістити посилання, ставиться команда-посилання на мітку.

Позначається будь-яке місце тексту за допомогою команди `\label`. Ця команда має один обов'язковий аргумент у фігурних дужках – «мітку». В якості мітки можна використати будь-яку послідовність букв, цифр і розділових знаків (без проміжків, фігурних дужок, символів та `\`). Наприклад, ця команда може мати вигляд:

```
\label{vash}
```

Посилання на сторінку, на якій розташована мітка, здійснюється командою `\pageref`. Вона має один обов'язковий аргумент – та ж мітка, на яку робиться посилання. Приклад:

Будьте обережні з грибами! До кошику разом із їстівними можуть потрапити й отруйні.

Як відомо (див. с. 48), збирання грибів – небезпечна справа.

Будьте обережні з грибами\label{vash}!
До кошику разом із їстівними можуть потрапити й отруйні.

Як відомо (див. с. \pageref{vash}),
збирання грибів - небезпечна справа.

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 48 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

Команда `\label` розташовується поруч із ключовим словом «грибами» без проміжку для впевненості, що буде позначена саме та сторінка, на яку потрапило це слово. Для одержання посилань необхідно дві трансляції. Якщо після двох запусків підряд з'явиться повідомлення про невизначене посилання, це свідчить, що у вихідному тексті є помилка (найімовірніше, неточність в аргументі команди `\pageref` або відсутність в тексті команди `\label`). На місце, позначене за допомогою команди `\label`, можна посилатися командою `\ref`. Тоді при друкуванні з'явиться не номер сторінки, а номер розділу документа, у якому перебуває мітка, або номер малюнка, або номер елемента в «нумерованому переліку» ... Позначити з можливістю посилання можна майже будь-який елемент документа.

Варто нагадати, для того, щоб формула була пронумерована, `\label` треба включати в оточення `equation` чи `eqnarray`, а також численні оточення, визначені в пакеті `amsmath: multiline, gather, align`. При цьому ще потрібно не забути вказати мітку, на яку буде здійснено посилання (див. стор. 64).

Також автоматично створюють номери для оточення «теорема». Можна також самостійно створювати команди, що дають автоматичну нумерацію (див. «Створення нових команд»).

Скажемо кілька слів про те, які символи можна використовувати в аргументі команди `\label`. Завжди можна користуватися цифрами і (рядковими і прописними) латинськими літерами; ні в якому разі не можна поміщати в аргумент `\label` фігурні дужки, а також символи `~` (тильда) або `\`. Інші символи в рядку команди `\label` іноді шкідливі, а іноді призводять до синтаксичних помилок. Не ставши T_EX-ніком, краще експерименти не проводити. Можливість використання українських літер в мітках залежить від рівня здійсненої українізації.

Здійснювати посилання можна простим розшуком в файлі потрібної формули і копіюванням аргументу команди `\label`, але є і більш зручний спосіб. На час підготовки документу варто підключити стильової пакет `showkeys`, при цьому над кожним місцем у тексті, позначеним за допомогою команди `\label` (або, скажімо, `\bibitem`), і над кожним місцем, де стоїть посилання – команда `\ref` (або, скажімо, `\cite`), буде наддрукована мітка – аргумент команд: `\label`, `\ref` тощо.

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 49 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

10. Основні принципи набору формул у системі L^AT_EX

При підготовці документів за допомогою T_EX'a, слід розрізняти математичні формули всередині абзацу й «відокремлені» (виділені в окремий рядок). Існує декілька способів набору математичного тексту всередині абзацу [11]:

```
$ математичний текст $
```

```
\( математичний текст \)
```

```
\begin{math} математичний текст \end{math}
```

Виокремлені формули обмежуються парами знаків долара \$\$ теж по обидва боки.

Особливості набору формул

При наведенні даного прикладу набір однієї й тієї ж формули $x^2 + y^2 = a^2$ здійснюється по-різному, тому для неї

$$x^2 + y^2 = a^2$$

виділено окремий рядок.

При наведенні даного прикладу набір однієї й тієї ж формули $x^2+y^2=a^2$ здійснюється по-різному, тому для неї $\$x^2+y^2=a^2\$$ виділено окремий рядок.

Однорядкові виносні формули, як зазначалося і буде зазначено (див. стор. 49, 64), формуються за допомогою оточення `equation`.


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)

[Стор. 50 із 189](#)
[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

При створенні виокремлення формули, розмір шрифту трохи збільшується з метою покращення читабельності. L^AT_EX має кілька стилів для оформлення математично-логічних формул. При бажанні можна вибрати необхідний стиль в ручну:

- `\displaystyle` – для виносних формул,
- `\textstyle` – рядкових формул,
- `\scriptstyle` – для набору індексів,
- `\scriptscriptstyle` – для набору індексів другого рівня.

Формулами вважаються як цілі формули, так і окремі цифри чи букви, у тому числі грецькі, а також верхні й нижні індекси, спецзнаки. Проміжки всередині тексту, що задає формулу, ігноруються, порожні рядки не дозволяються. Розстанівка проміжків у математичних формулах T_EX'ом виконується автоматично. Якщо необхідно залишити проміжок перед або після формули, що знаходиться всередині тексту, то його слід не знищувати перед або після обмежувача її знака долара. Це стосується й розділових знаків, що розташовуються поза формулою. Їх також потрібно ставити після знака долара, що закриває формулу. Кожна буква у формулі розглядається як ім'я змінної й набирається шрифтом «математичний курсив». На відміну від звичайного курсиву, у ньому збільшені відстані між сусідніми буквами.

Особливості набору формул

Переконайтеся *на власні очі!*

Good bye my love, good bye.

```
Переконайтеся {\sl на власні очі!}\\
\begin{center}{\large $Good$ $bye$ my love, {\sl good bye}.}\end{center}
```


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)


Стор. 51 із 189

[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

11. Степені й індекси

Степені й індекси набираються за допомогою знаків $_i^{\wedge}$ відповідно. Черговий приклад.

Особливості набору формул

Якщо за x_1 та x_2 позначити катети прямокутного трикутника, x_3 – гіпотенузу, то вони пов'язані рівністю

$$x_1^2 + x_2^2 = x_3^2,$$

виходячи з теореми Піфагора.

Якщо за x_1 та x_2 позначити катети прямокутного трикутника, x_3 – гіпотенузу, то вони пов'язані рівністю $x_1^2 + x_2^2 = x_3^2$, виходячи з теореми Піфагора.

Якщо індекс або показник степеня – вираз, що містить більше одного символу, то його необхідно взяти у фігурні дужки:

Особливості набору формул

Обчислити значення диференціала функції

$$y = 5^{\sin x \cos 2x},$$

при зміні x від $x = 30^\circ$ до $30^\circ 20'$.

Обчислити значення диференціала функції $y = 5^{\sin x \cos \{2x\}}$, при зміні x від $x = 30^\circ$ до $30^\circ 20'$.


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)

[Стор. 52 із 189](#)
[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

При розташовуванні верхніх і нижніх індексів на різних відстанях від виразу, до якого вони належать, оформлюється частина індексів до «порожньої формули» (пара фігурних дужок {}).

Особливості набору формул

Дійсно, Π_{it}^{jk} і Π_t^{jk} – різні формули.

Дійсно, Π_{it}^{jk} і $\Pi_{i}^{jk}_t$ -- різні формули.

Дроби, що позначаються похилою рисою, набираються безпосередньо.

Особливості набору формул

Знайти частинні похідні третього порядку функції $u = y/x$.

Знайти частинні похідні третього порядку функції $u=y/x$.

При використанні у формулах десяткових дробів, у яких дробова частина відокремлена від цілої за допомогою коми, слід дану кому взяти у фігурні дужки.

Особливості набору формул

У формулі $\pi \approx 3,14$ невеликого проміжку після коми немає тому, що вона знаходиться в фігурних дужках. Не зробивши цього, будемо мати $\pi \approx 3,14$.

У формулі $\pi \approx 3{,}14$ невеликого проміжку після коми немає тому, що вона знаходиться в фігурних дужках. Не зробивши цього, будемо мати $\pi \approx 3,14$.

Дроби, в яких чисельник розташований над знаменником, набираються за допомогою команди `\frac`, див. с. 70.


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)

[Стор. 53 із 189](#)
[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

12. Дужки

Круглі й квадратні дужки набираються звичайно, фігурні – генеруються командами `\{ i \}` (див. с. 34), для інших також є спеціальні команди, наприклад:

`<` – «ліва кутова дужка», `\langle`;

`>` – «права кутова дужка», `\rangle`.

Якщо обмежений дужками фрагмент формули займає багато місця по вертикалі (за рахунок дробів, степенів і т.п.), то й самі дужки повинні бути більшого розміру, чим звичайні. Команда `\left` перед відкриваючою дужкою в сукупності з командою `\right` перед відповідною їй закриваючою дужкою дозволяє автоматично вибрати потрібний розмір дужки.

Особливості набору формул

Формула, що визначає одиничну функцію Хевісайда:

$$\eta(t) = \begin{cases} 0, & t \in (-\infty, 0); \\ 1, & t \in [0, +\infty). \end{cases}$$

Формула, що визначає одиничну функцію Хевісайда: `$$ \eta(t)=\left\{\begin{array}{l} 0, & t \in (-\infty,0); \\ 1, & t \in [0,+\infty). \end{array}\right.` `$$`

Якщо команді `\left` у формулі не задано відповідну команду `\right`, то T_EX видає повідомлення про помилку. Разом із тим T_EX зовсім не вимагає, щоб «обмежувачі» при командах `\left` і `\right` були одного типу. Замість «обмежувача» після команди `\left` або `\right` можна поставити крапку. На місці цієї крапки після трансляції вихідного тексту нічого не надрукується, а



Сайт ПНПУ

Головна

Зміст



Стор. 54 із 189

Назад

Перегляд

Закрити

Вихід

Таблиця 4. Дужки

Дужка	Команда	Дужка	Команда	Дужка	Команда	Дужка	Команда
(({	\{	<	\langle]	\rfloor
))	}	\}	>	\rangle	[\lfloor
[[\]	\rceil	\	\backslash
]]	/	/	[\lceil		
		┌	\ulcorner	┐	\urcorner		

інший «обмежувач» буде необхідного розміру. У тексті прикладу створена ліва фігурна дужка потрібного розміру, без відповідної їй правої завдяки конструкції:

```
\left{ ... \right.
```

Конструкція з `\left` і `\right` застосовна не тільки до круглих дужок. У табл. 4 перераховані дужки та інші символи, які з `\left` і `\right` автоматично приймають потрібний розмір.

Можна указати розмір обмежника явно. Для цього є T_EX'івські команди `\bigl`, `\Bigl`, `\biggl` і `\Biggl` для лівих обмежників і `\bigr`, `\Bigr`, `\biggr` і `\Biggr` для правих обмежників.

Особливості набору формул

$$\left(x - 2 \cdot \left(x^2 + 3 \left(5 - x(x - 7) \right) \right) \right) = 3$$

```
$$\Biggl(x-2\cdot\biggl(x^2+3\Bigl(5-x\bigl(x-7\bigr)\Bigr)\biggr)\Bigg)=3$
```

Перераховано команди в порядку зростання розміру створюваного ними обмежника. Команди явного розміру обмежників, не зобов'язані з'являтися парами, як `\left` і `\right`.


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)


Стор. 55 із 189

[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

Таблиця 5. Різний вигляд набору трьох крапок

№ п/п	Розташування	Команда	Вигляд
1	Унизу рядка	<code>\ldots</code>	...
2	По центру рядка	<code>\cdots</code>	...
3	Вертикально	<code>\vdots</code>	⋮
4	По діагоналі	<code>\ddots</code>	⋱

13. Штрихи й три крапки

Штрихи в формулах позначаються знаком ' і не оформляються як верхні індекси.

Особливості набору формул

Крім задачі Коші, для рівнянь другого порядку можна сформулювати крайову задачу

$$y'' = f(x, y, y'), \quad y(a) = y_0, \quad y(b) = y_1.$$

Крім задачі Коші, для рівнянь другого порядку можна сформулювати крайову задачу
`$$y''=f(x,\,y,\,y'),\; y(a)=y_0,\; y(b)=y_1.$$`

Команди, що використані при наборі формул прикладу: `\,` та `\;` – генерують відступи певного розміру (див. с. 76). Даний приклад висвітлює ще один характерний елемент набору вихідного тексту. У відокремленій формулі наприкінці поставлена кома. Якби вона знаходилась після знаків `$$`, то з неї розпочався б абзац, розташований після формули.

У формулах T_EX розрізняє три крапки так, як представлено в табл. 5. Перші використовується при переліках, другі – коли потрібно замінити пропущені доданки або співмножники, треті та четверті – при наборі матриць розміром $m \times n$ чи визначників порядку n .


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)


Стор. 56 із 189

[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

Можна створювати власні моделі трьох крапок. У наступному прикладі запропоновано діагональну модель, що розміщена під кутом 45°:

Особливості набору формул

Три крапки $\cdot\cdot\cdot$ створені власноруч.

Три крапки $\mathinner{\mkern2mu \raise1pt\hbox{.}\mkern2mu \raise4pt\hbox{.}\mkern2mu \raise7pt\hbox{.}\mkern1mu}$ створені власноруч.

У наступному прикладі при наборі формул матричного числення, використано три крапки з табл. 5.

Особливості набору формул

Визначником n -го порядку називається вираз

$$\begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{vmatrix} = \sum_{k=1}^n a_{ik} A_{ik},$$

де A_{ik} – алгебраїчні доповнення до елементів a_{ik} ; $i = 1, 2, \dots, n$.

Визначником n -го порядку називається вираз $\left\langle \begin{array}{cccc} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{array} \right\rangle = \sum_{k=1}^n a_{ik} A_{ik}$, де A_{ik} – алгебраїчні доповнення до елементів a_{ik} ; $i=1, 2, \dots, n$.


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)


Стор. 57 із 189

[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

14. Таблиці спецзнаків

У даному розділі перелічено всі математичні знаки, надані L^AT_EX'ом. Знаків цих дуже багато, тому їх розбито на кілька груп.

14.1. Операції, відношення й просто значки

Ім'я команди, що задає грецьку букву, збігається з англійською назвою цієї букви (наприклад, буква α задається командою `\alpha`). Виключення становить буква o (вона називається «омикрон»): по накресленню вона збігається з курсивною латинською o , так що спеціальної команди для неї не передбачено, і для її набору досить просто написати o у формулі.

α	<code>\alpha</code>	β	<code>\beta</code>	γ	<code>\gamma</code>	δ	<code>\delta</code>	ϵ	<code>\epsilon</code>
ε	<code>\varepsilon</code>	ζ	<code>\zeta</code>	η	<code>\eta</code>	θ	<code>\theta</code>	ϑ	<code>\vartheta</code>
ι	<code>\iota</code>	κ	<code>\kappa</code>	λ	<code>\lambda</code>	μ	<code>\mu</code>	ν	<code>\nu</code>
ξ	<code>\xi</code>	π	<code>\pi</code>	ϖ	<code>\varpi</code>	ρ	<code>\rho</code>	ϱ	<code>\varrho</code>
σ	<code>\sigma</code>	ς	<code>\varsigma</code>	τ	<code>\tau</code>	υ	<code>\upsilon</code>	ϕ	<code>\phi</code>
φ	<code>\varphi</code>	χ	<code>\chi</code>	ψ	<code>\psi</code>	ω	<code>\omega</code>		

Більшість прописних грецьких букв по накресленню збігаються з латинськими, і для них спеціальних команд немає – треба просто набрати відповідну латинську літеру прямим шрифтом. Прописні грецькі букви, що не збігаються по накресленню з латинськими наведено нижче:

Γ	<code>\Gamma</code>	Δ	<code>\Delta</code>	Θ	<code>\Theta</code>
Λ	<code>\Lambda</code>	Ξ	<code>\Xi</code>	Π	<code>\Pi</code>
Σ	<code>\Sigma</code>	Υ	<code>\Upsilon</code>	Φ	<code>\Phi</code>
Ψ	<code>\Psi</code>	Ω	<code>\Omega</code>		

Особливість набору наведених прописних грецьких букв: ім'я команди пишеться з прописної літери.

Наступні символи визначають бінарні операції (на зразок знаків додавання, множення та


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)


Стор. 58 із 189

[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

інших). Ось їх список:

+	+	-	-	*	\ast
±	\pm	⊕	\uplus	◁	\triangleleft
⊞	\mp	⊐	\sqcap	▷	\triangleright
×	\times	⊔	\sqcup	⊕	\oplus
÷	\div	∨	\vee	⊖	\ominus
*	\star	∧	\wedge	⊗	\otimes
○	\circ	\	\setminus	⊘	\oslash
•	\bullet	ℓ	\wr	⊙	\odot
·	\cdot	◇	\diamond	◯	\bigcirc
∩	\cap	△	\bigtriangleup	†	\dagger
∪	\cup	▽	\bigtriangledown	∏	\amalg

Існують символи «бінарних відношень», навколо яких T_EX залишає додаткові проміжки.

Ці символи подано в таблиці:

<	<	>	>	=	=
≤	\leq	≥	\geq	⋈	\prec
⋈	\succ	⋇	\preceq	⋉	\succeq
⋈	\ll	⋉	\gg	⊂	\subset
⊃	\supset	⊆	\subseteq	⊇	\supseteq
⊆	\sqsubseteq	⊇	\sqsupseteq	∈	\in
∋	\ni	⊥	\vdash	⊥	\dashv
≡	\equiv	≠	\models	~	\sim
⊥	\perp	≈	\simeq		\mid
∞	\asymp	∥	\parallel	≈	\approx
∪	\smile	ℝ	\cong	∪	\frown
≠	\neq	∝	\propto	≐	\doteq

У результаті підключення стильового пакету latexsym (див. с.), отримуємо додаткові мо-


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)


Стор. 59 із 189

[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

жливості:

\triangleleft `\lhd` \triangleright `\rhd` \bowtie `\bowtie` \trianglelefteq `\unlhd` \trianglerighteq `\unrhd`

Стрілки різних видів та відповідні їм команди зібрані у таблиці:

\nearrow	<code>\nearrow</code>	\Uparrow	<code>\upuparrows</code>	\longrightarrow	<code>\longrightarrow</code>
\nwarrow	<code>\nwarrow</code>	\Downarrow	<code>\downdownarrows</code>	\longleftarrow	<code>\longleftarrow</code>
\searrow	<code>\searrow</code>	\rightarrow	<code>\rightarrow</code>	\longleftrightarrow	<code>\longleftrightarrow</code>
\swarrow	<code>\swarrow</code>	\Rightarrow	<code>\Rrightarrow</code>	\circlearrowleft	<code>\circlearrowleft</code>
\leftarrow	<code>\leftarrow</code>	\Leftarrow	<code>\Lleftarrow</code>	\circlearrowright	<code>\circlearrowright</code>
\Lleftarrow	<code>\Lleftarrow</code>	\Rightarrow	<code>\Rrightarrow</code>	\longleftrightarrow	<code>\longleftrightarrow</code>
\mapsto	<code>\mapsto</code>	\leftarrowtail	<code>\leftarrowtail</code>	\rightarrowtail	<code>\rightarrowtail</code>
\multimap	<code>\multimap</code>	\leftrightarrow	<code>\leftrightarrow</code>	\dashrightarrow	<code>\dashrightarrow</code>
\Downarrow	<code>\Downarrow</code>	\Leftrightarrow	<code>\Leftrightarrow</code>	\dashleftarrow	<code>\dashleftarrow</code>
\downarrow	<code>\downarrow</code>	\hookrightarrow	<code>\hookrightarrow</code>	\longmapsto	<code>\longmapsto</code>
\Uparrow	<code>\Uparrow</code>	\leftharpoonup	<code>\leftharpoonup</code>	\leftharpoondown	<code>\leftharpoondown</code>
\uparrow	<code>\uparrow</code>	\leftleftarrows	<code>\leftleftarrows</code>	\leftrightarrows	<code>\leftrightarrows</code>
\Lsh	<code>\Lsh</code>	\updownarrow	<code>\updownarrow</code>	\Longrightarrow	<code>\Longrightarrow</code>
\Rsh	<code>\Rsh</code>	\Updownarrow	<code>\Updownarrow</code>	\Longleftarrow	<code>\Longleftarrow</code>

14.2. Імена функцій, що верстаються прямим шрифтом

Функції на зразок \sin , \log і т.п., імена яких повинні бути надрукованими прямим шрифтом, набираються за допомогою спеціальних команд (зазвичай однойменних із позначеннями відповідних функцій).

Якщо користувачу потрібно в формулу ввести якусь власну функцію набрану прямим шрифтом, то для цього йому потрібно скористатися командою `\text{rm}{...}`.

У черговій таблиці наведено повний список цих функцій з командами для їх відтворення при наборі формул.


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)
[◀](#)
[▶](#)
[◀](#)
[▶](#)

Стор. 60 із 189

[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)



Сайт ПНПУ

Головна

Зміст



Стор. 61 із 189

Назад

Перегляд

Закрити

Вихід

ln	<code>\ln</code>	ker	<code>\ker</code>	tanh	<code>\tanh</code>	max	<code>\max</code>	sin	<code>\sin</code>
lg	<code>\lg</code>	arg	<code>\arg</code>	coth	<code>\coth</code>	det	<code>\det</code>	exp	<code>\exp</code>
log	<code>\log</code>	dim	<code>\dim</code>	sinh	<code>\sinh</code>	lim	<code>\lim</code>	arctan	<code>\arctan</code>
cot	<code>\cot</code>	tan	<code>\tan</code>	cosh	<code>\cosh</code>	inf	<code>\inf</code>	sup	<code>\sup</code>
csc	<code>\csc</code>	cos	<code>\cos</code>	arcsin	<code>\arcsin</code>	lim sup	<code>\limsup</code>	gcd	<code>\gcd</code>
sec	<code>\sec</code>	hom	<code>\hom</code>	arccos	<code>\arccos</code>	min	<code>\min</code>	lim inf	<code>\liminf</code>

У цій таблиці позначення `tan`, `arctan` і т.д. – не що інше, як прийняті в англомовній літературі позначення для тангенса, арктангенса й т.д. У вітчизняній літературі ці функції прийнято позначати `tg`, `arctg` і т.д. Команди для цих функцій необхідно визначати самому. У преамбулі документа треба написати

```
\newcommand{\tt tg}{\mathop{\rm} {\tt tg}}\nolimits}}
```

Для набору символу `mod`, що застосовується при записі порівнянь по модулю, теж використовується прямий шрифт. У таких випадках зручно користуватися командою `\pmod`, яка має досить цінну особливість, автоматично генерує необхідні дужки навколо символу `mod`:

Особливості набору формул

Знайти розв'язок наступної конгруенції $3x \equiv 5 \pmod{7}$.

```
Знайти розв'язок наступної конгруенції $3x\equiv 5\pmod{7}.$
```

У термінології T_EX надписи над і під знаком операції називають «межами». Три рядки попередньої таблиці містять список операцій, для яких «межі» визначаються так само, як індекси. Цей список потрібно поповнити такими:

Pr	<code>\Pr</code>	\odot	<code>\bigodot</code>	\bigcap	<code>\bigcap</code>
\sum	<code>\sum</code>	\oplus	<code>\bigoplus</code>	\bigcup	<code>\bigcup</code>
\int	<code>\int</code>	\otimes	<code>\bigotimes</code>	\bigvee	<code>\bigvee</code>
\oint	<code>\oint</code>	\wedge	<code>\bigwedge</code>	\biguplus	<code>\biguplus</code>
\prod	<code>\prod</code>	\amalg	<code>\coprod</code>		

Межі також можна одержати користуючись командою `\limits`, але кінцеві результати можуть відрізнятися:

Особливості набору формул

Набір формули здійснено для економії місця:

$$\int_0^1 x^3 dx = \frac{1}{4},$$

тому межі знаходяться збоку. Положення меж в інтегралі

$$\int_0^1 x^3 dx = \frac{1}{4}$$

змінено.

Набір формули здійснено для економії місця: `$$\int_0^1 x^3 dx = \frac{1}{4}$$`,
`$$\int\limits_0^1 x^3 dx = \frac{1}{4}$$` тому межі знаходяться збоку. Положення меж в інтегралі
`$$\int\limits_0^1 x^3 dx = \frac{1}{4}$$` змінено.


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)


Стор. 62 із 189

[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

14.3. Різне

Ряд значків, що не входять в жодний із розділів класифікації, але часто зустрічаються, зібрані в наступній таблиці.

$\ $	<code>\ </code>	\hbar	<code>\hbar</code>	\exists	<code>\exists</code>	<code>\exists</code>	<code>\exists</code>
\P	<code>\P</code>	\surd	<code>\surd</code>	∂	<code>\partial</code>	<code>\partial</code>	<code>\partial</code>
\S	<code>\S</code>	\flat	<code>\flat</code>	\natural	<code>\natural</code>	<code>\natural</code>	<code>\natural</code>
\Im	<code>\Im</code>	\ddagger	<code>\ddagger</code>	\triangle	<code>\triangle</code>	<code>\triangle</code>	<code>\triangle</code>
\Re	<code>\Re</code>	∞	<code>\infty</code>	⌘	<code>\maltese</code>	<code>\maltese</code>	<code>\maltese</code>
\wp	<code>\wp</code>	∇	<code>\nabla</code>	\clubsuit	<code>\clubsuit</code>	<code>\clubsuit</code>	<code>\clubsuit</code>
\yen	<code>\yen</code>	\angle	<code>\angle</code>	\emptyset	<code>\emptyset</code>	<code>\emptyset</code>	<code>\emptyset</code>
\top	<code>\top</code>	\aleph	<code>\aleph</code>	®	<code>\circledR</code>	<code>\circledR</code>	<code>\circledR</code>
ℓ	<code>\ell</code>	\prime	<code>\prime</code>	\checkmark	<code>\checkmark</code>	<code>\checkmark</code>	<code>\checkmark</code>
\Box	<code>\Box</code>	\jmath	<code>\jmath</code>	\heartsuit	<code>\heartsuit</code>	<code>\heartsuit</code>	<code>\heartsuit</code>
\bot	<code>\bot</code>	\sharp	<code>\sharp</code>	\backslash	<code>\backslash</code>	<code>\backslash</code>	<code>\backslash</code>
\dagger	<code>\dagger</code>	\imath	<code>\imath</code>	©	<code>\copyright</code>	<code>\copyright</code>	<code>\copyright</code>
\mho	<code>\mho</code>	\forall	<code>\forall</code>	\spadesuit	<code>\spadesuit</code>	<code>\spadesuit</code>	<code>\spadesuit</code>
\neg	<code>\neg</code>	\pounds	<code>\pounds</code>	\diamondsuit	<code>\diamondsuit</code>	<code>\diamondsuit</code>	<code>\diamondsuit</code>

Щоб одержати в математичній формулі зображення перекресленого символу, треба перед командою, що генерує цей символ, поставити команду `\not`.


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)
[⏪](#)
[⏩](#)
[◀](#)
[▶](#)

Стор. 63 із 189

[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

15. Нумерація формул

Нумерувати можна формули, тільки ті, які виділені в окремий рядок. Сама формула повинна бути оформлена як оточення `equation`. Кожна така формула на друці автоматично одержить номер. Щоб на нього можна було посилатися, треба формулу позначити: у будь-якому місці між `\begin{equation}` і `\end{equation}` поставити команду `\label`. Після цього команда `\ref` буде генерувати номер формули. Приклад:

Особливості набору формул

Перша важлива границя

$$\lim_{x \rightarrow 0} \frac{\sin x}{x} = 1 \quad (1)$$

знаходиться під номером (1).

Перша важлива границя `\begin{equation}\label{ww1} \lim_{x \rightarrow 0} \frac{\sin x}{x} = 1 \end{equation}` знаходиться під номером (`\ref{ww1}`).

Дужки навколо номера формули, згенерованого командою `\ref`, автоматично не ставляться. Вигляд номера формули залежить від стилю документа: у стилі `article` (стаття) формули мають суцільну нумерацію, а в стилі `book` (книга) нумерація формул розпочинається заново в кожному розділі. Тому номер формули згенерований оточенням `equation` попереднього прикладу має вигляд (1).

Автоматичну нумерацію можна й не використовувати, її можна здійснити власноруч за допомогою команди `\eqno`. Формулу в даному випадку необхідно оформляти за допомогою знаків `$$`; номером буде слугувати весь текст між `\eqno` і закриваючими формулу `$$` (текст обробляється T_EX'ом так само, як і математичні формули). Для створення нумерації зліва слугує команда `\leqno`:


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)


Стор. 64 із 189

[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

Особливості набору формул

Одна й таж формула

$$S = \iint_D dx dy; \quad (1.23)$$

(1.23)

$$S = \iint_D dx dy,$$

має різносторонню нумерацію.

Одна й таж формула `$$S=\int\limits_{\rule{2mm}{0pt}D} \!\!\!\int dx dy;`
`\eqno (1.23)$$` `$$S=\int\limits_{\rule{2mm}{0pt}D} \!\!\!\int dx dy,\leqno (1.23)$$`
 має різносторонню нумерацію.

У прикладі командами `\rule` (створює лінійку) і `\!` (зменшує відстань між символами) було зроблено потрібне місце положення знака інтеграла та межі інтегрування.

Ніяких автоматичних посилань на формулу, що генерується командами `\eqno` та `\leqno`, T_EX не створює.



[Сайт ПНПУ](#)

[Головна](#)

[Зміст](#)



Стор. 65 із 189

[Назад](#)

[Перегляд](#)

[Закрити](#)

[Вихід](#)

16. Переноси у формулах

При необхідності T_EX може перенести частину внутрішньотекстової формули на інший рядок. Такі переноси можливі після знаків «бінарних відношень», на зразок знака рівності або «бінарних операцій», на зразок знаків додавання або множення причому останній знак у рядку, всупереч вітчизняній традиції, не дублюється на початку наступної. Щоб уникнути таких переносів необхідно формулу або її частину взяти у фігурні дужки. Цей спосіб боротьби з невдалими переносами у формулах має один недолік: при цьому затруднюється верстка абзаців і зростає ймовірність появи неприємних повідомлень «`Overfull \hbox`» (див.). Більш гнучкий спосіб боротьби з переносами у формулах – записати в преамбулі файлу рядок

```
\binoppenalty=10000
```

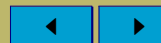
або рядок

```
\relpenalty=10000
```

Перший з цих рядків заборонить всі розриви рядків після знаків бінарних операцій, а другий – після знаків бінарних відношень, і при цьому перешкод верстці абзацу буде менше, ніж при обмеженні всієї формули фігурними дужками.

Якщо всетаки необхідно виконати перенесення формули на інший рядок, то варто пам'ятати, що розрив допустимий у першу чергу – на знаках співвідношень (=, >, ≈ та ін.), у другу – на знаках додавання і віднімання (+, −, ±, ∓), і в останню – на знаках множення множення або ділення (причому знаки повинені бути замінені на такі ×, ÷).

У L^AT_EX перенесення формул на новий рядок можна виконати звичайною командою `\\`, але це не позбавить користувача від необхідності використовувати оточення `eqnarray`.

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 66 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

17. Включення тексту у формули

У математичну формулу за допомогою L^AT_EX'овської команди `\mbox` можна включити фрагмент звичайного тексту. У наступному прикладі продемонстровано, як це можна зробити; у ньому використовується ще команда `\qquad`, що робить у тексті або формулі проміжок розміром 2em.

Особливості набору формул

$$\sqrt{x^3} = x \quad \text{для всіх } x.$$

```
$$\sqrt{x^3}=x \quad \qquad \mbox{для всіх } x.$$
```

Аргумент команди `\mbox` обробляється T_EX'ом як звичайний текст: проміжки не ігноруються, слова набираються не математичним курсивом, а тим же шрифтом. Весь текст, що є аргументом команди `\mbox`, буде надрукований в один рядок. У прикладі залишений проміжок перед закриваючою фігурною дужкою, щоб забезпечити проміжок між текстом і формулою. Команда `\qquad` була використана з аналогічної причини.

Враховуючи те, що L^AT_EX погано розуміє український текст у формулах, є й інший спосіб його туди додати. Тому для використання українського тексту у формулах потрібно підключити пакет `amsmath`, або `mathtext` з опцією `warn` (нагадаємо, це робиться в преамбулі документу командами `\usepackage{amsmath}` та `\usepackage[warn]{mathtext}`).

Особливості набору формул

$$a \xrightarrow{\text{Текст}} b$$

```
$$a\xrightarrow{\text{Текст}}b$$
```


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)

[Стор. 67 із 189](#)
[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

Таблиця 6. Надрядкові знаки

Знак	Команда	Знак	Команда	Знак	Команда	Знак	Команда	Знак	Команда
\tilde{a}	<code>\tilde{a}</code>	\hat{a}	<code>\hat{a}</code>	\grave{a}	<code>\grave{a}</code>	\ddot{a}	<code>\ddot{a}</code>	\breve{a}	<code>\breve{a}</code>
\check{a}	<code>\check{a}</code>	\bar{a}	<code>\bar{a}</code>	\acute{a}	<code>\acute{a}</code>	\dot{a}	<code>\dot{a}</code>	\vec{a}	<code>\vec{a}</code>

18. Надрядкові знаки, звичайні дроби

Якщо потрібно поставити додатковий значок над буквою або фрагментом формули: риску, «кришку», і т.п. У T_EX'ові для цих цілей є спеціальні команди, що перераховані в табл. 6, у якій, для прикладу, ці значки ставляться над буквою *a*. Надрядкові знаки, перераховані в табл. 6, можна ставити лише над окремими буквами: якщо сказати `\hat{a+b}`, то вийде некрасива формула $a \hat{+} b$. T_EX надає можливість поставити «кришку» необхідного розміру над цілим фрагментом формули за допомогою команди `\widehat`.

Особливості набору формул

Тотожність $\widehat{f * g} = \hat{f} \cdot \hat{g}$ означає, що перетворення Фур'є переводить згортку в добуток.

Тотожність `\widehat{f*g}=\hat{f}\cdot\hat{g}` означає, що перетворення Фур'є переводить згортку в добуток.

Існує команда `\overrightarrow`, призначена для постановки стрілки над формулою:

Особливості набору формул

Із теореми маємо: $\overrightarrow{AB} \perp \vec{a}$.

Із теореми маємо: `\overrightarrow{AB} \perp\!\!\!\perp\vec{a}`.


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)


Стор. 68 із 189

[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

Аналогічна їй команда `\overleftarrow` ставить над формулою стрілку, спрямовану вліво, а не вправо. Поставити горизонтальну риску над будь-яким фрагментом формули можна командою `\overline`.

Особливості набору формул

Часто користуються формулою

$$\overline{a_n a_{n-1} \dots a_1 a_0} = \sum_{k=0}^n 10^k a_k,$$

при написанні науково-популярної літератури.

Часто користуються формулою `$$\overline{a_{n-1}\dots a_1a_0}=\sum_{k=0}^n10^ka_k$$` при написанні науково-популярної літератури.

Команди `\overbrace` і `\underbrace` створюють довгі горизонтальні фігурні дужки відразу над або під виразами.

Особливості набору формул

$$\underbrace{5 + 5 + \dots + 5}_{25} = 5 \cdot 25$$

`$$\underbrace{5+5+\cdots+5}_{25}=5\cdot 25$$`

За допомогою команди `\sqrt` набирається квадратний корінь, обов'язковим аргументом якої є підкореневий вираз. Корінь довільного степеня набирається за допомогою тієї ж команди `\sqrt` із необов'язковим аргументом – показником кореня. Необов'язковий аргумент цієї команди ставиться перед обов'язковим. Це виглядає так:



[Сайт ПНПУ](#)

[Головна](#)

[Зміст](#)



Стор. 69 із 189

[Назад](#)

[Перегляд](#)

[Закрити](#)

[Вихід](#)

Особливості набору формул

Знайти інтеграл

$$\int \frac{dx}{\sqrt{x}(1 + \sqrt[3]{x})^2}$$

скориставшись підстановками Чебишева.

Знайти інтеграл $\int \frac{dx}{\sqrt{x}(1 + \sqrt[3]{x})^2}$

скориставшись підстановками Чебишева.

Неважко помітити, що у даному прикладі підінтегральний вираз записано у вигляді звичайного дробу, який отримано на основі команди `\frac`. Її стандартна форма запису така:

`\frac{чисельник}{знаменник}`

Ця команда має два обов'язкові аргументи: перший — чисельник, другий — знаменник. Наступний приклад демонструє взаємоузгодженість у розміщенні чисельників та знаменників двох різних дробів, що генеруються командами `\frac`.

Особливості набору формул

Лінії потоку й потенціалу

$$\varphi_i = \frac{1}{2\pi} \frac{mx_i}{x_1^2 + x_2^2},$$

де $i = 1, 2$, є сім'єю ортогональних кіл.

Лінії потоку й потенціалу $\varphi_i = \frac{1}{2\pi} \frac{mx_i}{x_1^2 + x_2^2}$,

де $i=1,2$, є сім'єю ортогональних кіл.



Сайт ПНПУ

Головна

Зміст



Стор. 70 із 189

Назад

Перегляд

Закрити

Вихід

Окрім команди `\frac{...}{...}` існує й інша можливість `{...}\over{...}`. Варто звернути увагу на те, що результати використання цих команд (хоча їх подано в порядку викладення) будуть однаковими.

$$E = \frac{m_1 v^2}{2} \quad E = \frac{m_1 v^2}{2}$$

Якщо T_EX вибрав для чисельника й знаменника дроби занадто дрібний шрифт (наприклад, коли дріб входить у чисельник або знаменник іншого дроби), то у стильовому пакеті `amsmath` доступні команди `\tfrac` і `\dfrac` (дріб в «текстовому» стилі – `textstyle` та в «виключному» стилі – `displaystyle`), тобто

```
\tfrac{чисельник}{знаменник}
```

```
\dfrac{чисельник}{знаменник}
```

Відмінності у результаті використання команд `\frac` та `\dfrac` наведено в наступних прикладах (їх можна бачити неозброєним оком):

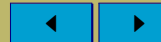
Особливості набору формул

Дріб набрано з використанням команди `\frac`:

$$\frac{x}{x+y}$$

$$x - y$$

Дріб набрано з використанням команди `\verb"\frac"`: `$$\frac{\frac{x}{x+y}}{x-y}$$`


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)


Стор. 71 із 189

[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

Особливості набору формул

Помітьте суттєву різницю! Її отримано завдяки `\dfrac`:

$$\frac{x}{\frac{x+y}{x-y}}$$

Помітьте суттєву різницю! Її отримано завдяки `\verb"\dfrac"`:

`$$\frac{\dfrac{x}{x+y}}{x-y}$$`

Як і у випадку зі звичайною командою `\frac`, можна не брати у фігурні дужки чисельник й/або знаменник, що складається з однієї букви.

У вище зазначеному стильовому пакеті для набору «ланцюгових дробів» (див. також с. 77) є спеціальна команда `\cfrac`. Як нею користуватися, повинно бути зрозуміло з наступного прикладу:

Особливості набору формул

$$\frac{7}{25} = \frac{1}{3 + \frac{1}{1 + \frac{1}{1 + \frac{1}{3}}}}$$

`$$\frac{7}{25}=\cfrac{1}{3+\cfrac{1}{1+\cfrac{1}{1+\cfrac{1}{3}}}}$$`

До речі, якщо необхідно, щоб якийсь із чисельників у ланцюговому дробові був не центрований, а виключений уліво або вправо, то треба замість `\cfrac` вказати команди `\cfrac[l]` або


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)


Стор. 72 із 189

[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

`\cfrac[r]` відповідно. У наступному прикладі свідомо зроблені виключення чисельників дробів для демонстрації дії зазначених команд.

Особливості набору формул

Увага! Лівостороннє та правостороннє виключення чисельників дробів

$$\frac{7}{25} = \frac{1}{3 + \frac{1}{1 + \frac{1}{1 + \frac{1}{3}}}}$$

```
\textbf{Увага!} \emph{Лівостороннє та правостороннє виключення
чисельників дробів}$$\frac{7}{25}=\frac{1}{3+\cfrac[1]{1}{1+ \cfrac[r]{1}{1+
\cfrac{1}{3}}}}$$
```

Для набору узагальненого дробу передбачена команда `\genfrac` із шістьма аргументами. Перший і другий аргументи цієї команди – це лівий і правий обмежники відповідно; третій аргумент – товщина дробової риски (якщо товщина нульова, то дробова риска не друкується); четвертий аргумент містить вказівки щодо розміру шрифту для чисельника й знаменника (якщо залишити його порожнім, написавши просто {}, замість {0} чи {1}, то T_EX вибере розмір самостійно; цифра 0 означає `displaystyle`, цифра 1 – `textstyle`); нарешті, п'ятий і шостий аргументи – це властиво чисельник і знаменник.

Якщо залишити третій аргумент порожнім, написавши просто {} замість фігурних дужок, у яких записана товщина, то буде обрана товщина дробової риски за замовчуванням (вона дорівнює 0.4 pt). Якщо залишити перший і другий аргументи порожніми, то обмежників не буде. При зазначенні лівого обмежника повинен бути зазначений і правий! Наприклад:


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)

[Стор. 73 із 189](#)
[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

Особливості набору формул

$\frac{x}{y}$ – це те ж саме, що $\frac{x}{y}$.

`$\frac{x}{y}$` -- це те ж саме, що `$\frac{x}{y}$` .

Звичайно, команда `\genfrac` гарна не сама по собі, а як сировина для визначення макросів, пристосованих до конкретних потреб.

Команди, що дозволяють ставити значки необхідного розміру над фрагментом формули (нехай це три букви *abc*), можна узагальнити у вигляді таблиці:

\widetilde{abc}	<code>\widetilde{abc}</code>	\overleftarrow{abc}	<code>\overleftarrow{abc}</code>
\widehat{abc}	<code>\widehat{abc}</code>	\overrightarrow{abc}	<code>\overrightarrow{abc}</code>
\underbrace{abc}	<code>\underbrace{abc}</code>	\sqrt{abc}	<code>\sqrt{abc}</code>
\underline{abc}	<code>\underline{abc}</code>	$\sqrt[n]{abc}$	<code>\sqrt[n]{abc}</code>
\overbrace{abc}	<code>\overbrace{abc}</code>	$\frac{x}{abc}$	<code>\frac{x}{abc}</code>
\overline{abc}	<code>\overline{abc}</code>		


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)


Стор. 74 із 189

[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

19. Набір матриць

Щоб набрати за допомогою L^AT_EX'a матрицю, треба скористатися оточенням `array`. Будь-яка матриця складається з рядків і стовпців. Рядки матриці розділяються за допомогою команди `\\` (останній рядок закінчувати командою `\\` не потрібно), а елементи всередині одного рядка, що відносяться до різних стовпців, відокремлюються один від одного за допомогою символу `&`. Потім, після `\begin{array}`, що відкриває оточення, необхідно написати преамбулу матриці, що задає кількість і тип стовпців матриці. Якщо преамбула являє собою чотири букви `cccc`. Це означає, що в матриці 4 стовпці (по літері на стовпець), і що вміст кожного з них буде розташований по центру стовпця (`c` – від слова «centered»). Крім `c`, у преамбулі може стояти буква `l`, що дозволяє відповідний стовпець вирівняти по лівому краю (`left`), або `r` – стовпець буде вирівняний по правому краю (`right`).

Для створення дужок матриці слід написати `\left(` перед `\begin{array}` і `\right)` після `\end{array}`. Внутрішня частина матриці аналогічна визначнику приклад якого наведено на с. 56.

Матриці з різними типами оточуючих дужок.

$$\begin{matrix} 1 & 0 & \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} & \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} & \begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix} & \left\| \begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix} \right\| \end{matrix}$$

```
[\begin{matrix}1&0 \\ 0&1 \end{matrix} \begin{pmatrix} 1&0 \\ 0&1 \end{pmatrix} \begin{bmatrix} 1&0 \\ 0&1 \end{bmatrix} \begin{vmatrix} 1&0 \\ 0&1 \end{vmatrix} \left\| \begin{matrix} 1&0 \\ 0&1 \end{matrix} \right\|]
```

При наборі матриць часто виникає необхідність застосовувати три крапки певного типу. Способи їх задання розглянуто на тій же с. 56.


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)

[Стор. 75 із 189](#)
[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

20. Проміжки в математичній моді

Якщо проміжки між символами у формулах, вибрані T_EX'ом автоматично, виглядають невдало, то у формулу можна включити команди, що задають проміжки в явному виді.

[\!	«Від'ємний тонкий проміжок»
]	\,	«Тонкий проміжок»
[\:	«Середній проміжок»
]	\;	«Товстий проміжок»
]	[\quad	Проміжок у 1em
]	[\qquad	Проміжок у 2em

У наступному прикладі зібрані типові випадки використання даних команд.

Особливості набору формул

Зверніть увагу на символ потрійного інтеграла

$$\iiint f(x, y, z)dv.$$

Завдяки команді \! відстані між знаками інтегралів різні.

Зверніть увагу на символ потрійного інтеграла $\iiint f(x, y, z)dv$.

Завдяки команді \verb"!" відстані між знаками інтегралів різні.

Команди \quad та \qquad часто застосовуються для відокремлення тексту, що входить до складу формули.

У більшості випадків T_EX автоматично вибирає розмір символів у формулах. Якщо слід змінити стиль набору формули, його можна вказати в явному вигляді за допомогою T_EX'овських команд:


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)

[Стор. 76 із 189](#)
[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)



Сайт ПНПУ

Головна

Зміст



Стор. 77 із 189

Назад

Перегляд

Закрити

Вихід

`displaystyle` «виключний» стиль;
`textstyle` «текстовий» стиль;
`scriptstyle` стиль для індексів;
`scriptscriptstyle` стиль для індексів до індексів.

«Виключний» і «текстовий» стилі використовують однакові шрифти, але формули в текстовому стилі виглядають «скромніше». Наприклад:

— у виключному стилі верхні індекси піднімаються вище, а нижні — опускаються нижче, ніж у текстовому;

— у текстовому стилі «межі» операцій записуються не зверху, а збоку;

— у стилі для індексів використовуються менші шрифти, ніж у виключному або текстовому, а в стилі індексів до індексів — ще дрібніші.

Вибір стилів набору формул такий:

- виключна формула набирається у виключному стилі, а внутрітекстова — у текстовому;
- у момент дії якогось зі стилів при зустрічі `\frac` (або `\atop`), для набору чисельника й знаменника T_EX перемикається на наступний по черзі стиль із вищенаведених у таблиці;
- якщо в момент дії виключного чи текстового стилю зустрінеться верхній або нижній індекс, то він починає набиратися стилем для індексів;
- при дії стилю для індексів або індексів до індексів, індекс набирається в стилі «індекси до індексів».

Стилю `scriptscriptscriptstyle` і подальших не передбачено, тому індекси третього й вищих порядків набираються тими ж шрифтами, що й індекси другого порядку. Автоматичний контроль перемикання стилів не завжди може задовольняти вимоги користувача. У такому випадку для зміни вигляду формул стилі необхідно вказувати за допомогою T_EX'овських команд. Імена цих команд співпадають із відповідними англійськими назвами стилів `\displaystyle ... \scriptscriptstyle`. Наприклад, при запису «ланцюгових дробів» маємо:

Особливості набору формул

Бажаний результат важко отримати не здійснюючи зміну стилів у формулі:

$$\frac{7}{25} = \frac{1}{3 + \frac{1}{1 + \frac{1}{1 + \frac{1}{3}}}}.$$

Бажаний результат важко отримати не здійснюючи зміну стилів у формулі: `\[\frac{7}{25}=\frac{1}{3+\frac{1}{1+\frac{1}{1+\frac{1}{3}}}}.\]`

Вигляд формули можна поліпшити доопрацювавши її безпосередню генерацію:

Особливості набору формул

Здійснивши зміну стилів у формулі, отримуємо результат

$$\frac{7}{25} = \frac{1}{3 + \frac{1}{1 + \frac{1}{1 + \frac{1}{3}}}}.$$

Здійснивши зміну стилів у формулі, отримуємо результат `\[\frac{7}{25}=\frac{1}{\displaystyle 3+\frac{1}{\displaystyle 1+\frac{1}{\displaystyle 1+\frac{1}{3}}}}.\]`

Слід відмітити, що бажаний результат можна отримати використовуючи команду `\cfrac` стильового пакету `amsmath` (див. с. 72).


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)

[Стор. 78 із 189](#)
[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

21. Додаткові проміжки навколо формул

При виникненні необхідності можна зробити, щоб кожна формула була оточена додатковими проміжками праворуч і ліворуч від неї. Для цього в T_EX'ові передбачений параметр `\mathsurround`. Значення цього параметру – розмір додаткового проміжку, що ставиться по обидві сторони від кожної внутрітекстової математичної формули. При запуску T_EX'а значення цього параметра дорівнює нулю, так що відстані між формулами й оточуючим текстом такі ж, як і між словами в тексті. Однак, параметру `\mathsurround` можна присвоїти значення ненульової довжини. Наприклад, якщо написати в преамбулі

```
mathsurround=2pt
```

то кожна формула буде оточена додатковими проміжками по 2 пункти з обох сторін. [10]

Якщо потрібно зробити додаткові горизонтальні відбиття навколо однієї певної формули, то необхідно помістити `\mathsurround` із потрібним значенням в саму формулу між знаки, що її обмежують.

Коли вбудований режим математики використовується для цілей, відмінних від вставки формули в текст, то це хороша практика, щоб встановити `\mathsurround` рівним нулю для уникнення паразитних просторів.

За розмір вертикальних відступів навколо формул L^AT_EX відповідають наступні команди:

`\abovedisplayskip` – вертикальний інтервал перед формулою після довгого рядка;

`\abovedisplayshortskip` – вертикальний інтервал перед формулою після короткого рядка;

`\belowdisplayskip` – вертикальний інтервал після формули перед довгим рядком;

`\belowdisplayshortskip` – вертикальний інтервал після формули перед коротким рядком.

Наголосимо ще раз, змінювати розміри вертикальних відступів можна двома способами: пропорційно існуючим, задаючи конкретні значення.


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)

[Стор. 79 із 189](#)
[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

22. Обробка помилок

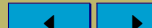
У вихідних T_EX'овських текстах, які потрібно підготувати, неминуче будуть присутні помилки. Усі повідомлення, які T_EX видає на екран у процесі трансляції вихідного тексту, усі відповіді на ці повідомлення, взагалі все, що в процесі трансляції з'являється на екрані, записується в спеціальний файл – протокол трансляції. Зазвичай файл-протокол має те ж ім'я, що й оброблюваний T_EX'ом файл, і розширення log, тому протокол трансляції називається log-файлом. При завершенні трансляції log-файл можна переглянути й проаналізувати, що відбулося.

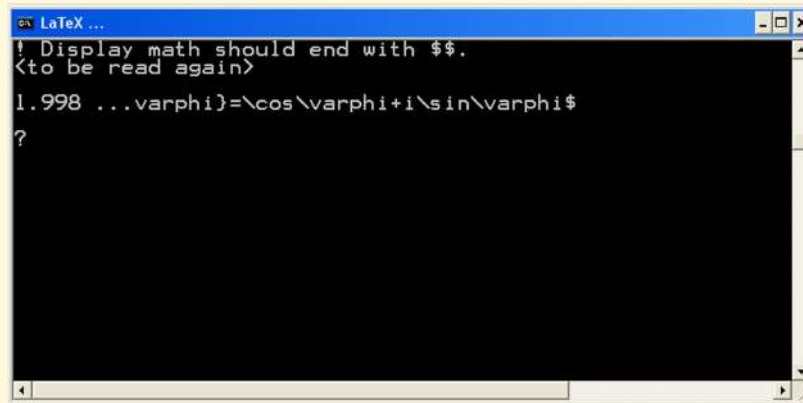
Частина інформації, що видається при трансляції на екран і в log-файл, являє собою попередження, при видачі яких трансляція не переривається. Однак, у випадку виявлення T_EX'ом синтаксичної помилки у вихідному тексті, трансляція припиняється, а на екран виводиться повідомлення про помилку. Щоб проілюструвати сказане на прикладі, при наборі наступної формули свідомо зроблено помилку – не набрано \$:

```
$$\exp^{i\varphi}=\cos\varphi+i\sin\varphi.$
```

При трансляції матимемо повідомлення представлене на рис. 4

Натискати «Enter» у відповідь на повідомлення про помилку — найпоширеніша на практиці реакція в 90% випадків і цього цілком достатньо. Якщо прийняти рішення про постійне натискування на клавішу «Enter» у відповідь на всі повідомлення про помилки, то можна після появи на екрані першого ж із цих повідомлень натиснути на S, або s, а потім на «Enter». При виявленні подальших помилок трансляція перериватися не буде. На моніторі пронесуться повідомлення про неточності, після чого можна їх вивчити, переглянувши log-файл. Для того щоб внести виправлення із клавіатури, слід натиснути і або I, а потім «Enter». На екрані з'явиться таке запрошення:

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 80 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)



```
LaTeX ...
! Display math should end with $$
<to be read again>
1.998 ...varphi)=\cos\varphi+i\sin\varphi$
?
```

Рис. 4. Повідомлення про помилку

```
insert>
```

У відповідь на це запрошення варто ввести той текст або команди, які потрібно вставити в текст, що читається T_EX'ом. Можуть виникати нетрадиційні способи реакції на помилки. По-перше, у відповідь на запрошення ? можна набрати **h** або **H** і натиснути «Enter». У цьому випадку T_EX видасть на екран додаткову інформацію із приводу допущеної помилки, а потім ще раз запрошення ?. По-друге, можна набрати **r** або **R** і «Enter». Результат буде такий же, як і при натисканні на клавішу **s**, із тією різницею, що якщо аргументом команди `\input` слугує неіснуючий файл, то не виникне жодних питань, а трансляція просто перерветься. Нарешті, можна набрати **Q** або **q** і «Enter». Результат аналогічний натисканню **R**, лише на екрані не буде ніякого повідомлення (все запишеться в log-файлі).

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[◀](#) [▶](#)[◀](#) [▶](#)

Стор. 81 із 189

[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

23. Спеціальні та діактричні знаки

Більшість розділових знаків (крапка, кома, двокрапка та інші) набираються звичайним чином. Наприклад, крапці у вихідному тексті відповідає типографська крапка на друкові.

Щоб при друку одержати дефіс, коротке чи довге тире, потрібно у вихідному тексті набрати один, два або три знаки - відповідно. В українському й російському текстах рекомендується використовувати довге тире в якості звичайного, а коротке тире — у сполученнях типу «я повернуся через 2--3 години». У вихідному тексті це виглядає так: «я повернуся через 2--3 години»; пробіли навколо тире відсутні. Довге тире, за прийнятими типографськими правилами, напаци, необхідно оточувати проміжками по обидва боки.

В англійських текстах лапки, що відкриваються, зображуються у вихідному тексті двома підряд зворотними апострофами, а ті, що закриваються – двома звичайними.

The “definitions” are translations rather than explanations.

The “definitions” are translations rather than explanations.

У вітчизняних текстах використовуються лапки типу «ялинка» і „лапки“. Стандартний комплект T_EX'овських шрифтів може їх і не включати. У русифікованому L^AT_EX'ові «ялинка» задаються командами (з іменами з російських букв!) `\лк` і `\пк`, а „лапки“ – командами `\glqq` і `\grqq` (після команди `\grqq` слід ставити «пусту групу» `{}`, щоб між лапками і наступним словом був проміжок).

Якщо в тексті зустрічаються лапки всередині лапок, то, згідно типографських правил, внутрішні лапки повинні відрізнятися від зовнішніх. Можна, наприклад, зовні поставити «ялинка», а всередині „лапки“.

Для набору трьох крапок є спеціальні команди `\ldots` (див. с. 56) та `\dots`.

Замість «...» пишемо: ні, щось тут не так ...


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)


Стор. 82 із 189

[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

Замість «...» пишемо: ні, щось тут не так `\dots`

Підкреслення тексту здійснюється за допомогою команди `\underline` (див. розділ 18, с. 74), що має один обов'язковий аргумент – підкреслюваний текст:

Зверніть увагу як отримано підкреслення

Останнє слово буде `\underline{підкреслено}`.
`\underline{підкреслено}` буде підкреслено.

Підкреслений текст повинен розміщуватися в одному рядку.

Щоб взяти частину тексту в рамку, використовується команда `\fbox`:

Слово буде знаходитися `\fbox{в рамці}`. Слово буде знаходитися `\fbox{в рамці.}`

Команда `\fbox` дозволяє взяти в рамку тільки фрагмент тексту, що вміщається в один рядок.

У багатьох мовах використовуються букви з додатковими значками, що розміщуються над або під буквою. Вони називаються діакритичними знаками. У табл. 7 наведені команди для набору букв спеціального виду.

Наступна таблиця (див. табл. 8) містить команди необхідні для набору букв різних алфавітів.

Слід зауважити, що команди `\i` та `\j` цієї таблиці потрібні для того, щоб ставити діакритичні знаки над буквами *i* та *j*.

Якщо набрати <code>\=i</code> , то матимемо \bar{i} , хоча це не те, що потрібно. Правильно писати <code>\=i</code> , щоб одержати \bar{i} .	Якщо набрати <code>\verb"\=i</code> , то матимемо <code>\=i</code> , хоча це не те, що потрібно. Правильно писати <code>\verb"\=i</code> , щоб одержати <code>\=i</code> .
--------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------

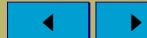
Для набору звичайних російських текстів фактично потрібні тільки дві команди: `\'` (щоб ставити наголоси) та `\"` (для букви *ё*).



Сайт ПНПУ

Головна

Зміст



Стор. 83 із 189

Назад

Перегляд

Закрити

Вихід

Таблиця 7. Діакритичні знаки

Набрано	Отримано	Набрано	Отримано
<code>\`{e}</code>	è	<code>\u{e}</code>	ě
<code>\' {e}</code>	é	<code>\v{e}</code>	ě
<code>\~{e}</code>	ê	<code>\H{e}</code>	ě
<code>\e"</code>	ë	<code>\c{e}</code>	ę
<code>\~{e}</code>	ẽ	<code>\d{e}</code>	ę
<code>\={e}</code>	ē	<code>\b{e}</code>	ë
<code>\. {e}</code>	è	<code>\t{e}</code>	ë

Таблиця 8. Букви різних алфавітів

Набрано	Отримано	Набрано	Отримано
<code>\oe</code>	œ	<code>\OE</code>	Œ
<code>\ae</code>	æ	<code>\AE</code>	Æ
<code>\aa</code>	å	<code>\AA</code>	Å
<code>\o</code>	ø	<code>\O</code>	Ø
<code>\l</code>	ł	<code>\L</code>	Ł
<code>\i</code>	ı	<code>\j</code>	Ј
<code>\ss</code>	ß		


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)


Стор. 84 із 189

[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

24. Проміжки між словами та реченнями

Проміжки між словами T_EX виставляє автоматично, дотримуючись принципу мінімального розміру. Часто потрібно, щоб два сусідніх слова не потрапили на різні рядки, тому необхідно між ними вставити «символ нерозривного пробілу» `~`. Так як, згідно вітчизняним поліграфічним правилам, рядок не повинен починатися з тире, при потребі усунення подібної проблеми використовується `~`.

Проміжки між реченнями самі по собі є ширшими і довшими, ніж між словами всередині речення. В українських текстах проміжки між словами й реченнями відрізнятися не повинні. Для цього варто включити в преамбулу команду `\frenchspacing`. Якщо серед українського тексту зустрічається фрагмент, написаний англійською мовою, то командою `\nonfrenchspacing` можна відновити дію англійського правила відносно проміжків між словами та реченнями, що відрізняються за довжиною. По закінченню англійського тексту, слід повернути дію українського правила вище зазначеною командою.

Для встановлення проміжків власноруч можна скористатися командами `\enskip` (створює проміжок у два рази менший, ніж команда `\quad`), та `\hspace*`. За допомогою двох останніх команд, вказавши їх обов'язковий аргумент (довжину)

```
\hspace{довжина};  
\hspace*{довжина},
```

можна встановлювати проміжки певної величини. Зокрема, команда `\hspace*` зберігає відповідний проміжок на початку (в кінці) рядка.

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 85 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

25. Шрифти та їх зміна в тексті

Зрозуміло, що при роботі з текстом для виділення тих або інших його фрагментів необхідні команди оперативної зміни шрифтів. Всі ці команди формалізовані так, що користувач не має потреби в інформації, де і як зберігаються шрифти і як здійснюється доступ до них.

Основні групи команд роботи зі шрифтами:

- команди, що декларують типи (накреслення) й розміри нематематичних шрифтів;
- команди уведення математичних символів (див. розділ 10).

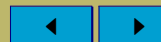
Параметри команд задання розмірів шрифту залежать від вибору нормального шрифту документа, і обчислюються у відповідних файлах опису стилю, а команди декларації типу шрифту залежать від конфігурації схеми вибору шрифтів, що була прийнята при поточній компіляції системи.

Нормальний шрифт документу, призначений для виведення основного тексту, вибирається за замовчуванням або вказівкою певного файлу коригуючого стилю: 10pt, 11pt або 12pt (за замовчуванням вибирається стиль 10pt). Це завжди прямий шрифт. Розмір нормального шрифту документа є найважливішою характеристикою, що визначає вибір шрифтів для заголовків рубрик різного рівня, виноска і інших стандартних елементів публікації (у табл. 9 перераховані абсолютні розміри шрифтів у стандартних класах).

Властиво розмір шрифту зазначений над похилою рисою, а під нею наведена відстань між рядками з вказівкою одиниць розмірності, що повинна з'являтися в команді `\size` (див. с. 95).

До стандартного комплекту L^AT_EX'а входять накреслення й розміри шрифтів, що подані у табл. 10 та табл. 11 разом із командами, що задають перемикання на них.

Нормальний розмір шрифту 14pt не є стандартним. Задання опції у вигляді короткої форми (14pt) відповідного файлу, що входить до списку команди `\documentclass` при стандартних базових стилях неможливе, варто вказувати повну назву файлу, `art14`.

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 86 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)



Таблиця 9. Абсолютні розміри шрифтів в стандартних класах.

Команда L ^A T _E X'a	Нормальний розмір шрифту			
	10pt	11pt	12pt	14pt
<code>\tiny</code>	5/6pt	6/7pt	6/7pt	7/8.4pt
<code>\scriptsize</code>	7/8pt	8/9.5pt	8/9.5pt	9/11pt
<code>\footnotesize</code>	8/9.5pt	9/11pt	10/12pt	11/13.6pt
<code>\small</code>	9/11pt	10/12pt	11/13.6pt	12/14.5pt
<code>\normalsize</code>	10/12pt	11/13.6pt	12/14.5pt	14/18pt
<code>\large</code>	12/14pt	12/14pt	14/18pt	17/20pt
<code>\Large</code>	14/18pt	14/18pt	17/22pt	20/25pt
<code>\LARGE</code>	17/22pt	17/22pt	20/25pt	25/30pt
<code>\huge</code>	20/25pt	20/25pt	25/30pt	–
<code>\Huge</code>	25/30pt	25/30pt	–	–

Сайт ПНПУ

Головна

Зміст

Стор. 87 із 189

Назад

Перегляд

Закрити

Вихід

Таблиця 10. Накреслення шрифтів

Команда	Назва накреслення
<code>\bf</code>	напівжирний шрифт (boldface)
<code>\it</code>	<i>курсив</i> (italic)
<code>\sl</code>	<i>похилий шрифт</i> (slanted)
<code>\sf</code>	рубаний шрифт (sans serif)
<code>\sc</code>	КАПТЕЛЬ (SMALL CAPS)
<code>\tt</code>	імітація друкарської машинки (typewriter)
<code>\rm</code>	прямий світлий шрифт (roman)

Будучи введеною в текст, команда задання розміру чи типу шрифту буде діяти до кінця документа або до тих пір, поки не буде вказано нової аналогічної команди. Для набору обраним шрифтом фрагмента тексту варто створити групу (фігурні дужки; див. підрозділ 7 с. 41) до якої входять, і команда вибору шрифту, і сам фрагмент (команда вибору шрифту діє тільки в межах групи).

Важлива особливість L^AT_EX 2_ε полягає в тім, що атрибути шрифту незалежні. Це означає, що можна давати команди зміни розміру або навіть сімейства шрифту, зберігаючи при цьому установки атрибутів нахилу або насиченості.

У математичному режимі для перемикання на інший шрифт існує окремий набір команд (див. табл. 12). При використанні цього набору непотрібно тимчасово виходити із математичного режиму для введення звичайного тексту певним шрифтом.

У роботі з командами зміни розміру шрифту, часто трапляються помилки. Якщо набраний цілий абзац шрифтом зміненого розміру (скажімо, `\small` або `\footnotesize`), то в той момент, коли T_EX бачить порожній рядок або команду `\par`, цей шрифт ще не повинен бути переключений на звичайний, інакше інтервали між рядками будуть неправильними.



Сайт ПНПУ

Головна

Зміст



Стор. 88 із 189

Назад

Перегляд

Закрити

Вихід

Таблиця 11. Розміри шрифтів

Команда	Назва розміру
<code>\tiny</code>	Мікроскопічний
<code>\scriptsize</code>	Дуже маленький (як індекси)
<code>\footnotesize</code>	Маленький (як посилання)
<code>\small</code>	Малий
<code>\normalsize</code>	Нормальний
<code>\large</code>	Великий
<code>\Large</code>	Зовсім великий
<code>\LARGE</code>	Занадто великий
<code>\huge</code>	Величезний
<code>\Huge</code>	Грандіозний

Таблиця 12. Математичні шрифти

Команда	Приклад	
	До компіляції	Після компіляції
<code>\mathbf{...}</code>	<code>\$\$\sum x=\mathbf{V}\$\$</code>	$\sum x = \mathbf{V}$
<code>\mathit{...}</code>	<code>\$\$\mathit{ffi}\neq ffi\$\$</code>	$ffi \neq ffi$
<code>\mathcal{...}</code>	<code>\$\$\mathcal{A}=\left(a_{ij}\right)\$\$</code>	$\mathcal{A} = (a_{ij})$
<code>\mathsf{...}</code>	<code>\$\$\mathsf{G\times R}\$\$</code>	$G \times R$
<code>\mathnormal{...}</code>	<code>\$\$\mathnormal{Q^{21}}\neq Q^{21}\$\$</code>	$Q^{19} \neq Q^{19}$
<code>\mathtt{...}</code>	<code>\$\$\mathtt{L}(x,y)\$\$</code>	$L(x, y)$
<code>\mathrm{...}</code>	<code>\$\$\mathrm{K}_{2005}\$\$</code>	K_{2005}



Сайт ПНПУ

Головна

Зміст



Стор. 89 із 189

Назад

Перегляд

Закрити

Вихід

У даному випадку, групу непотрібно закривати перед порожнім рядком, бо матимемо небажаний результат.

Шрифт став нормального розміру.

```
{\footnotesize У даному випадку, групу
непотрібно закривати перед порожнім
рядком, бо матимемо небажаний результат.
```

```
}
Шрифт став нормального розміру.
```

Якщо, все таки, закрити групу перед порожнім рядком, то неправильність інтервалів забезпечена.

Шрифт всеодно став нормального розміру.

```
{\footnotesize Якщо, все таки, закрити
групу перед порожнім рядком, то
неправельність інтервалів забезпечена.}
```

```
Шрифт всеодно став нормального розміру.
```

Цей ефект можливий тому, що визначення розміру між рядками здійснюється командами перемикаючими розмір шрифту на заключному етапі верстки абзацу.

Таблиця 13 теж містить команди зміни шрифтів у тексті, але вони використовуються лише в L^AT_EX 2_ε.

L^AT_EX і L^AT_EX надають можливості для вибору й завантаження будь-якого шрифту чи його символу, дозволяючи користувачеві працювати з усією колекцією шрифтів, але для цього необхідно знати, як улаштована бібліотека шрифтів.

25.1. Класифікація шрифтів

Шрифти поєднуються в сімейства – аналоги гарнітур класичної поліграфії.

Кожне сімейство має коротку назву, що використовується в командах опису й завантаження шрифтів. Наприклад, `cmr` — назва сімейства шрифтів Computer Modern Roman, (розробленого



Сайт ПНПУ

Головна

Зміст



Стор. 90 із 189

Назад

Перегляд

Закрити

Вихід

Таблиця 13. Команди зміни шрифтів у тексті.

Без аргументу:	Із аргументом:	На друкові маємо:
<code>\rmfamily Шрифт</code>	<code>\textrm{Шрифт}</code>	Шрифт
<code>\sffamily Шрифт</code>	<code>\textsf{Шрифт}</code>	Шрифт
<code>\ttfamily Шрифт</code>	<code>\texttt{Шрифт}</code>	Шрифт
<code>\mdseries Шрифт</code>	<code>\textmd{Шрифт}</code>	Шрифт
<code>\bfseries Шрифт</code>	<code>\textbf{Шрифт}</code>	Шрифт
<code>\upshape Шрифт</code>	<code>\textup{Шрифт}</code>	Шрифт
<code>\itshape Шрифт</code>	<code>\textit{Шрифт}</code>	<i>Шрифт</i>
<code>\slshape Шрифт</code>	<code>\textsl{Шрифт}</code>	<i>Шрифт</i>
<code>\scshape Шрифт</code>	<code>\textsc{Шрифт}</code>	ШРИФТ

Д.Кнотом).

До найчастіше використовуваних сімейств при підготовці публікацій, входять шрифти різних видів – нормальні, жирні, курсивні й деякі інші, і кожний вид представлений набором шрифтів різних розмірів.

Слід зауважити, що сімейство може складатися й тільки з одного шрифту, наприклад, сімейства екзотичних шрифтів.

- Текстові шрифти Computer Modern.


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)


Стор. 91 із 189

[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

<code>cmr</code>	Computer Modern Roman	Сімейство шрифтів, базове для більшості підготовлюваних на T _E X'ові документів.
<code>cmcyr</code>	Computer Modern Cyrillic	Підмножина кириличних шрифтів у сімействі Computer Modern Roman.
<code>cmtt</code>	Computer Modern Typewriter	Сімейство рівних по ширині шрифтів.
<code>cmctt</code>	Computer Modern Cyrillic Typewriter	Підмножина кириличних шрифтів у сімействі <code>cmtt</code>
<code>cmss</code>	Computer Modern Sans Serif	Сімейство рубаних (без зарубок) шрифтів.
<code>cmcss</code>	Computer Modern Cyrillic Sans Serif	Підмножина кириличних шрифтів у сімействі рубаних шрифтів.

- **Математичні шрифти Computer Modern.**

<code>cmi</code>	Math italic	Шрифти з буквами грецького й латинського алфавітів курсивного накреслення, цифрами й деякими спеціальними символами.
<code>cmsy</code>	Math symbols	Шрифти з математичними символами.

- **Екзотичні шрифти Computer Modern.**

<code>cmf</code>	Funny i Funny Italic.
<code>cmtex</code>	T _E X Extended ASCII Type.
<code>cmdunh</code>	Dunhill Roman Type.

- **AMS-T_EX шрифти.**

- Euler – три родинних сімейства шрифтів. Ці шрифти розробив Германн Запф, вони належать Станфордському університету як частина T_EX-проекту.


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)

[Стор. 92 із 189](#)
[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

euf	Euler Fraktur (German)	Містить букви й інші символи готичного накреслення.
eur	Euler Roman (cursive)	Містить букви грецького й латинського алфавітів прямого курсивного накреслення.
eus	Euler Script	Містить букви латинського алфавіту каліграфічного накреслення.

– Шрифти з математичними символами.

msa AMS symbol font A.

msb AMS symbol font B.

– WNCyr Cyrillic fonts — сімейство шрифтів кириличного алфавіту. Розроблено в Університеті штату Вашингтон (м.Сіетл). Містять практично повний алфавіт для більшості слов'янських мов.

Ім'я шрифту, що належить до певного сімейства, ідентифікує його щодо інших шрифтів, наявних у бібліотеці. Воно будується за форматом: перші дві – чотири букви являють собою абrevіатуру сімейства, потім ідуть букви, що вказують на накреслення й форму конкретного шрифту. Цифри наприкінці – це розмір у типографських пунктах. Наприклад, ім'я шрифту `cmtt12` є абrevіатурою для «Computer Modern Typewriter Type розміром в 12 пунктів».

25.2. Найпростіші команди вибору шрифту й окремого символу

Найпростіший формат команди T_EX'а завантаження шрифту має вигляд

```
\font\ім'я = необхідний шрифт
```

де `\ім'я` – нова команда, асоційована з даним шрифтом, а параметр `необхідний шрифт` – бібліотечна назва шрифту.

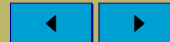
Завантаження шрифту виконується по команді L^AT_EX'а



Сайт ПНПУ

Головна

Зміст



Стор. 93 із 189

Назад

Перегляд

Закрити

Вихід

```
\newfont{\ім'я}{ім'я шрифту}
```

У ній визначається ім'я команди для вибору шрифту, бібліотечну назву якого зазначено в другому параметрі.

Друк окремого символу поточного шрифту виконується по команді T_EX'а або L^AT_EX'а:

```
\charномер
```

```
\symbol{номер}
```

де номер є числовий код, що визначається за таблицями. У наступному прикладі висвітлена реалізація вище перерахованих команд.

Команди вибору шрифта

```
Insert 12345 \LaTeX Label 12345 \LaTeX ABC 12345 LATEX dvips ΔΘΛΠ @=#
```

```
\font\eufrb=eufb8 \font\eufrb=eufm10 \font\eufrb=eufb10 \newfont{\cmdunh}{cmdunh10}
{\eufrb Insert\quad 12345\quad LATEX}\quad{\eufrb Label\quad 12345\quad LATEX}\quad
\eufrb\char65\char66\char67}\quad{\cmdunh 12345\quad LATEX\quad dvips\quad \symbol{1}
\symbol{2}\symbol{3}\symbol{4} \symbol{5}\quad\symbol{64} \symbol{61}\symbol{35}}
```

Щоб одержати таблицю, що представляє символи шрифта й їхні коди, варто запустити T_EX для обробки системного файлу `testfont.tex` командою

```
TEX testfont
```

Буде викликана назва шрифта, а потім тип операції. На останній запит вводимо команду

```
\table\end
```


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)


Стор. 94 із 189

[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

У результаті з'явиться файл `testfont.dvi`, що містить шукану таблицю з яким можна працювати як з будь-яким `dvi` файлом.

У L^AT_EX 2_ε запроваджена нова схема вибору шрифтів (New Font Selection Scheme – NFSS). Вона більш стандартизована й зручна у використанні, особливо для розробників стилів.

У новій схемі вибору шрифтів реалізовані наступні важливі моменти:

- команда завантаження шрифту може бути видана в будь-якому місці документа;
- команди зміни характеристик шрифту повністю незалежні й можуть використовуватися в довільному порядку.

Нова схема оперує наступними характеристиками шрифтів:

family *сімейство* – набір шрифтів, що мають загальні характеристики в малюнку символу, такі, наприклад, як висота окремих груп символів або особливе накреслення деяких деталей.

Сімейство містить у собі шрифти різних розмірів, форм, насиченості й ширини.

series *серія* – комбінація двох взаємозалежних характеристик, насиченості й ширини.

По насиченості найбільше часто вживані шрифти розділяються на світлі, нормальні та жирні, по ширині – на вузькі, нормальні й широкі. Існують також інші варіанти, наприклад, напівжирний або занадто жирний шрифти;

shape *форма* шрифту може бути однією з наступних: пряма, курсив, капітель, похила, прямий курсив;

size *розмір* шрифту та відстань між базовими лініями задається в типографських пунктах.

Для опису характеристик шрифтів слід використовувати команди:

```
\family{ім'я} \shape{код} \series{код} \size{число1}{число2}
```

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 95 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

У таблиці 9 наведені параметри команди `\size`, реалізовані в стандартних макрокомандах L^AT_EX'a, для різних розмірів нормального шрифту документа. Класифікація часто використовуваних текстових шрифтів Computer Modern розміром 10pt наведена в таблиці 14.

Під гарнітурою слід розуміти набір шрифтів, що зберігають стилеві ознаки основного малюнку, але відрізняються по накресленнях і розмірам. Зміна вигляду шрифту в межах однієї гарнітури можуть створити накреслення, що розрізняються:

- по нахилу – пряме, похиле вправо або вліво;
- по щільності – вузьке, нормальне, широке;
- по насиченості – світле, жирне.

25.3. Збільшення шрифтів

Команда завантаження для масштабування шрифту має два формати:

```
\font\ім'я = зовнішнє ім'я шрифту scaled \magstepi
```

```
\font\ім'я = зовнішнє ім'я шрифту at бажаний розмір
```

де параметр *i* слід вибрати згідно табл. 15.

У принципі в цій команді можна вказати довільний розмір шрифту, однак варто мати на увазі, що можливість масштабування шрифтів закладена не в усі драйвери й, імовірно, вибір буде обмежений набором шрифтів, призначених для конкретної програми. Якщо немає шрифту замовленого розміру, то драйвером буде використаний шрифт, розмір якого досить близький до замовленого. Але не виключена можливість, що драйвер припинить роботу з видачею повідомлення про відсутність шрифтів необхідного розміру.

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 96 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

Таблиця 14. Класифікація шрифтів Computer Modern розміром в 10pt

family	series	shape(s)	Імена шрифтових файлів
<i>Гарнітура CM roman</i>			
cmr	m	n, it, sl, sc, u	cmr10, cmti10, cmcsc10, cmu10
cmr	bx	n, it, sl	cmbx10, cmbxti10, cmbxsl10
cmr	b	n	cmb10
<i>Гарнітура CM sans serif</i>			
cmss	m	n, it	cmss10, cmssi10
cmss	bx	n	cmssbx10
cmss	sbc	n	cmssdc10
<i>Гарнітура CM typewriter</i>			
cmtt	m	n, it, sl, sc	cmtt10, cmitt10, cmstt10, cmtcsc10

Таблиця 15. Коефіцієнти збільшення шрифту.

Параметр	Коефіцієнт
<code>\magstep0</code>	1
<code>\magstephalf</code>	1.095
<code>\magstep1</code>	1.2
<code>\magstep2</code>	1.44
<code>\magstep3</code>	1.728
<code>\magstep4</code>	2.074
<code>\magstep5</code>	2.488



Сайт ПНПУ

Головна

Зміст



Стор. 97 із 189

Назад

Перегляд

Закрити

Вихід

26. Посилання

Щоб зробити посилання до певного місця в тексті, використовується команда `\footnote` з одним обов'язковим аргументом — текстом посилання. У стандартних стилях L^AT_EX'а посилання¹¹ нумеруються підряд протягом усього розділу або навіть (у стилі `article`) усього документа. У вихідному тексті попередній фрагмент виглядав так:

```
посилання\footnote{Для прикладу} нумеруються ...
```

Якщо після слова, до якого робиться посилання, повинен стояти розділовий знак, то його слід поставити після закриваючої фігурної дужки, що обмежує аргумент команди `\footnote`.

Якщо текст посилання складається з декількох абзаців, то вони розділяються порожнім рядком. У L^AT_EX'ові не передбачена можливість розпочинання нумерації заново на кожній сторінці. Якщо знехтувати автоматичною нумерацією посилань, то можна скористатися командою `\footnote` з необов'язковим аргументом, який ставиться у квадратних дужках перед обов'язковим¹⁹⁶⁸. Попередній фрагмент виглядав би у вихідному тексті так:

```
обов'язковим \footnote[1968]{Ось так!}. Попередній ...
```

При використанні команди `\footnote` з необов'язковим аргументом автоматична нумерація посилань «не збивається».

Текст посилання друкується внизу сторінки і відділяється від основного тексту горизонтальною лінією, яка створюється командою `\footnoterule`. За замовчуванням `\footnoterule` займає третину ширини сторінки і задається наступним набором команд:

```
\newcommand{\footnoterule}{\vspace*{-3pt}\hrule width .4\columnwidth\vspace*{2.6pt}}.
```

¹¹Для прикладу
¹⁹⁶⁸Ось так!


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)

[Стор. 98 із 189](#)
[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

27. Абзаци

Для верстки абзацу T_EX'ові не потрібно додавати жодних спеціальних зусиль: досить залишити у вихідному тексті порожній рядок, що вказує на кінець абзацу.

Зазвичай, абзаци вирівнюються по правому краю. При необхідності проміжки між словами розтягуються або стискаються, а в словах ставляться переноси. T_EX із усіх варіантів розбивки тексту абзацу на рядки вибирає оптимальний, урахувуючи межі, що є для стиску й розтягу проміжків між словами. Межа стиснення рядка не перевищується T_EX'ом ні за яких умов. Рядки, що не вдається стиснути, залишаються надто довгими й виходять за поля документа. Межа розтяжності, навпаки, може бути перевищена. У результаті цього з'являються рядки, які поліграфісти називають «рідкими», або «розрідженими»:

Маємо досить і досить рідкий рядок.

У процесі трансляції T_EX повідомляє про такі недоліки, записуючи повідомлення в log-файл (див. підрозділ 22). Наприклад, повідомлення

```
Overfull \hbox (1.77339pt too wide) in paragraph at 7--12
```

вказує на «переповнення» (`overfull`) рядка. У дужках зазначено, на яку саме відстань рядок виходить за край: на 1.77339 пунктів. Далі сказано, що переповнення відбулося при верстці абзацу (слова «in paragraph»), а потім указані номери рядків вихідного файлу, у яких був записаний цей абзац. Також у log-файл записується фрагмент невдалого рядка поблизу кінця якого, відбулося переповнення. Повідомлення про розріджені рядки виглядають так:

```
Underfull \vbox (badness 2318) in paragraph at lines          940--942
\elvrm спис-ку ну-ме-ру-ют-ся
```


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)

[Стор. 99 із 189](#)
[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

(у log-файлі ці повідомлення мають і продовження). Головних елементів у повідомленнях три:

- саме слово `Underfull` означає, що мова йде про рідкий рядок;
- чисельна характеристика того, наскільки розріджений рядок (по-англійському це число називається `badness`, дорівнює 2318);
- вказівка на місце знаходження абзацу з рідким рядком (940-942).

Вигляд при друкуванні розріджених рядків із різними значеннями міри `badness`:

Так виглядає розріджений рядок.	<code>badness = 0</code>
Так виглядає розріджений рядок.	<code>badness = 187</code>
Так виглядає розріджений рядок.	<code>badness = 1215</code>
Так виглядає розріджений рядок.	<code>badness = 2050</code>
Так виглядає розріджений рядок.	<code>badness = 3907</code>
Так виглядає розріджений рядок.	<code>badness = 10000</code>

В останньому рядку значення `badness` дорівнює 10000, це граничне значення заходів розрідженості рядків. Якщо розтягнути пробіли в рядку ще сильніше, `badness` не збільшиться. L^AT_EX вважає подібні рядки однаково поганими, тому не робить між ними різниці. Повідомлення про `underfull`-і з'являються, коли `badness` перевищуватиме 1000.

При розбитті абзацу на рядки L^AT_EX не може створити такі рядки, `badness` яких більше, ніж значення параметру `\tolerance`. Якщо цю умову виконати не вдається, виникає `overfull`. При мірі розрідженості рядка `badness`, що не перевищує значення `\tolerance`, буде створено розріджений рядок.

За замовчуванням `\tolerance=200`. Максимально можливе значення `\tolerance=10000`. Розумним вважається значення `\tolerance`, що рівне 300 або 400.


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)

[Стор. 100 із 189](#)
[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

28. Переноси

При роботі з текстами необхідно користуватися режимом, у якому перенос двох останніх літер був би допустимий. Установлюється цей режим за допомогою зміни параметра

```
\righthyphenmin.
```

Значення параметра – ціле число, рівне найменшій кількості букв слова, які можна переносити на наступний рядок. Запис у файлі

```
\righthyphenmin=2
```

означає, що при обробці всіх абзаців, що закінчуються після цієї команди, будуть дозволені переноси з відривом двох останніх букв. Для англійського тексту перед його завершальним порожнім рядком потрібна команда

```
\righthyphenmin=3
```

для уникнення неправильних переносів в англійських словах. Після внесення змін – проблеми з переносами зникають. Якщо цього не відбувається, існує 2 способи вирішення проблеми:

1. «одноразовий» спосіб – передбачає вказування T_EX'у місць переносів у слові командою \- Наприклад: команда \- означає, що дане слово можна переносити тільки у вказаних місцях;
2. за допомогою команди \hyphenation (її аргументом є слова, у яких дефісами позначені дозволені місця переносів) – задається T_EX'у в яких місцях можна переносити слова в межах даного документу.

Якщо в слові, що є аргументом команди \hyphenation, не поставити дефісів, то це означає, що його взагалі не можна переносити. Місце для \hyphenation – преамбула документу:

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 101 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

```
\hyphenation{своїми розбіжність}  
\hyphenation{до-ро-га вір-ність}
```

Слова, указані в аргументі команди `\hyphenation`, повинні розділятися проміжками.

Якщо користувачу необхідно зробити переноси в словах, в яких використовується слеш, то для цього необхідно записати у слові замість знаку / команду `\slash`, і при необхідності L^AT_EX автоматично зробить перенесення в місці, де розташований слеш.

Параметр `\hyphenpenalty` що впливає на частоту переносів. За замовчуванням рівний 50. Чим більше значення `\hyphenpenalty`, тим рідше переноси. При максимальному значенні параметру `hyphenpenalty=10000` переноси заборонені. L^AT_EX зменшує кількість переносів за рахунок розтягування рядків, тому, при збільшенні значення `\hyphenpenalty`, можливо, буде необхідно збільшити і такі параметри, як `\tolerance` та `\emergencystretch`.

`\exhyphenpenalty` відповідає за переноси в словах, де є дефіс або `\slash`. Чим більше значення цього параметра, тим рідше переноси. При максимальному значенні `exhyphenpenalty=10000` переноси в словах з дефісом і `\slash` будуть заборонені.

Параметр `\doublehyphendemerits` відповідає за кількість послідовних рядків, які завершуються переносами. Чим більше значення цього параметра, тим менше рядків підряд будуть завершуватися переносами. За замовчуванням `\doublehyphendemerits=10000`. Щоб послідовних рядків, які завершуються переносами, не було взагалі, вкажіть `\doublehyphendemerits=1000000`.

Параметр `\finalhyphendemerits` відповідає за переноси слів у передостанніх рядках абзаців. За замовчуванням `\finalhyphendemerits=5000`. Чим більше значення цього параметру, тим рідше будуть переноси в передостанніх рядках абзаців.

Для заборони переносів у всіх словах, що розпочинаються з прописної літери, необхідно вказати `\uchyph=0`. Щоб переноси в подібних словах знову дозволити: `\uchyph=1`.

Ці команди можна задавати або глобально – в преамбулі документа, або локально – в групі. В останньому випадку необхідно після абзацу перед закриваючою групу фігурною дужкою залишити порожній рядок, який «дасть зрозуміти» L^AT_EX-у, що абзац завершений.

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 102 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

29. Команда `\sloppy` і параметр `\emergencystretch`

Щоб позбутися виникаючих переповнень досить включити у файл команду `\sloppy`. Розташована всередині групи, команда `\sloppy` діє на абзац тоді, коли закриваюча фігурна дужка слідує після порожнього рядка, що завершує абзац.

Команда `sloppy`

Якщо промовець і слухач близькі за своїми поглядами, то промова сприймається позитивно. Натомість якщо розбіжність у поглядах велика, то надії на те, що промовець зуміє переконати слухачів і змінити їх погляди, мало.

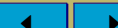
```
{\sloppy Якщо промовець і слухач близькі за своїми поглядами, то промова
сприймається позитивно. Натомість якщо розбіжність у поглядах велика, то надії на
те, що промовець зуміє переконати слухачів і змінити їх погляди, мало.}
```

Її все ж не рекомендується розміщувати в преамбулі, бо в 90% випадків грубі недоліки, пов'язані з переповненнями зникають, але проблеми розріджень рядків можуть виникнути.

Повернутися до звичайного режиму можливо за допомогою команди `\fussy`. Є також інший спосіб – не концентрувати всю розрідженість в одному рядку, яка може з'явитися в результаті дії команди `\sloppy`, а розподілити її більш-менш рівномірно по всьому абзацу. Для цього використовується параметр `\emergencystretch` – деяка довжина, за замовчуванням дорівнює нулю. При встановленні його значення рівним приблизно 3–5 пунктам, наприклад,

```
\emergencystretch=5pt
```

TeX спробує зробити всі рядки абзацу більш розрідженими у випадку, коли без переповнень зверстати абзац не вдається. Точний розмір `\emergencystretch` підбирається експериментально.


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)


Стор. 103 із 189

[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

30. Ручне керування розривами рядків

Ще один спосіб заборонити T_EX'у робити перенос у даному слові й у даному місці: потрібно наприкінці слова без проміжку поставити команду `\-`, наприклад:

Слово мистецтво\ - у даному своєму входженні переносів не матиме.

Скориставшись командою `\\` можна одержати рядок, що не доходить до краю й розтягу не отримує.

Особливості керування розривами рядків

Цей і наступний рядки зазнали розриву. Вони не розріджені, хоча праворуч лишилося порожнє місце.

Цей і `\\` наступний рядки `\\` зазнали розриву. Вони не розріджені, хоча праворуч лишилося порожнє місце.

Можна також використати команду `\linebreak`. При цьому обірваний рядок буде вирівняний по правому краю, навіть, якщо його прийдеться розтягти. Якщо рядок дійсно виявиться розтягнутим, під час трансляції про це буде повідомлено. Якщо абзац довгий, а команда `\linebreak` розташована не занадто близько до його початку або кінця, то швидше за все ніяких розтягнутих рядків не буде.

Існує також команда `\pagebreak`, формально аналогічна команді `\linebreak`. Якщо дати її без аргументів, то сторінку у вказаному місці буде розірвано; при цьому не виключено, що буде зроблена спроба вирівняти її по висоті з іншими сторінками за рахунок розтягування тих вертикальних інтервалів, які можна розтягнути – як правило, це інтервали між абзацами. (Команда `\newpage` такої спроби не робить.) Якщо дати команду `\pagebreak` із необов'язковим


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)


Стор. 104 із 189

[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

аргументом (цілим числом від 0 до 4), то цей аргумент буде виражати ступінь бажаності розриву сторінки в даному місці: якщо 0, то це всього лише дозвіл розірвати сторінку, якщо 4, то розрив обов'язковий, в інших випадках ступінь бажаності зростає із зростанням аргументу від 1 до 3.

Особливості керування розривами рядків

Цей, а також наступний рядки зазнали розриву. Вони розріджені, але вирівняні по лівому краю.

Цей, а також `\linebreak` наступний рядки `\linebreak` зазнали розриву. Вони розріджені, але вирівняні по лівому краю.

Команди `\\[m]`, `\linebreak[n]` можуть використовуватися з необов'язковим аргументом, де m – дійсне число, що визначає відстань вертикального відступу після «обірваного рядка», а n – ціле від 0 до 4, вказує наскільки бажаний перехід на новий рядок.

Є також команда `\nolinebreak`, що діє протилежно розглянутій. Для неї теж передбачено необов'язковий аргумент – ціле число (0–4), що посилює заборону розриву рядка у вказаному місці. Команда `\nolinebreak` задана без аргументу забороняє розрив рядка.

Особливості керування розривами рядків

Розірвано рядок із метою створення вертикального відступу.

Розірвано рядок `\\[15pt]` із метою створення вертикального відступу.


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)


Стор. 105 із 189

[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

31. Поняття про режими

У процесі обробки вихідного тексту T_EX завжди перебуває в одному з трьох режимів: горизонтальному, вертикальному або математичному. Тобто:

- у процесі обробки тексту (від появи першої ж букви до команди «закінчити абзац», наприклад, порожнього рядка) T_EX знаходиться в горизонтальному режимі;
- між абзацами, а також на початку роботи (наприклад, у процесі обробки преамбули до L^AT_EX'овського файлу) T_EX працює у вертикальному режимі;
- при обробці математичних формул T_EX перебуває в математичному режимі.

У вертикальному режимі всі проміжки й порожні рядки ігноруються, тому не слід зважати на зайві або відсутні пробіли між порожнім рядком, який завершує абзац, і новою порцією тексту.

В якості команди "закінчити абзац" можна використовувати, поряд з відомим користувачу порожнім рядком, команду `\par`. Якщо йдуть підряд кілька команд `\par`, то все це рівносильно одному порожньому рядку або одній команді `\par` (точно так само, як кілька порожніх рядків рівносильні одному), додаткового проміжку між абзацами користувач таким чином не створить. Підсумувати сказане в поняттях режиму можна так: у вертикальному режимі команда `\par` нічого не робить.

Математичний режим в L^AT_EX реалізується трьома різними способами: рівняння всередині абзацу (див. 50), рівняння розміщене на окремому рядку, рівняння розміщене на окремому рядку з автоматичною нумерацією.

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 106 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

32. Вилучення абзацного відступу

Іноді виникає необхідність створити абзац, у якому відсутній абзацний відступ. Для цього зручно користуватися командою `\noindent`.

Особливості керування абзацним відступом

У цьому абзаці відступу не буде.

Наступний абзац зверстано з відступом.

```
\noindent У цьому абзаці відступу не буде.
```

```
Наступний абзац \noindent зверстано з відступом.
```

Команда `\noindent` діє тільки на той абзац, що нею розпочинається. Якщо її помістити всередині абзацу, то взагалі нічого не відбудеться. Між `\noindent` і абзацом, до якого вона ставиться, не повинно бути порожнього рядка. Інакше команда виявиться поставленою до «порожнього абзацу», який буде закінчуватися цим же порожнім рядком. Користуючись поняттям режиму, можна сказати, що у вертикальному режимі команда `\noindent` означає «почати новий абзац без абзацного відступу», а в горизонтальному (і математичному) режимі вона означає нічого не робити».

Якщо користувачу потрібно створити абзац з «від'ємним» абзацним відступом, у якому всі рядки, крім першого, починаються на відстані одного сантиметра від полів. Цього можна досягти наступним чином:

```
\hangindent=1cm \noindent
```

Тут TeX'івський параметр `\hangindent` означає величину відступу від полів у всіх рядках абзацу, крім першого (за замовчуванням значення цього параметра дорівнює нулю).

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 107 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

33. Керування розривами сторінок

Заборона розриву сторінки після кінця абзацу досягається командою `\nopagebreak` тоді, коли стоятиме після нього. Якщо після кінця абзацу одночасно присутні команда `\nopagebreak` і команда для додаткових вертикальних проміжків, то команда `\nopagebreak` повинна йти першою, оскільки в іншому випадку вона не подіє.

Команда `\nopagebreak` може приймати необов'язковий аргумент — ціле число від 0 до 4. Разом із цим аргументом вона не забороняє розриву сторінки в зазначеному місці, але робить його менш вигідним із погляду T_EX'a (чим більший аргумент, тим менше вигідний). Команда `\nopagebreak[4]` означає повну заборону розриву так, якби команда була подана взагалі без аргументу. Якщо аргумент дорівнює 0, то в даному місці в принципі сторінку можна розірвати.

Після команди `\samepage` розриви сторінок стануть можливі тільки між абзацами, але не всередині абзаців і не між текстом і виключною формулою. Поставивши команду `\samepage` усередині групи, після кінця групи її дія припиняється. Для примусового розриву сторінок у T_EX'ові існують команди

```
\newpage, \clearpage, \cleardoublepage.
```

Усі перераховані вище команди для розриву сторінки діють навіть у тому випадку, коли команда `\samepage` раніше його заборонила. Існує команда `\pagebreak`. Якщо її ввести без аргументів, то сторінка в цьому місці буде розірвана, при чому може бути зроблена спроба вирівняти її по висоті з іншими сторінками за рахунок розтягування тих вертикальних інтервалів, які можна розтягти (як правило, це інтервали між абзацами). Зрозуміло, вигляд у такої сторінки ще гірший, ніж в «укороченої» сторінки з нормальними інтервалами. Якщо дати команду `\pagebreak` із необов'язковим аргументом (цілим числом від 0 до 4), то цей аргумент буде виражати ступінь бажаності розриву сторінки в даному місці. Вибір 0 – дозвіл розірвати сторінку, вибір 4 – вказівка на обов'язковість розриву. В інших випадках ступінь бажаності розриву зростає з ростом аргументу від 1 до 3.

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 108 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

34. Вертикальні проміжки

Для задання проміжків між абзацами використовуються такі команди:

- `\smallskip` задає проміжок величиною \smallskip ;
- `\medskip` задає проміжок величиною \medskip ;
- `\bigskip` задає проміжок величиною \bigskip .

Найпростіше поставити ці команди безпосередньо після порожнього рядка або команди `\par`, що завершує абзац:

Особливості керування вертикальними проміжками

Після цього абзацу буде залишено додатковий проміжок.

Це дійсно так.

Після цього абзацу буде залишено додатковий проміжок. `\par\medskip` Це дійсно так.

Якщо необхідно задати розмір вертикального проміжку в явному вигляді, можна скористатися командою `\vspace`, у якій є один обов'язковий аргумент – величина проміжку. Наприкладі проілюструємо, як отримати між рядками вертикальний проміжок розміром вісім міліметрів. Команда `\vspace` ставиться після кінця абзацу, при необхідності отримання вертикального відступу між абзацами (подібно таким командам, як `\smallskip`), бо, як свідчить приклад, коли вона розміщена посеред абзацу, додатковий проміжок буде між рядками.

Якщо набрати команду `\vspace` відразу ж після `\newpage` або `\clearpage`, то вертикальний відступ на початку нової сторінки не з'явиться. Відступ, створюваний `\vspace`, зникне у випадку, якщо він розташовується на початку нової, автоматично створеної сторінки.


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)


Стор. 109 із 189

[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

Особливості керування вертикальними проміжками

У даному фрагменті тексту

здійснено вертикальний відступ.

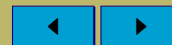
У даному фрагменті тексту `\vspace{8mm}` здійснено вертикальний відступ.

Щоб вертикальний відступ на початку сторінки не зникав, необхідно скористатися варіантом із зірочкою після ім'я команди: `\vspace*{розмір відступу}`. Якщо написати `\vspace*{1cm}`, то буде створено вертикальний проміжок в 1cm, що не пропадає навіть у тому випадку, коли команду дано відразу після `\newpage` або `\clearpage` або в цьому місці стався розрив сторінки.

Можна змусити команду `\vspace` створити проміжок не фіксованої величини, а змінної довжини. Саме, в самому загальному вигляді ця команда записується так:

```
\vspace{x plus y minus z}.
```

Тут x , y і z – довжини, виражені в T_EX'івських одиницях, а `plus` і `minus` – так звані “ключові слова” T_EX'а (на відміну від інших команд, перед ними не треба ставити `backslash`). При цьому x позначає “природну” величину відступу: якщо при верстці сторінок вертикальні інтервали не доводиться розтягувати або стискати (наприклад, у випадку, коли користувач дозволив T_EX'у залишати внизу сторінки порожнє місце), то буде зроблений пробіл розміром x . При необхідності, однак (наприклад, задля того, щоб всі сторінки мали однакову висоту), цей інтервал можна буде змінити: y вказує степінь розтяжності, а z – степінь стиснення інтервалу.

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 110 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

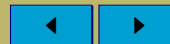
35. Інтерліньяж

У поліграфії словом «нтерліньяж» називається інтервал між рядками. L^AT_EX'овські команди на зразок `\small`, що встановлюють розмір шрифту, автоматично визначають і розмір інтервалу між рядками. Для збільшення інтервалів між рядками на 1%, тобто в 1.01 рази, варто написати так:

```
\renewcommand{\baselinestretch}{1.01}}
```

Для подвійного інтервалу L^AT_EX необхідно вказати число 1,65, а для полуторного – 1.25. Так само можна використовувати команду, де задати потрібний множник: для полуторних інтервалів `\linespread{1.3}`.

Замість десяткової крапки можна використовувати й десяткову кому. Цю команду слід поміщати в преамбулу. Між абзацами можна організувати додаткові вертикальні інтервали. При написанні `\parskip=3mm` між абзацами буде відступ у 3 мм (на додаток до звичайного міжрядкового інтервалу). Без особливої необхідності не треба присвоювати параметру `\parskip` нове значення, оскільки воно цілком розумно встановлюється в стандартних L^AT_EX'овських стилях. Насправді в стандартних стилях `\parskip` є розтяжною довжиною. Сам природний розмір `\parskip` дорівнює нулю, але в нього є ще plus-компонента, рівна одному пункту. Якщо вертикальні інтервали на сторінці не варіюються, то ніякий додатковий інтервал між абзацами не створюється. Якщо при верстці сторінку доводиться розтягувати по вертикалі, то кожний із інтервалів між абзацами може бути збільшеним (максимум на один пункт). При бажанні можна змінювати як природний розмір, так і розтяжний компонент параметра `\parskip` за допомогою команди `\setlength`.

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 111 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

36. Набір у дві колонки

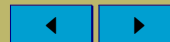
Якщо необхідно набирати весь документ у дві колонки, то це потрібно зробити, указавши в команді `\documentclass` відповідну «стильову опцію»

```
\documentclass[twocolumn]
```

Команда `\twocolumn` дозволяє набирати не весь текст у дві колонки, а тільки його частину. Діє вона так: спочатку виконується команда `\clearpage` (див. с. 108), а потім із нової сторінки, створеної цією командою, починається набір у дві колонки. Іноді буває необхідно на початку нової сторінки помістити один або кілька абзаців тексту на всю ширину сторінки, а текст, що залишився, на цій сторінці набрати у дві колонки. Для цього можна використати команду `\twocolumn` із необов'язковим аргументом у квадратних дужках – це той текст, що буде надрукований на всю ширину сторінки. Якщо він складається з декількох абзаців, то вони, як звичайно, розділяються порожніми рядками. Команда `\onecolumn` здійснює перехід від двоколонного набору до одноколонного (попередньо вона знов-таки виконує команду `\clearpage`).

Ширина інтервалу між стовцями тексту в стилі `\twocolumn` визначається за допомогою команди `\columnsep`. Ширина вертикальної лінії, яка розміщується між двома колонками в стилі `\twocolumn` визначається командою `\columnseprule`. За замовчуванням її значення рівне нулю.

Більш гнучкого форматування друкованого документа дозволяє досягти пакет `multicol` із колекції `tools`. Описаними вище засобами неможливо сформулювати сторінку зі змінним числом колонок, так як команди `\twocolumn` і `\onecolumn` завжди починають друкувати текст з нової сторінки. Крім того, колонки виявляються незбалансованими, тобто мають різну ширину.

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 112 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

37. Стили та класи документів, їх стильові опції, пакети

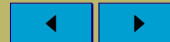
Команда `\documentstyle` (розпочинаючи з L^AT_EX 2_ε – `\documentclass`), із якої починається будь-який T_EX'овський файл, має один обов'язковий аргумент – назву основного стилю – і один необов'язковий, що розташовується перед обов'язковим — список <стильових опцій> (див. с. 36).

Як зазначалося, стиль `article` зручно застосовувати для статей, стиль `report` – для більших статей, розбитих на розділи, або невеликих книг, стиль `book` – суто для книг. У табл. 16 перераховані деякі особливості оформлення, властиві стандартним стилям. Зміст знаків у ній такий:

- <+> – означає <завжди присутній>;
- <-> – означає <завжди відсутній>;
- <∓> – означає <за замовчуванням відсутній, але буде присутній, якщо задати стильову опцію або спеціальну команду>;
- <±> – означає <за замовчуванням є присутнім, але можна скасувати за допомогою стильової опції або спеціальної команди>.

Відомо, що список стильових опцій через кому ставиться у квадратних дужках перед основним аргументом команди `\documentstyle`. Найважливіші й часто вживані стильові опції – це `11pt` і `12pt`. Вони означають, що основний текст документа буде набраний шрифтом кегля 11 або 12 відповідно. Якщо цих опцій не вказувати, то буде шрифт кегля 10 (див. табл. 2).

Стильова опція `twoside` задає друк з різними полями на непарних і парних сторінках (як у книгах). Її можна вказувати тільки для стилів `article` і `report`; у стилі `book` опція виходить автоматично, і вказувати її не потрібно. Разом з опцією `twoside`, в L^AT_EXе всі три основних класи документів допускають опцію `oneside`. Якщо її вказати, то поля на парних і непарних сторінках будуть однакові.


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)

[Стор. 113 із 189](#)
[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

Таблиця 16. Особливості оформлення стандартних стилів.

	article	report	book
Автоматична нумерація розділів	+	+	+
Розбиття на розділи	–	+	+
Розбиття на частини	+	+	+
<Двостороній>* друк	∓	∓	+ (\pm у L ^A T _E X 2 _ε)
Титульний аркуш	∓	+	+
Колонтитули	∓	±	±
Висота всіх сторінок однакова	∓	∓	±
Набір у два стовпчики	∓	∓	∓

* Різні поля для парних і непарних сторінок.

Стильову опцію `\twocolumn` можна задавати для кожного із трьох основних стилів. Вона означає, що набір тексту буде проведено у два стовпчики. Так, як абзаци при цьому будуть виходити досить вузькі, слід при користуванні цією опцією заодно збільшити параметр `\tolerance`, інакше буде багато рядків виходити за межі стовпчиків.

Стильову опцію `titlepage` необхідно застосовувати при використанні стилю `article`. При її заданні в документі буде надрукований титульний аркуш, (у стилях `report` і `book` титульний аркуш робиться завжди). Поряд з опцією `titlepage`, в L^AT_EX всі три основних класи документів допускають опцію `notitlepage`. Якщо її вказати, то окремого титульного аркуша друкуватися не буде (навіть у документі класу `book`).

Опція `draft` придатна для кожного з трьох стилів. Якщо вона включена, то кожний рядок, що виходить за межі поля (рядок, про який видається повідомлення `<Overfull \hbox>`), позначається на полях значком ■.


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)


Стор. 114 із 189

[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

Режим, при якому різні сторінки можуть мати різну висоту, задається наступною командою `\raggedbottom`. За замовчуванням такий режим встановлюється стилями `article` і `report`, якщо тільки не зазначена стильова опція `twoside` (<двосторонній> друк). У всіх інших випадках L^AT_EX буде, за замовчуванням, робити всі сторінки однакової висоти.

Наступні дві стильові опції, застосовні до кожного з основних стилів, впливають на оформлення <відокремлених> математичних формул (див. підрозділ 10). Опція `fleqn` означає, що <відокремлені> формули, задані за допомогою оточень `equation`, `eqnarray` і `displaymath`, а також пари команд `\[` і `\]`, будуть надруковані не в центрі рядка, а в її лівій частині. Опція `leqno` означає, що номери формул, генеровані оточеннями `equation` і `eqnarray`, будуть друкуватися не праворуч, а ліворуч.

Все вище сказане про стиль оформлення документа, застосовне й у L^AT_EX 2_ε, треба тільки говорити <клас документа> замість <основний стиль> і писати `\documentclass` замість `\documentstyle` (див. підрозділ 5).

Отже, клас визначає вид і структуру документа. Клас – це база, яку можна правити з допомогою додаткових стильових файлів. У класі задається геометрія сторінки та визначені команди секціонування. Клас може бути просто невеликою модифікацією вже наявного класу, а може бути принципово новою реалізацією представлення друкованого чи електронного T_EXнічного слова.

Якщо стандартні класи не подобаються користувачу, він намагається їх поліпшити усуваючи деякий конкретний недолік. Це зроблено в наборі класів `extsizes` (`extarticle`, `extbook`, `extletter`, `extproc`, `extreport`), які відрізняються від стандартних тільки можливістю задати базовий розмір шрифту, відмінний від звичайного 10-12pt. Є набори класів, які створено з певною метою. Прикладом такого підходу є класи від *AMS* (`amsart`, `amsbook`, `amsproc`), які використовуються для матеріалів, призначених для публікації в журналах Американського математичного співтовариства. Класи з набору `ntgclass` представляють із себе «героїчну» спробу німецькомовних голландців зробити те ж, що і в стандарті, але істотно різними способами.

Останнім часом все більше уваги звертає на себе набір класів **KOMA-script** розробника-

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)

Стор. 115 із 189

[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

ми яких є німці. Слід враховувати, що європейські традиції поліграфії (в основному французькі), все-таки ближче до нас, ніж американські, на які традиційно орієнтувалося L^AT_EX-співтовариство. Для статей передбачається використовувати `scrartcl`, для книг `scrbook`, для листів `scrllttr2`, а для звітів `scrreprt`.

На відміну від стандарту, класи з **КОМА-script** дозволяють використовувати базові розміри шрифту в 9pt, 14pt і 17pt. Величезні поля, що мають місце в стандартних класах, класах **КОМА-script** значно зменшені. Зазнали змін і інші елементи. Якщо оформлення за замовчуванням не здається адекватним, то **КОМА-script** надає великий набір високорівневих налаштувань. Докладна документація на більш ніж двохстах сторінках «The KOMA-Script bundle» ([scrguien.pdf](#)) дозволяє настроїти всі необхідні параметри. [1]

У процесі набору документа часто виявляється, що в деяких областях базовий L^AT_EX не може вирішити виникаючих проблем. Наприклад тоді, коли є необхідність включити в документ графіку, кольоровий текст або вихідний код програми із зовнішнього файлу. Для цього потрібно буде розширити можливості L^AT_EX'a. Саме для цього у L^AT_EX 2_ε поряд зі стильовими опціями використовуються й так звані стильові пакети [2]. На файл, що розпочинається командою `\documentclass` може впливати одна або декілька команд `\usepackage`, в аргументі яких стоїть, через кому, список стильових пакетів. Класична форма активізації пакета є такою:

```
\usepackage[опції]{пакет}
```

де `пакет` – це ім'я пакета, а `опції` – список ключових слів, що включають спеціальні властивості пакета. Одні пакети є стандартними і розповсюджуються разом із L^AT_EX 2_ε, інші надаються окремо. Ось деякі з стандартних пакетів:

- `doc` – дозволяє документувати програми на L^AT_EX;
- `exscale` – надає масштабовані версії розширених математичних шрифтів;
- `fontenc` – указує, яке кодування шрифту повинен використати L^AT_EX;

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)

Стор. 116 із 189

[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

- `ifthen` – надає команди виду «якщо ..., то виконувати..., інакше виконувати ...»;
- `latexsym` – використовується для підключення шрифту, що містить спеціальні символи L^AT_EX'a (\square , \diamond , \boxtimes , \rightsquigarrow , \triangleleft , \mathcal{U} , \triangleright , \sqsubset , \supset , \leq , \geq);
- `amssymb` – великий ресурс математичних символів і команд, для запису формул;
- `makeidx` – надає можливість використовувати набір команд для генерації покажчиків;
- `amscd` – для малювання діаграм;
- `graphicx` – дозволяє включати картинки [3];
- `color` – для створення кольорових тексту та тла в документі;
- `lettrine` – дозволяє вставляти Буквиці;
- `acromake` – використовується для створення абревіатур;
- `ulem` – різні види підкреслень окремих слів і тексту (приклади на с. 122);
- `fancybox` – для створення рамок: із тінню, із округлими кутами та подвійних;
- `array` – використовується для вертикального вирівнювання абзаців у таблицях;
- `cases` – для набору формул, як із цифровою (див. с. 128), так і буквенною нумерацією в кожному рядку;
- `deleq` – для створення частинних номерів формул (увага, працює тільки в L^AT_EX 2_ε; приклад див. с. 130);
- `eqname` – дозволяє створювати довільні імена або номери для формул (див. с. 132);

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 117 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

- `syntonly` – обробляє документ, не друкуючи його;
- `inputenc` – вказує вхідне кодування, таке як ASCII, ISO Latin-1, ISO Latin-2, 437/850 IBM code pages, Apple Macintosh, Next, ANSI-Windows або обумовлене користувачем;
- `afterpage` – дає можливість «розчищення завалу» без побічного ефекту у вигляді видачі скороченої сторінки: якщо замість `\clearpage` написати `\afterpage{\clearpage}`, «застрягли» плаваючі об'єкти будуть надруковані після того, як скінчиться поточна сторінка. Цим засобом не можна користуватися при друкуванні у дві колонки;
- `html,htmllist` – розширює можливості конвертора по оформленню HTML-сторінок.

Наприклад, якщо включити в преамбулу рядок

```
\usepackage{amsmath}
```

то відкриються додаткові L^AT_EX'овські можливості набору математичних формул; якщо ж написати

```
\usepackage[noamsfont3]{amsmath}
```

то будуть всі ці можливості, крім використання готичного й ажурного шрифтів.

Досить частою є проблема набору формул, що містять літери нашого вітчизняного алфавіту. Щоб позбутися даної проблеми, слід використати допоміжні пакети, а саме, це можна зробити так:

```
\usepackage{mathtext}
\usepackage[T2A]{fontenc}
\DeclareSymbolFont{T2Aletters}{T2A}{cmr}{m}{it}
```

Наприклад, щоб підключити векторні шрифти, можна написати:

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 118 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

```
\usepackage{psycyr}
\usepackage{mathtext}
\usepackage[T2A]{fontenc}
\DeclareSymbolFont{T2Aletters}{T2A}{ftm}{m}{n}
```

При використанні команди

```
\usepackage[ukrainian]{babel}
```

підключається до документа пакет `babel` з опцією `ukrainian`. Після цього L^AT_EX 2_ε розумітиме українську мову, використовуватиме правила переносів. До речі, для кожного слова можна самостійно вказати бажані місця його переносу: `ма\ -те\ -ма\ -ти\ -ка` або `математика\ -`. В останньому випадку переносів не буде зроблено взагалі.

Стосовно пакету `lettrine`, можливості якого досить великі особливо в плані обтікання буквиці з похилим краєм. Хитрість у тому, що потрібно використовувати не тільки пакет `lettrine`, але і шрифт (наприклад, `Type 1`, `TrueType` або `OpenType`). Працює в pdfL^AT_EX.

Особливості використання стилю `lettrine`

ДАЛІ наведено кілька простих прикладів використання стилю `lettrine`. Зверніть увагу, що для цього нам необхідно в преамбулі документа, підключити пакети `\usepackage {type1cm}` та `\usepackage {lettrine}`.

```
\lettrine {Д}{алі} наведено кілька простих прикладів використання стилю
\verb|lettrine|. Зверніть увагу, що для цього нам необхідно в преамбулі документа,
підключити пакети \verb|\usepackage {type1cm}| та \verb|\usepackage {lettrine}|.
```


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)


Стор. 119 із 189

[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

Особливості використання стилю `lettrine`

У другому прикладі наголошується, що з налаштуваннями пакет можна зробити ще краще. Для цього нам знадобляться деякі необов'язкові аргументи. А саме: `lines` – вказує кількість рядків для буквиці; `slope` – регуляція нахилу тексту, використовується для букв А, У і т.д.; `findent` – горизонтальний проміжок між буквицею та текстом. Цей приклад показує використання різних накреслень буквиці й основного тексту.

```
\lettrine[lines=3,slope=-6pt,findent=-3pt]{У}{}
```

другому прикладі наголошується, що з налаштуваннями пакет можна зробити ще краще. Для цього нам знадобляться деякі необов'язкові аргументи. А саме: `\verb|lines|` - вказує кількість рядків для буквиці; `\verb|slope|` - регуляція нахилу тексту, використовується для букв А, У і т.д.; `\verb|findent|` - горизонтальний проміжок між буквицею та текстом. Цей приклад показує використання різних накреслень буквиці й основного тексту.

Особливості використання стилю `lettrine`

ШРИФТ буквиці та основного тексту один і той же у третьому прикладі. Щоб повністю візуалізувати ефекти аргументів, скористаємося `nindent` для регуляції відступу рядків починаючи з другого.

```
\lettrine[lines=3,slope=10pt,nindent=42pt]{Ш}{рифт}
```

буквиці та основного тексту один і той же у третьому прикладі. Щоб повністю візуалізувати ефекти аргументів, скористаємося `\verb|nindent|` для регуляції відступу рядків починаючи з другого.

У наступному прикладі реалізовано можливості пакету `acromake`, що дозволяє створювати аббревіатури. Перед тим, як використовувати аббревіатуру в тексті, її необхідно визначити.


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)

[Стор. 120 із 189](#)
[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

Нехай готується стаття про оперативну пам'ять комп'ютера. Тоді буде доречною аббревіатура: Random Access Memory (RAM), що під час набору тексту визначається введенням рядка

```
\acromake{RAM}{\textbf{RAM}}{Random Access Memory}
```

Насиченість та накреслення шрифту аббревіатури вибирається за потребою, у даному випадку зазначена напівжирна опція.

Приклад створення аббревіатур

Оперативна пам'ять призначена для зберігання змінної інформації, тобто вона допускає зміну свого вмісту під час виконання мікропроцесором обчислювальних операцій. Цей вид пам'яті забезпечує режими запису, читання і зберігання інформації. Так як у будь-яку миттєвість часу доступ може виконуватися до будь-якої комірки пам'яті, то цей її вид називають пам'яттю з довільною вибіркою Random Access Memory (**RAM**). Для побудови запам'ятовуючих пристроїв типу **RAM** (see Page 121) використовують мікросхеми статичної (SRAM – Static **RAM**) та динамічної (DRAM – Dynamic **RAM**) пам'яті.

Оперативна пам'ять призначена для зберігання змінної інформації, тобто вона допускає зміну свого вмісту під час виконання мікропроцесором обчислювальних операцій. Цей вид пам'яті забезпечує режими запису, читання і зберігання інформації. Так як у будь-яку миттєвість часу доступ може виконуватися до будь-якої комірки пам'яті, то цей її вид називають пам'яттю з довільною вибіркою \RAM. Для побудови запам'ятовуючих пристроїв типу \RAM використовують мікросхеми статичної (SRAM - Static \RAM) та динамічної (DRAM - Dynamic \RAM) пам'яті.

Слід зауважити, що при посиланні на сторінку де безпосередньо визначається аббревіатура за замовчуванням використовуються англійські слова see Page, замість потрібних див. с. Щоб усунути дану проблему, необхідно відредагувати файл `acromake.sty` (це можна зробити за допомогою текстового редактора типу Блокнот), що знаходиться (орієнтовно) у папці

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 121 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

C:\texmf\tex\latex\ltxmisc, замінивши see Page на див. с.

За допомогою пакету `ulem` можна створити різні підкреслення, як окремих слів, так і тексту.

Приклад створення підкреслень

Зверніть увагу, що у даному випадку підкреслення тексту здійснене з переходом на інший рядок.

```
\emph{Зверніть увагу}, що у даному випадку \emph{підкреслення тексту здійснене з переходом на інший рядок.}
```

Наступні приклади демонструють різні стилі підкреслення.

`\uline`

Просте підкреслення (under-line).

```
Просте \uline{підкреслення (under-line)}.
```

`\uuline`

Підкреслення подвійною лінією (double under-line).

```
Підкреслення \uuline{подвійною лінією (double under-line)}.
```

`\uwave`

Підкреслення хвилястою лінією (under-wave).

```
Підкреслення \uwave{хвилястою лінією (under-wave)}.
```

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 122 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

\soutЗакреслення тексту лінією крізь текст (~~strike out~~).Закреслення тексту `\sout{лінія крізь текст (strike out)}`.**\xout**Закреслення тексту похилими лініями (~~cross out~~, ~~X out~~).Закреслення тексту `\xout{похилими лініями (cross out, X out)}`.

Використовуючи пакет `fancybox` можна отримати красиві рамки. Наступні приклади наглядно демонструють його можливості.

Бокс із тінню`\shadowbox{\bf Бокс із тінню}`**Подвійний бокс**`\doublebox{\bf Подвійний бокс}`

Необхідно звернути увагу на два овальних бокси, що відрізняються один від одного товщиною лінії (лінія навколо 2-го – товща).

Овальний бокс`\ovalbox{\bf Овальний бокс}`**Товстий овальний бокс**`\Ovalbox{\bf Товстий овальний бокс}`

Стиль `hhline` вводить нові типи горизонтальних ліній ("рамок") у таблицях.

На кожен стовпчик *повинні* бути задані елементи горизонтальної лінії:

- = – подвійна горизонтальна лінія
- – одинарна горизонтальна лінія
- ~ – пропуск (немає горизонтальних ліній)

Ці елементи розділяються вертикальними лініями й "куточками":



Сайт ПНПУ

Головна

Зміст



Стор. 123 із 189

Назад

Перегляд

Закрити

Вихід

- || – подвійна вертикальна лінія (без розриву)
- | – одиночна вертикальна лінія (без розриву)
- |: – подвійна вертикальна лінія (с розгалуженням вправо)
- :| – подвійна вертикальна лінія (с розгалуженням уліво)
- b – нижній подвійний куточок
- t – верхній подвійний куточок
- # – перетинання двох подвійних ліній

Попередження! При використанні пакету `hhline` користувач може зіткнутися, як і автор, з проблемою несприйняття символу `|` компілятором PDFLaTeX. Даної проблеми можна позбутися скориставшись командою `\DeleteShortVerb{\|}`

Простий приклад.

Приклад застосування пакету `hhline`

A	B	C	D
A	B	B	Г

```
\begin{tabular}{|c|cc|c|} \hline A & B & C & D \\ \hhline{|=|~~|=|} A & B & B & Г \\ \hhline{--} \end{tabular}
```

Більш складний приклад.

Приклад застосування пакету `hhline`

1	2	3	4
6	7	8	9
У	К	Р	А
Ї	Н	А	!

```
\begin{tabular}{||cc||cc||} \hhline{t===t===t|}
1 & 2 & 3 & 4 \\ 6 & 7 & 8 & 9 \\ \hhline{#===:===|} У & К & Р & А \\ \hhline{||-||~} Ї & Н & А & ! \\ \hhline{|b===b===b|} \end{tabular}
```

При використанні пакету `color` вибираючи колір для тексту чи тла слід пам'ятати, що


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)


Стор. 124 із 189

[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

кольори є двох типів: стандартні та визначені користувачем.

Визначення кольорів користувачем

```
\documentclass[11pt]{book}
\usepackage{color}
\begin{document}
\definecolor{light-blue}{rgb}{0.8,0.85,1}
\definecolor{gray}{gray}{0.50}
\definecolor{mygray}{gray}{0.75}
.....
\end{document}
```

Стандартні та визначені користувачем у вище наведеному прикладі кольори, а також їх візуалізацію можна переглянути у табл. 17.

```
{\color{blue}Стандартні та \textcolor{red}{визначені користувачем} у вище
наведеному прикладі кольори,} а також їх \fcolorbox{red}{mygray}{візуалізацію} можна
переглянути у табл.\ref{rr11}.
```

У L^AT_EXові існує стильовий пакет `array`. Його підключення відкриває різні дрібні (але нерідко корисні) додаткові можливості. Наприклад, при користуванні командою `\hline` горизонтальні лінії іноді занадто щільно примикають до тексту (особливо якщо текст містить прописні букви). Для боротьби із цим треба привласнити ненульове значення параметру `\extrarowheight`. Це – величина, що додається до висоти кожного рядка таблиці. Цьому параметру можна привласнювати значення так само, як і будь-якому іншому параметру зі значенням довжини; за замовчуванням його величина дорівнює нулю, для кращого відділення ліній від тексту необхідно привласнити йому значення 2-3 пункти.


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)

[Стор. 125 із 189](#)
[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

Таблиця 17. Кольори пакету color

Колір	Приклад
Стандартні кольори	
black	Текст чорного кольору.
white	Білий колір на сірому тлі.
red	Текст червоного кольору.
green	Текст зеленого кольору.
blue	Текст синього кольору.
cyan	Текст кольору ціан (голубий).
magenta	Текст кольору фуксин (пурпурний).
yellow	Жовтий колір на сірому тлі.
Кольори визначені користувачем	
light-blue	Текст голубого кольору.
gray	Текст сірого кольору.
mygray	Для порівняння з попереднім.

Приклад порушення вирівнювання

а б
в г

а	б
в	г

При використанні оточенням **tabular** з необов'язковим аргументом **t**, що задає вирівнювання таблиці як |букви| по верхньому рядку, відбувається порушення вирівнювання, якщо таблиця починається з горизонтальної лінії.

Щоб вирівнювання відбувалося не по лінії, а по першому рядку тексту, треба задати верхню лінію командою `\firsthline`, тоді все вийде. Щоб при користуванні **tabular** з необов'язковим аргументом **b** вирівнювання таблиці як цілого йшло по нижньому рядку тексту, а не по нижній


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)


Стор. 126 із 189

[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

Таблиця 18. Застосування пакету `array`

	<code>b{3cm}</code>	<code>m{3cm}</code>	<code>p{3cm}</code>
Текст –	Для ефективного рішення винахідницьких задач вищих рівнів потрібний евристичний алгоритм.	Такий алгоритм не може бути створений на основі досвіду окремого винахідника або навіть групи винахідників.	Щоб його одержати, потрібно: виявити об'єктивні закономірності розвитку технічних об'єктів досліджуючи великі масиви патентної інформації.

лінійці, треба нижню горизонтальну лінійку задати командою `\lasthline`.

Пакет `array` вводить також нові типи аргументів у `tabular`:

`b{}` – абзац вирівнюється з іншим рядком по *нижньому краю* тексту;

`m{}` – абзац вирівнюється по *центру* тексту;

`p{}` – абзац вирівнюється з іншим рядком по *верхньому краю* тексту (стандарт).

Перераховані можливості пакету `array` наглядно відображено в табл. 18, що в `tex`-файлі мала вигляд:


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)


Стор. 127 із 189

[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)


```

\begin{table}
\centering \caption{Застосування пакету \texttt{array}}\label{array}
\begin{tabular}{|c|b{4.5cm}|m{4.5cm}|p{4.5cm}|} \hline & \multicolumn{1}{c|}
{\texttt{b\{4.5cm\}}} & \multicolumn{1}{c|}{\texttt{m\{4.5cm\}}} &
\multicolumn{1}{c|}{\texttt{p\{4.5cm\}}} \\ \hline Текст - & Для ефективного
рішення винахідницьких задач вищих рівнів потрібний евристичний алгоритм. &
Такий алгоритм не може бути створений на основі досвіду окремого винахідника
або навіть групи винахідників. & Щоб одержати його, потрібно: виявити об'єктивні
закономірності розвитку технічних об'єктів досліджуючи великі масиви патентної
інформації. \\ \hline \end{tabular} \end{table}

```

Чудовою знахідкою для кожного, хто працює з формулами, буде пакет `cases`. При його підключенні, оточення `numcases` дозволяє створити цифрову нумерацію кожного рядка формули.

Оточення `numcases`

$$|x| = \begin{cases} x, & \text{якщо } x \in [0, +\infty) \\ -x, & \text{якщо } x \in (-\infty, 0) \end{cases} \quad (2)$$

(3)

```

\begin{numcases}{|x|=} x, & \text{якщо } \$x \in [0, +\infty)$ \\ -x, & \text{якщо } \$x \in (-\infty, 0)$

```

У свою чергу оточення `subnumcases` можна застосовувати, якщо нумерація рядків формули повинна бути буквенна.



Сайт ПНПУ

Головна

Зміст



Стор. 128 із 189

Назад

Перегляд

Закрити

Вихід

Оточення `subnumcases`

$$f(x) = \begin{cases} 1, & \text{при } x \in \mathbb{Q} \\ -x, & \text{при } x \notin \mathbb{Q} \end{cases} \quad (4a)$$

(4b)

```
\begin{subnumcases}{f(x)=} 1, & при $x\in \mathbb{Q}$ \\
-1, & при $x\not\in \mathbb{Q}$ \end{subnumcases}
```

Якщо прописати у рядку `\nonumber`, то нумерацію у ньому буде відмінено.

Оточення `numcases`

$$f(x) = \begin{cases} \frac{\sin x}{x}, & x \neq 0 \\ 1, & x = 0 \end{cases} \quad (5)$$

```
\begin{numcases}{f(x)=} \frac{\sin x}{x}, & $x\neq 0$ \\
1, & $x=0$ \nonumber \end{numcases}
```

У даному випадку для створення посилань на формули користуються стандартними командами `\label{}` та `\ref{}` (див. с. 48), яку розміщують у лівій частині при посиланні на всю формулу – (див. 6), а в рядкові тоді, коли посилання здійснюється на окрему її частину, наприклад – (6a).



Сайт ПНПУ

Головна

Зміст



Стор. 129 із 189

Назад

Перегляд

Закрити

Вихід

Оточення subnumcases

$$g(x, y) = \begin{cases} 0, & x \neq y \\ 1, & x = y = 0 \\ -1, & x = y \neq 0 \end{cases} \quad \begin{array}{l} (6a) \\ (6b) \\ (6c) \end{array}$$

```
\begin{subnumcases} {\label{w} g(x,y)=} 0, & $x\neq y$\label{wz} \\
1, & $x=y=0$ \\
-1, & $x=y\neq 0$ \end{subnumcases}
```

Продовжуючи тему створення частинних номерів для формул, розглянемо приклади їх створення на основі пакета `deleq`. Як відомо, використовуючи звичайне оточення `equation` (див. с. 64), матимемо стандартну нумерацію.

Стандартна нумерація

$$\sin^2 \alpha + \cos^2 \alpha = 1. \quad (7)$$

Якщо аналогічно `equation` використати оточення `deqn`, то у формули з'явиться частковий номер, тобто:

Оточення deqn

$$\sin(-\alpha) = -\sin \alpha \quad (8a)$$

```
\begin{deqn} \sin(-\alpha) = - \sin\alpha \end{deqn}
```

Продовження часткової нумерації можна одержати з оточенням `ddeqn` (`deqn` для 1-го номеру, `ddeqn` для всіх наступних):



Сайт ПНПУ

Головна

Зміст



Стор. 130 із 189

Назад

Перегляд

Закрити

Вихід

Оточення ddeqn

$$\cos(-\alpha) = \cos \alpha \quad (8b)$$

```
\begin{ddeqn} \cos(-\alpha) = \cos\alpha \end{ddeqn}
```

Під час набору масивів рівнянь використовуються оточення: `deqarr` для 1-го масиву (див. приклад 9b), `ddeqar` для наступних (див. 9c):

Оточення deqarr

$$\sin 2\alpha = 2 \sin \alpha \cos \alpha \quad (9a)$$

$$\cos 2\alpha = \cos^2 \alpha - \sin^2 \alpha \quad (9b)$$

```
\begin{deqarr} \sin 2\alpha & = & 2\sin \alpha \cos \alpha \quad \\\ \cos 2\alpha & = & \cos^2 \alpha - \sin^2 \alpha \label{d1} \end{deqarr}
```

Оточення ddeqar

$$\cos^2 \alpha = \frac{1 + \cos 2\alpha}{2} \quad (9c)$$

$$\sin^2 \alpha = \frac{1 - \cos 2\alpha}{2} \quad (9d)$$

```
\begin{ddeqar} \cos^2\alpha & = & \frac{1+\cos 2\alpha}{2} \label{d2} \\\ \sin^2\alpha & = & \frac{1-\cos 2\alpha}{2} \end{ddeqar}
```


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)


Стор. 131 із 189

[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

Щоб створити для формули певний номер чи ім'я слід скористатися пакетом `eqname`. Нехай необхідно мати формули за номером **2005** та ім'ям **ТАК!**.

Створення власного номера чи ім'я

$$1 + 1 = 2 \quad (2005)$$

$$1 + 2 = 3 \quad (\text{ТАК!})$$

$$1 + 3 = 4 \quad (2005)$$

```
\setcounter{equation}{2004} \begin{eqnarray}
1+1=2 \eqname{2005}\label{e1}\ 1+2=3 \eqname{ТАК!}\label{e2}\
1+3=4 \label{e3} \end{eqnarray}
```

Тепер можна робити посилання.

Посилання на власно створений номер

Запам'ятай (2005) і (ТАК!) ...

Запам'ятай (`\ref{e1}`) і (`\ref{e2}`) ...

Також слід запам'ятати наступне при використанні пакета `eqname`:

- `\label` повинна стояти після команди `\eqname`;
- для зміни номера формули в її поточній нумерації (у даному прикладі рівність **2005**), потрібно перед `\begin{eqnarray}` прописати `\setcounter{equation}{k-1}`, де **k** – необхідний номер;


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)


Стор. 132 із 189

[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

- пакет не працює з `leqno.sty` (опція `leqno` означає, що номери формул, генеровані оточеннями `equation` та `eqnarray`, будуть друкуватися не праворуч, а ліворуч).

Слід відмітити, що L^AT_EX 2_ε надає досить суттєві додаткові можливості. Крім трьох основних класів `article`, `report` і `book`, у яких з'явилися нові стильові опції, в основний набір L^AT_EX'a 2_ε входить клас `proc`, а також класи `amsart`, `amsproc` та `amsbook`, що відносяться до $\mathcal{A}\mathcal{M}\mathcal{S}$ -L^AT_EX'a.

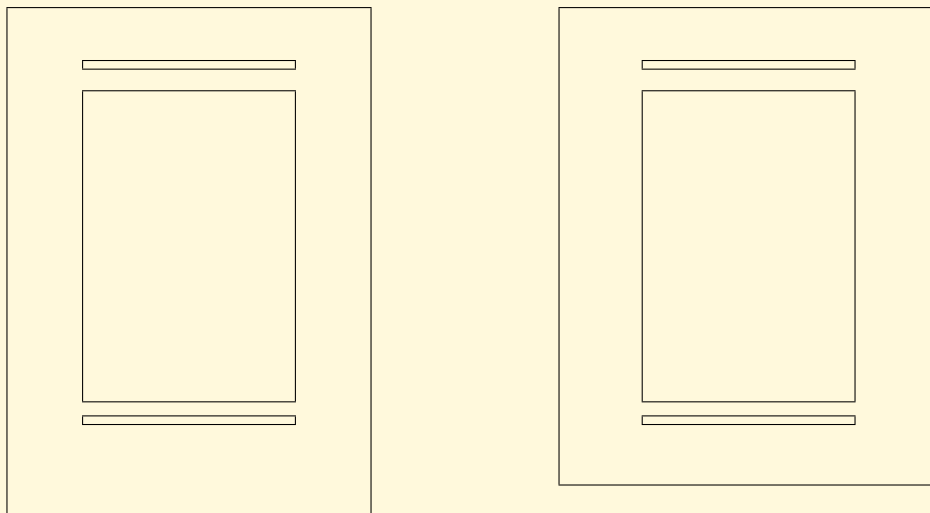
У L^AT_EX'ові 2.09 розміри тексту та полів устанавлюються автоматично і залежать тільки від стилю, що погано підходять до вітчизняних стандартів. L^AT_EX 2_ε є більш динамічним у даному відношенні: вказати стильову опцію, що задає формат використовуваного паперу, T_EX розрахує розміри тексту й полів так, щоб вони максимально відповідали цьому формату. Ці опції такі:

- `a4paper` 210 × 297 міліметрів – найбільш застосовується у нашій країні;
- `a5paper` 148 × 210 міліметрів;
- `b5paper` 176 × 250 міліметрів;
- `legalpaper` 8.5 × 14 дюймів;
- `executivepaper` 7.25 × 10.5 дюймів.

Якщо жодна із цих опцій не зазначена, L^AT_EX 2_ε вважає, що розмір паперу дорівнює 8.5 × 11 дюймів (див. табл. 2).

Коли L^AT_EX готує сторінку вихідного документу, він може поміщати текст у три її різні області, а саме, у заголовок, тіло й основу. Якщо створюється біжучий заголовок, то він разом з поточним номером сторінки відправляється в область її заголовка; якщо ж біжучий заголовок відсутній, то поточний номер сторінки міститься у центрі її області основи. Помітимо, що зноски – якщо вони є – розташовуються в області *тіла* вихідної сторінки. На мал. 5, стор.

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 133 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)



Європейський формат A4

Американський формат $11 \times 8\frac{1}{2}$ дюймів

Рис. 5. Розташування заголовка, тіла й основи сторінки.

134¹² показане розміщення цих трьох областей на аркуші паперу розміру $11 \times 8\frac{1}{2}$ дюйма й на форматі a4 у припущенні, що задано стиль документа `article` й опція `10pt` або `11pt`. На цих схемах зовнішній прямокутник – в обох випадках – це фізичний аркуш паперу, а три внутрішніх прямокутники (зверху вниз) – це заголовок, тіло й основа. Помітимо, що на аркуші формату

¹²Розмір аркуша паперу формату A4 дорівнює 297×210 мм. Стиль документа `article` є *однобічним* стилем. В однобічному стилі документа всі сторінки (концептуально) є правобічними або непарнономерованими. Це означає, що заголовок, тіло й основа як непарних, так і парних сторінок на фізичному аркуші паперу розташовуються щодо лівої й верхньої границі сторінки однаково.

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)

Стор. 134 із 189

[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

A4 тело області тексту не центровано. Це відбувається тому, що T_EX та L^AT_EX припускають, що друкуватися вихідний документ буде скоріше на стандарті, прийнятому в Америці, ніж на форматі a4, стандартному для Європи.

Клас документів `proc` призначений для друку статей, що входять у збірники доповідей на конференціях. У цьому класі друк проводиться завжди у два стовпчики, зі зменшеними полями. Опції `a5paper`, `b5paper`, `onecolumn` і `titlepage` у класі `proc` використовувати не можна.

При користуванні класом `proc` унизу кожної сторінки буде надруковане слово Page (<сторінка>) і номер сторінки. Якщо ви хочете, щоб замість Page друкувалося щось інше, скажемо, <с.>, то потрібно в преамбулі перевизначити команду `\pagename`, тобто написати:

```
\renewcommand{\pagename}{с.}
```

Якщо передбачається розташувати текст так, щоб він розміщувався паралельно широкому, а не вузькому краю паперу, то можна вказати опцію `landscape`: у цьому випадку T_EX буде обчислювати розміри тексту й полів, вважаючи, що ширина й висота аркуша папери помінялися ролями. Слід зауважити, що задання опції `landscape` саме по собі текст на 90° не поверне: він буде зверстаний T_EX'ом виходячи з відповідних розмірів, але далі необхідно мати принтер або dvi-драйвер, здатні забезпечити друк тексту в такій орієнтації (опція `landscape` не здатна перетворити вузький принтер у широкий). За замовчуванням вважається, що рядки паралельні вузькому краю аркуша.

У класів `report` і `book` є опції `openright` і `openany`. Якщо зазначено опцію `openright`, то кожний розділ починається обов'язково з непарної сторінки (при необхідності друкується додаткова порожня сторінка; на розвороті непарна сторінка буде правою). Опція `openany`, дозволяє починати розділ як з парної, так і з непарної сторінки, і зайвих порожніх сторінок заради початку розділу L^AT_EX не робить. За замовчуванням у класі `report` L^AT_EX діє так, ніби було встановлено `openany`, а в класі `book` — `openright`. У L^AT_EX'ові 2.09 немає легального способу вплинути на те, з яких сторінок будуть починатися розділи.

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 135 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

Таблиця 19. Стили оформлення сторінки

Опція	Стиль оформлення
<code>empty</code>	Немає ні колонтитулів, ні номерів сторінок.
<code>plain</code>	Номера сторінок ставляться внизу по середині рядка, колонтитулів немає.
<code>headings</code>	Присутні колонтитули, що містять номери сторінок.
<code>myheadings</code>	Присутні колонтитули, що містять номери сторінок, текст (у стандартному випадку це номери й назви розділів документа). Відмінність від попереднього випадку в у тім, що текст колонтитулів не породжуються L ^A T _E X'ом автоматично, а задається користувачем у явному вигляді.

38. Стиль оформлення сторінки

Для задання стилю оформлення сторінки в L^AT_EX'ові передбачена команда `\pagestyle`, що має один обов'язковий аргумент — слово, що позначає цей стиль. При користуванні стандартними стилями L^AT_EX'a це слово повинне бути одним з наведених у табл. 19. Якщо основний стиль – `article`, то за замовчуванням сторінки оформляються стилем `plain`, у двох інших основних стилях – стилем `headings`.

Поряд з командою `\pagestyle`, що задає стиль оформлення всіх сторінок, є також команда `\thispagestyle`, що задає стиль оформлення однієї окремо взятої сторінки. Вона приймає такі ж аргументи, що й `\pagestyle`, але оформлення буде здійснюватися тільки тієї сторінки, на яку потрапив текст, що оточує цю команду. Заздалегідь угадати, на яку сторінку потрапить даний фрагмент тексту, звичайно неможливо. Щоб мати від цієї команди передбачувані результати, необхідно її використовувати після `\newpage` або `\clearpage`.


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)


Стор. 136 із 189

[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

39. Поля, розмір сторінки

Стандартні L^AT_EX'овські стилі самостійно встановлюють значення ширини й висоти сторінки.

Розміри тексту на сторінці, полів та ін. задаються різними параметрами зі значенням довжини, які можна змінювати лише в преамбулі документа.

Ширина тексту на сторінці задається параметром `\textwidth`. Якщо набір здійснюється у два стовпчики, то `\textwidth` містить у собі ширину обох стовпчиків і проміжок між ними. Якщо потрібно, щоб ширина тексту на сторінці рівнялася 7 сантиметрів, то в преамбулі потрібно написати так:

```
\textwidth=7cm
```

При зміні ширини тексту часто доводиться міняти й поля. У L^AT_EX'ові передбачений параметр, що регулює розмір лівого поля. Спосіб завдання лівого поля залежить від того, чи є набір у даному стилі <двостороннім> чи ні.

При односторонньому наборі величина лівого поля задається параметром `\oddsidemargin`. Поле в даному випадку відраховується не від самого краю аркуша, а від попередньо зробленого відступу в один дюйм. Отже, якщо в преамбулі написано

```
\oddsidemargin=0pt
```

то текст буде починатися на відстані один дюйм від краю, а якщо буде задано

```
\oddsidemargin=4,6mm
```

то відступ від краю аркуша буде 30 мм (один дюйм дорівнює 2.54 см). При наданні параметру `\oddsidemargin` негативних значень, відстань від краю аркуша до початку тексту буде, відповідно, менше дюйма.

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 137 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

Параметр `\oddsidemargin` також використовується при двосторонньому наборі, але зміст його інший: він задає розміри лівого поля тільки для сторінок із непарними номерами. Для сторінок з парними номерами розмір лівого поля задається параметром `\evensidemargin`.

Верхнє поле коригується параметром `\topmargin`. Як і у випадку з лівим полем, це відстань не безпосередньо від краю аркуша, а від лінії, паралельної краю й віддаленої від нього на один дюйм. При цьому треба усвідомлювати, що `\topmargin` — це відстань до колонтитула. Якщо колонтитула немає (непередбачено стилем), то вгорі сторінки буде залишено порожній простір. Він відводиться на колонтитул, розмір якого задається параметром `\headheight`, плюс порожній простір, рівний відступу між колонтитулом і основним текстом (позначається `\headsep`).

Висота тексту задається параметром `\textheight` який не враховує ні номери сторінок, ні колонтитули. Отже, слід пам'ятати, Якщо вони передбачені стилем, повна висота тексту на сторінці буде більше, ніж `\textheight`.

Висоту сторінки можна змінювати, привласнюючи в преамбулі параметру `\textheight` нове значення. Але якщо стилем передбачена однакова висота сторінок, то висоту тексту не можна встановлювати довільно: необхідно погодити її значення з параметрами `\topskip` і `\baselineskip`. Перший із цих параметрів визначає відстань від низу першого рядка до <верхнього обрізу> основного тексту сторінки, параметр `\baselineskip` визначає відстань між рядками й залежить від використовуваного шрифту. Значення `\textheight` слід вибрати таким, щоб відношення

$$\frac{\text{\textheight} - \text{\topskip}}{\text{\baselineskip}}$$

було цілим числом.

У стандарті L^AT_EX `\topskip` завжди дорівнює 10 пунктам, а `\baselineskip` — 12.


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)


Стор. 138 із 189

[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

40. Розділи документа

Команди рубрикації надають можливість розбивати текст на розділи й підрозділи різних рівнів, а саме:

<code>\part</code>	— частина	<code>\subsubsection</code>	— підпідрозділ
<code>\chapter</code>	— глава	<code>\paragraph</code>	— параграф
<code>\section</code>	— розділ	<code>\subparagraph</code>	— підпараграф
<code>\subsection</code>	— підрозділ	<code>\appendix</code>	— додаток

У цьому переліку кожна наступна команда позначає більше дрібний підрозділ, ніж попередня. Слід зауважити, що команда `\chapter` (<глава>) у стилі `article` не визначена (завдяки цій обставині статтю легко переробити в главу книги), інші команди визначені у всіх трьох стандартних стилях.

Стандартні стилі забезпечують нумерацію розділів, при якій більше дрібний розділ <підлеглий> більшому: коли, наприклад, починається новий розділ `\section`, нумерація розділів `\subsection` і більше дрібних починається заново. Виключенням із цього правила є команда `\part` (<частина>): якщо частина 2 кінчається главою 5, то перша із глав частини 3 буде мати номер 6, а не 1. При модифікації стандартних стилів можна міняти як принцип нумерації розділів, так і вид цих <номерів> на друкові (наприклад, якщо ми захочемо, щоб розділи позначалися послідовними буквами алфавіту).

У розділах, створюваних описаними вище командами, перший абзац набирається без абзацного відступу (за винятком самого дрібного розділу `\subparagraph`).

Команди рубрикації (крім `\appendix`, формат і дія якої залежить від базового стилю документа) мають однаковий формат із одним обов'язковим параметром, що є заголовок рубрики. Виключенням є команда `\part`, заголовок у якій не обов'язковий.

Зразок формату:


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)

[Стор. 139 із 189](#)
[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

```
\section[заголовок для змісту]{заголовок}
```

Необов'язковий параметр використовується рідко. У ньому вказується короткий заголовок, що буде використаний у змісті й колонтитулі – біжучому заголовку зверху кожної сторінки. Якщо цього параметра немає, то обов'язковий аргумент команди рубрикації буде виданий і в тексті, і в змісті, якщо цього вимагає формат сторінки, та колонтитул.

Якщо необхідно мати заголовок без номера, користуються *-формою команди рубрикації:

```
\chapter*[заголовок для змісту]{заголовок}
```

Розміри шрифтів для заголовків рубрик, наявність біжучого заголовка, а також розміщення у змісті рубрик, заданих в *-формі, – визначаються у файлі опису стилю. Базові стилі не виводять біжучого заголовка і автоматично вносять у зміст тільки нумеровані рубрики.

Щоб розпочати розділ, озаглавлений <Теорія груп Лі>, для цього у тексті пишуть так:

```
\section{Теорія груп Лі}
```

Команда `\section` приймає один обов'язковий аргумент – назва розділу (ця ж назва буде і в колонтитулах, якщо такі передбачені стилем, і в змісті, якщо дано команду <створити зміст>). Проміжки між розділами та їхня нумерація виставляться автоматично.

Крім обов'язкового аргументу в команді `\section` передбачений і необов'язковий. Необов'язковий аргумент іде перед обов'язковим; у ньому записується варіант заголовка, призначений для змісту й колонтитулів. Наприклад:

```
\section[Групи Лі]{Теорія груп Лі}
```

Розділ позначимо через `\label`. Тоді команда `\ref` буде видавати його номер.

Команда `\section*`, що має тільки обов'язковий аргумент, починає розділ, не нумеруючи його. Такого типу рубрикації до змісту й колонтитулів не попадають (докладніше див. с. 145).


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)

[Стор. 140 із 189](#)
[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

41. Зміна стандартних заголовків

Якщо необхідно щоб у документі замість слова <Chapter> друкувалося <Глава>, то для цього в преамбулу документу треба внести наступну команду:

```
\renewcommand{\chaptername}{Глава}
```

Команда `\renewcommand`, використовується для перевизначення значень уже існуючих команд.

Далі наведений список заголовків разом з їх українськими перекладами й командами, які треба перевизначити за допомогою `\renewcommand` для того, щоб при друкуванні з'являлися саме ці переклади.

<code>\chaptername</code>	Chapter	Глава
<code>\contentsname</code>	Contents	Зміст
<code>\listfigurename</code>	List of Figures	Списки ілюстрацій
<code>\listtablename</code>	List of Tables	Список таблиць
<code>\abstractname</code>	Abstract	Анотація
<code>\refname</code>	References	Список літератури
<code>\bibname</code>	Bibliography	Список літератури
<code>\indexname</code>	Index	Предметний покажчик
<code>\figurename</code>	Figure	Рис.
<code>\tablename</code>	Table	Таблиця
<code>\partname</code>	Part	Частина
<code>\appendixname</code>	Appendix	Додаток


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)


Стор. 141 із 189

[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

42. Анотація, додаток

У стилях (класах) `article` і `report` передбачена можливість оформити анотацію до всього документа. Це робиться за допомогою оточення `abstract`. Як правило, текст анотації розміщують до початку основного тексту. Він розташовується між командами `\begin{abstract}` і `\end{abstract}`. Цей текст буде автоматично озаглавлений `<Abstract>`, якщо не міняти стилю.

Команда `\appendix` означає, що із цього місця починається додаток до документа. Цією командою передбачено:

- нову нумерацію розділів документа;
- для `<самих великих>` розділів документа (`\section` у стилі (класі) `article` і класі `proc`, `\chapter` у двох інших основних стилях (класах)) передбачена нумерація не цифрами, а прописними латинськими літерами;
- при визначенні команди `\chapter`, глави називати не `<Chapter>`, а так, як визначено в команді `\appendixname` ;
- при використанні стильової опції `russcorr` встановлюється нумерація самих великих розділів документа прописними російськими літерами (замість прописних латинських в стандартних стилях).

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 142 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

43. Титульний аркуш, зміст, список літератури

Титульний лист документу в цілому генерується за допомогою команди

```
\maketitle
```

У зв'язку з тим, що команда `\maketitle` генерує текст, її не можна розміщувати в преамбулі документу.

Створення титульного аркушу за допомогою команди `\maketitle` безпосередньо пов'язане зі стилем документу. Так, для стилів `report` і `book` це можливо завжди, а для стилю `article` – якщо зазначено стильову опцію `titlepage`).

Якщо титульний аркуш не передбачений, то команда `\maketitle` розмістить задану вами інформацію про автора, заголовок і інше на першій сторінці (вибравши підходящі шрифти й зробивши належні відступи між титульною інформацією й текстом).

Зміст титульного аркушу створюється командами

```
\author, \title, \date,
```

які можна вказувати в будь-якому порядку, але обов'язково до команди `\maketitle`, що генерує заголовок до всього документу.

Автор задається за допомогою команди `\author`, що приймає єдиний обов'язковий аргумент – прізвище, ім'я та по батькові автора (у бажаному вигляді). Якщо авторів декілька, їх імена повинні бути розділені командою `\and`. У факультативному аргументі, тобто між самою командою `\author` і прізвищем, ім'ям та по батькові автора, що знаходяться у фігурних дужках, можна вказувати коротку форму обов'язкового аргументу з метою розміщення в колонтитулах, наприклад,

```
\author[Ю. Подошвелев]{Ю.Г. Подошвелев}
```

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 143 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

Заголовок створюється за допомогою команди `\title`. Якщо заголовок довгий, то можна самому розбити його на рядки за допомогою команди `\\`; якщо цього не зробити, заголовок буде розбито на центровані рядки автоматично, так само, як абзац в оточенні `center`. Крапку в кінці заголовку ставити непотрібно. У допоміжному аргументі вказується коротка назва статті для використання в колонтитулі.

Наступний елемент інформації для титулу – команда `\date`. Вона має один обов'язковий аргумент, у якому можна задати будь-який текст (наприклад, дату, згідно з перекладом слова *date*), що буде розміщений на титульному аркуші в одній або декількох центрованих рядках, наприклад `\date{22 січня 2006}`. Якщо дати вказувати не потрібно (або взагалі не потрібно використовувати команду `\date`), то застосовують команду `\date{}`. Але якщо цієї команди не вказати, то L^AT_EX надрукує на титулі дату свого запуску, причому англійською мовою.

Можна не використовувати стиль оформлення титульного аркушу, що пропонується по замовчуванню L^AT_EX'ом, а створити індивідуальний титульний аркуш скориставшись оточенням `\begin{titlepage} i \end{titlepage}`.^[10]

L^AT_EX створює зміст, беручи заголовки розділів і номери сторінок з попереднього проходу по документу. Команда

```
\tableofcontents
```

вставляє зміст у те місце, де вона викликана. Щоб одержати правильний зміст, новий документ повинен бути оброблений L^AT_EX двічі. Саме після другого запуску команда `\tableofcontents` опрацює весь документ і запише результат у файл із розширенням `*.toc`, а потім згенерує зміст документу. В особливих випадках може бути необхідний і третій прохід (L^AT_EX попереджує про це уразі потреби).

Слід відмітити, що команди `\listoffigures` і `\listoftables` створюють список малюнків і список таблиць, відповідно. Вони працюють так само, як команда `\tableofcontents`, за винятком того, що L^AT_EX при цьому використовує інші робочі файли.

Як уже відомо, всі команди секціонування існують також у варіантах із зірочкою. Такий ва-

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 144 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

ріант виходить додаванням * до імені команди. Вони генерують заголовки розділів, що не нумеруються й не включаються до змісту. Наприклад, якщо замість команди `\chapter{Умовні позначення}` задати команду `\chapter*{Умовні позначення}`, то пункт рубрикації |Умовні позначення| до змісту не попаде. Примусово розмістити пункти такого типу у змісті можна скориставшись командою

```
\addcontentsline,
```

яку необхідно вказати після команди рубрикації, що містить *.

Це робиться наступним чином

```
\tableofcontents

\chapter*{Умовні позначення}
\addcontentsline{toc}{chapter}{Умовні позначення}
... текст переліку умовних позначень ...
\chapter*{Вступ}
\addcontentsline{toc}{chapter}{Вступ}
... текст вступу ...
```

Зазвичай заголовки розділів з'являються у змісті точно в тому ж вигляді, у якому їх було введено в текст. Іноді це неможливо через те, що заголовок занадто довгий для змісту. Елемент змісту може в цьому випадку вказуватися необов'язковим аргументом перед заголовком.

```
\chapter[Довгі заголовки - не проблема!]{Цей заголовок занадто довгий для
відображення у змісті}
```

L^AT_EX надає можливість оформити список літератури, елементи якого нумеруються авто-


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)


Стор. 145 із 189

[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

матично; у тексті при цьому можна посилатися на ці номери.

Список літератури оформляється як оточення `thebibliography`. Це оточення має обов'язковий аргумент – номер елемента бібліографії.

Кожне джерело вводиться командою `\bibitem`. У неї є один обов'язковий аргумент – умовна позначка, що являє собою будь-яку послідовність із букв і цифр.

У тексті посилання на джерело робиться за допомогою команди `\cite`. У неї є обов'язковий аргумент – умовна позначка того джерела, на яке виконується посилання. Можна посилатися відразу на декілька джерел – для цього в аргументі команди `\cite` треба вказати через кому позначки відповідних джерел посилання.

Списку літератури в цілому L^AT_EX автоматично дає заголовок, обумовлений командою `\refname`, який можна перевизначити.

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 146 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

44. Використання баз даних при складанні бібліографії

Великий список літератури або список, основна частина якого інтенсивно використовується, доцільно зберігати в базі даних.

Очевидна перевага роботи з базами даних полягає в тому, що бібліографічна інформація набирається тільки один раз, при занесенні її в базу даних. Кожний пункт бібліографії оформляється як окремий запис із унікальною міткою. Вставка цього пункту в список літератури будь-якого документу з посиланням на нього в тексті або без посилання виконується простими по форматі командами, що вказують цю мітку.

Складання списку літератури здійснюється за допомогою двох описових команд, одна із яких визначає стиль бібліографічного списку, а друга – місце розташування списку (це місце видачі даної команди) і використовувані бази даних.

44.1. Команди роботи з бібліографією, що зберігається в базі даних

Витяг інформації в конкретну публікацію виконується за допомогою вже відомої користувачу команди `\cite` і команди `\nocite`. Як обов'язкові параметри цих команд використовуються мітки записів бібліографічних баз даних.

```
\cite [текст]{список міток}
```

Команда посилання на один або кілька пунктів списку літератури. Обов'язковий параметр повинен містити одну або кілька міток, розділених комами або тире. Необов'язковий параметр *текст* буде представлений як коментар до пункту списку літератури.

```
\nocite [текст]{список міток}
```

Команда вказує бібліографічну інформацію, на яку немає посилань у тексті, але яку варто

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 147 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

внести в список літератури. Обов'язковий параметр повинен містити одну або кілька міток, розділених комами. Необов'язковий параметр *<текст>* також буде представлений як коментар до пункту списку літератури.

При роботі з бібліографічними базами даних замість командних дужок `\thebibliography` використовуються дві інші команди. Перша з них задає стиль бібліографії, друга вказує бази даних, з яких будуть вибиратися необхідні пункти (ті, чиї мітки зазначені в документі чи командах `\cite` і `\nocite`).

```
\bibliographystyle{<стиль>}
```

задає стиль бібліографії, що визначає вид списку літератури – як стиль документа визначає формат документа. Команда `\bibliographystyle` повинна обов'язково з'являтися *після* команди `\begin{document}`.

Файли опису бібліографічних стилів мають розширення `.bst`. Доступні наступні стандартні бібліографічні стилі.

Стиль plain

Елементи відсортовані за алфавітом і позначені числами, імена й прізвища авторів зазначені повністю.

Стиль unsrt

Порядок розташування елементів відповідає розташуванню посилань у тексті, імена й прізвища авторів зазначені повністю.

Стиль alpha

Елементи відсортовані за алфавітом і позначені символічними мітками, як правило складеними з використанням імені автора й року публікації.

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 148 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

Стиль abbr

елементи відсортовані за алфавітом і позначені числами, прізвища авторів, назва місяця й журналу дані в скороченому вигляді.

```
\bibliography{⟨файл1⟩,⟨файл2⟩...}
```

Обов'язковий параметр містить імена (або ім'я) файлів використовуваних бібліографічних баз даних. Імена файлів варто вказувати *без розширення*.

L^AT_EX дозволяє використати одночасно кілька баз даних.

44.2. Процедура одержання списку літератури з бібліографічних баз даних

Процедура одержання бібліографії з файлів баз даних складається з декількох кроків.

- При обробці документа L^AT_EX'ом створюється робочий файл із розширенням `aux`, що буде містити всі ключі для посилань і всі аргументи команд `\bibliographystyle`, `\cite` і `\nocite`, `\bibliography`. Цей файл варто обробити програмою , що викликається командою `\normalsize`

```
BIBTEX ⟨ім'я файлу без розширення⟩
```

Програма читає інформацію з файлу `aux` і створює однойменний файл із розширенням `bb1`, що містить команди L^AT_EX'а для одержання списку літератури. Одночасно створюється протокольний файл, що має розширення `blg`.

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 149 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

- При повторній обробці вихідного файлу (після роботи програми BibT_EX), L^AT_EX читає bbl-файл (по команді `\bibliography`) і вставляє в текст список літератури.
- Нарешті при третьому проході L^AT_EX виконає розміщення посилань.

44.3. Формат файлу бібліографічних даних

База даних – це файл, розділений на записи. Кожний запис зберігає бібліографічну інформацію однієї роботи.

Файл бази даних для L^AT_EX'а готується будь-яким редактором. Імена файлів бібліографічних баз даних повинні мати розширення `bib`.

Запис починається символом `@`. За ним слідує вказівка *типу* роботи, що представляє запис – `article` (стаття), `book` (книга) та інше.

За типом запису у фігурних дужках треба власна інформація про роботу. Вона складається з *ідентифікатора* і *полів*.

Ідентифікатор є унікальна мітка запису. Саме він повинен бути використаний як обов'язковий аргумент команди `\cite` або `\nocite` при посиланні на дану роботу.

Кожне поле складається з імені, знака рівності й текстового рядка, обмежувачем якого є фігурні дужки або подвійні лапки. Якщо текст повністю складається із цифр, дужки або лапки можуть бути опущені.

Поля класифікуються в такий спосіб:

обов'язкове відсутність поля викличе повідомлення про помилку;

необов'язкове інформація в цьому полі буде використана, якщо вона присутня в записі, але її відсутність не викличе повідомлення про помилку чи проблем при друці. Деякі бібліографічні стилі можуть залишати без уваги певні поля;

ігнороване якщо BibT_EX не розпізнає поле як обов'язкове або необов'язкове, то це поле ігнорується. Тому користувач може включати в запис будь-які поля, для збе-

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 150 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

рігання потрібної йому інформації (наприклад, **ANNOTATION** – поле, у якому можна зберігати анотацію роботи).

Поля розділяються комами. Порядок їхнього уведення неістотний.

VibTeX не розрізняє прописних і малих літер в ідентифікаторах записів або назвах полів.

44.3.1. Типи записів

Тут перераховані основні типи записів із вказівкою їх *обов'язкових* і *необов'язкових* полів, які використовуються в стандартних бібліографічних стилях.

- article — журнальна стаття.

Обов'язкові поля

author (автор)
title (заголовок)
journal (журнал)
year (рік)

Необов'язкові поля

volume (том)
number (номер)
pages (сторінки)
month (місяць)
note (примітка)

- book – книга.

Обов'язкові поля

author (автор) / editor (редактор)
title (заголовок)
publisher (видавництво)
year (рік)

Необов'язкові поля

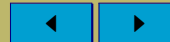
volume (том)
series (серія)
address (адреса)
edition (видання)
month (місяць)
note (примітка)



Сайт ПНПУ

Головна

Зміст



Стор. 151 із 189

Назад

Перегляд

Закрити

Вихід

- booklet – буклет (друкована праця в плетінні без посилання на видавництво).

Обов'язкові поля

title (заголовок)

Необов'язкові поля

author (автор)

howpublished (спосіб видання)

address (адреса)

month (місяць)

year (рік)

note (примітка)

- conference – конференція. Див. нижче inproceedings.

- inbook – витримка із книги.

Обов'язкові поля

author (автор) / editor (редактор)

title (заголовок)

publisher (видавництво)

chapter (глава) і/або pages (сторінки)

year (рік)

Необов'язкові поля

volume (том)

series (серія)

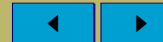
address (адреса)

edition (видання)

month (місяць)

note (примітка)

- incollection – витримка із книги із заголовком.

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)

Стор. 152 із 189

[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

Обов'язкові поля

`author` (автор)
`title` (заголовок)
`booktitle` (заголовок книги)
`publisher` (видавництво)
`year` (рік)

- `inproceedings` – стаття в працях конференції.

Обов'язкові поля

`author` (автор)
`title` (заголовок)
`booktitle` (загальний заголовок)
`year` (рік)

- `manual` – технічна документація.

Обов'язкові поля

`title` (заголовок)

Необов'язкові поля

`editor` (редактор)
`chapter` (глава)
`pages` (сторінки)
`address` (адреса)
`month` (місяць)
`note` (примітка)

Необов'язкові поля

`editor` (редактор)
`organization` (організація-спонсор)
`pages` (сторінки)
`publisher` (видавництво)
`address` (адреса)
`month` (місяць)
`note` (примітка)

Необов'язкові поля

`author` (автор)
`organization` (організація)
`edition` (видання)
`address` (адреса)
`year` (рік)
`month` (місяць)
`note` (примітка)

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 153 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

- masterthesis – дисертація магістра.

Обов'язкові поля

author (автор)
title (заголовок)
school (навчальний заклад)
year (рік)

- misc – різне.

Обов'язкові поля

- phdthesis – докторська дисертація.

Обов'язкові поля

author (автор)
title (заголовок)
school (навчальний заклад)
year (рік)

- proceedings – праці конференції.

Необов'язкові поля

address (адреса)
month (місяць)
note (примітка)

Необов'язкові поля

author (автор)
title (заголовок)
howpublished (спосіб видання)
year (рік)
month (місяць)
note (примітка)

Необов'язкові поля

address (адреса)
month (місяць)
note (примітка)



Сайт ПНПУ

Головна

Зміст



Стор. 154 із 189

Назад

Перегляд

Закрити

Вихід

Обов'язкові поля

`title` (заголовок)
`year` (рік)

Необов'язкові поля

`editor` (редактор)
`publisher` (видавництво)
`organization` (організація-спонсор)
`address` (адреса)
`month` (місяць)
`note` (примітка).

- `techreport` – звіт, що видається навчальною або якою-небудь іншою установою й звичайно нумерований по серіях.

Обов'язкові поля

`author` (автор)
`title` (заголовок)
`institution` (установа)
`year` (рік)

Необов'язкові поля

`type` (тип звіту)
`number` (номер)
`month` (місяць)
`note` (примітка)

- `unpublished` – не опублікований офіційно документ, що має протек ім'я автора й назву.

Обов'язкові поля

`author` (автор)
`title` (заголовок)
`note` (примітка)

Необов'язкові поля

`month` (місяць)
`year` (рік)

Як додаток до перерахованих вище полів, кожний тип запису може містити ще й необов'язкове поле `key`, що використовується деякими стилями для сортування.

44.3.2. Преамбула

База даних може починатися з особливого типу запису – преамбули. У цьому записі користувач може визначити засобами L^AT_EX'а нові макрокоманди, які полегшать побудову кон-


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)


Стор. 155 із 189

[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

кретного списку літератури.

Означаючи нові команди їх розміщують в загальні групові дужки. Кожне визначення береться в подвійні лапки, при цьому одне від одного відокремлюються символом #.

```
@preamble{|\newcommand{\printfirst}[2]{#1} |
# |\def\n#1{\hbox{#1}}|
# |\newcommand{\singleletter}[1]{#1} |
# |\newcommand{\switchargs}[2]{#2#1} | }
```

44.3.3. Символьні послідовності

Ще один допустимий тип запису призначений для полегшення уведення повторюваних послідовностей символів, наприклад, інформації про автора, що багато разів включений у базу даних робіт.

Аргумент запису повинен знаходитися в круглих дужках і містити назву послідовності та її значення, що з'єднуються знаком рівності. Значення послідовності укладається в подвійні обмежувачі, наприклад, лапки.

```
@STRING( Подошвелев = |Юрій Георгійович Подошвелев| )
```

44.3.4. Уведення інформації про авторів і редакторів

При уведенні інформації в поля `author` і `editor` важливо дотримувати наступних правил.

- для запису повних даних – прізвища, імені й, можливо, по батькові – існує два формати:

```
|Юрій Георгійович Подошвелев| |Подошвелев, Юрій Георгійович|
```


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)


Стор. 156 із 189

[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

У першому форматі, у якому відсутня кома, прізвищем вважається останнє слово, а попередні будуть уведені в список літератури перед прізвищем у вигляді початкових букв і крапок за ними. При другому форматі прізвищем вважається слово, відділене комою, а інші слова будуть представлені лише ініціалами;

- якщо в базу даних вводяться прізвища й ініціали авторів, то між ініціалами повинен бути пробіл:

```
|Ю. Г. Подошвелев | |Подошвелев, Ю. Г. |
```

- якщо авторів декілька, між їхніми іменами варто вставляти символи `and`;
- якщо авторів дуже багато, в список їх можна додавати зі словами `and others`, які в списку літератури дадуть стандартне позначення `et al.`

44.4. Кілька зауважень

- Матеріал нашої книги по використанню бібліографічних баз даних є лише загальне знайомство з найпростішою технікою роботи з ними. Для вивчення всіх можливостей програми BibTeX варто ознайомитися з технічною документацією по цій програмі.
- У цей час немає закінченого програмного продукту, що представляє українізовані бібліографічні стильові файли з урахуванням особливостей складання списків літератури в публікаціях українських видавництв. Тому в представленому прикладі беруть участь лише основні пункти літератури.

Як приклад наводимо частину бази даних, що є основою списку літератури нашої книги. Стильовий файл компонування даних з бази даних у список літератури взято `gost2003s`, який було модифіковано з метою її викладення до останніх вимог ВАК.

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 157 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

```
@book{Stoljarov,  
author = {Столяров, А. В.},  
title = {Сверстай диплом красиво \LaTeX\ за три дня},  
year = {2010},  
isbn = {978-5-317-03440-5},  
publisher = {МАКС Пресс},  
address = {М.},  
numpages = {101},  
pagetotal = {101},  
language = {russian},  
OPTnote = {},  
url = {http://www.stolyarov.info/books/pdf/latex3days.pdf}  
}
```

```
@book{Rog,  
author = {Роженко, А. И.},  
title = {Искусство верстки в \LaTeX'e. - 3-е изд.},  
year = {2005},  
isbn = {5-901548-25-6},  
publisher = {ИВМиМГ СО РАН},  
address = {Новосибирск},  
numpages = {398},  
pagetotal = {398},  
language = {russian}  
}
```

```
@book{Gussens-1999,  
author = {М. Гуссенс and Ф. Миттельбах and А. Самарин},  
title = {Путеводитель по пакету \LaTeX{} и его
```

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 158 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

```
расширению \LaTeXe: Пер. с англ.},
year = {1999},
isbn = {5-03-003325-4},
publisher = {Мир},
address = {М.},
numpages = {606},
pagetotal = {606},
language = {russian},
OPTnote = {},
url = {elibrary.bsu.az/kitablar/1020.PDF}
}
```

```
@book{Lvovskij,
title={Набор и вёрстка в системе \LaTeX{.} - 3-е изд.},
author={Львовский, С. М.},
year={2003},
isbn = {5-940570-91-7},
publisher = {МЦНМО},
address = {М.},
numpages = {448},
pagetotal = {448},
language = {russian},
OPTnote = {},
url = {www.mccme.ru/free-books/llang/newllang.pdf}
}
```

```
@book{Knuth1,
title={Все про METAFONT: Пер. с англ. М.Р. Саит-Аметова.},
author={Кнут, Д. Э.},
```

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 159 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)


```
year={2003},  
isbn = {5-8459-0382-3.},  
publisher = {Вильямс},  
address = {М.},  
numpages = {560},  
pagetotal = {560},  
language = {russian},  
OPTnote = {},  
url = {publ.lib.ru/ARCHIVES/K/KNUT_Donal'd_Ervin/_Knut_D.E..html#006}  
}
```

```
@book{Knuth2,  
title={Все про \TeX: Пер. с англ. Л.Ф. Козаченко.},  
author={Кнут, Д. Э.},  
year={2003},  
isbn = {5-8459-0382-3},  
publisher = {Вильямс},  
address = {М.},  
numpages = {543},  
pagetotal = {543},  
language = {russian},  
OPTnote = {},  
url = {publ.lib.ru/ARCHIVES/K/KNUT_Donal'd_Ervin/_Knut_D.E..html#006}  
}
```

```
@book{Knuth3,  
title={Искусство программирования. Том 1. Основные алгоритмы: Пер. с англ. Ю.В. Козаченко.},  
author={Кнут, Д. Э.},  
year={2002},
```

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 160 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

```
isbn = {0-201-89683-4},
publisher = {Вильямс},
address = {М.},
numpages = {720},
pagetotal = {720},
language = {russian},
OPTnote = {},
url = {publ.lib.ru/ARCHIVES/K/KNUT_Donal'd_Ervin/_Knut_D.E..html#006}
}
```

```
@book{Knuth4,
title={Искусство программирования. Том 2. Получисленные алгоритмы:
Пер. с англ. Ю.В. Козаченко.},
author={Кнут, Д. Э.},
year={2007},
isbn = {0-201-89684-2},
publisher = {Вильямс},
address = {М.},
numpages = {832},
pagetotal = {832},
language = {russian},
OPTnote = {},
url = {publ.lib.ru/ARCHIVES/K/KNUT_Donal'd_Ervin/_Knut_D.E..html#006}
}
```

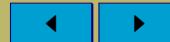
```
@book{Knuth5,
title={Искусство программирования. Том 3. Сортировка и поиск:
Пер. с англ. Ю.В. Козаченко.},
author={Кнут, Д. Э.},
```

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 161 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

```
year={2007},  
isbn = {0-201-89685-0},  
publisher = {Вильямс},  
address = {М.},  
numpages = {824},  
pagetotal = {824},  
language = {russian},  
OPTnote = {},  
url = {www.proklondike.com/books/thalg/knut_art3.html}  
}
```

```
@book{Knuth6,  
title={Искусство программирования. Том 4. Комбинаторные алгоритмы:  
Пер. с англ. И.В. Красикова.},  
author={Кнут, Д. Э.},  
year={2013},  
isbn = {978-5-8459-1744-7},  
publisher = {Вильямс},  
address = {М.},  
numpages = {960},  
pagetotal = {960},  
language = {russian},  
OPTnote = {},  
url = {publ.lib.ru/ARCHIVES/K/KNUT_Donal'd_Ervin/_Knut_D.E..html#006}  
}
```

```
@book{Gussens1,  
author = {М. Гуссенс and С. Ратц and Ф. Миттельбах},  
title = {Путеводитель по пакету \LaTeX{} и его
```

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 162 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

графическим расширениям. Иллюстрирование документов при помощи \TeX'a и PostScript'a: Пер. с англ.},

```
year = {2002},  
isbn = {5-03-003388-2},  
publisher = {Мир: Бином ЛЗ},  
address = {М.},  
numpages = {621},  
pagetotal = {621},  
language = {russian},  
OPTnote = {},  
url = {www.books4all.ru/description/1589.html}  
}
```

```
@book{Baldin,  
author = {Балдин, Е. М.},  
title = {Компьютерная типография \LaTeX.},  
year = {2008},  
isbn = {978-5-9775-0230-6},  
publisher = {БХВ-Петербург},  
address = {Новосибирск},  
numpages = {304},  
pagetotal = {304},  
language = {russian},  
url = {http://elib.sbras.ru:8080/jspui/bitstream/SBRAS/9015/1/ctex.pdf}  
}
```

Оточення `thebibliography` генерує бібліографію. Кожний елемент починається з

```
\bibitem{маркер}
```

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[◀](#) [▶](#)[◀](#) [▶](#)[Стор. 163 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

Потім маркер використовується для подальших посилань на книгу, статтю або працю.

```
\cite{маркер}
```

Елементи бібліографії нумеруються автоматично. Відмітимо, що параметр після команди `\begin{thebibliography}` устанавлює максимальну ширину цих номерів. [12]

Для виклику Bib_TE_Xа достатньо замінити оточення `thebibliography` командами

```
\bibliographystyle{gost2003s} %% стилевий файл для оформлення згідно ГОСТу  
\bibliography{biblio} %% ім'я бібліографічної бази (bib-файлу)
```

Представимо стильові файли, розроблені М. Поляковим:

gost71u.bst — ГОСТ 7.1-1984, без сортування за авторами;

gost71s.bst — ГОСТ 7.1-1984, з сортуванням за авторами;

gost780u.bst — ГОСТ 7.80-2000, без сортування за авторами;

gost780s.bst — ГОСТ 7.80-2000, з сортуванням за авторами.

Варіації цих файлів, доповнені К. Воронцовим для підтримки анотацій і інших додаткових можливостей, що використовуються конвертером `bbl2html`:

gost71uv.bst — ГОСТ 7.1-1984, без сортування за авторами;

gost71sv.bst — ГОСТ 7.1-1984, з сортуванням за авторами;

gost780uv.bst — ГОСТ 7.80-2000, без сортування за авторами;

gost780sv.bst — ГОСТ 7.80-2000, з сортуванням за авторами.

Додаткові можливості конвертера `bbl2html`.

Для форматування списку авторів уведено команду `\BibAuthor`. Для форматування назви публікації введена команда `\BibTitle`. Для форматування посилання на URL уведено команду `\BibUrl`. Для форматування анотації введена команда `\BibAnnote`. Для форматування заголовків введена команда `\BibSection`. Всі ці команди можна перевизначити. Додано тип публікації `bibsection` із допустимими полями `title` і `annote`, щовикористовується разом зі стилями `gost71uv` і `gost780uv` для розміщення заголовків у `bbl`-файл перед його обробкою конвертером.

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 164 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

45. Пакет pdfscreen та його завантаження

Складних інструкцій з використання для пакету `pdfscreen` не потрібно, оскільки він є розширенням `hyperref.sty` Себастьяна Ратца, що застосовується для обробки перехресних посилань команд в L^AT_EX при створенні гіпертекстових посилань в документі. Основне призначення пакету – зміна розмірів ширини й висоти сторінки для забезпечення ідеального масштабування при перегляді документу на екрані. Таким чином, всі команди, що контролюють розміри сторінки перевизначаються з метою формування динамічних розмірів заповненості екрану.

Для завантаження пакету необхідно в преамбулі документу вказати команду:

```
\usepackage[screen,panelleft]{pdfscreen}
```

Немає необхідності завантажувати пакет `\usepackage{hyperref}` з його варіантами, так як останній завантажується `pdfscreen`. У ранніх версіях це можна було зробити так: завантажити попередньо `hyperref.sty` та вказати в `pdfscreen` необхідні опції.

Драйвер бекенд за замовчуванням для `pdfscreen` є PdfT_EX. Тим не менш, користувач може вказати свій драйвер бекенд в якості опції. Пакет `pdfscreen` краще завантажувати в преамбулі, з метою уникнення подальшого перевизначення команд, що використовуються в ньому.

45.1. Опції пакету pdfscreen

Доступні наступні опції:

1. `screen` – формує екранізацію, використовується при створенні екранних презентацій.
2. `print` – формує друк, вигляд якого схожий з DVI-файлом.
3. `panelleft` – панель навігації в лівій частині.

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[◀◀](#) [▶▶](#)[◀](#) [▶](#)

Стор. 165 із 189

[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

4. `panelright` – панель навігації в правій частині.
5. `nopanel` – приховує панель.
6. `paneltoc` – генерує зміст на панелі. Не слід використовувати команду `\tableofcontents` в документі одночасно з `paneltoc`.
7. `sectionbreak` – здійснює перенесення на нові сторінки основних структурних частин документа.
8. `cod` – команда, що може бути використана для друкування лістингу програми.
9. `Backend drivers: dvips, dvipsone, . . . , vtex` – вказуються в якості додаткового внутрішнього інтерфейсу back-end драйверу. PdfTeX використовується за замовчуванням.
10. `Color schemes` – визначає шість кольорових схем – *bluelace, blue, gray, orange, palegreen та chocolate* – для панелі та кнопок, які можна вказати опціями для пакету. За замовчуванням синій.
11. `Foreign language support` – підтримуються тільки 15 європейських мов на даний момент. Якщо мова не підтримується, то пакет буде відображати назви кнопок англійською мовою.
12. `ukrainian` – чи будь-яка інша підтримувана мова. Текст на навігаційних кнопках буде виводитися на відповідній мові. Ця опція не залежить від опції пакету `babel`. Якщо вказана мова не підтримується `pdfscreen`, то локалізувати кнопки можна за допомогою `pdfscreen.cfg`, див., наприклад, `pdfscreen.cfg.specimen`.
13. `nocfg` – дає можливість подавити конфігураційний файл, якщо користувач не хоче використовувати власні параметри.

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 166 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

45.2. Інші параметри

Щоб зробити `pdfscreen` більш функціональним, необхідно використовувати наступні параметри:

```
\emblem{<ім'я графічного файлу>}
```

Ім'я графічного файлу, який відображається на панелі навігації.

```
\urlid{<URL ім'я>}
```

Кнопка домашньої сторінки в навігаційній панелі буде пов'язана з URL.

```
\screensize{<висота>}{<ширина>}
```

Ця команда необхідна для того, щоб вказати розміри екрану вихідного формату PDF. Оскільки розміри екрану за замовчуванням не доступні, то користувач повинен вказати їх у явному вигляді. Зауважимо, що немає ніяких обмежень на розміри екрану. На відміну від попередніх версій, користувач може вільно вибрати будь-який розмір. За замовчуванням ширина панелі становить 15 % від ширини екрану.

```
\margins{<зліва>}{<справа>}{<зверху>}{<знизу>}
```

За допомогою цієї команда можна встановити поля документа. Вони повинні вказуватися в явному вигляді в преамбулі документа.

Із `\margin` і `\screensize` задається розмір екрану в преамбулі документа, `pdfscreen` після цього буде використовувати розміри екрану і полів, що вказані користувачем.

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 167 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

45.3. Типова преамбула

Типову преамбулу наведено нижче (саме за допомогою цієї преамбули відформатовано вихідний документ):

```
\documentclass[a4paper,11pt]{article}
\usepackage{xspace,colortbl}
\usepackage[cp1251]{inputenc}
\usepackage[T1,OT1,TS1,T2A]{fontenc}
\usepackage[english,ukrainian]{babel}
\usepackage\0arg{screen,panelleft,gray,paneltoc}\Arg{pdfscreen}

\margins\Arg{.75in}\Arg{.75in}\Arg{.75in}\Arg{.75in}
\screensize\Arg{6.25in}\Arg{8in}
\overlay\Arg{lightsteelblue.pdf}
\begin{document}

\input{Rozdil_1}
\renewcommand{\@biblabel}[1]{#1.}

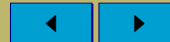
\makeatother
\addcontentsline{toc}{section}{Список використаних джерел}
\renewcommand{\refname}{Список використаних джерел}
\bibliographystyle{gost2003s}
\bibliography{biblio}
\end{document}
```

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 168 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

45.4. Пакети, що потрібні для запуску pdfscreen

Наступні пакети необхідні для правильної компіляції (завантажити їх можна зі CTAN):

Пакет	Опис пакету
1	2
<code>hyperref.sty</code>	Пакет застосовується для обробки перехресних посилань команд в L ^A T _E X при створенні гіпертекстових посилань в документі.
<code>comment.sty</code>	Вибірково включити/виключити фрагменти тексту, що дозволяє користувачеві визначати нові, окремо керовані, версії коментарів. Весь текст між <code>\comment ... \endcomment</code> і <code>\begin{comment} ... \end{comment}</code> відкидається. Це середовище працює з будь-якою кількістю коментарів, і коментар може бути довільний текст.
<code>truncate.sty</code>	Пакет для вирізання тексту заданої ширини. Пакет визначає команду <code>\truncate[⟨маркер⟩]{⟨ширина⟩}{⟨текст⟩}</code> . Якщо текст занадто великий, щоб поміститися у зазначеній ширині, то він обрізається, і продовжується маркером визначеним в команді. За замовчуванням маркеру співставляється три крапки. Пакет дозволяє вихідному тексту мати свою природну ширину. Текст не буде обрізається в середині слова.
<code>graphicx.sty</code>	Пакет визначає стратегії для включення графіки в T _E X документи. Розширює можливості стандартних команд типу <code>\includegraphics</code> при вставці графіки. Створено декларацію <code>\graphicspath{⟨dir-list⟩}</code> за допомогою якої можна вказати список каталогів, у яких необхідно здійснювати пошук графічних файлів.
<code>color.sty</code>	Автор пакету Фред Кнайт. Створений для реалізації відтінків сірого або кольорової гама в PostScript. Необхідний для чорно-білих і кольорових принтерів PostScript. Кольори визначаються за замовчуванням, але можна додати будь-який колір, якщо знати його RGB значення.


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)


Стор. 169 із 189

[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

1	2
<code>colortbl.sty</code>	Пакет дозволяє використовувати ефекти кольору при створення таблиць. Саме цей рядок таблиці демонструє використання розглядуваного пакету.
<code>calc.sty</code>	Пакет здійснює перевизначення L ^A T _E X команд <code>\setcounter</code> , <code>\addtocounter</code> , <code>\setlength</code> , і <code>\addtolength</code> . Можливості цих команд розширено допуском арифметики в інфіксі позначення. Наприклад, масштабування фігури можна здійснювати так <code>\setlength{\newYsize}{\Ysize*\ratio{\textwidth}{\Xsize}}</code> .
<code>amssymb.sty</code>	Стиль визначає математичну символіку, що міститься в шрифтах Американського математичного товариства (AMS).
<code>amsbsy.sty</code>	Стиль реалізується командами <code>\pmb</code> та <code>\boldsymbol</code> , що дозволяють перетворювати стандартизовану математичну символіку в bold символіку.
<code>shortvrb.sty</code>	Стиль Френка Міттельбаха. Пакет визначає команду <code>\MakeShortVerb</code> , що робить один символ активним. Це стосується команди <code>\verb</code> . Таким чином, після того, як застосовано команду <code>\MakeShortVerb{\ }</code> , текст <code> матеріал </code> має той же ефект, що і <code>\verb матеріал </code> .
<code>fancybox.sty</code>	Стиль для створення боксів із тіннями, подвійними і овальними рамками. Основні команди є такими: <code>\shadowbox</code> , <code>\doublebox</code> , <code>\ovalbox</code> (із <code>\thinlines</code>) і <code>\Ovalbox</code> (із <code>\thicklines</code>). Користувачу надається можливість керувати динамічними критеріями боксових структур для досягнення бажаної мети.


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)


Стор. 170 із 189

[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

46. Панель навігації

Конструкція панелі навігації, як правило, знаходиться повністю зліва, але, все таки, залежить від фантазії користувача і може бути створена за його смаком. Команда `\panel` відповідає за те, щоб панель була додана і за замовчуванням розташована вертикально з кнопками навігації. Існує також команда:

```
\addButton{<довжина>}{<текстовий рядок>}
```

яку може використовувати користувач для створення власних кнопок. Нижче наведено приклад того, як створюється кнопка **Наступна сторінка**:

```
\Acrobatmenu{NextPage}{\addButton{1.25in}{Наступна сторінка}}
```

Це створить таку кнопку навігації



а клацання по ній приведе користувача до наступної сторінки. Аналогічним чином, можна створити кнопки з графічним зображеннями. Для їх створення використовується команда

```
\imageButton{<ширина>}{<висота>}{<ім'я графічного файлу>}
```

матимемо



Сайт ПНПУ

Головна

Зміст



Стор. 171 із 189

Назад

Перегляд

Закрити

Вихід

Панель зліва	⇒ <code>\usepackage[screen,panelleft]{pdfscreen}</code>
Панель справа	⇒ <code>\usepackage[screen,panelright]{pdfscreen}</code>
Без панелі	⇒ <code>\usepackage[screen,nopanel]{pdfscreen}</code>
Портрет	Зміна розміру екрана за допомогою команди: <code>\screensize{⟨висота⟩}{⟨ширина⟩}</code>
Квадрат	Зміна розміру екрана за допомогою команди: <code>\screensize{⟨висота⟩}{⟨ширина⟩}</code>
Широка панель	⇒ <code>\panelwidth=⟨розмір⟩</code>
Версія для друку	⇒ <code>\usepackage[print]{pdfscreen}</code>

Рис. 6. Різні типи позиціонування панелі

Кнопка герба школи зображення генерується за допомогою команди:

```
\href{pnpu.edu.ua/ua/}{\imageButton{.5in}{!}{PNPU.jpg}}
```

Якщо користувач натисне на цю кнопку, то перейде на веб-сайт 26 школи м. Полтави, що визначається адресою <http://pnpu.edu.ua/ua/>.

Навігаційну панель може бути встановлено за бажанням користувача, злівої чи справа сторони. Різні типи позиціонування панелі представлено на рис. 6.

Ширину панелі може бути змінено явним заданням розміру в преамбулі документу, як `\panelwidth=⟨розмір⟩`. Значення за замовчуванням становить 15% від ширини екрану, проте, є мінімальне значення 1 in, якщо 15% від ширини екрану буде меншим ніж 1 in.

Користувач може визначити свою власну панель, головне щоб вона вписалася в документ. Панель навігації розглядуваного електронного посібника, що представлена в нижньому колон-



Сайт ПНПУ

Головна

Зміст



Стор. 172 із 189

Назад

Перегляд

Закрити

Вихід

титулі, створено з використанням пакету `\usepackage{fancyhdr}`. Нижче представлено частину коду за допомогою якого згенеровано бокову панель:

```
\panelwidth=1.3in
\def\panel{\colorbox{panelbackground}
{\begin{minipage}[t][\paperheight][b]{\panelwidth}
\centering\null\vspace*{12pt}
\includegraphics[width=.75in]{customLogo.jpg}\par\vfill
\href{\@urlid}{\addButton{.85in}{\@Panelhomepage}}\par\vfill
\Acrobatmenu{FirstPage}{\addButton{.85in}
{\FBlack\@Paneltitlepagename}}\par\vfill
\Acrobatmenu{FirstPage}{\addButton{.2in}
{\FBlack\scalebox{.8}[1.4]{\bt1\bt1}}}\hspace{-3pt}
\Acrobatmenu{PrevPage}{\addButton{.2in}
{\FBlack\scalebox{.8}[1.4]{\bt1}}}\hspace{-3pt}
\Acrobatmenu{NextPage}{\addButton{.2in}
{\LBlack\scalebox{.8}[1.4]{\rtl}}}\hspace{-3pt}
\Acrobatmenu{LastPage}{\addButton{.2in}
{\LBlack\scalebox{.8}[1.4]{\rtl\rtl}}}\par\vfill
\Acrobatmenu{GoToPage}{\addButton{.85in}{\@Panelgotopagename}}\par\vfill
\Acrobatmenu{GoBack}{\addButton{.85in}{\@Panelgobackname}}\par\vfill
\Acrobatmenu{FullScreen}{\addButton{.85in}{Full Screen}}\par\vfill
\Acrobatmenu{Close}{\addButton{.85in}{\@Panelclosename}}\par\vfill
\Acrobatmenu{Quit}{\addButton{.85in}{\@Panelquitname}}\par
\null\vspace*{12pt}
\end{minipage}}}
```

Неважко помітити, що значення імен посилань із меню програми Acrobat програмуються


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)


Стор. 173 із 189

[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

англійською мовою. Задати розміри документу та перейменувати кнопки панелі можна так:

```
\margins{15mm}{15mm}{15mm}{10mm}
\screensize{180mm}{240mm}
\textheight=140mm
\headsep=8mm \footskip=8mm
\emblem{customLogo.jpg}
\overlay{overlay1.pdf}
\paneloverlay{but.pdf}
\urlid{school26poltava.at.ua/}
\def\pfill{\vskip8pt}
\bottombuttons %\nobottombuttons
\paneltitlepagename{Головна}
\panelabstractname{Аногація}
\panelcontentsname{Зміст}
\panelfullscreenname{Перегляд}
\panelhomepagename{Сайт ЗШ №26}
\panelgobackname{Назад}
\panelclosename{Закрити}
\panelquitname{Вихід}
\panelpagename{Стор.}
\panelofname{із}
```

Замість шрифту на кнопках можна використати деяку картинку, наприклад, sz кодом

```
\Acrobatmenu{GoBack}{\includegraphics{backpic}}
```



Сайт ПНПУ

Головна

Зміст



Стор. 174 із 189

Назад

Перегляд

Закрити

Вихід

47. Інші можливості пакету pdfscreen

47.1. Фон

Фон області екрану може бути створено на основі графічного файлу за допомогою команди `\overlay{ім'я графічного файлу}`. В якості альтернативи, ви можете вказати колір фону, вказавши `\backgroundcolor{color}`, де *color* визначається на основі кольорів пакету `color.sty`.

Фон панелі також можуть бути забезпечений за допомогою графічного файлу командою `\paneloverlay{графічний файл}`. Якщо не вказано накладення графіки для панелі, за замовчуванням вступить в силу колір `panelbackground`. Хоча можна перевизначити `panelbackground` колір за вибором користувача, чи внести зміни в файл `pdfscreen.cfg`. Команди `\overlayempty` і `\paneloverlayempty` допомагають уникнути накладань на будь-якому етапі.

47.2. Нижні та верхні кнопки

Меню, що знаходиться в нижньому колонтитулі документу, викликається за допомогою команди `\bottombuttons`. Закривається командою `\nobottombuttons`. Так само викликати і закрити меню у верхньому колонтитулі, цьому слугують команди `\topbuttons` і `\notopbuttons`. Користувач може викликати обидва меню одночасно, хоча це виглядатиме дивно.

47.3. Зміст в панелі

Опціональний пакет `paneltoc` дозволить створити зміст (заголовки розділів) у панелі навігації, якщо в документі не використано команди `\@starttoc{toc}`. В іншому разі зміст із заголовків розділів замінить розширений зміст. Використовувати цей пакет потрібно із обережністю, бо велика кількість розділів спотворить навігаційну панель. Проте, ручне втручання за допомогою команди `\clearpage` у відповідному місці може виправити справу певною мірою.

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 175 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

47.4. Файл конфігурації

Типовий файл конфігурації поставляється з пакетом. Користувач може коригувати конфігураційний файл `pdfscreen.cfg`. Як зазначалося раніше, саме в цьому файлі здійснюється переклад тексту кнопок, якщо мова, що вказана в `babel`, не підтримується пакетом `pdfscreen`, а також вказуються нові колірні схеми, URL ідентифікатори, дата, ім'я графічного файлу логотипу/емблеми. На с. 177 наведено зміст конфігураційного файлу `pdfscreen.cfg` з метою візуалізації місць перекладу тексту кнопок.

47.5. Створення форм

Існують різні підходи до створення форм, і запропонований підхід не єдиний спосіб використання об'єктів типу «форма» в PDF. Наприклад, побудова форм в специфічних пакетах `acrotext` (Д.П. Сторі) та `contex` (Ганс Хаген) спирається на використання мови JavaScript.

Наведемо приклад створення форми, яка відображатиме елемент вибору альтернативи типу «комбінований список». Елемент такого вигляду дає можливість або обрати необхідне значення, або безпосередньо увести таке значення.

```
\begin{Form}[action=mailto:srahtz,method=post]
\ChoiceMenu[combo,default=Author,name=whatdwarf]
{Автор твору <Наталка Полтавка>}
{Котляревський,Квітка-Основ'яненко,Гоголь}\end{Form}
```

У результаті матимемо:

Автор твору «Наталка Полтавка»

Створення форм для тестування з використанням JavaScript, розглянуто в додатку [Використання пакету Exerquiz в pdfscreen](#).

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[«](#) [»](#)[«](#) [»](#)[Стор. 176 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

```

%% ----- language specific commands -----
%% \paneltitlepage{Title Page} \panlabstractname{Abstract}
%% \panelcontentsname{Contents} \panelfullscreenname{Full Screen}
%% \panelhomepagename{Home Page} \panelgobackname{Go Back} \panelclosename{Close}
%% \panelquitname{Quit} \panelpagename{Page} \panelofname{of}
%% ----- various colors -----
\definecolor{panelbackground}{gray}{.8} \definecolor{buttonbackground}{gray}{.6}
\definecolor{buttonshadow}{gray}{.2} \definecolor{orange}{rgb}{1,.549,0}
\definecolor{orange1}{rgb}{1,.5,0} \definecolor{section0}{rgb}{0,.5,.1}
\definecolor{section1}{rgb}{0,.5,1} \definecolor{section2}{rgb}{0,.5,.5}
\definecolor{section3}{rgb}{0,.5,.4} \definecolor{section4}{rgb}{.4,.5,.2}
\definecolor{section5}{rgb}{.5,.5,.3}
%%\screensize{6.25in}{8in} %%\screensize{height}{width}
\emailid{cvr@river-valley.com} \urlid{www.river-valley.com} \def\@urlcolor{magenta}
\IfFileExists{univ.jpg}{\emblem{univ.jpg}}{\def\@pagecolor{red}\def\@menucolor{red}
\def\@linkcolor{red}\def\@anchorcolor{black}\def\@filecolor{cyan}\def\@citecolor{blue}
\def\@secntformat#1{\llap{\scshape\color{section\thesection@level}
\csname the#1\endcsname.\hspace*{6pt}}}\def\pdfscreen{\tt pdfscreen.sty}\xspace}
\definecolor{backgroundcolor}{rgb}{.889941,.917973,.9931}
\definecolor{buttonbackground}{rgb}{.902,.902,.980} \definecolor{green}{rgb}{0,.7,0}
\definecolor{buttonbackground}{rgb}{.902,.902,.980} \definecolor{Red}{rgb}{1,.347,0}
\definecolor{buttonshadow}{rgb}{.412,.412,.412} \pagedissolve{R /D 0}
\definecolor{seashell}{rgb}{.285,.286,.28510} \definecolor{buttonshadow}{rgb}{0,0,0}
\definecolor{lcyan}{rgb}{.895,.93,.916} \urlid{www.river-valley.com}
\backgroundcolor{backgroundcolor} \endinput
%% end of file 'pdfscreen.cfg'

```


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)

[Стор. 177 із 189](#)
[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

48. Слайди

При створенні слайдових презентацій слід використовувати оточення

```
\begin{slide}
Тут повинен знаходитися матеріал слайду
\end{slide}
```

Слайд – це поле, що охоплює ширину і висоту області тексту, в межах якого матеріал буде вертикально по центру.

48.1. Шрифти

Всі атрибути шрифту переглядаються з метою їх динамізації з середовищем слайду. Проте, якщо користувач забажає повернутися до початкових розмірів, йому доведеться додати команду `\realnormalsize`, перед командою розміру шрифту `\normalsize`, для шрифту розміром `\large` додається `\reallarge` і так далі.

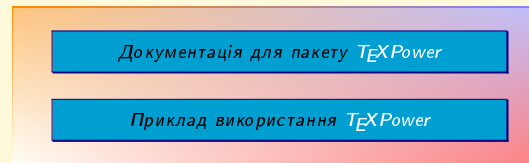
48.2. Подальша обробка

Отримати додаткові можливості при обробці слайдових презентацій можна за допомогою постпроцесора, що використовує віртуальну машину Java а саме, пакету **PPower4**. Вказаний пакет складається з декількох стильових файлів і доступний на CTAN. Основне завдання **PPower4** це додати деякі динамічні можливості для презентації. Додаткові функції включають в себе відображення частини сторінки в будь-якому порядку, видалення елементів зі сторінок та вставки переходів або фону. **PPower4** може бути використаний для презентацій при обробці документів, створених з допомогою `pdflatex`, `vlatex` або `dvipdfm`.

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 178 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

Варто звернути увагу на презентаційний пакет T_EXPower Стефана Лехмке, що є досить потужним і унікальним програмним продуктом, орієнтований на візуальні ефекти. Він містить клас `powersem`.

Нижче створено кнопки, натиснувши на які користувач зможе завантажити з Інтернету документацію по пакету T_EXPower та приклади його застосування.



48.3. Вбудовування інтерактивних Flash- і 3D-об'єктів, аудіо- та відеофайлів

Мультимедійні вклучення в Adobe Reader-9/X дозволяє здійснити пакет `media9`.



Аудіофайл (Едвін Мартон “Без тебе”) вставлено за допомогою наступного коду:

```
\includemedia[height=2em, width=17.3em, addresource=Without_you.mp3,
flashvars={source=Without_you.mp3 & autoPlay=true}]{\large Edvin Marton
\hs{12mm}\textquotedblleft Without you\textquotedblright}{APlayer.swf}
```

Команда `\includemedia` є тут ключовим об'єктом.

```
\includemedia[опції]{текст}{шлях до Flash (SWF) чи URL / 3D (PRC, U3D) файлу}
```


[Сайт ПНПУ](#)
[Головна](#)
[Зміст](#)


Стор. 179 із 189

[Назад](#)
[Перегляд](#)
[Закрити](#)
[Вихід](#)

Останній аргумент $\{\langle \text{шлях до Flash (SWF) файлу чи URL / 3D (PRC, U3D) файлу} \rangle\}$ є головним при вставці мультимедіа в PDF. Це може бути локальний SWF-файл, або URL файлу розміщеного на YouTube. Слід відмітити, що локальний файл стане частиною PDF-файлу, а для перегляду флеш-вмісту з URL потрібне підключення до Інтернет, коли користувач активує його в програмі Adobe Reader. URL-адреса має бути повною, тобто розпочинатися з `http[s]://` або `ftp://`. Що стосується 3D-контенту, Adobe Reader підтримує тільки U3D або PRC файли. При використанні `\includemedia` для відтворення відео або звукових файлів в Adobe Reader якості основного додатки використовується SWF-медіа-плеєр.

Теорема Піфагора

Нижче представлено код вставки відеоконтенту з YouTube (доведення теореми Піфагора англійською мовою), що демонструє широкі можливості використання розподілених систем.

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 180 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

```
\includegraphics[activate=onclick, width=0.6\linewidth,height=0.3375\linewidth,
activate=pageopen,flashvars={modestbranding=1 & autohide=1 & showinfo=0 & rel=0}]{
{http://www.youtube.com/v/r382kfkqAF4?rel=0}}
```

48.4. Вкладення інтерактивних 3D-об'єктів в PDF-документи

Поряд з відео корисним інструментом при створенні наукової публікації є вбудовані в PDF тривимірні графічні об'єкти. Інтерактивні тривимірні ілюстрації значно покращують сприйняття інформації читачем. Тут він може самостійно налаштувати тривимірну сцену: відстань до об'єкта, кутові координати, відображення поверхонь і полігональної сітки, ефекти освітлення, прозорість, масштаб та розмір об'єкта. З'являється можливість представити як загальний план, так і найдрібніші деталі 3D-об'єкта в межах однієї інтерактивної ілюстрації.

При необхідності автор може акцентувати увагу на певних ділянках об'єкта, визначивши набір видів, які читач буде перемикає в контексті одного 3D-об'єкта. Вони задаються невеликим набором параметрів, практично не збільшуючи розмір документа, на відміну від аналогічного набору статичних зображень.

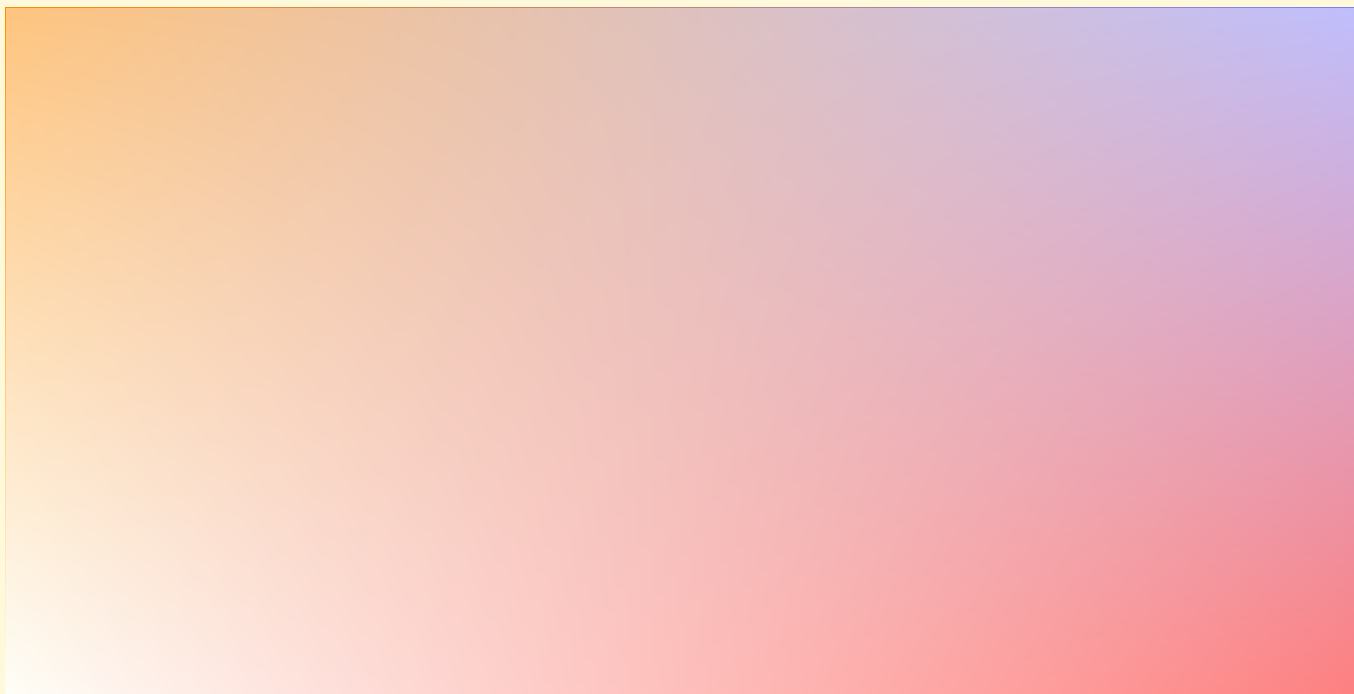
Тривимірні графічні об'єкти можуть бути вбудовані в PDF-документ в форматах U3D (Universal 3D) і PRC (Product Representation Compact). Кращу компресію полігональних сіток і більш точне уявлення поверхонь забезпечує формат PRC. Тим не менш, формат U3D більш поширений і доступний у вигляді Open Source бібліотек. Багато середовищ моделювання та рендерингу надають вбудовані інструменти для збереження тривимірних об'єктів у форматі U3D.

При використанні L^AT_EX для верстки та вбудовування 3D необхідно виконати дві операції:

- вбудувати 3D-об'єкт в документ .tex і скопіювати PDF;
- отриманий PDF відкрити в Adobe Reader, сформувати види для об'єкта, скопіювати

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 181 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

параметри видів в документ .vws, підключити їх до документу .tex, додати кнопки і перекомпілювати.



Файл U3D розроблено на основі VRML моделі Пітера Уайтхауса, <http://www.wonko.info/vrml/index.htm>; перетворення в U3D зроблено з використанням DeepExploration. Зазначимо, що файл `dice.vws` містить керування видами. Кнопки створено на основі `\mediabutton` із опцією `3Dgotoview`.

Із метою повідомити читачеві про наявність інтерактивного 3D-об'єкту, на сторінці рекомендується додавати кнопку "3D що вмикає/вимикає тривимірний об'єкт у вікні.

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 182 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

49. Перехід між сторінками

В Acrobat можна вибрати один із декількох переходів між сторінками з тим, щоб зробити процес заміни сторінки більш привабливим. На жаль, ці ефекти не можуть бути встановлені безпосередньо в Acrobat. Хоча можна вибрати перехід між сторінками в розділі "Файл" Налаштування "Повноекранний режим але ця установка буде відноситись до всього документу, а не до окремих сторінок у файлі. Так, як керування переходами між окремими сторінками не підтримуються в інтерфейсі акробата, користувач може задавати переходи між сторінками задалегідь. Варто лише вказати за допомогою команди `\pagedissolve{опції}` свій вибір.

Переходи між сторінками завжди активуються при відкритті сторінки, незалежно від попередньої сторінки. Тому не має значення, чи буде відображатися сторінка за допомогою ручного навігації, чи буде відкрита за номером сторінки або посилання.

Отже, перехід між сторінками визначаються для поточної сторінки. Зверніть увагу, що тільки пряма вказівка сторінки може бути використана для переходів сторінок.

Ключі варіантів переходів наведено в табл. 20.

У табл. 21 представлено параметри ключів, що взяті з добре відомої книги, **Веб-публікації з Acrobat/PDF Томаса Мерза**. Для деяких із переходів, можуть бути вказані додаткові параметри. Код, зазначений нижче додає роздільний ефект ліній, що переміщуються по горизонталі (/H), від внутрішньої частини сторінки до зовнішньої частини (/O). Тривалість ефекту становить дві секунди (/D).

Приклад команди, що задає переходи між сторінками, наведено нижче.

```
\pagedissolve{Dissolve /D 1 /Di /H /DM /O}
```

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 183 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

Ключі	Пояснення
<i>/Split</i>	Дві лінії розгортки на екрані, що необхідні для відкриття нової сторінки, все схоже на відкриття завіси.
<i>/Blinds</i>	Подібно <i>/Split</i> , але з лініями, що нагадують «жалюзі»
<i>/Box</i>	Прямокутник розширюється від центру старої сторінки, щоб відкрити нову.
<i>/Wipe</i>	Окрема лінія «втирає» стару сторінку, щоб відкрити нову.
<i>/Dissolve</i>	Стара сторінка «розчиняється», щоб відкрити нову.
<i>/Glitter</i>	Similar to <i>/Wipe</i> Ефект зачищення старої сторінки від одного краю до іншого.
<i>/R (Replace)</i>	Стара сторінка просто замінюється новою без будь-якого спеціального ефекту. Це значення використовується за замовчуванням.

Таблиця 20. Ключі переходів між сторінками

Ключі	Пояснення
<i>/D</i>	Тривалість перехідного ефекту в секундах (застосовується до всіх ефектів)
<i>/Di (Direction)</i>	Напрямок руху (кратність 90° тільки). Значення збільшуються в напрямку проти годинникової стрілки від 0° вправо (для <i>/Wipe</i> і <i>/Glitter</i>).
<i>/Dm (Dimension)</i>	Можливі значення <i>/H</i> або <i>/V</i> для горизонтального або вертикального ефекту, відповідно (для <i>/Split</i> і <i>/Blinds</i>).
<i>/M (Motion)</i>	Визначає, чи буде ефект здійснюється в напрямку від центру чи від краю. Можливі значення: <i>/I</i> – від центру, <i>/O</i> – від краю (для <i>/Split</i> і <i>/Box</i>).

Таблиця 21. Розширені можливості для переходів між сторінками



Сайт ПНПУ

Головна

Зміст



Стор. 184 із 189

Назад

Перегляд

Закрити

Вихід

50. Використання пакету Exerquiz в Pdfscreen

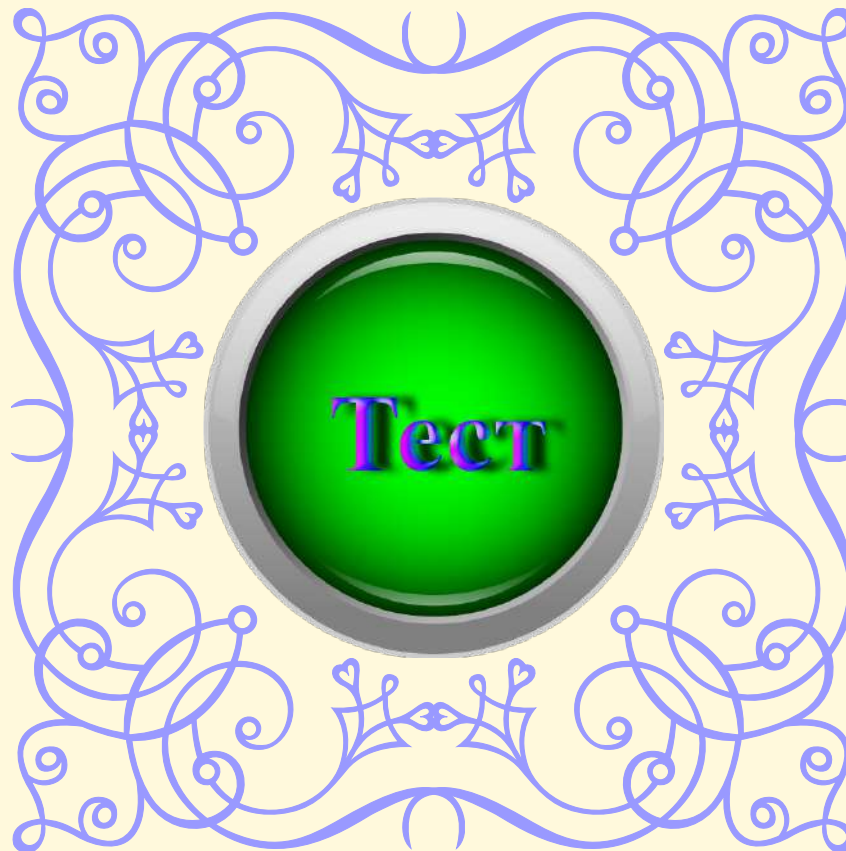
Сучасне Інтернет-покоління потребує цікавого і наочного навчання, в якому є колір, руху і елементи гри. Для потреб викладання математики і фізики потрібна така комп'ютерна система, яка забезпечує зручність при написанні формул, і в той же час, надає зручне середовище для виконання креслень лінійкою і циркулем. Видавнича система L^AT_EX повністю відповідає цим вимогам. Вона надає зручне середовище і пакети для форматування складних математичних формул, а її графічні розширення дозволяють ефективно та зручно створювати графіку.

Однією з потреб мультимедійного навчання, є створення інтерактивних, естетично витриманих, гіпертекстових презентацій. Система L^AT_EX пропонує багато різноманітні пакети для їх створення, при цьому вихідні результати подаються у форматі pdf, і є доступними для перегляду за допомогою програми Adobe Acrobat Reader.

Використовуючи пакет `pdfscreen`, можна легко досягти вище зазначених результатів. Крім того даний пакет дозволяє разом із пакетом `exerquiz` створювати професійні інтерактивні тести.

Для метою перевірки сумісності пакетів в системі L^AT_EX та демонстрації їх спільної роботи, було створено інтерактивні тестові завдання, що розміщені в окремому документі. Завантажити цей документ можна натиснувши кнопку із назвою «Тест», що знаходиться на наступній сторінці. Відкривши тестові завдання, користувач їх може як виконати, так і переглянути код створення.

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 185 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)



Для перегляду тестів та механізму їх створення необхідно натиснути кнопку «Тест». Вікторіанське оформлення кнопки здійснене за допомогою пакету `pgfornament`. Зауважимо, що даний пакет також використано при оформленні титульних сторінок.

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)

Стор. 186 із 189

[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

Список використаних джерел

1. Балдин Е. М. Компьютерная типография L^AT_EX. / Е. М. Балдин. — Новосибирск : БХВ-Петербург, 2008. — 304 с. — ISBN: 978-5-9775-0230-6. — URL: <http://elib.sbras.ru:8080/jspui/bitstream/SBRAS/9015/1/ctex.pdf>.
2. Гуссенс М. Путеводитель по пакету L^AT_EX и его расширению L^AT_EX 2_ε: Пер. с англ. / М. Гуссенс, Ф. Миттельбах, А. Самарин. — М. : Мир, 1999. — 606 с. — ISBN: 5-03-003325-4. — URL: elibrary.bsu.az/kitablar/1020.PDF.
3. Гуссенс М. Путеводитель по пакету L^AT_EX и его графическим расширениям. Иллюстрирование документов при помощи T_EX'a и PostScript'a: Пер. с англ. / М. Гуссенс, С. Ратц, Ф. Миттельбах. — М. : Мир: Бином ЛЗ, 2002. — 621 с. — ISBN: 5-03-003388-2. — URL: www.books4all.ru/description/1589.html.
4. Кнут Д. Э. Искусство программирования. Том 1. Основные алгоритмы: Пер. с англ. Ю.В. Козаченко. / Д. Э. Кнут. — М. : Вильямс, 2002. — 720 с. — ISBN: 0-201-89683-4. — URL: publ.lib.ru/ARCHIVES/K/KNUT_Donal'd_Ervin/_Knut_D.E..html#006.
5. Кнут Д. Э. Все про METAFONT: Пер. с англ. М.Р. Сайт-Аметова. / Д. Э. Кнут. — М. : Вильямс, 2003. — 560 с. — ISBN: 5-8459-0382-3. — URL: publ.lib.ru/ARCHIVES/K/KNUT_Donal'd_Ervin/_Knut_D.E..html#006.

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)[Стор. 187 із 189](#)[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

6. Кнут Д. Э. Все про T_EX: Пер. с англ. Л.Ф. Козаченко. / Д. Э. Кнут. — М. : Вильямс, 2003. — 543 с. — ISBN: 5-8459-0382-3. — URL: publ.lib.ru/ARCHIVES/K/KNUT_Donal'd_Ervin/_Knut_D.E..html#006.
7. Кнут Д. Э. Искусство программирования. Том 2. Получисленные алгоритмы: Пер. с англ. Ю.В. Козаченко. / Д. Э. Кнут. — М. : Вильямс, 2007. — 832 с. — ISBN: 0-201-89684-2. — URL: publ.lib.ru/ARCHIVES/K/KNUT_Donal'd_Ervin/_Knut_D.E..html#006.
8. Кнут Д. Э. Искусство программирования. Том 3. Сортировка и поиск: Пер. с англ. Ю.В. Козаченко. / Д. Э. Кнут. — М. : Вильямс, 2007. — 824 с. — ISBN: 0-201-89685-0. — URL: www.proklondike.com/books/thalg/knut_art3.html.
9. Кнут Д. Э. Искусство программирования. Том 4. Комбинаторные алгоритмы: Пер. с англ. И.В. Красикова. / Д. Э. Кнут. — М. : Вильямс, 2013. — 960 с. — ISBN: 978-5-8459-1744-7. — URL: publ.lib.ru/ARCHIVES/K/KNUT_Donal'd_Ervin/_Knut_D.E..html#006.
10. Львовский С. М. Набор и вёрстка в системе L^AT_EX. — 3-е изд. / С. М. Львовский. — М. : МЦНМО, 2003. — 448 с. — ISBN: 5-940570-91-7. — URL: www.mcsme.ru/free-books/llang/newllang.pdf.
11. Столяров А. В. Сверстай диплом красиво L^AT_EX за три дня / А. В. Столяров. — М. : МАКС Пресс, 2010. — 101 с. — ISBN: 978-5-317-03440-5. — URL: <http://www.stolyarov.info/books/pdf/latex3days.pdf>.

[Сайт ПНПУ](#)[Головна](#)[Зміст](#)

Стор. 188 із 189

[Назад](#)[Перегляд](#)[Закрити](#)[Вихід](#)

12. Роженко А. И. Искусство верстки в L^AT_EX'e. – 3-е изд. / А. И. Роженко. — Новосибирск : ИВМиМГ СО РАН, 2005. — 398 с. — ISBN: 5-901548-25-6.



Сайт ПНПУ

Головна

Зміст



Стор. 189 із 189

Назад

Перегляд

Закрити

Вихід