

Не надто короткий вступ до L^AT_EX 2_ε

Або L^AT_EX 2_ε за 98 хвилин

Tobias Oetiker

Hubert Partl, Irene Hyna та Elisabeth Schlegl

Version 3.22, 10 October, 2002

Переклад: **Максим Поляков**

15 листопада 2002 р.

Copyright © 2000–2002 Tobias Oetiker and all the Contributors to LShort. All rights reserved.

This document is free; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This document is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this document; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Copyright © 2000–2002 Tobias Oetiker та усі контрибутори до LShort. Усі права застережені.

Цей документ є вільним; Ви можете розповсюджувати його та/або вносити зміни відповідно до умов Загальної Публічної Ліцензії GNU у тому вигляді, у якому вона була опублікована Фондацією Вільного Програмного Забезпечення; або 2-ї версії Ліцензії, або (на ваш розсуд) будь-якої більш пізньої версії.

Цей документ розповсюджується із сподіванням, що він виявиться корисним, але БЕЗ БУДЬ-ЯКОЇ ГАРАНТІЇ, без навіть УЯВНОЇ ГАРАНТІЇ КОМЕРЦІЙНОЇ ПРИДАТНОСТІ чи ВІДПОВІДНОСТІ БУДЬ-ЯКОМУ ПЕВНОМУ ЗАСТОСУВАННЮ. Зверніться до Загальної Публічної Ліцензії GNU за подробицями. Ви мали отримати копію Загальної Публічної Ліцензії GNU разом з цим документом; якщо ні — напишіть за адресою: Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Подяка

Джерелом значної частини матеріалу, використаного у цьому документі, є австрійський вступ до L^AT_EX 2.09 німецькою мовою, авторами якого є:

Hubert Partl <partl@mail.boku.ac.at>

Zentraler Informatikdienst der Universität für Bodenkultur Wien

Irene Hyna <Irene.Hyna@bmf.ac.at>

Bundesministerium für Wissenschaft und Forschung Wien

Elisabeth Schlegl <no email>

in Graz

Якщо ви маєте бажання ознайомитись із німецькою версією документа, ви можете знайти версію, яку Jörg Knappen оновив для L^AT_EX 2_ε, за адресою CTAN:/tex-archive/info/lshort/german.

Готуючи цей документ, я отримав багато відгуків від учасників `comp.text.tex`. Перелічені нижче особи допомогли з виправленнями, порадами, та пропозиціями щодо покращень. Вони доклали багато зусиль, щоб допомогти мені довести документ до його теперішнього стану. Я щиро їм усім дякую. Природно, всі помилки, які ви знайдете в цій книзі, — мої. Якщо ви випадково знайдете вірно написане слово, це виключна заслуга котрогось із них.

Rosemary Bailey, Friedemann Brauer, Jan Busa, Markus Brühwiler, David Carlisle, José Carlos Santos, Mike Chapman, Pierre Chardaire, Christopher Chin, Carl Cerecke, Chris McCormack, Wim van Dam, Jan Dittberner, Michael John Downes, David Dureisseix, Elliot, Daniel Flipo, David Frey, Hans Fugal, Robin Fairbairns, Jörg Fischer, Erik Frisk, Frank, Kasper B. Graversen, Arlo Griffiths, Alexandre Guimond, Cyril Goutte, Greg Gamble, Neil Hammond, Rasmus Borup Hansen, Joseph Hilferty, Björn Hvittfeldt, Martien Hulsen, Werner Icking, Jakob, Eric Jacoboni, Alan Jeffrey, Byron Jones, David Jones, Johannes-Maria Kaltenbach, Michael Koundouros, Andrzej Kawalec, Alain Kessi, Christian Kern, Jörg Knappen, Kjetil Kjernsmo, Maik Lehardt, Alexander Mai, Martin Maechler, Aleksandar S Milosevic, Claus Malten, Kevin Van Maren, Lenimar Nunes de Andrade, Demerson Andre Polli, Maksym Polyakov Hubert Partl, John Reffing, Mike Ressler, Brian Ripley, Young U. Ryu, Bernd Rosenlecher, Chris Rowley, Risto Saarelma, Hanspeter Schmid, Craig Schlenker, Christopher Sawtell, Geoffrey Swindale, Josef Tkadlec, Scott Veirs, Didier Verna, Fabian Wernli, Carl-Gustav Werner, David Woodhouse, Chris York, Fritz Zaucker, Rick Zaccane, and Mikhail Zotov.

Перелічені нижче особи допомогли зауваженнями, виправленнями, порадами щодо українського перекладу.

Тарас Бейко, Дмитро Ковальов, Володимир Лісівка, Анатолій Маляренко, Олександр Мороцький, Дмитро Редчук, Валентин Соломко.

Передмова

L^AT_EX [1] — система комп'ютерного набору, призначена для створення наукових та математичних документів високої типографської якості. Система також цілком придатна для створення інших видів документів, починаючи від звичайних листів, і закінчуючи цілими книгами. Механізмом форматування для L^AT_EX слугує програма T_EX[2].

Цей короткий *Вступ* описує L^AT_EX 2_ε і повинен бути достатнім для більшості застосувань L^AT_EX. Повний опис системи L^AT_EX можна знайти в [1, 3].

Цей *Вступ* розбито на п'ять розділів:

Розділ 1 розповідає про базову структуру документів L^AT_EX та дає коротке знайомство з його історією. Після читання цього розділу ви повинні мати загальне уявлення про L^AT_EX. Воно допоможе вам об'єднати в єдине ціле відомості, які ви отримуєте, читаючи наступні розділи.

Розділ 2 заглиблюється в деталі набору. Він пояснює більшість важливих команд і оточень L^AT_EX. Після прочитання цього розділу ви зможете створювати перші документи.

Розділ 3 пояснює, як набирати формули в L^AT_EX. Велика кількість прикладів допоможе вам зрозуміти, як використовувати цю, одну із найважливіших можливостей L^AT_EX. Наприкінці цього розділу ви знайдете таблиці з переліком більшості відомих у L^AT_EX математичних символів.

Розділ 4 розповідає про створення предметного покажчика і бібліографії, включення EPS графіки, та про деякі інші корисні речі.

Розділ 5 містить потенційно небезпечні відомості про те, як змінювати стандартний макет документа. Необережне використання цих можливостей може призвести до небажаного результату.

Важливо читати розділи послідовно. Зрештою, брошура не така вже й велика. Уважно читайте приклади, які містять велику кількість корисної інформації.

Л^AT_EX працює на більшості комп'ютерів, починаючи з IBM PC чи Mac, і закінчуючи потужними системами UNIX чи VMS. У багатьох університетських мережах система уже встановлена і готова до роботи. Інформація про те, як використовувати локальну установку Л^AT_EX, можна знайти в *Local Guide* [4]. Якщо у вас будуть проблеми з початком роботи, зверніться по допомогу до того, хто надав вам цей буклет. Мета даного *Вступу* полягає не у тім, щоб навчити вас встановлювати і настроювати систему Л^AT_EX, а щоб навчити писати документи так, щоб вони могли бути оброблені Л^AT_EX.

Якщо вам буде потрібний будь-який, інший матеріал стосовно Л^AT_EX, пошукайте його у одному з архівів CTAN¹. Домашня сторінка CTAN має адресу <http://www.ctan.org>. Усі пакети можна отримати з ftp архіву <ftp://www.ctan.org> та з його дзеркал по усьому світу. Їх можна знайти за адресами <ftp://ctan.tug.org> (США), <ftp://ftp.dante.de> (Німеччина), <ftp://ftp.tex.ac.uk> (Великобританія). Якщо ви в іншій країні, виберіть найближчий до вас архів.

На протязі цієї книги ви знайдете й інші посилання на CTAN, особливо стосовно програмного забезпечення та документів, які ви, можливо, захочете звантажити. Замість написання повних URL, я просто писав CTAN:, і слідом — відповідну адресу в межах дерева CTAN.

Якщо ви бажаєте мати Л^AT_EX на вашому власному комп'ютері, подивіться, що можна знайти за адресою <CTAN:/tex-archive/systems>.

Якщо у вас є думки з приводу того, що варто додати, вилучити, чи змінити в цьому документі, будь ласка, дайте мені знати. Я особливо зацікавлений у відгуках новачків Л^AT_EX стосовно того, які частини *Вступу* легко зрозуміти, і що варто було б пояснити краще.

Tobias Oetiker <oetiker@ee.ethz.ch>

Department of Electrical Engineering,
Swiss Federal Institute of Technology

Остання версія цього документа доступна за адресою
<CTAN:/tex-archive/info/lshort>.

¹Comprehensive T_EX Archive Network — Всеохоплююча Мережа Архівів T_EX.

Зміст

Подяка	iii
Передмова	v
1 Це варто знати	1
1.1 The Name of the Game	1
1.1.1 \TeX	1
1.1.2 \LaTeX	1
1.2 Основи	2
1.2.1 Автор, дизайнер і набірник	2
1.2.2 Дизайн макета	3
1.2.3 Переваги і недоліки	4
1.3 Вхідні файли \LaTeX	5
1.3.1 Пробіли	5
1.3.2 Зарезервовані символи	5
1.3.3 Команди \LaTeX	6
1.3.4 Коментарі	6
1.4 Структура вхідного файлу	7
1.5 Типова сесія	8
1.6 Макет документа	10
1.6.1 Класи документів	10
1.6.2 Пакети	12
1.6.3 Стили сторінки	12
1.7 Файли, які вам можуть трапитися	12
1.8 Великі проекти	15
2 Набір тексту	17
2.1 Структура тексту і мови	17
2.2 Поділ на рядки і сторінки	19
2.2.1 Вирівнювання абзаців	19
2.2.2 Переноси	20
2.3 Ready made Strings	21
2.4 Спеціальні літери і символи	21

2.4.1	Знаки лапок	21
2.4.2	Тире і дефіси	22
2.4.3	Тильда (\sim)	22
2.4.4	Символ градуса ($^\circ$)	22
2.4.5	Трикрапка (\dots)	23
2.4.6	Лігатури	23
2.4.7	Акценти і спеціальні символи	23
2.5	Підтримка багатомовності	24
2.5.1	Українська мова	26
2.6	Пробіли між словами	27
2.7	Заголовки, частини і розділи	28
2.8	Перехресні посилання	30
2.9	Виноски	31
2.10	Виділення	31
2.11	Оточення	32
2.11.1	Список, перелік та опис	32
2.11.2	Вирівнювання ліворуч, праворуч і по центру	33
2.11.3	Цитати та вірші	33
2.11.4	Буквальне відтворення	34
2.11.5	Таблиці	35
2.12	Плаваючі об'єкти	37
3	Набір математичних формул	41
3.1	Загальні відомості	41
3.2	Групування в математичному режимі	43
3.3	Складові математичної формули	43
3.4	Математичні пробіли	47
3.5	Вертикально розташований матеріал	48
3.6	Фантом	50
3.7	Розмір математичного шрифту	51
3.8	Теореми, закони, ...	51
3.9	Жирні символи	53
3.10	Список математичних символів	54
4	Корисні можливості	61
4.1	Включення EPS графіки	61
4.2	Бібліографія	63
4.3	Покажчики	64
4.4	Вибагливі колонтитуди	65
4.5	Пакет <code>verbatim</code>	66
4.6	Звантаження і встановлення пакетів <code>L^AT_EX</code>	67

5	Налаштування ЛАТЕХ	69
5.1	Нові команди, оточення і пакети	69
5.1.1	Нові команди	70
5.1.2	Нові оточення	71
5.1.3	Ваш власний пакет	71
5.2	Шрифти та розміри	72
5.2.1	Команди зміни шрифту	72
5.2.2	Увага, небезпека	75
5.2.3	Порада	76
5.3	Інтервали	76
5.3.1	Інтервали між рядками	76
5.3.2	Форматування абзаців	76
5.3.3	Горизонтальні інтервали	77
5.3.4	Вертикальні інтервали	78
5.4	Макетування сторінки	80
5.5	Ще про довжини	81
5.6	Блоки	81
5.7	Лінійки і розміри	84
	Бібліографія	85
	Показчик	87

Перелік ілюстрацій

1.1	Складові системи \TeX	2
1.2	Мінімальний файл \LaTeX	9
1.3	Приклад реалістичної журнальної статті	9
4.1	Приклад налаштування <code>fancyhdr</code>	66
5.1	Приклад пакета	72
5.2	Параметри макета сторінки	79

Перелік таблиць

1.1	Класи документів	10
1.2	Опції класів документів	11
1.3	Деякі з пакетів, що розповсюджуються з L ^A T _E X	13
1.4	Стандартні стилі сторінки L ^A T _E X	14
2.1	Акценти і спеціальні символи	24
2.2	Деякі з команд, визначених українською опцією пакета <code>babel</code>	27
2.3	Ключі розміщення плаваючого об'єкта	37
3.1	Акценти в математичному режимі	54
3.2	Маленькі грецькі літери	54
3.3	Великі грецькі літери	54
3.4	Бінарні відношення	55
3.5	Бінарні оператори	55
3.6	Великі оператори	56
3.7	Стрілки	56
3.8	Розділювачі	56
3.9	Великі розділювачі	56
3.10	Різні символи	57
3.11	Не-математичні символи	57
3.12	Розділювачі <i>AMS</i>	57
3.13	Грецькі та іврит літери <i>AMS</i>	57
3.14	Бінарні відношення <i>AMS</i>	58
3.15	Стрілки <i>AMS</i>	58
3.16	Заперечення бінарних відношень і стрілок <i>AMS</i>	59
3.17	Бінарні оператори <i>AMS</i>	59
3.18	<i>AMS</i> різне	60
3.19	Математичний алфавіт	60
4.1	Назви ключів пакета <code>graphicx</code> .	62
4.2	Приклади синтаксису ключів покажчика	65
5.1	Шрифти	73

5.2	Розміри шрифтів	73
5.3	Абсолютні розміри шрифтів у стандартних класах	74
5.4	Математичні шрифти	74
5.5	Одиниці виміру \TeX	77

Розділ 1

Це варто знати

Перша частина цього розділу містить короткий огляд філософії та історії \LaTeX . Друга частина зосереджується на основних структурах документів $\LaTeX 2_{\epsilon}$. Після читання цього розділу ви повинні мати загальне уявлення про те, як працює \LaTeX . Надалі це допоможе вам об'єднати всю нову інформацію в єдину картину.

1.1 The Name of the Game

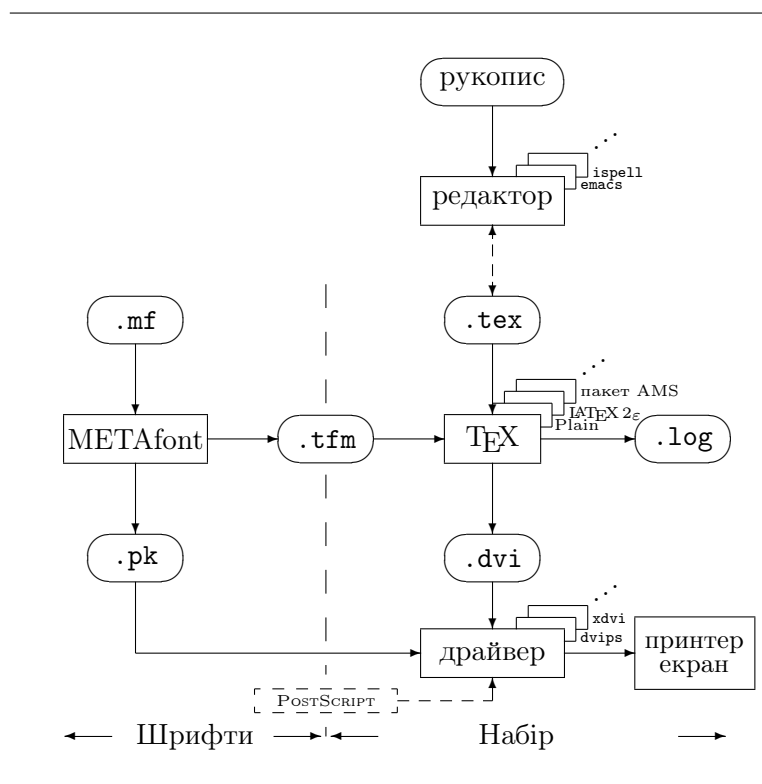
1.1.1 \TeX

\TeX — це комп'ютерна програма, створена Дональдом Кнутом (Donald E. Knuth) [2]. Вона призначена для верстки тексту і математичних формул. Кнут почав писати \TeX у 1977 році, щоб дослідити потенціал цифрового друкарського обладнання, яке в той час набувало поширення в поліграфічній промисловості. Він сподівався, що зможе переломити тенденцію деградації типографської якості, від чого, на його думку, потерпали його власні книги і статті. \TeX , у його теперішньому вигляді, вийшов у світ у 1982 році, з деякими вдосконаленнями, доданими у 1989 році для кращої підтримки 8-бітних символів та багатомовності. \TeX відомий надзвичайною стабільністю, здатністю працювати на багатьох комп'ютерних платформах і операційних системах, а також практично повною відсутністю помилок. Номер версії \TeX сходиться до числа π і зараз дорівнює 3.14159.

\TeX вимовляється «тех». У середовищі ASCII \TeX пишеться `TeX`.

1.1.2 \LaTeX

\LaTeX — макропакет, що дозволяє авторам здійснювати набір і друкувати їхні роботи з високою типографською якістю за допомогою професійно розроблених макетів. \LaTeX був написаний Леслі Лампортом (Leslie Lamport) [1]. Він використовує \TeX як набірника.

Рис. 1.1: Складові системи Т_ЕX

У 1994 році пакет L^AT_EX був оновлений командою L^AT_EX3 на чолі з Франком Міттельбахом (Frank Mittelbach), з метою включення деяких давно очікуваних поліпшень, а також об'єднання усіх вдосконалених версій, що з'явилися з часу виходу версії L^AT_EX 2.09 багато років тому. Щоб не плутати цю нову версію зі старою, вона називається L^AT_EX 2_ε. Цей документ описує саме L^AT_EX 2_ε.

L^AT_EX вимовляється «лейтех» або «латех». Якщо ви посилаєтесь на L^AT_EX у середовищі ASCII, пишіть LaTeX. L^AT_EX 2_ε пишеться LaTeX2ε.

Рисунок 1.1 на сторінці 2 показує, як працюють разом T_EX і L^AT_EX 2_ε. Схему запозичено з CTAN:/tex-archive/info/whatstex/wots.tex (автор Kees van der Laan).

1.2 Основи

1.2.1 Автор, дизайнер і набірник

Для того, щоб опублікуватися, автори віддають свої рукописи до видавництва. Там один з дизайнерів визначає макет документа (ширину стов-

пчиків, шрифти, інтервали перед та після заголовків, та ін.). Дизайнер записує свої інструкції у рукопис і віддає його набірнику, який здійснює набір книги згідно з цими інструкціями.

Дизайнер–людина намагається зрозуміти, що автор мав на увазі, коли створював рукопис. Покладаючись на свій професійний досвід та виходячи із змісту рукопису, він вирішує, які елементи є заголовками розділів, цитатами, прикладами, формулами, і т. ін.

У середовищі \LaTeX , \LaTeX бере на себе роль дизайнера книги, використовуючи \TeX як набірника. Але \LaTeX — це «лише» комп'ютерна програма, і тому потребує чітких інструкцій. Автор повинен надати додаткову інформацію, яка описує логічну структуру його роботи. Ця інформація записується в текст у вигляді «команд \LaTeX ».

Це в суттєво відрізняється від підходу WYSIWYG¹, прийнятого в більшості сучасних текстових процесорів, таких як *MS Word* чи *Corel WordPerfect*. У цих програмах автори форматують документ інтерактивно одночасно з набором тексту на комп'ютері. У процесі роботи вони можуть бачити на екрані, який вигляд матиме документ, коли його буде надруковано.

Використовуючи \LaTeX , як правило, неможливо побачити кінцеву картину під час набору тексту. Відформатований документ, однак, можна проглянути на екрані після обробки файлу \LaTeX . При потребі, виправлення можна зробити перед, власне, роздруковкою.

1.2.2 Дизайн макета

Типографський дизайн — це професія. Недосвідчені автори часто припускаються серйозних помилок форматування, гадаючи, що дизайн книги — це здебільшого питання естетики: «якщо документ виглядає художньо, він має гарний дизайн». Але, через те, що документ призначений для читання, а не для вивішування в картинній галереї, зручність його читання і розуміння набагато важливіша, ніж краса. Наприклад:

- Розмір шрифтів і нумерація заголовків повинні вибиратися так, щоб зробити структуру розділів і підрозділів зрозумілою для читача.
- Рядок повинен бути досить коротким, щоб не напружувати ока читача, і досить довгим для красивого заповнення сторінки.

Користуючись системами WYSIWYG, автори часто створюють естетично привабливі документи зі слабко вираженою або непослідовною структурою. \LaTeX дозволяє запобігти таких помилок форматування, примушуючи автора декларувати логічну структуру його документа. Потім уже \LaTeX вибирає найбільш придатний макет.

¹What you see is what you get — що ви бачите, те й отримуєте.

1.2.3 Переваги і недоліки

Коли люди зі світу WYSIWYG зустрічаються з користувачами L^AT_EX, вони часто обговорюють «переваги L^AT_EX над звичайним текстовим процесором», чи навпаки. Найкраще, що ви можете зробити, коли починається така дискусія, — не висовуватися, тому, що вона часто виходить з-під контролю. Однак, іноді ви не зможете уникнути . . .

Ну, то ось вам деякі аргументи для використання в подібних дискусіях. Основні переваги L^AT_EX над звичайними текстовими процесорами:

- Існують професійно виконані макети, що надають документам вигляду «як з видавництва».
- Зручність набору математичних формул.
- Користувачу потрібно вивчити лише кілька зрозумілих команд, що визначають логічну структуру документа. Йому практично ніколи не потрібно возитися з власне макетом документа.
- Легко створюються навіть складні структури, такі, як примітки, зміст, бібліографія, предметний покажчик, та інше.
- Для вирішення багатьох типографських задач, які не підтримуються прямо базовим L^AT_EX, є вільно розповсюджені додаткові пакети. Наприклад, існують пакети для включення POSTSCRIPT графіки, або для форматування бібліографії у точній відповідності з конкретними стандартами. Багато які з цих додаткових компонентів описані в *The L^AT_EX Companion* [3].
- L^AT_EX заохочує авторів писати добре структуровані документи, тому що саме так L^AT_EX і працює — визначаючи структуру.
- T_EX, механізм форматування L^AT_EX 2_ε, — надзвичайно мобільний і вільно доступний. Тому система працює практично на всіх існуючих платформах.

L^AT_EX має також і деякі недоліки, але, я гадаю, мені буде важко знайти серед них помітні, хоча, я впевнений, інші вам налічать сотні ; -)

- L^AT_EX погано працює у того, хто продав свою душу . . .
- Хоча деякі параметри можуть бути налаштовані в межах заздалегідь виготовлених макетів, створити цілком новий макет документа не дуже просто, і це займає багато часу.²
- Дуже складно писати неструктуровані і неорганізовані документи.

²Ходять чутки, що це — одна з основних цілей майбутньої системи L^AT_EX3.

- Незважаючи на деякі підбадьорливі перші кроки, ви ніколи не зможете повністю збагнути суть Логічної Розмітки.

1.3 Вхідні файли L^AT_EX

Вхідними даними для L^AT_EX є звичайний текстовий ASCII файл. Його можна створити в будь-якому текстовому редакторі. Він містить текст документа разом з командами, що вказують L^AT_EX, як здійснювати набір.

1.3.1 Пробіли

«Порожні» символи, такі, як пробіл або табуляція, L^AT_EX трактує однаково, як «пробіл». *Кілька послідовних* порожніх символів трактуються як *один* «пробіл». Порожні символи на початку рядка звичайно ігноруються, а одиничний кінець рядка сприймається як «пробіл».

Порожній рядок між двох рядків тексту визначає кінець абзаца. *Кілька* порожніх рядків трактуються так само, як *один* порожній рядок. Нижче наведено приклад. Ліворуч — текст вхідного файлу, праворуч — форматований вивід.

Немає значення, вводите ви
один чи кілька
пробілів між словами.

Порожній рядок починає
новий абзац.

Немає значення, вводите ви один чи кілька
пробілів між словами.

Порожній рядок починає новий абзац.

1.3.2 Зарезервовані символи

Наступні символи є зарезервованими символами, тобто такими, що або мають у L^AT_EX спеціальне значення, або наявні не у всіх шрифтах. Якщо ви введете їх у текст прямо, вони звичайно не надрукуються, а змусять L^AT_EX робити непередбачені речі.

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

Як ви пізніше побачите, ці символи можна використовувати у ваших документах, додавши попереду «\»:

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

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

Інші символи, і ще багато чого, можна надрукувати з використанням спеціальних команд. Знак «\» *не можна* отримати за допомогою комбінації (\\), оскільки ця команда використовується для розриву рядка³.

³Скористайтеся натомість командою `$$backslash$`. Вона дає «\».

1.3.3 Команди \LaTeX

Команди \LaTeX чутливі до регістра і мають один із двох наступних форматів:

- Вони починаються символом «\» і продовжуються ім'ям, що складається тільки з літер. Кінцевою межею імені команди є пробіл, цифра, чи будь-яка інша «не-літера».
- Вони складаються з «\» і рівно однієї «не-літери».

\LaTeX ігнорує пробіли після команд. Щоб отримати пробіл після команди, потрібно увести «{» і пробіл, або спеціальну команду пробілу після імені команди. «{» не дає \LaTeX «з'їсти» всі пробіли після імені команди.

Я читав, що Кнут поділяє
людей, що працюють з \TeX
на \TeX ників та \TeX пертів.\\
Сьогодні "--- \today

Я читав, що Кнут поділяє людей, що працюють з \TeX на \TeX ників та \TeX пертів.
Сьогодні — 15 листопада 2002 р.

Деяким командам потрібні параметри, що повинні бути задані між фігурними дужками «{ }» після імені команди. Деякі команди підтримують необов'язкові параметри, що додаються після імені команди в квадратних дужках «[]». Наступний приклад показує використання кількох команд \LaTeX . Не хвилюйтесь, вони будуть розтлумачені пізніше.

Ви можете на мене
 $\text{\textsl{покластися}}!$

Ви можете на мене *покластися!*

Будь ласка, почніть новий
рядок прямо отут! \newline
Дякую!

Будь ласка, почніть новий рядок прямо отут!
Дякую!

1.3.4 Коментарі

Коли, обробляючи вхідний файл, \LaTeX зустрічає символ %, він ігнорує залишок та кінець поточного рядка, і всі пробіли на початку наступного рядка.

Це корисно для додавання у вхідний файл коментарів, які не повинні виводитися на друк.

Символом % можна також скористатися, щоб розбити довгі рядки в тих місцях, де не дозволяються пробіли або кінець рядка.

```
Це "--- % безглуздий
% Краще: інструктивний <----
приклад: полісульфоніл%
          пиперидиніл%
          метиленгідроксид
```

Це — приклад: полісульфонілпиперидиніл-метиленгідроксид

Для довгих коментарів слід скористатися оточенням `comment`. Це оточення визначене в пакеті `verbatim`, тому для використання `comment` вам необхідно додати команду `\usepackage{verbatim}` до преамбули документа.

```
Це "--- інший
\begin{comment}
можливо, також безглуздий,
але корисний
\end{comment}
приклад вбудовування
коментарів у ваш документ.
```

Це — інший приклад вбудовування коментарів у ваш документ.

Пам'ятайте, це не працюватиме у середині складних оточень, наприклад математичних.

1.4 Структура вхідного файлу

Коли $\text{\LaTeX} 2_{\epsilon}$ обробляє вхідний файл, він очікує відповідності певній структурі. Зокрема, кожен документ повинен починатися командою

```
\documentclass{...}
```

Вона вказує, документ якого типу ви збираєтеся писати. Після цього можна включити команди, що впливають на стиль документа в цілому, чи завантажити пакети, що додають системі \LaTeX нові якості. Для завантаження такого пакета використовується команда

```
\usepackage{...}
```

Коли все настроювання завершено⁴, власне текст починається командою

```
\begin{document}
```

Тепер ви вводите текст із командами \LaTeX . Наприкінці документа слід додати команду

```
\end{document}
```

⁴Область між `\documentclass` і `\begin{document}` називається *преамбулою*.

Усе, що йде після цієї команди, L^AT_EX ігнорує.

Рис. 1.2 показує вміст мінімального файлу для L^AT_EX 2_ε. Трохи складніший вхідний файл наведено на рис. 1.3⁵.

1.5 Типова сесія

Б'юся об заклад, ви не можете дочекатися моменту, коли зможете спробувати роботу L^AT_EX на прикладі маленького гарненького файлу, зображеного на сторінці 9. Ось кілька порад. L^AT_EX не має графічного середовища з вишуканими меню та кнопками. Це просто комп'ютерна програма, що обробляє ваш вхідний файл. Деякі дистрибутиви L^AT_EX мають графічний інтерфейс, де можна, клацнувши мишкою, примусити L^AT_EX компілювати вхідний файл. Але «Справжні Чоловіки не Клацають Мишкою», тому ось, як умовити L^AT_EX компілювати ваш вхідний файл у системі з алфавітно-цифровим терміналом. Зважте, цей опис виходить з припущення, що L^AT_EX уже встановлений на вашому комп'ютері.

1. Створіть/відредагуйте ваш вхідний L^AT_EX файл. Він повинен бути простим текстовим ASCII файлом. В Unix усі редактори створюють саме такі файли. Під Windows вам потрібно переконатися, що файл записано саме в форматі ASCII, або *Plain Text*. Вибираючи ім'я файла, переконайтеся, що він має розширення `.tex`.
2. Запустіть L^AT_EX з вашим вхідним файлом

```
latex foo.tex
```

У випадку успішного завершення ви отримаєте `.dvi` файл. Вам можливо доведеться запістити L^AT_EX кілька разів для того, щоб отримати прасильні перехресні посилання. Якщо у вхідному файлі є помилки, L^AT_EX повідомить про це і припинить його обробку.

3. Тепер його можна продивитися

```
xdvi foo.dvi
```

або конвертувати в POSTSCRIPT

```
dvips -Pcmz foo.dvi -o foo.ps
```

`xdvi` та `dvips` є вільні (open-source) засоби для роботи з `.dvi` файлами. Перша показує їх на дисплеї в середовищі X11, а інша — ство-

⁵Для того, щоб документи могли містити українські літери, необхідно підключити пакет українізації L^AT_EX. Пакет `babel` є стандартним засобом інтернаціоналізації і підтримує, серед інших, українську мову. (*Прим. пер.*)

```
\documentclass{article}
\usepackage[ukrainian]{babel}
\begin{document}
Мале є прекрасним.
\end{document}
```

Рис. 1.2: Мінімальний файл L^AT_EX

```
\documentclass[a4paper,11pt]{article}
\usepackage[T2A]{fontenc}
\usepackage[koi8-u]{inputenc}
\usepackage[ukrainian]{babel}
% Визначаємо заголовок
\author{Г.~Паргл}
\title{Мінімалізм}
\begin{document}
\frenchspacing
\begin{document}
% Генеруємо заголовок
\maketitle
% Створюємо зміст
\tableofcontents
\section{\cyr{}Початок}
Ось тут і починається моя чудова стаття.
\section{\cyr{}Кінець}
\ldots{} а отут вона закінчується.
\end{document}
```

Рис. 1.3: Приклад реалістичної журнальної статті

рює POSTSCRIPT файл для друку. Якщо ви працюєте не в системі Unix, для роботи з .dvi файлами можуть існувати інші засоби.

1.6 Макет документа

1.6.1 Класи документів

Перше, що L^AT_EX повинен знати, обробляючи вхідний файл, це тип створеного автором документа. Він задається командою

```
\documentclass[опції]{клас}
```

Тут *клас* визначає тип створюваного документа. Таблиця 1.1 перелічує класи документів, розглянуті у цьому *Вступі*. До складу L^AT_EX 2_ε входять додаткові класи для інших документів, у тому числі листи і слайди. Параметр *опції* змінює поведінку класу документа. Опції повинні бути розділені комами. Найбільш часто вживані опції стандартних класів документів перелічені у таблиці 1.2.

Приклад: вхідний файл для документа L^AT_EX може починатися рядком

```
\documentclass[11pt,twoside,a4paper]{article}
```

Ця команда змушує L^AT_EX набирати документ як *статтю*, з базовим розміром шрифту *одинадцять пунктів* і формувати документ для *двостороннього* друку на аркушах *формату А4*.

Табл. 1.1: Класи документів

article	для статей у наукових журналах, презентацій, коротких звітів, програмної документації, запрошень. . .
report	для довших звітів з кількома розділами, невеликих книжок, дисертацій. . .
book	для справжніх книг
slides	для слайдів. Використовує великі літери без зарубок. Замість нього можна використовувати FoilT _E X ^a .

^aCTAN:/tex-archive/macros/latex/contrib/supported/foiltex

Табл. 1.2: Опції класів документів

<code>10pt, 11pt, 12pt</code>	Встановлює розмір основного шрифту документа. Якщо жодна з цих опцій не зазначена, використовується <code>10pt</code> .
<code>a4paper, letterpaper...</code>	Визначає розмір аркуша. Якщо не вказано, використовується <code>letterpaper</code> . Аналогічно можуть бути задані <code>a5paper, b5paper, executivepaper</code> і <code>legalpaper</code> .
<code>fleqn</code>	Формули будуть не відцентровані, а вирівняні ліворуч.
<code>leqno</code>	Формули нумеруються не праворуч, а ліворуч.
<code>titlepage, notitlepage</code>	Вказують, розміщувати заголовок на окремому (титульному) аркуші, чи ні. Стандартно класи <code>report</code> і <code>book</code> формують титульну сторінку, починаючи наступний матеріал з нової, а <code>article</code> — ні.
<code>twocolumn</code>	Форматування документа у два стовпчики.
<code>twoside, oneside</code>	Форматує сторінки для друку на двох, чи на одній стороні аркуша. Якщо не вказано, <code>article</code> і <code>report</code> використовують односторонній вивід, а клас <code>book</code> — двосторонній вивід. Зважте, опція <code>twoside</code> впливає на формат сторінок, а <i>не</i> вказує принтеру друкувати з двох сторін.
<code>openright, openany</code>	Розділи починаються тільки з правої (непарної), чи з будь-якої сторінки. Для класу <code>article</code> ці опції не мають сенсу, оскільки там немає розділів. Клас <code>report</code> стандартно починає розділ з наступної сторінки, а клас <code>book</code> — з непарної.

1.6.2 Пакети

У процесі створення деяких документів, вам, ймовірно, зустрінуться проблеми, які не можна вирішити за допомогою базового \LaTeX . Якщо ви захочете включити у документ графіку, кольоровий текст чи вхідний код програми з зовнішнього файлу, можливості \LaTeX потрібно буде розширити. Такі розширення називаються пакетами. Пакети активізуються командою

```
\usepackage[опції]{пакет}
```

Де *пакет* — це ім'я пакета, а *опції* — список ключових слів, що активізують спеціальні властивості пакета. Деякі пакети включені в основний дистрибутив $\LaTeX 2_{\epsilon}$ (див. таблицю 1.3), інші розповсюджуються окремо. Додаткова інформація про встановлені у вас пакети може міститися в *Local Guide* [4]. Основне джерело інформації про пакети \LaTeX — це *The \LaTeX Companion* [3]. Ця книга містить описи сотень пакетів разом з інформацією про те, як писати ваші власні розширення для $\LaTeX 2_{\epsilon}$.

1.6.3 Стили сторінки

\LaTeX підтримує три стандартні комбінації верхнього та нижнього колонтитулів — так звані стилі сторінки. Параметр *стиль* команди

```
\pagestyle{стиль}
```

визначає, який із них використовувати. Стандартні стилі сторінки перелічені в таблиці 1.4.

Стиль поточної сторінки можна змінити командою

```
\thispagestyle{стиль}
```

Опис того, як створювати ваші власні стилі колонтитулів, дивіться в *The \LaTeX Companion* [3], а також у розділі 4.4 на сторінці 65.

1.7 Файли, які вам можуть трапитися

Працюючи з \LaTeX , ви скоро опинитесь в лабіринті файлів з різними розширеннями, і, можливо, без поняття. Нижче наведено список типів файлів, які ви можете зустріти, працюючи з \LaTeX . Ця таблиця не претендує на виключну повноту, але якщо котресь із розширень, важливих, на вашу думку, тут відсутнє, дайте, будь ласка, знати.

.tex Вхідний файл \LaTeX або \TeX . Може бути компільований за допомогою `latex`.

Табл. 1.3: Деякі з пакетів, що розповсюджуються з L^AT_EX

<code>doc</code>	Дозволяє документувати програми на L ^A T _E X. Описаний у <code>doc.dtx</code> ^a і в <i>The L^AT_EX Companion</i> [3].
<code>exscale</code>	Надає масштабовані версії додаткових математичних шрифтів. Описаний у <code>ltxscale.dtx</code> .
<code>fontenc</code>	Вказує кодування шрифту L ^A T _E X. Описаний у <code>ltoutenc.dtx</code> .
<code>ifthen</code>	Надає команди виду «якщо ..., то виконувати ..., інакше виконувати ...». Описаний у <code>ifthen.dtx</code> і в <i>The L^AT_EX Companion</i> [3].
<code>latexsym</code>	Щоб підключити шрифт спеціальних символів L ^A T _E X, потрібно використовувати пакет <code>latexsym</code> . Описаний у <code>latexsym.dtx</code> і в <i>The L^AT_EX Companion</i> [3].
<code>makeidx</code>	Надає команди для створення покажчиків. Описаний у розділі 4.3 і в <i>The L^AT_EX Companion</i> [3].
<code>syntonly</code>	Обробляє документ, не здійснюючи власне набору, що зручно для швидкої перевірки на помилки. Описаний у <code>syntonly.dtx</code> і в <i>The L^AT_EX Companion</i> [3].
<code>inputenc</code>	Дозволяє вказати вхідне кодування, наприклад, ASCII, ISO Latin-1, ISO Latin-2, 437/850 IBM code pages, Apple Macintosh, Next, ANSI-Windows чи визначене користувачем. Описаний у <code>inputenc.dtx</code> .

^aЦей файл повинен бути встановлений у вашій системі, і ви можете отримати `dvi` файл, набравши `latex doc.dtx` у будь-якому каталозі, де ви маєте право на запис. Те саме стосується усіх інших файлів, згаданих у цій таблиці.

- `.sty` ЛАТ_EX макро пакет. Такий файл можна підключити до вашого ЛАТ_EX документа за допомогою команди `\usepackage`.
- `.dtx` Документований Т_EX. Це основний формат розповсюдження стильових файлів ЛАТ_EX. Якщо обробити `.dtx` файл, то буде створено документований код макро пакета ЛАТ_EX, що міститься у файлі `.dtx`.
- `.ins` Установник для файлів, що містяться у відповідному файлі `.dtx`. Завантаживши пакет ЛАТ_EX з мережі, ви звичайно отримаєте файли `.dtx` та `.ins`. Для того, щоб розпакувати `.dtx` файл, виконайте ЛАТ_EX з файлом `.ins`.
- `.cls` Файли класів, які визначають вигляд вашого документа. Потрібний клас можна вибрати командою `\documentclass`.

Під час обробки вашого документа ЛАТ_EX створює наступні файли:

- `.dvi` Device Independent file (пристрій-незалежний файл). Це — головний результат компілювання ЛАТ_EX. ви можете продивитися його зміст програмою перегляду DVI, або надіслати на друк за допомогою `dvips` чи аналогічної програми.
- `.log` Дає детальний звіт про те, що трапилося під час останнього проходу ЛАТ_EX.
- `.toc` Містить назви розділів та інших структурних елементів. Він читається під час наступного проходу ЛАТ_EX, і використовується для побудови змісту.
- `.lof` Те ж саме, що `.toc`, але для списку ілюстрацій.

Табл. 1.4: Стандартні стилі сторінки ЛАТ_EX

plain	друкує номери в середині нижнього колонтитула. Цей стиль установлений за замовчуванням.
headings	друкує назву поточного розділу і номер сторінки у верхньому колонтитулі, а нижній залишається порожнім. (Цей стиль використано у даному документі.)
empty	робить верхні та нижні колонтитули порожніми. (Цей стиль автоматично застосовується до першої сторінки кожного розділу.)

- .lot Те саме для списку таблиць.
- .aux Ще один файл, що передає інформацію від одного проходу до наступного. Серед інших речей, .aux файл використовується для зберігання інформації, пов'язаної з перехресними посиланнями.
- .idx Якщо ваш документ містить покажчик, у цьому файлі \LaTeX тримає слова, які потрібно туди занести. Обробіть цей файл програмою `makeindex`. Дивіться розділ 4.3 на сторінці 64 для більш детальної інформації про створення покажчиків.
- .ind Оброблений .idx файл, готовий для включення у ваш документ у наступному циклі компілювання.
- .ilg Звіт про роботу `makeindex`.

1.8 Великі проекти

При роботі з великими документами буває зручно розділити вхідний файл на кілька частин. \LaTeX має дві команди, які допоможуть це зробити. Команду

```
\include{файл}
```

можна використовувати в тілі документа, щоб включити у нього вміст файлу з назвою *файл.tex*. Пам'ятайте, що \LaTeX почне нову сторінку, перш, ніж обробляти матеріал з *файл.tex*.

Для вибіркового читання файлів, включених командами `\include`, можна скористатися командою

```
\includeonly{файл,файл,...}
```

Після виконання в преамбулі документа цієї команди \LaTeX буде включати тільки файли, перераховані в її аргументі. Зверніть увагу, між іменами файлів і комами не повинно бути пробілів.

Команда `\include` починає набір включеного тексту з нової сторінки. Це зручно при використанні `\includeonly`, тому що, навіть коли деякі включені файли пропущені, межі сторінок не будуть зміщуватися, і заголовки структурних елементів пропущених файлів залишаться у змісті. Проте іноді це небажано, і тоді ви можете використати команду

```
\input{файл.tex}
```

Вона просто включає вміст зазначеного файлу. Зверніть увагу, що у цьому випадку потрібно вказувати розширення файла.

Для того, щоб \LaTeX швидко перевірів ваш документ, можна скористатися пакетом `syntonly`. Це примусить \LaTeX продивитися документ, перевіряючи тільки синтаксис і правильність використання команд, без створення вихідного `.dvi` файла. У цьому режимі \LaTeX працює набагато швидше і ви можете заощадити цінний час. Використання дуже просте:

```
\usepackage{syntonly}  
\syntaxonly
```

Коли ви захочете отримати вивід, просто закоментуйте другий рядок (додавши знак процента).

Розділ 2

Набір тексту

Після прочитання попереднього розділу ви повинні мати уявлення, з чого складається документ \LaTeX . У цьому розділі ви отримаєте знання, необхідні для створення реальних документів.

2.1 Структура тексту і мови

Основна мета написання тексту (за виключенням деякої сучасної UVA¹ літератури), — передати читачу ідеї, інформацію, чи знання. Читач краще зрозуміє текст, якщо ці ідеї структуровано, і набагато краще побачить і відчує цю структуру, якщо типографська форма відбиває логічну і семантичну структуру документа.

\LaTeX відрізняється від інших систем комп'ютерного набору тим, що вам потрібно лише задавати йому логічну і семантичну структуру тексту. Він потім вибирає типографську форму відповідно до «правил», заданих у файлі класу документа та в різних стильових файлах.

Найважливіший елемент тексту в \LaTeX (і в типографії взагалі) — абзац. Він називається «одиницею тексту», тому що абзац є тією типографською формою, яка повинна відображувати одну зв'язну думку або ідею. Далі ви довідаєтеся, як можна отримати розрив рядка, наприклад, за допомогою \backslash , або початок абзаца, наприклад, залишивши порожній рядок. Тому, якщо починається нова думка, повинен починатися новий абзац, а, якщо ні, — використовуйте розрив рядка. Якщо ви маєте сумнів щодо поділу на абзаци, подумайте про ваш текст, як про носія ідей і думок. Якщо ви почали новий абзац, продовжуючи стару думку, — заберіть початок нового абзаца. Якщо протягом абзаца починається зовсім нова думка — розбийте його на два.

Більшість людей недооцінюють важливість належної організації абзаців. Багато хто навіть не розуміє значення абзацних розривів, і, особливо

¹Швейцарською німецькою «не схожий будь-якою ціною» (Um's Verrecken Anders).

в \LaTeX , починає новий абзац, навіть не усвідомлюючи цього. Останню помилку особливо легко зробити, коли в тексті є формули. Подивіться на наступні приклади і спробуйте зрозуміти, чому іноді в них використовуються порожні рядки (розриви абзаців), а іноді — ні. (Якщо ви ще не знаєте усіх команд на рівні, достатньому для розуміння прикладів, прочитайте, будь ласка, цей і наступний розділи, і потім поверніться до прикладів знову.)

```
% Приклад 1
\ldots коли Ейнштейн вивів свою формулу
\begin{equation}
  e = m \cdot c^2 \ ; \ ,
\end{equation}
яка є одночасно найширше відомою
і найменш зрозумілою фізичною формулою.
```

```
% Приклад 2
\ldots з чого випливає закон струму Кірхгофа:
\begin{equation}
  \sum_{k=1}^n I_k = 0 \ ; \ .
\end{equation}
```

Закон напруги Кірхгофа можна вивести \ldots

```
% Приклад 3
\ldots який має кілька переваг.

\begin{equation}
  I_D = I_F - I_R
\end{equation}
є ядром зовсім іншої моделі транзистора.
\ldots
```

Наступна, менша, одиниця тексту — речення. У англійському тексті після крапки, що завершує речення, ставиться довший пробіл, ніж після крапки, яка стоїть після скорочення. \LaTeX намагається визначити, яку з них ви мали на увазі. Якщо він помиляється, ви повинні йому підказати. Це пояснено далі у цьому розділі.

Належна організація тексту стосується і окремих частини речень. Більшість мов використовує дуже складні правила пунктуації, але в деяких (включаючи німецьку і англійську) ви розставите майже усі коми на місця, просто пам'ятаючи, що вони значать: коротку зупинку в потоці викладу. Якщо ви не впевнені у своїх комах, прочитайте речення вголос, роблячи коротку паузу на кожній комі. Якщо десь це пролунає дивно,

заберіть цю кому, якщо ви відчуєте необхідність зупинки в іншому місці, поставте кому там.

Нарешті, абзаци тексту повинні бути також логічно структуровані на більш високому рівні поєднанням у розділи, підрозділи, і так далі. Однак, типографський ефект написання, наприклад, `\section{Структура тексту і мови}`, настільки очевидний, що повинно бути одразу зрозуміло (для тих, хто знає англійську мову:-), як використовувати ці високорівневі структури.

2.2 Поділ на рядки і сторінки

2.2.1 Вирівнювання абзаців

Набір книг часто здійснюється так, щоб усі рядки мали однакову довжину. \LaTeX вставляє необхідні розриви рядків і пробіли між словами, оптимізуючи форматування усього абзацу. При необхідності він також переносить слова, які не можна розмістити в рядку. Форматування абзаців залежить від класу документа. Звичайно кожен абзац починається з відступу, а між двома послідовними абзацами не робиться додаткового інтервалу. Подробиці дивіться у розділі [5.3.2](#).

У деяких випадках може бути необхідно вказати \LaTeX , що потрібно розірвати рядок у певному місці:

```
\ \ чи \newline
```

починають новий рядок, не починаючи нового абзацу.

```
\ \*
```

забороняє, крім того, розрив сторінки після вставленого розриву рядка.

```
\newpage
```

починає нову сторінку.

Команди

```
\linebreak[n], \nolinebreak[n], \pagebreak[n] і \nopagebreak[n]
```

призведуть, відповідно, до розриву рядка, заборони розриву рядка, розриву сторінки і заборони розриву сторінки. Необов'язковий аргумент n дозволяє автору впливати на їхні дії. Він може приймати значення від нуля до чотирьох. Значення n менше від 4 залишає \LaTeX можливість ігнорувати вашу команду, якщо результат виглядав би дуже погано. Не

плутайте команди «розриву» (`break`) з командами «початку» (`new`). Навіть якщо ви задасте команду «розриву», \LaTeX спробує вирівняти правий край сторінки і загальну висоту сторінки, як описано в наступному підрозділі (секції). Якщо ви дійсно хочете почати «новий рядок» (`new line`), використовуйте відповідну команду. Вгадайте, як вона називається!

\LaTeX завжди намагається оптимальним чином розбити абзац на рядки. Якщо він не може знайти спосіб розбити рядки у відповідності зі своїми стандартами, він дозволяє одному рядку виступати з абзацу вправо. Тоді \LaTeX скаржиться на «`overflow hbox`». Найчастіше це трапляється, коли він не може знайти місця для переносу слова². Можна примусити \LaTeX трохи послабити свої стандарти, давши команду `\sloppy`. Це дозволить запобігти появі рядків, що виступають, збільшуючи інтервали між словами, навіть якщо кінцевий результат буде далеким від оптимального. У цьому випадку користувач отримує попередження про «`underfull hbox`». В багатьох випадках абзац матиме не дуже гарний вигляд. Команда `\fussy` дозволить повернути початкові високі стандарти поділу рядків.

2.2.2 Переноси

У разі потреби \LaTeX розбиває слова для переносу на наступний рядок. Якщо алгоритм переносів не знаходить правильних місць переносу, ви можете виправити положення, повідомивши \TeX про винятки за допомогою наступних команд.

Якщо задати команду

```
\hyphenation{список слів}
```

то слова, перераховані у її аргументі, будуть переноситися тільки у місцях, позначених «-». Ця команда повинна містити тільки слова, що складаються зі звичайних літер або, радше, знаків, які \LaTeX вважає звичайними літерами. Винятки переносів зберігаються для мови, котра є активною в момент виконання команди `\hyphenation`. Це значить, що якщо ви використаєте цю команду в преамбулі документа, вона впливатиме на правила переносу для англійської мови. Якщо вона буде вжита після `\begin{document}` і ви використовуєте пакет для підтримки національних мов, наприклад, `babel`, винятки переносів стосуватимуться мови, активованої за допомогою `babel`.

Наступний приклад дозволяє переноси в слові «`hyphenation`», так само, як і в слові «`Hyphenation`», і забороняє переноси в словах «`FORTRAN`», «`Fortran`» і «`fortran`»:

²Хоча \LaTeX і дає попередження, такі рядки не завжди легко знайти. Якщо в команді `\documentclass` використати опцію `draft`, такі рядки будуть відзначені жирною чорною рисою на правому полі.

`\hyphenation{FORTRAN Hy-phen-a-tion}`

Команда `\-` вставляє в слово можливі точки переносу. Ця команда особливо корисна для слів, що містять спеціальні символи (наприклад, символи з діакритичними акцентами), тому що \LaTeX не переносить такі слова автоматично.

I think this is: su`\-`per`\-`cal`\-`
i`\-`frag`\-`i`\-`lis`\-`tic`\-`ex`\-`pi`\-`
al`\-`i`\-`do`\-`ci`\-`ous

I think this is: supercalifragilisticexpialidocious

Кілька слів можна втримати разом на одному рядку командою

`\mbox{текст}`

Аргумент команди не буде перенесено за будь-яких обставин.

Номер мого телефону незабаром зміниться на `\mbox{0116 291 2319}`.

Параметр `\mbox{\emph{назва файлу}}` повинен містити назву файлу.

Номер мого телефону незабаром зміниться на 0116 291 2319.

Параметр *назва файлу* повинен містити назву файлу.

Команда `\fbox` аналогічна до `\mbox`, але, на додачу, навколо вмісту буде зображено прямокутник.

2.3 Ready made Strings

В деяких прикладах на попередніх сторінках ви бачили кілька простих команд \LaTeX для набору спеціальних рядків тексту:

Команда	Приклад	Опис
<code>\today</code>	15 листопада 2002 р.	Поточна дата поточною мовою
<code>\TeX</code>	\TeX	Ім'я улюбленого набірника
<code>\LaTeX</code>	\LaTeX	The name of the Game
<code>\LaTeXe</code>	$\text{\LaTeX} 2\epsilon$	Сучасна реінкарнація \LaTeX

2.4 Спеціальні літери і символи

2.4.1 Знаки лапок

Для набору лапок *ніколи не використовуйте* знак `"`, як на друкарській машинці. В типографії існують спеціальні знаки відкриваючої і закри-

ваючої лапок. У \LaTeX використовують два знаки ‘ для відкриваючих лапок і два знаки ’ (апострофи) для закриваючих³.

‘Будь ласка, натисніть
клавішу ‘x’ ’

“Будь ласка, натисніть клавішу ‘x’ ”

2.4.2 Тире і дефіси

\LaTeX знає про чотири види тире. Три з них ви можете одержати різною кількістю послідовних знаків -. Четверте насправді зовсім не тире, а математичний знак мінус:

X-промені\\
сторінки 13--67\\
{\Eng yes---or no?}\\
\$0\$, \$1\$ та \$-1\$

X-промені
сторінки 13–67
yes—or no?
0, 1 та −1

Ці тире називаються так: «-» дефіс, «–» коротке тире, «—» довге англійське тире⁴, і «−» знак мінуса.

2.4.3 Тильда (~)

Символ, котрий часто можна побачити як частину веб-адреси, називається тильдою. В \LaTeX його можна отримати за допомогою \LaTeX , але результат: ~ може бути це не зовсім тим, чого б ви хотіли. Спробуйте це:

http://www.rich.edu/~{bush} \\
http://www.clever.edu/\$\sim\$demo

http://www.rich.edu/~bush
http://www.clever.edu/~demo

2.4.4 Символ градуса (°)

Як надрукувати символ градуса в \LaTeX ?

Уже \LaTeX , \LaTeX \\
скоро настане
надпровідність.

Уже \LaTeX , скоро настане надпровідність.

³В українській мові звичайно замість “таких” і ‘таких’ лапок використовують «такі» і „такі“. Вони задаються командами \LaTeX , \LaTeX (замість цієї пари часто використовують лігатури \LaTeX , \LaTeX), і \LaTeX , \LaTeX . Див. також розділ 2.5.1. (Прим. пер.)

⁴В українській мові використовується довге кирилічне тире «—», яке трошки коротше ніж англійське, і обмежується пробілами. Дивіться розділ 2.5.1 і таблицю 2.2. (Прим. пер.)

2.4.5 Трикрапка (...)

На друкарській машинці крапка чи кома займають стільки ж місця, як і будь-яка інша літера. Якщо ці символи надруковані в книзі, то вони займають дуже мало місця і набираються дуже близько до попередньої літери. Ви не зможете отримати трикрапку, просто надрукувавши три крапки, тому що відстані між ними будуть неправильним. Тому для трикрапки є спеціальна команда, що називається

```
\ldots
```

Не так ..., а отак:\
Нью-Йорк, Токіо, Будапешт, \ldots

Не так ..., а отак:
Нью-Йорк, Токіо, Будапешт, ...

2.4.6 Лігатури

Деякі комбінації літер виводяться на друк не просто як послідовності окремих літер, а з використанням спеціальних символів.

ff fi fl ffi... замість ff fi fl ffi ...

Ці так звані лігатури можуть бути заборонені вставкою `\mbox{}` між двома відповідними літерами. Це може бути необхідно для слів, утворених із двох частин.

Не “shelfful”,\
а “shelf\mbox{ }ful”

Не “shelfful”,
а “shelfful”

2.4.7 Акценти і спеціальні символи

LaTeX підтримує використання акцентів і спеціальних символів багатьох мов. Таблиця 2.1 показує всілякі акценти у застосуванні до літери o. Зрозуміло, що на її місці можуть бути й інші літери.

Щоб помістити знак акценту над літерами i чи j, крапки над ними повинні бути вилучені. Це досягається набором `\i` та `\j`.

```
H\^otel, na\"i ve, \'el\'eve,\  
sm\o rrebr\o d, !'Se\~norita!,\  
Sch\"onbrunner Schlo\ss\}  
Stra\ss e
```

Hôtel, naïve, élève,
smørrebrød, ¡Señorita!,
Schönbrunner Schloß Straße

Табл. 2.1: Акценти і спеціальні символи

ò	\‘o	ó	\’o	ô	\^o	õ	\~o
ō	\=o	ò	\.o	ö	\"o	ç	\c c
ǒ	\u o	ǒ	\v o	ǒ	\H o	ø	\c o
ø	\d o	ø	\b o	oo	\t oo		
œ	\oe	Œ	\OE	æ	\ae	Æ	\AE
å	\aa	Å	\AA				
ø	\o	Ø	\O	l	\l	L	\L
ı	\i	J	\j	i	!‘	ı	?‘

2.5 Підтримка багатомовності

Якщо вам потрібно створювати документи мовами, відмінними від англійської, L^AT_EX повинен бути відповідним чином сконфігурованим:

1. Усі генеровані автоматично текстові послідовності⁵ повинні бути перекладені потрібною мовою. Для багатьох мов ці зміни досягаються використанням пакета `babel` (автор Johannes Braams).
2. L^AT_EX повинен знати правила переносу для нової мови. Підключення правил переносу до L^AT_EX є більш складним. Воно включає перебудування форматного файлу з шаблонами переносів потрібних мов. ваш *Local Guide* [4] повинен містити більше інформації про це.

Якщо ваша система уже відповідним чином сконфігурована, ви можете активізувати пакет `babel` додаванням команди

```
\usepackage[мова1,мова2,мова3]{babel}
```

після команди `\documentclass`. В *Local Guide* [4] також може бути зазначено, які мови підтримує ваша система. Якщо ви використовуєте `babel` з кількома мовами, остання з *мов* стає основною мовою документа, а команду

```
\selectlanguage{мова}
```

можна використовувати для встановлення поточної мови всередині документа. `Babel` автоматично активізує правильні правила переносу

⁵Зміст, Список ілюстрацій, Бібліографія ...

для обраної вами мови. Якщо форматний файл вашого \LaTeX не підтримує переносів для обраної мови, `babel` буде працювати, але заборонить переноси, що негативно позначиться на зовнішньому вигляді документа.

Для деяких мов `babel` вводить нові команди, що спрощують набір спеціальних символів. Наприклад, німецька мова містить безліч діакритичних знаків (äöü). З використанням `babel` можна вводити ö, друкуючи "o замість \".

Деякі комп'ютерні системи дозволяють вам вводити символи національних алфавітів, які не містяться в ASCII, прямо з клавіатури. \LaTeX може обробляти такі символи. Починаючи з випуску $\text{\LaTeX} 2_{\epsilon}$ від грудня 1994 р., підтримка декількох вхідних кодувань є частиною базової поставки $\text{\LaTeX} 2_{\epsilon}$. Ознайомтеся з пакетом `inputenc`:

```
\usepackage[encoding]{inputenc}
```

При використанні цього пакета ви повинні розуміти, що інші можуть неправильно бачити ваші вхідні файли на своєму комп'ютері через використання іншого кодування. Наприклад, німецька літера ä на IBM PC кодується як 132, а на деяких Unix системах, що використовують ISO-LATIN 1, — як 228. Тому використовуйте цю можливість з обережністю.

Операційна система	кодування
Mac	<code>applemac</code>
Unix	<code>latin1</code>
Windows	<code>ansinew</code>
OS/2	<code>cp850</code>

Інша річ — кодування шрифту. Воно визначає, у якій позиції в межах шрифту \TeX знаходиться кожна літера. Оригінальний \TeX шрифт Computer Modern містить тільки 128 символів зі старого 7-бітного набору символів ASCII. Коли потрібні акцентовані літери, \TeX створює їх, комбінуючи звичайну літеру з акцентом. Незважаючи на те, що результат виглядає прекрасно, цей підхід не дає автоматичним переносам працювати усередині слів, що використовують акцентовані літери.

На щастя, більшість сучасних поставок \TeX містять шрифти EC. Ці шрифти схожі на Computer Modern, але містять спеціальні символи для більшості акцентованих літер, що використовуються у європейських мовах. Використовуючи ці шрифти, ви можете поліпшити переноси в документах мовами, що використовують латиницю і мають літери з діакритичними акцентами. Шрифти EC активізуються включенням у преамбулі

вашого документа пакету `fontenc`.

```
\usepackage[T1]{fontenc}
```

Нажаль, не існує високоякісних ЕС шрифтів у форматі `POSTSCRIPT`⁶. Зверніть увагу на пакет `ae`, відомий також як *Almost European Computer Modern*, який використовує символи шрифтів `Computer Modern`, але перевпорядковує їх відповідно кодуванню ЕС.

2.5.1 Українська мова⁷

Декілька порад для тих, хто створює документи `LATEX` українською мовою. Більшість сучасних дистрибутивів `TEX/LATEX` містять пакет `babel` із стандартною підтримкою української мови, кириличні `LN` шрифти, і таблиці українських переносів.

Якщо ви збираєтеся використовувати кирилицю в математичних формулах, спочатку у преамбулі документа необхідно підключити пакет

```
\usepackage{mathtext}
```

Далі потрібно задати кодування шрифту^{8 9}, воно повинно збігатися з кодуванням таблиці українських переносів¹⁰, з якою згенеровано файл формату (`latex.fmt`):

```
\usepackage[T2A]{fontenc}
```

Потім слід вибрати вхідне кодування. В залежності від платформи, на якій ви працюєте, це може бути `koi8-u` (`UNIX`, `Linux`), `sr866nav`

⁶Володимир Волович створив набір `cm-super`, що містить шрифти `ЕС/ТС`, `ЕС Concrete`, `ЕС Bright` та `LN` у форматі `Type1`. Він входить до дистрибутивів `TEXLive7` та `MikTEX`. Нажаль, ці шрифти гіршої типографської якості, ніж `Type1` шрифти `CM`.

⁷Цей розділ додано під час перекладу на заміну розділу про підтримку німецької мови.

⁸Український алфавіт підтримують кодування `T2A`, `LCY`, `X2`, і, частково, `OT2`.

⁹Якщо ви використовуєте пакети `AMS-LATEX`, завантажуйте їх перед пакетами `fontenc` і `babel`.

¹⁰Про те, як встановити українські переноси, читайте в [CTAN:/tex-archive/language/hyphenation/ukrhyph/README](https://ctan.org/tex-archive/language/hyphenation/ukrhyph/README) та [12], де описані також інші аспекти організації `LATEX`.

(DOS), cp1251 (Windows), або macukr (Mac):

```
\usepackage[koi8-u]{inputenc}
```

і, власне, завантажити підтримку української мови:

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

Остання команда вмикає українські переноси (якщо ваш \LaTeX відповідним чином налаштований) а також змінює всі автоматичні тексти українською, наприклад, «Chapter» стає «Розділ». Крім того, активізуються деякі установки, наприклад, `\frenchspacing`, та стають доступними ряд команд, що допомагають створювати документи українською мовою у відповідності з типографськими стандартами (див. табл. 2.2).

Табл. 2.2: Деякі з команд, визначених українською опцією пакета `babel`

" -	вставляє в слово можливу точку переносу, але, на відміну від <code>\-</code> , не забороняє переноси в інших точках слова згідно до діючих правил.
" ---	довге кирилічне тире для тексту. Воно трошки коротше, ніж англійське, обмежується короткими жорсткими пробілами, і не відривається від попереднього слова.
" =	дефіс в складних словах, що дозволяє перенос в цьому місці і в решті слова (в \LaTeX звичайний дефіс не дозволяє розрив складного слова для переносу).
" ‘ ” ’	«німецькі» ліві і праві подвійні лапки („ „“).
" < " >	ліві і праві подвійні лапки «ялинка» (« »).
\dq	прямі подвійні лапки (" ").

2.6 Пробіли між словами

Для отримання рівного правого краю \LaTeX змінює ширину інтервалів між словами. Інтервал наприкінці речення він робить дещо більшим, роблячи текст більш читабельним¹¹. \LaTeX вважає, що речення закінчуються крапками, знаками питання, чи знаками оклику. Якщо крапка йде після великої літери, вона не вважається кінцем речення, тому що крапки після великих літер звичайно використовуються для скорочень.

Будь-які винятки з цих припущень повинні бути явно зазначені в тексті. Знак «\» перед пробілом дає в результаті пробіл, який не буде

¹¹У відповідності з традиціями набору, прийнятими в англійській мові. (Прим. пер.)

збільшений. Знак «~» дає пробіл, що не може збільшитися і який, крім того, забороняє розрив рядка. Команда \@ перед крапкою вказує, що ця крапка закінчує речення, незважаючи на те, що вона йде після великої літери.

```
Mr.~Smith was happy to see her\\
cf.~Fig.~5\\
I like BASIC\@. What about you?
```

```
Mr. Smith was happy to see her
cf. Fig. 5
I like BASIC. What about you?
```

Додатковий пробіл після крапок можна заборонити командою

```
\frenchspacing
```

яка вказує L^AT_EX не робити пробіл після крапки більшим, ніж після звичайних символів. Це звичайно для мов, відмінних від англійської, за винятком бібліографій. Якщо ви використовуєте `\frenchspacing`, команда \@ не потрібна¹².

2.7 Заголовки, частини і розділи

Щоб допомогти читачу орієнтуватися у вашому документі, його потрібно розділяти на частини, розділи, та інші структурні одиниці. L^AT_EX підтримує такий поділ за допомогою спеціальних команд, аргументами яких є заголовки відповідних структурних одиниць. Ваша справа — використовувати їх у належному порядку.

В класі `article` є такі команди групування:

```
\section{...}           \paragraph{...}
\subsection{...}       \subparagraph{...}
\subsubsection{...}    \appendix
```

У класах `report` і `book` ви можете використовувати дві додаткові команди: `\part{...}`. Якщо потрібно розділити документ на частини, не впливаючи на нумерацію розділів та частин, можна використати команду `\chapter{...}`.

Оскільки в класі `article` немає розділів (chapters), статті досить легко об'єднати як розділи книги. Інтервали між розділами, нумерація і розмір шрифту заголовків L^AT_EX встановлює автоматично.

Існує дві особливі команди секціонування:

- Команда `\part` не впливає на послідовність нумерування розділів.

¹²Якщо ви використовуєте пакет `babel`, українська мова автоматично активізує `\frenchspacing`. (Прим. пер.)

- Команда `\appendix` аргументу не має. Вона просто починає нумерувати розділи літерами замість цифр¹³ і замінює «Розділ» на «Додаток».

Л^AT_EX створює зміст, беручи заголовки розділів і номери сторінок з попереднього проходу по документу. Команда

```
\tableofcontents
```

створює зміст у тому місці, де вона викликана. Щоб отримати правильний зміст, новий документ повинен бути оброблений Л^AT_EX двічі. В особливих випадках може бути необхідний і третій прохід. Л^AT_EX попередить вас, коли це буде потрібно.

Усі перераховані вище команди групування існують також у варіантах із зірочкою. Такі варіанти можна отримати додаванням `*` до імені команди. Вони генерують заголовки розділів, що не нумеруються і не включаються в зміст. Наприклад, замість команди `\section{Довідка}` слід використати `\section*{Довідка}`.

Як правило, заголовки розділів з'являються в змісті точно у тому ж вигляді, в якому вони набрані в тексті. Іноді це неможливо через те, що заголовок задовгий для змісту. Варіант заголовку розділу для змісту в таких випадках можна вказати як необов'язковий аргументом перед власне заголовком.

```
\chapter[Заголовок для змісту]{Довгий  
дуже нудний заголовок, показаний в тексті}
```

Титульний аркуш усього документа генерується за допомогою команди

```
\maketitle
```

Його вміст визначається командами

```
\title{...}, \author{...} і, при бажанні, \date{...}
```

перед тим, як буде дано команду `\maketitle`. Аргумент команди `\author` може містити кілька імен, розділених командами `\and`.

Приклад використання деяких із згаданих команд можна знайти на рис. 1.3.

¹³У класі `article` міняється нумерація підрозділів (sections).

Крім описаних вище команд групування, \LaTeX 2 ϵ уводить три додаткові команди для використання з класом `book`.

```
\frontmatter, \mainmatter і \backmatter
```

Вони корисні для поділу вашої публікації. Команди змінюють заголовки розділів і нумерацію сторінок так, як це очікується від книги:

`\frontmatter` повинна бути першою командою після `\begin{document}`.

Вона активізує нумерування сторінок римськими цифрами. Звичайним є використання команд секціонування з зірочками (наприклад, `\chapter*{Передмова}`) для вступної частини книги, оскільки в такому випадку \LaTeX не нумеруватиме їх.

`\mainmatter` іде якраз перед першим розділом книги. Вона обнулює лічильник сторінок і починає нумерувати їх арабськими цифрами.

`\appendix` вказує на початок додатків вашої книги. Після цієї команди розділи будуть пронумеровані літерами.

`\backmatter` повинна стояти перед останніми елементами вашої книги, такими, як бібліографія та предметний покажчик. В стандартних класах документів це не матиме візуального ефекту.

2.8 Перехресні посилання

У книгах, звітах і статтях часто зустрічаються перехресні посилання на ілюстрації, таблиці й окремі частини тексту. \LaTeX надає наступні команди для таких посилань:

```
\label{мітка}, \ref{мітка} і \pageref{мітка}
```

де *мітка* — обраний користувачем ідентифікатор. \LaTeX заміняє `\ref` номером розділу, підрозділу, ілюстрації, таблиці, чи теореми, де була використана відповідна команда `\label`. `\pageref` друкує номер сторінки, на якій знаходиться команда `\label`¹⁴. Так само, як і у випадку з заголовками розділів, тут використовуються номери з попереднього проходу.

Посилання на цей розділ~`\label{sec:this}` має такий вигляд: <<див. розділ~`\ref{sec:this}` на ст.~`\pageref{sec:this}`.>>

Посилання на цей розділ має такий вигляд: <<див. розділ 2.8 на ст. 30.>>

¹⁴Зверніть увагу, що ці команди не знають, на що саме вони посилаються. `\label` просто зберігає останній автоматично згенерований номер.

2.9 Виноски

Команда

```
\footnote{текст виноски}
```

друкує виноску внизу поточної сторінки. Виноски завжди повинні міститися після слова, або речення, якого вони стосуються. Виноска, що стосується речення або його частини, повинна йти після коми чи крапки^{15 16}.

```
Користувачі \LaTeX{}
часто вживають
виноски\footnote{%
  Це "--- виноска.}.
```

Користувачі ІАТЄХ часто вживають виноски^a.

^aЦе — виноска.

2.10 Виділення

У машинописних документах важливі слова виділяють підкресленням.

```
\underline{текст}
```

У друкованих виданнях ці слова виділяються *курсивом*. В ІАТЄХ для цього слугує команда

```
\emph{текст}
```

Її аргументом є текст для виділення. Що насправді робить ця команда, залежить від контексту:

```
Якщо ви використовуєте
\emph{виділення в уже
виділеному тексті},
\LaTeX{} використовує
\emph{прямий} шрифт.}
```

Якщо ви використовуєте *виділення в уже виділеному тексті*, ІАТЄХ використовує *прямий шрифт*.

Зважте на відмінність між командами *виділення* і *зміни шрифту*:

¹⁵Це в англійській мові, а в українській виноски йдуть безпосередньо після слова, перед розділовими знаками. (*Прим. пер.*)

¹⁶Зважте, виноски відволікають читача від основної частини документа. В кінці кінців, усі читають виноски, ми допитливі істоти. То, чому не інтегрувати усе, що ви бажаєте сказати, в тіло документа?¹⁷

¹⁷Вказівник не обов'язково йде туди куди він вказує :-).

```
\textit{Ви можете також
\emph{виділити} текст,
набравши його курсивом,}
\textsf{шрифтом без
\emph{зарубок,}}
\texttt{чи в стилі
\emph{друкарської машинки}.}
```

Ви можете також виділити текст, набравши його курсивом, шрифтом без зарубок, чи в стилі друкарської машинки.

2.11 Оточення

Для набору спеціальних видів тексту L^AT_EX визначає безліч оточень для різних типів форматування:

```
\begin{оточення} текст \end{оточення}
```

де *оточення* — назва оточення. Оточення можна викликати усередині інших оточень, дотримуючись порядку виклику і повернення:

```
\begin{aaa}... \begin{bbb}... \end{bbb}... \end{aaa}
```

У наступних розділах розповідається про усі важливі оточення.

2.11.1 Список, перелік та опис

Оточення `itemize` підходить для простих списків, оточення `enumerate` — для нумерованих списків, а оточення `description` — для описів.

```
\flushleft
\begin{enumerate}
\item Ви можете на ваш
смак змішувати
оточення списків:
\begin{itemize}
\item Але це може виглядати
нерозумно.
\item[--] З тире.
\end{itemize}
\item Тому пам'ятайте:
\begin{description}
\item[Дурниці] не стануть
розумнішими якщо їх
розмістити списком.
\item[Розумні] речі, однак,
цілком можна представити
у вигляді списку.
\end{description}
\end{enumerate}
```

1. Ви можете на ваш смак змішувати оточення списків:
 - Але це може виглядати нерозумно.
 - З тире.
2. Тому пам'ятайте:

Дурниці не стануть розумнішими якщо їх розмістити списком.

Розумні речі, однак, цілком можна представити у вигляді списку.

2.11.2 Вирівнювання ліворуч, праворуч і по центру

Оточення `flushleft` і `flushright` форматують абзаци, вирівняними ліворуч чи праворуч. Оточення `center` дає відцентрований текст. Якщо ви не використовуєте `\` щоб вказати розриви рядків, `LATEX` визначить їх автоматично.

```
\begin{flushleft}
  Цей текст\ вирівняний ліворуч.
  \LaTeX{} не намагається зробити
  усі рядки однакової довжини.
\end{flushleft}
```

Цей текст
вирівняний ліворуч. `LATEX` не намагається
зробити усі рядки однакової довжини.

```
\begin{flushright}
  Цей текст\ вирівняний праворуч.
  \LaTeX{} не намагається зробити
  усі рядки однакової довжини.
\end{flushright}
```

Цей текст
вирівняний праворуч. `LATEX` не
намагається зробити усі рядки однакової
довжини.

```
\begin{center}
  У центрі\Землі
\end{center}
```

У центрі
Землі

2.11.3 Цитати та вірші

Оточення `quote` корисно для цитат, важливих фраз і прикладів.

Типографське правило для
довжини рядка:
`\begin{quote}`
В середньому, довжина рядка
не повинна бути більшою
від 66~символів.
`\end{quote}`
Ось чому `\LaTeX{}` робить
поля сторінок такими
широкими, а в газетах
часто застосовують набір
у декілька шпальт.

Типографське правило для довжини рядка:

В середньому, довжина рядка
не повинна бути більшою від
66 символів.

Ось чому `LATEX` робить поля сторінок та-
кими широкими, а в газетах часто застосо-
вують набір у декілька шпальт.

Існують ще два схожих оточення: `quotation` і `verse`. Для цитат, що охоплюють кілька абзаців, корисне оточення `quotation`, тому що воно робить абзацний відступ. Оточення `verse` використовують для віршів, де важливі розриви рядків. Рядки розділяються за допомогою `\` наприкінці рядка, і порожнього рядка після кожної строфи.

```
Я знаю тільки один англійський
віршик напам'ять:
\begin{flushleft}
\begin{verse}
Humpty Dumpty sat on a wall:\\
Humpty Dumpty had a great fall.\\
All the King's horses and all
the King's men\\
Couldn't put Humpty together
again.
\end{verse}
\end{flushleft}
```

Я знаю тільки один англійський віршик напам'ять:

```
Humpty Dumpty sat on a wall:
Humpty Dumpty had a great
fall.
All the King's horses and all
the King's men
Couldn't put Humpty together
again.
```

2.11.4 Буквальне відтворення

Текст, розміщений між `\begin{verbatim}` і `\end{verbatim}` буде відтворений так, наче він надрукований друкарською машинкою, із усіма пробілами і розривами рядків, без виконання будь-яких команд \LaTeX .

Усередині абзацу аналогічну функцію виконує команда

```
\verb+текст+
```

Тут «+» — це тільки приклад символ-обмежувача. Ви може використовувати будь-який символ, крім літер, «*» чи пробілу. Багато прикладів \LaTeX у цьому буклеті набрані з використанням цієї команди.

Команда `\verb|\ldots| \ldots`

```
\begin{verbatim}
10 PRINT "HELLO WORLD ";
20 GOTO 10
\end{verbatim}
```

Команда `\ldots ...`

```
10 PRINT "HELLO WORLD ";
20 GOTO 10
```

```
\begin{verbatim*}
варіант оточення
verbatim з
зірочкою виділяє
пробіли в тексті
\end{verbatim*}
```

```
варіант_оточення
verbatim_з
зірочкою_виділяє
пробіли_в_тексті
```

Команду `\verb` теж можна використовувати аналогічним чином із зірочкою:

```
\verb*|Ось так :-)|
```

```
Ось_так_:-)|
```


Оточення `verbatim` і команду `\verb` не можна використовувати у середині параметрів інших команд.

2.11.5 Таблиці

Оточення `tabular` використовують для набору таблиць з можливістю формування горизонтальних і вертикальних ліній. \LaTeX автоматично визначає ширину стовпчиків.

Аргумент *специфікація* команди

```
\begin{tabular}[поз]{специфікація}
```

визначає формат таблиці. Використовуйте `l` для стовпчика тексту, вирівняного вліво, `r` — для тексту, вирівняного вправо і `c` — для центрального тексту; `p{ширина}` — для стовпчика зазначеної ширини, що містить вирівняний текст із переносом рядків, і `|` — для вертикальної лінії. *поз* визначає вертикальну позицію усього середовища `tabular`, `t`, `b` та `c` вказують, вирівнювати по верхній, нижній межі, чи по центру середовища.

У середині оточення `tabular` стовпчики розділяються знаком «&», команда `\\` починає новий рядок, а `\hline` вставляє горизонтальну лінію. Ви можете вставити часткову горизонтальну лінію використовуючи `\cline{j-i}`, де *j* та *i* — номери стовпчиків, на які лінія повинна розповсюджуватися.

```
\begin{tabular}{|r|l|}
\hline
54 & шістнадцяткове \\
124 & вісімкове \\
1010100 & двійкове \\
\hline \hline
84 & десяткове \\
\hline
\end{tabular}
```

54	шістнадцяткове
124	вісімкове
1010100	двійкове
<hr/>	
84	десяткове

```
\begin{tabular}{|p{4.7cm}|}
\hline
Ласкаво просимо до абзацу
у рамочці. Сподіваємося, вам усім
тут сподобається.\\
\hline
\end{tabular}
```

Ласкаво просимо до абзацу у рамочці. Сподіваємося, вам усім тут сподобається.

Роздільник стовпчиків можна задати конструкцією `@{...}`. Ця команда видаляє пробіл між стовпчиками і заміняє його на те, що включено у фігурні дужки. Одне з частих використань цієї команди буде показано далі, коли йтиметься про вирівнювання по десятковій крапці.

Інше можливе використання — для заборони ведучого пробілу в таблиці за допомогою `@{}`:

```
\begin{tabular}{@{} l @{}}
\hline
немає ведучого пробілу\\
\hline
\end{tabular}
```

немає ведучого пробілу

```
\begin{tabular}{l}
\hline
ведучий пробіл ліворуч і праворуч\\
\hline
\end{tabular}
```

ведучий пробіл ліворуч і праворуч

Оскільки в \LaTeX не існує стандартного способу вирівняти стовпчики чисел по десятковій крапці¹⁸, ми можемо «обдурити» \TeX і домогтися цього за допомогою двох стовпчиків: вирівняної вправо цілої частини і вирівняної вліво дробової. Команда `@{.}` у рядку `\begin{tabular}` замінює нормальний пробіл між стовпчиками просто на «.», даючи ефект одного стовпчика, вирівняного по десятковій крапці. Не забудьте замінити у ваших числах крапку на роздільник стовпчиків (`&`)! Мітку стовпчика можна помістити над нашим числовим «стовпчиком» командою `\multicolumn`:

```
\begin{tabular}{c r @{.} l}
Вираз із  $\pi$  & & \\
\multicolumn{2}{c}{Значення} & \\
\hline
 $\pi$  & 3&1416 & \\
 $\pi^\pi$  & 36&46 & \\
 $(\pi^\pi)^\pi$  & 80662&7 & \\
\end{tabular}
```

Вираз із π	Значення
π	3.1416
π^π	36.46
$(\pi^\pi)^\pi$	80662.7

```
\begin{tabular}{|c|c|}
\hline
\multicolumn{2}{|c|}{Ene} \\
\hline
Mene & Muh! \\
\hline
\end{tabular}
```

Ene	
Mene	Muh!

Матеріал, набраний за допомогою середовища `tabular`, завжди залишається на одній сторінці. Якщо потрібно набирати довгі таблиці, зверніть увагу на середовища `supertabular` і `longtabular`.

¹⁸Якщо на вашій системі встановлений комплект 'tools', зверніть увагу на пакет `dcolumm`.

2.12 Плаваючі об'єкти

Більшість публікацій у наші дні містять безліч ілюстрацій і таблиць. Ці елементи потребують особливого поводження, оскільки їх не можна розбити між сторінками. Одним із рішень було б починати нову сторінку щоразу, коли ілюстрація чи таблиця занадто велика, щоб поміститися на поточній сторінці. Це залишало б сторінки частково порожніми, що виглядає не дуже гарно.

Вирішення цієї проблеми є «перепливання» будь-якої ілюстрації чи таблиці, що не уміщується на поточній сторінці, на наступну, і заповнення поточної текстом. \LaTeX пропонує для плаваючих об'єктів два оточення, одне — для таблиць, і одне — для ілюстрацій. Щоб у повній мірі використати їхні переваги, важливо хоч приблизно уявляти, як \LaTeX обробляє плаваючі об'єкти. Інакше вони можуть стати джерелом розчарування, коли \LaTeX розміщуватиме їх не там, де ви хочете.

Давайте спочатку розглянемо команди \LaTeX , призначені для плаваючих об'єктів.

Будь-який матеріал, включений в оточення `figure` чи `table`, трактується як плаваючий. Обидва оточення

```
\begin{figure}[специфікація розміщення] чи
\begin{table}[специфікація розміщення]
```

мають обов'язковий параметр, названий *специфікацією розміщення*. Цей параметр використовується для вказівки, куди \LaTeX може переміщувати даний плаваючий об'єкт. *Специфікація розміщення* будується складанням у рядок *ключів розміщення плаваючого об'єкта*. Див. таблицю 2.3.

Табл. 2.3: Ключі розміщення плаваючого об'єкта

Ключ	Дозволяє розміщувати об'єкт . . .
h	<i>тут (here)</i> , у тому самому місці тексту, де він з'явився. Звичайно використовується для маленьких об'єктів.
t	<i>угорі (top)</i> сторінки
b	<i>унизу (bottom)</i> сторінки
p	на <i>спеціальній сторінці</i> , що містить тільки плаваючі об'єкти.
!	не звертати увагу на більшість внутрішніх параметрів ^a , що можуть запобігти розміщенню цього об'єкта згідно з вашими специфікаціями.

^aТаких, як максимальне число плаваючих об'єктів, дозволених на одній сторінці

Наприклад, таблицю можна почати наступним рядком:

```
\begin{table}[!hbp]
```

Специфікація розміщення `[!hbp]` дозволяє \LaTeX розмістити таблицю прямо тут (**h**), внизу тієї ж сторінки (**b**), чи на окремій сторінці (**p**), і навіть у тому випадку, якщо усе це не буде виглядати вельми добре (!). Якщо ніякої специфікації розміщення не задано, стандартні класи використовують `[tbp]`.

\LaTeX розміщує кожен плаваючий об'єкт у відповідності із заданою автором специфікацією. Якщо об'єкт не можна помістити на поточній сторінці, він розміщується у чергу ілюстрацій чи в чергу таблиць¹⁹. Коли починається нова сторінка, \LaTeX спершу перевіряє, чи можна заповнити спеціальну сторінку плаваючими об'єктами з черг. Якщо ні, то перший об'єкт з кожної черги вважається таким, що тільки що зустрівся у тексті: \LaTeX знову намагається розмістити їх у відповідності із специфікаціями (за винятком 'h', що вже неможливо). Нові плаваючі об'єкти розміщуються у відповідні черги. \LaTeX зберігає порядок, у якому зустрілися плаваючі об'єкти відповідного типу. Тому ілюстрація, яку не вдалося розмістити, відштовхує всі подальші ілюстрації до кінця документа. Отже:

Якщо \LaTeX не розміщує плаваючі об'єкти так, як ви цього очікуєте, то часто це тому, що якийсь об'єкт влаштував застримку в одній із двох черг.

Незважаючи на те, що специфікацію розміщення з одним варіантом задати можливо, робити цього не слід, оскільки це може призвести до проблем. Якщо плаваючий об'єкт не може бути розміщений в заданій позиції, він «застряє» і блокує наступні плаваючі об'єкти. Зокрема, ніколи не використовуйте опцію `[h]`: це настільки погано, що останні версії \LaTeX навіть автоматично замінюють її на `[ht]`.

Після пояснення найскладнішої частини, залишається ще кілька зауважень стосовно оточень `table` і `figure`. Командою

```
\caption{текст заголовка}
```

ви можете задати заголовок для об'єкта. \LaTeX додасть рядок «Таблиця» чи «Рис.» і поточний номер.

¹⁹Які є чергами типу *fifo*: «першим увійшов — першим вийшов».

Розділ 3

Набір математичних формул

От тепер ви готові! У цьому розділі ми ознайомимося з найсильнішим місцем \TeX : набором математики. Але майте на увазі, що цей розділ дає тільки поверхневий огляд. Хоча для багатьох з вас викладених тут речей буде досить, не впадайте у відчай, якщо ви не зможете знайти рішення, яке б задовольнило потреби набору вашої математики. Дуже вірогідно, що ваша проблема може бути вирішена за допомогою $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}^1$ ² чи якогось іншого пакета.

3.1 Загальні відомості

$\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ має спеціальний режим для набору математики. Математичний текст усередині абзацу вводиться між $\backslash($ і $\backslash)$, між $\$$ і $\$,$ або між \backslashbegin{math} і \backslashend{math} .

Додайте $\$a\$$ у квадраті і $\$b\$$ у квадраті, щоб отримати $\$c\$$ у квадраті. Або, мовою математики:
 $\$c^{\{2\}}=a^{\{2\}}+b^{\{2\}}\$$

Додайте a у квадраті і b у квадраті, щоб отримати c у квадраті. Або, мовою математики: $c^2 = a^2 + b^2$

$\backslash\text{TeX}\{\}$ вимовляється як $\backslash\tau\epsilon\chi$.
 $\backslash\tau\epsilon\chi\backslash\chi$.
 100 м^3 води.
Від усього \heartsuit .

TeX вимовляється як $\tau\epsilon\chi$.
 100 м^3 води.
Від усього \heartsuit .

Великі математичні рівняння чи формули бажано «експонувати», тобто набирати як окремий елемент між абзацами. Для цього їх потрібно

¹CTAN:/tex-archive/macros/latex/required/amslatex

²Якщо ви використовуєте пакети $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ в документах українською мовою, завантажуйте їх перед пакетами `fontenc` і `babel`. (Прим. пер.)

розмістити між `\[i \]` або всередині оточення `displaymath`. Це дасть вам нумеровані формули. Якщо ви хочете, щоб \LaTeX їх нумерував, користуйтеся оточенням `equation`.

Додавши `a` у квадраті і `b` у квадраті, отримаємо `c` у квадраті. Або, мовою математики:

```
\begin{displaymath}
c^2=a^2+b^2
\end{displaymath}
```

І ще один рядок

Додавши a у квадраті і b у квадраті, отримаємо c у квадраті. Або, мовою математики:

$$c^2 = a^2 + b^2$$

І ще один рядок

За допомогою `\label` і `\ref` ви можете робити в тексті посилання на рівняння.

```
\begin{equation}
\label{eq:eps}
\epsilon > 0
\end{equation}
З (\ref{eq:eps})
впливає \ldots
```

$$\epsilon > 0 \quad (3.1)$$

З (3.1) впливає ...

Зверніть увагу, що вирази у експонованих формулах будуть набрані в іншому стилі:

```
$$\lim_{n \to \infty}
\sum_{k=1}^n \frac{1}{k^2}
= \frac{\pi^2}{6}$$
```

$$\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$$

```
\begin{displaymath}
\lim_{n \to \infty}
\sum_{k=1}^n \frac{1}{k^2}
= \frac{\pi^2}{6}
\end{displaymath}
```

$$\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$$

Математичний режим відрізняється від *текстового режиму*. Наприклад, у *математичному режимі*:

1. Більшість пробілів і кінців рядків не приймаються до уваги, тому що всі пробіли або залежать від логіки математичних виразів, або повинні бути задані явно такими командами, як `\,`, `\quad`, чи `\qquad`.
2. Порожні рядки неприпустимі. Кожна формула займає тільки один абзац.

3. Кожна літера вважається ім'ям змінної, і відповідно набирається. Якщо потрібно ввести у формулу нормальний текст (нормальний прямий шрифт із нормальними пробілами), то слід користуватися командами `\textrm{...}`.

```
\begin{equation}
  \forall x \in \mathbf{R}:
  \quad x^2 \geq 0
\end{equation}
```

$$\forall x \in \mathbf{R}: \quad x^2 \geq 0 \quad (3.2)$$

```
\begin{equation}
  x^2 \geq 0 \quad \text{для усіх } x \in
  \mathbf{R}
\end{equation}
```

$$x^2 \geq 0 \quad \text{для усіх } x \in \mathbf{R} \quad (3.3)$$

Математики бувають дуже вибагливі щодо використовуваних символів: у цьому випадку було б традиційним використання «blackboard bold» символів, для отримання яких служить команда `\mathbb{R}` з пакетів `amsmath` чи `amssymb`. Останній приклад тепер матиме такий вигляд:

```
\begin{displaymath}
  x^2 \geq 0 \quad \text{для всіх } x \in
  \mathbb{R}
\end{displaymath}
```

$$x^2 \geq 0 \quad \text{для всіх } x \in \mathbb{R}$$

3.2 Групування в математичному режимі

Більшість команд математичного режиму діє тільки на наступний символ. Так що, якщо ви хочете, щоб команда стосувалася кількох символів, вам потрібно згрупувати їх разом за допомогою фігурних дужок: `{...}`.

```
\begin{equation}
  a^{x+y} \neq a^{x+y}
\end{equation}
```

$$a^x + y \neq a^{x+y} \quad (3.4)$$

3.3 Складові математичної формули

У цьому розділі будуть описані найбільш важливі команди, що використовуються в математичному наборі. Детальний перелік команд для набору математичних символів дивіться у розділі 3.10 на сторінці 54.

Малі грецькі літери вводяться як `\alpha`, `\beta`, `\gamma`, ..., великі літери — як `\Gamma`, `\Delta`, ...³

`\lambda, \xi, \pi, \mu, \Phi, \Omega`

$\lambda, \xi, \pi, \mu, \Phi, \Omega$

Верхні і нижні індекси вводяться за допомогою символів «`^`» та «`_`».

`a_{1} \quad x^{2} \quad $e^{-\alpha t}$ \quad a_{ij}^3`
`$e^{x^2} \neq e^{x^2}$`

$a_1 \quad x^2 \quad e^{-\alpha t} \quad a_{ij}^3$
 $e^{x^2} \neq e^{x^2}$

Квадратний корінь вводиться як `\sqrt`, корінь n -ного ступеня — за допомогою `\sqrt[n]`. Розмір знака кореня вибирається L^AT_EX автоматично. Якщо потрібен один тільки знак, використовуйте `\surd`.

`\sqrt{x}` \quad `\sqrt{x^2 + \sqrt{y}}` \quad `\sqrt[3]{2}`
`\sqrt[3]{2}` \quad `\surd[x^2 + y^2]`

$\sqrt{x} \quad \sqrt{x^2 + \sqrt{y}} \quad \sqrt[3]{2}$
 $\sqrt[3]{2}$
 $\surd[x^2 + y^2]$

Команди `\overline` і `\underline` створюють горизонтальні лінії відразу над чи під вираженням.

`\overline{m+n}`

$\overline{m+n}$

Команди `\overbrace` і `\underbrace` створюють довгі горизонтальні фігурні дужки відразу над чи під виразом.

`\underbrace{ a+b+\cdots+z }_{26}`

$\underbrace{a + b + \cdots + z}_{26}$

Для додавання знаків математичних акцентів, таких, як маленька стрілка чи знак тильда, до змінних, можна скористатися командами, перерахованим у таблиці 3.1. Широки «капелюшки» і тильди, що охоплюють кілька символів, генеруються командами `\widetilde` і `\widehat`. Символ «`'`» дає знак похідної.

³У L^AT_EX_{2_ε} велика «альфа» не визначена, тому що вона виглядає так само, як латинська «A». Коли буде завершено розробку нового кодування математичних символів, це буде змінено.

```
\begin{displaymath}
y=x^2\quad y'=2x
\quad y''=2
\end{displaymath}
```

$$y = x^2 \quad y' = 2x \quad y'' = 2$$

Вектори часто вказуються додаванням маленьких стрілок над змінною. Це робиться командою `\vec`. Для позначення вектора від A до B корисні дві команди `\overrightarrow` і `\overleftarrow`.

```
\begin{displaymath}
\vec a\quad
\overrightarrow{AB}
\end{displaymath}
```

$$\vec{a} \quad \overrightarrow{AB}$$

Для позначення оператора множення крапка звичайно не набирається. Проте, іноді її корисно використовувати, щоб допомогти оку читача згрупувати формулу. В такому випадку слід вживати `\cdot`.

```
\begin{displaymath}
v = {\sigma}_1 \cdot {\sigma}_2 \tau_1 \cdot \tau_2
\end{displaymath}
```

$$v = \sigma_1 \cdot \sigma_2 \tau_1 \cdot \tau_2$$

Назви log-подібних функцій набираються прямим шрифтом, а не курсивом, як змінні. Тому \LaTeX містить наступні команди для набору імен найбільш важливих функцій:

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

```
\[\lim_{x \rightarrow 0}
\frac{\sin x}{x}=1\]
```

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

Для функції модуля існує дві команди: `\bmod` для бінарного оператора « $a \bmod b$ » і `\pmod` для таких виразів як « $x \equiv a \pmod{b}$ ».

Дроби набираються командою `\frac{...}{...}`. Іноді перевагу надають дробам з похилою рисою $1/2$, оскільки вона краще виглядає при невеликій кількості «дробового матеріалу».

```
\frac{1}{2}$~години
\begin{displaymath}
\frac{x^2}{k+1}\quad
x^{\frac{2}{k+1}}\quad
x^{1/2}
\end{displaymath}
```

$1\frac{1}{2}$ години

$$\frac{x^2}{k+1} \quad x^{\frac{2}{k+1}} \quad x^{1/2}$$

Для набору біноміальних коефіцієнтів чи аналогічних структур можна користуватися `{... \choose ...}`, або `{... \atop ...}`. Друга команда дає той же результат, але без дужок⁴.

```
\begin{displaymath}
{n \choose k} \qquad \{x \atop y+2\}
\end{displaymath}
```

$$\binom{n}{k} \quad \{x \atop y+2\}$$

Для відображення бінарних відносин може бути корисним розміщення символів один над другим. `\stackrel{!}{=}` розміщує символ, даний у першому аргументі, з використанням шрифту розміру індексу, над другим, що розміщений на звичайній позиції.

```
\begin{displaymath}
\int f_N(x) \stackrel{!}{=} 1
\end{displaymath}
```

$$\int f_N(x) \stackrel{!}{=} 1$$

Оператор інтеграла вводиться командою `\int`, а **оператор суми** — командою `\sum`. Верхні і нижні границі вказуються за допомогою знаків «`^`» та «`_`», так само, як верхні і нижні індекси⁵.

```
\begin{displaymath}
\sum_{i=1}^n \quad \int_0^{\frac{\pi}{2}}
\int_{0}^{\frac{\pi}{2}} \quad \int_0^{\frac{\pi}{2}}
\end{displaymath}
```

$$\sum_{i=1}^n \quad \int_0^{\frac{\pi}{2}}$$

Для **дужок** та інших обмежувачів у `TeX` існує безліч символів (скажімо, `[` `<` `||` `↓`). Круглі і квадратні дужки можна уводити за допомогою відповідних клавіш, фігурні — `\{`, інші обмежувачі — спеціальними командами (наприклад, `\updownarrow`). Список доступних обмежувачів дивіться в таблиці 3.8 на сторінці 56.

```
\begin{displaymath}
\{a,b,c\} \neq \{a,b,c\}
\end{displaymath}
```

$$a, b, c \neq \{a, b, c\}$$

Якщо помістити команду `\left` перед відкриваючим обмежувачем, або команду `\right` — перед закриваючим, то `TeX` автоматично вибере правильний розмір обмежувача. Зважте, що Ви повинні кожен `\left` закривати відповідним `\right`, і що розмір визначається коректно, тільки

⁴Зверніть увагу, що використання цих команд «в старому стилі» явно заборонено пакетом `amsmath`. Вони замінені на `\binom` та `\genfrac`. Остання є надмножиною споріднених конструкцій, наприклад, ви можете отримати аналогічну до `\atop` конструкцію за допомогою `\newcommand{\newatop}[2]{\genfrac{}{}{0pt}{1}{#1}{#2}}`.

⁵`AMS-TeX`, крім того, дає можливість використовувати багаторядкові верхні і нижні індекси.

якщо обидва набрані в одному рядку. Якщо вам не потрібний правий обмежувач, використовуйте невидимий — «`\right.`»!

```
\begin{displaymath}
1 + \left( \frac{1}{1-x^2} \right)^3
\end{displaymath}
```

$$1 + \left(\frac{1}{1-x^2} \right)^3$$

У деяких випадках необхідно вказати коректний розмір математичного обмежувача вручну, це можна зробити, використовуючи `\big`, `\Big`, `\bigg`, і `\Bigg`, як префікси до більшості команд обмежувачів⁶.

```
$$\Big( (x+1)(x-1) \Big)^2$$\quad
$\big(\big(\big(\Big(\quad
$\big\}\Big\}\bigg\}\Bigg\}$\quad
$\big\|\Big\|\bigg\|\Bigg\|\$
```

$$\left((x+1)(x-1) \right)^2$$

$$\left(\left(\left(\left\} \right\} \right\} \right\} \right\} \left\| \left\| \left\| \left\| \right\| \right\| \right\| \right\| \right\|$$

Щоб ввести у формулу **трикрапки**, існує кілька команд. `\ldots` набирає крапки на базовій лінії, `\cdots` — центровані. Крім того, існують команди `\vdots` для вертикальних і `\ddots` для діагональних трикрапок. У розділі 3.5 наведено інший приклад.

```
\begin{displaymath}
x_1, \ldots, x_n \quad \quad
x_1 + \cdots + x_n
\end{displaymath}
```

$$x_1, \dots, x_n \quad x_1 + \cdots + x_n$$

3.4 Математичні пробіли

Якщо обрані Т_ЕX пробіли усередині формул незадовільні, ви можете їх налаштувати з використанням спеціальних команд. Команди для маленьких пробілів: `\`, для $\frac{3}{18}$ quad (U), `\:` для $\frac{4}{18}$ quad (U) і `\;` для $\frac{5}{18}$ quad (U). Екранований символ пробілу `_` дає пробіл середніх розмірів, а `\quad` (L) і `\qquad` (L) дають великі пробіли. Розмір `\quad` приблизно відповідає ширині літери «М» у поточному шрифті. Команда `\!` робить від'ємний пробіл розміром $-\frac{3}{18}$ quad (U).

⁶Ці команди не працюють як слід, якщо використовуються команди зміни розміру шрифту, чи якщо зазначена опція `11pt` чи `12pt`. Для виправлення цієї поведінки скористуйтеся пакетами `exscale` чи `amsmath`.

```

\newcommand{\ud}{\mathrm{d}}
\begin{displaymath}
\int\!\!\!\int\int_{D} g(x,y)
\ , \ \ud x\ , \ \ud y
\end{displaymath}
замість
\begin{displaymath}
\int\int_{D} g(x,y)dxdy
\end{displaymath}

```

$$\iint\!\!\!\int_D g(x,y) dx dy$$

замість

$$\iint_D g(x,y) dx dy$$

Зверніть увагу, «d» у диференціалі звичайно набирається прямим шрифтом.

$\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ пропонує інший спосіб точного настроювання пробілів між кількома знаками інтегралів а саме, команди `\iint`, `\iiint`, `\iiiint` і `\idotsint`. Якщо пакет `amsmath` завантажений, попередній приклад можна набирати так:

```

\newcommand{\ud}{\mathrm{d}}
\begin{displaymath}
\iint_{D} \ , \ \ud x \ , \ \ud y
\end{displaymath}

```

$$\iint_D dx dy$$

Деталі дивіться в електронному документі `textmath.tex` (розповсюджує з $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$) або в розділі 8 «The $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ Companion» [3].

3.5 Вертикально розташований матеріал

Для набору **матриць** користуйтеся оточенням `array`. Воно працює подібно до оточення `tabular`. Для розриву рядка використовується команда `\\`.

```

\begin{displaymath}
\mathbf{X} =
\left( \begin{array}{ccc}
x_{11} & x_{12} & \dots \\
x_{21} & x_{22} & \dots \\
\vdots & \vdots & \ddots
\end{array} \right)
\end{displaymath}

```

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & \dots \\ x_{21} & x_{22} & \dots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

Оточення `array` можна також використовувати для набору виразів, що мають один великий обмежувач, підставивши «.» замість невидимого правого обмежувача:

```

\begin{displaymath}
y = \left\{ \begin{array}{l}
a & \text{\texttrm{якщо } $d>c$} \\
b+x & \text{\texttrm{вранці}} \\
l & \text{\texttrm{решту дня}}
\end{array} \right.
\end{displaymath}

```

$$y = \begin{cases} a & \text{якщо } d > c \\ b + x & \text{вранці} \\ l & \text{решту дня} \end{cases}$$

Як і у середовищі `tabular`, в середовищі `array` можна також креслити лінії, наприклад для відокремлення елементів матриці:

```

\begin{displaymath}
\left( \begin{array}{c|c}
1 & 2 \\ \hline
3 & 4
\end{array} \right)
\end{displaymath}

```

$$\left(\begin{array}{c|c} 1 & 2 \\ \hline 3 & 4 \end{array} \right)$$

Для формул, що займають кілька рядків, або для систем рівнянь, замість `equation` користуйтеся оточеннями `eqnarray` і `eqnarray*`. Кожен рядок `eqnarray` отримує окремий номер рівняння. У `eqnarray*` номери не ставляться.

Оточення `eqnarray` і `eqnarray*` працюють на зразок таблиці з трьох стовпчиків формату `{rc1}`, де середній стовпчик використовується для знака рівності, знака нерівності, чи іншого придатного знака. Команда `\\` розбиває рядки.

```

\begin{eqnarray}
f(x) & = & \cos x & \\
f'(x) & = & -\sin x & \\
\int_0^x f(y)dy & = & \sin x & \\
\end{eqnarray}

```

$$f(x) = \cos x \quad (3.5)$$

$$f'(x) = -\sin x \quad (3.6)$$

$$\int_0^x f(y)dy = \sin x \quad (3.7)$$

Зверніть увагу, що по обидва боки знаків рівності забагато вільного місця. Його можна зменшити установкою `\setlength\arraycolsep{2pt}`, як у наступному прикладі.

Довгі рівняння не будуть автоматично розбиватися на правильні частини. Автор повинен вказати, де їх розбивати і скільки відступати. Найчастіше для цього використовують наступні методи:

```
\setlength\arraycolsep{2pt}
\begin{eqnarray}
\sin x & = & x - \frac{x^3}{3!} + \frac{x^5}{5!} - \\
& & \frac{x^7}{7!} + \dots
\end{eqnarray}
```

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots \quad (3.8)$$

```
\begin{eqnarray}
\lefteqn{ \cos x = 1 } \\
& - \frac{x^2}{2!} + \\
& \frac{x^4}{4!} - \\
& \frac{x^6}{6!} + \dots
\end{eqnarray}
```

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots \quad (3.9)$$

Команда `\nonumber` змушує \LaTeX не генерувати номер для цього рівняння.

Такими методами може бути складно отримати вертикально вирівняні рівняння, що виглядали б правильно; більш потужні альтернативи надає пакет `amsmath`. (Див. середовища `align`, `flalign`, `gather`, `multline` та `split`).

3.6 Фантом

Ми не можемо бачити фантоми, але вони займають певне місце в думках багатьох людей. \LaTeX нічим не відрізняється. Ми можемо скористатися цим для деяких цікавих трюків.

Вирівнюючи текст вертикально за допомогою `^` та `_`, \LaTeX іноді намагається бути занадто запобігливим. Використовуючи команду `\phantom`, можна зарезервувати місце для символів, які не з'являються в кінцевому виводі. Щоб краще зрозуміти, подивитися наступні приклади.

```
\begin{displaymath}
{}^{\phantom{12}}_6\text{C} \\
\quad \quad \quad \text{замість} \\
{}^{12}_6\text{C}
\end{displaymath}
```

$${}^{}_6\text{C} \quad \text{замість} \quad {}^{12}_6\text{C}$$

```
\begin{displaymath}
\Gamma_{ij}^{\phantom{k}} \\
\quad \quad \quad \text{замість} \\
\Gamma_{ij}^k
\end{displaymath}
```

$$\Gamma_{ij}^{} \quad \text{замість} \quad \Gamma_{ij}^k$$

3.7 Розмір математичного шрифту

У математичному режимі \TeX вибирає розмір шрифту в залежності від контексту. Наприклад, індекси набираються меншим шрифтом. Якщо потрібно додати до рівняння звичайний текст, не користуйтеся командою `\textrm`, тому, що механізм переключення розміру працювати не буде, оскільки `\textrm` тимчасово виходить у текстовий режим. Щоб механізм переключення розміру працював як належить, використовуйте команду `\mathrm`. Але майте на увазі, `\mathrm` буде добре працювати тільки з короткими елементами. Пробіли, як і раніше, не діють, і акцентовані символи⁷ не працюють⁸.

```
\begin{equation}
2^{\textrm{-й}} \quad \quad \quad
2^{\mathrm{-й}}
\end{equation}
```

$$2^{-i} \quad 2^{-j} \quad (3.10)$$

Проте, іноді вам може бути потрібно вказати \LaTeX точний розмір шрифту. У математичному режимі розмір встановлюється чотирма командами:

```
\displaystyle (123), \textstyle (123), \scriptstyle (123) і
\scriptscriptstyle (123).
```

Зміна стилів впливає також на спосіб зображення границь.

```
\begin{displaymath}
\mathop{\mathrm{corr}}(X,Y)=
\frac{\displaystyle
\sum_{i=1}^n(x_i-\overline{x})
(y_i-\overline{y})}{
\biggl[
\sum_{i=1}^n(x_i-\overline{x})^2
\sum_{i=1}^n(y_i-\overline{y})^2
\biggr]^{1/2}}
\end{displaymath}
```

$$\mathrm{corr}(X, Y) = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\left[\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2 \right]^{1/2}}$$

Це один із прикладів, коли нам потрібні більші дужки, ніж надані стандартними `\left[` `\right]`.

3.8 Теорема, закони, ...

При написанні математичних документів, вам, імовірно, потрібний спосіб набору «лем», «визначень», «аксіом» і аналогічних структур. \LaTeX

⁷А також кирилиця. (Прим. пер.)

⁸При підключенні $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ (пакет `amssym`) команда `\textrm` починає працювати зі зміною розміру шрифту.

підтримує це за допомогою команд

```
\newtheorem{назва}[лічильник]{текст}[розділ]
```

Аргумент *назва* — це коротке ключове слово, використовуване для ідентифікації «теореми». Аргументом *текст* ви визначаєте дійсну назву «теореми», під якою вона буде друкуватися в документі.

Аргументи в квадратних дужках необов'язкові. Вони використовуються для визначення того, як нумерувати «теорему». Аргументом *лічильник* ви може вказати *назву* попередньо визначеної «теореми». Нова «теорема» буде тоді нумеруватися в тій же послідовності. Аргумент *розділ* дозволяє вам вказати розділ, в межах якого ви хочете нумерувати вашу «теорему».

Після використання в преамбулі документа команди `\newtheorem`, ви можете користуватися наступними командами:

```
\begin{назва}[текст]
Це — моя цікава теорема.
\end{назва}
```

На цьому теорії повинно бути досить. Подальші приклади повинні розвіяти останні сумніви, і остаточно переконати вас, що оточення `\newtheorem` занадто складне, щоб його можна було зрозуміти:

```
% визначення для преамбули
% документа
\newtheorem{law}{Law}
\newtheorem{jury}[law]{Jury}
% в документі
\begin{law} \label{law:box}
Don't hide in the witness box
\end{law}
\begin{jury}[The Twelve]
It could be you! So beware and
see law \ref{law:box}\end{jury}
\begin{law}No, No, No\end{law}
```

Law 1 *Don't hide in the witness box*

Jury 2 (The Twelve) *It could be you! So beware and see law 1*

Law 3 *No, No, No*

Теорема «Jury» використовує той же лічильник, що і теорема «Law». Отже, вона отримає номер у послідовності з іншими теоремами «Law». За допомогою аргумента у квадратних дужках можна вказати назву теореми, чи щось аналогічне.

```

\flushleft
\newtheorem{mur}{Мерфі}[section]
\begin{mur}
Якщо існує два і
більше способів зробити
щось, і один їх цих
способів може призвести
до катастрофи, то хтось
обов'язково це зробить.
\end{mur}

```

Мерфі 3.8.1 *Якщо існує два і більше способів зробити щось, і один їх цих способів може призвести до катастрофи, то хтось обов'язково це зробить.*

Теорема «Мерфі» отримує номер, пов'язаний з номером поточного розділу. Ви можете також використовувати іншу структурну одиницю, наприклад, частину чи підрозділ.

3.9 Жирні символи

У \LaTeX досить непросто одержати жирні символи; це, ймовірно, зроблено навмисно, тому що непрофесіонали занадто часто зловживають ними. Команда зміни шрифту `\mathbf` дає напівжирні символи, але вони звичайні (прямі), тоді як математичні символи звичайно курсивні. Існує команда `\boldmath`, але вона може використовуватися тільки поза математичним режимом. Це ж стосується і символів.

```

\begin{displaymath}
\mu, M \quad \mathbf{\mu} \quad \mathbf{M}
\end{displaymath}

```

$\mu, M \quad \mathbf{\mu} \quad \mathbf{M}$

Зверніть увагу, що кома теж жирна, що може бути небажаним.

Пакет `amsbsy` (включається пакетом `amsmath`), а також `bm` з набору `tools` спрощує це, надаючи команду `\boldsymbol`.

```

\begin{displaymath}
\mu, M \quad \boldsymbol{\mu} \quad \boldsymbol{M}
\end{displaymath}

```

$\mu, M \quad \boldsymbol{\mu} \quad \boldsymbol{M}$

3.10 Список математичних символів

В наступних таблицях ви зможете знайти усі символи, доступні в *математичному режимі*.

Щоб використовувати символи, перелічені в таблицях 3.12–3.16⁹, необхідно завантажений пакет `amssymb` в преамбулі документа, а також мати встановлені математичні шрифти AMS¹⁰.

Табл. 3.1: Акценти в математичному режимі

\hat{a}	<code>\hat{a}</code>	\check{a}	<code>\check{a}</code>	\tilde{a}	<code>\tilde{a}</code>	\acute{a}	<code>\acute{a}</code>
\grave{a}	<code>\grave{a}</code>	\dot{a}	<code>\dot{a}</code>	\ddot{a}	<code>\ddot{a}</code>	\breve{a}	<code>\breve{a}</code>
\bar{a}	<code>\bar{a}</code>	\vec{a}	<code>\vec{a}</code>	\widehat{A}	<code>\widehat{A}</code>	\widetilde{A}	<code>\widetilde{A}</code>

\hat{a}

Табл. 3.2: Маленькі грецькі літери

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

Табл. 3.3: Великі грецькі літери

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

⁹Ці таблиці були отримані з файла `symbols.tex` (автор — David Carlisle) і суттєво модифіковані, як порекомендував Josef Tkađlec.

¹⁰Доступні за адресою CTAN:/tex-archive/fonts/amssymb.

Табл. 3.4: Бінарні відношення

Ви можете відтворити відповідні заперечення додаючи команду `\not` як префікс перед наступними символами.

$<$	<code><</code>	$>$	<code>></code>	$=$	<code>=</code>
\leq	<code>\leq</code> or <code>\le</code>	\geq	<code>\geq</code> or <code>\ge</code>	\equiv	<code>\equiv</code>
\ll	<code>\ll</code>	\gg	<code>\gg</code>	\doteq	<code>\doteq</code>
\prec	<code>\prec</code>	\succ	<code>\succ</code>	\sim	<code>\sim</code>
\preceq	<code>\preceq</code>	\succeq	<code>\succeq</code>	\simeq	<code>\simeq</code>
\subset	<code>\subset</code>	\supset	<code>\supset</code>	\approx	<code>\approx</code>
\subseteq	<code>\subseteq</code>	\supseteq	<code>\supseteq</code>	\cong	<code>\cong</code>
\sqsubset ^a	<code>\sqsubset</code> ^a	\sqsupset ^a	<code>\sqsupset</code> ^a	\Join ^a	<code>\Join</code> ^a
\sqsubseteq	<code>\sqsubseteq</code>	\sqsupseteq	<code>\sqsupseteq</code>	\bowtie	<code>\bowtie</code>
\in	<code>\in</code>	\ni , \owns	<code>\ni</code> , <code>\owns</code>	\propto	<code>\propto</code>
\vdash	<code>\vdash</code>	\dashv	<code>\dashv</code>	\models	<code>\models</code>
$ $	<code>\mid</code>	\parallel	<code>\parallel</code>	\perp	<code>\perp</code>
\smile	<code>\smile</code>	\frown	<code>\frown</code>	\asymp	<code>\asymp</code>
$:$	<code>:</code>	\notin	<code>\notin</code>	\neq or \ne	<code>\neq</code> or <code>\ne</code>

^aДля отримання цього символу скористайтеся пакетом `latexsym`

Табл. 3.5: Бінарні оператори

$+$	<code>+</code>	$-$	<code>-</code>	
\pm	<code>\pm</code>	\mp	<code>\mp</code>	\triangleleft
\cdot	<code>\cdot</code>	\div	<code>\div</code>	\triangleright
\times	<code>\times</code>	\setminus	<code>\setminus</code>	\star
\cup	<code>\cup</code>	\cap	<code>\cap</code>	\ast
\sqcup	<code>\sqcup</code>	\sqcap	<code>\sqcap</code>	\circ
\vee , \lor	<code>\vee</code> , <code>\lor</code>	\wedge , \land	<code>\wedge</code> , <code>\land</code>	\bullet
\oplus	<code>\oplus</code>	\ominus	<code>\ominus</code>	\diamond
\odot	<code>\odot</code>	\oslash	<code>\oslash</code>	\uplus
\otimes	<code>\otimes</code>	\bigcirc	<code>\bigcirc</code>	\amalg
\triangleleft	<code>\triangleleft</code>	\bigtriangledown	<code>\bigtriangledown</code>	\dagger
\lhd ^a	<code>\lhd</code> ^a	\rhd ^a	<code>\rhd</code> ^a	\ddagger
\unlhd ^a	<code>\unlhd</code> ^a	\unrhd ^a	<code>\unrhd</code> ^a	\wr

Табл. 3.6: Великі оператори

\sum	<code>\sum</code>	\bigcup	<code>\bigcup</code>	\bigvee	<code>\bigvee</code>	\bigoplus	<code>\bigoplus</code>
\prod	<code>\prod</code>	\bigcap	<code>\bigcap</code>	\bigwedge	<code>\bigwedge</code>	\bigotimes	<code>\bigotimes</code>
\coprod	<code>\coprod</code>	\bigsqcup	<code>\bigsqcup</code>			\bigodot	<code>\bigodot</code>
\int	<code>\int</code>	\oint	<code>\oint</code>			\biguplus	<code>\biguplus</code>

Табл. 3.7: Стрілки

\leftarrow	<code>\leftarrow</code> or <code>\gets</code>	\longleftarrow	<code>\longleftarrow</code>	\uparrow	<code>\uparrow</code>
\rightarrow	<code>\rightarrow</code> or <code>\to</code>	\longrightarrow	<code>\longrightarrow</code>	\downarrow	<code>\downarrow</code>
\leftrightarrow	<code>\leftrightarrow</code>	\longleftrightarrow	<code>\longleftrightarrow</code>	\updownarrow	<code>\updownarrow</code>
\Leftarrow	<code>\Leftarrow</code>	\Lleftarrow	<code>\Lleftarrow</code>	\Uparrow	<code>\Uparrow</code>
\Rightarrow	<code>\Rightarrow</code>	\Rrightarrow	<code>\Rrightarrow</code>	\Downarrow	<code>\Downarrow</code>
\Leftrightarrow	<code>\Leftrightarrow</code>	\Llongleftrightarrow	<code>\Llongleftrightarrow</code>	\Updownarrow	<code>\Updownarrow</code>
\mapsto	<code>\mapsto</code>	\longmapsto	<code>\longmapsto</code>	\nearrow	<code>\nearrow</code>
\hookrightarrow	<code>\hookrightarrow</code>	\hookrightarrow	<code>\hookrightarrow</code>	\searrow	<code>\searrow</code>
\leftharpoonup	<code>\leftharpoonup</code>	\rightharpoonup	<code>\rightharpoonup</code>	\swarrow	<code>\swarrow</code>
\leftharpoondown	<code>\leftharpoondown</code>	\rightharpoondown	<code>\rightharpoondown</code>	\nwarrow	<code>\nwarrow</code>
\rightleftharpoons	<code>\rightleftharpoons</code>	\iff (bigger spaces)	<code>\iff</code> (bigger spaces)	\leadsto	<code>\leadsto</code> ^a

^aДля отримання цього символу скористайтеся пакетом `latexsym`

Табл. 3.8: Розділювачі

$($	<code>(</code>	$)$	<code>)</code>	\uparrow	<code>\uparrow</code>	\Uparrow	<code>\Uparrow</code>
$[$	<code>[</code> or <code>\lbrack</code>	$]$	<code>]</code> or <code>\rbrack</code>	\downarrow	<code>\downarrow</code>	\Downarrow	<code>\Downarrow</code>
$\{$	<code>\{</code> or <code>\lbrace</code>	$\}$	<code>\}</code> or <code>\rbrace</code>	\updownarrow	<code>\updownarrow</code>	\Updownarrow	<code>\Updownarrow</code>
\langle	<code>\langle</code>	\rangle	<code>\rangle</code>	$ $	<code> </code> or <code>\vert</code>	$\ $	<code>\ </code> or <code>\Vert</code>
\lfloor	<code>\lfloor</code>	\rfloor	<code>\rfloor</code>	\lceil	<code>\lceil</code>	\rceil	<code>\rceil</code>
$/$	<code>/</code>	\backslash	<code>\backslash</code>	.	(dual. empty)		

Табл. 3.9: Великі розділювачі

$\left($	<code>\lgroup</code>	$\right)$	<code>\rgroup</code>	$\left[$	<code>\lmoustache</code>	$\right]$	<code>\rmoustache</code>
$\left $	<code>\arrowvert</code>	$\right $	<code>\Arrowvert</code>	$\left\{$	<code>\bracevert</code>	$\right\}$	

Табл. 3.10: Різні символи

\dots	<code>\dots</code>	\cdots	<code>\cdots</code>	\vdots	<code>\vdots</code>	\ddots	<code>\ddots</code>
\hbar	<code>\hbar</code>	\imath	<code>\imath</code>	\jmath	<code>\jmath</code>	ℓ	<code>\ell</code>
\Re	<code>\Re</code>	\Im	<code>\Im</code>	\aleph	<code>\aleph</code>	\wp	<code>\wp</code>
\forall	<code>\forall</code>	\exists	<code>\exists</code>	\mho	<code>\mho</code> ^a	∂	<code>\partial</code>
$'$	<code>'</code>	$'$	<code>\prime</code>	\emptyset	<code>\emptyset</code>	∞	<code>\infty</code>
∇	<code>\nabla</code>	\triangle	<code>\triangle</code>	\square	<code>\Box</code> ^a	\diamond	<code>\Diamond</code> ^a
\perp	<code>\bot</code>	\top	<code>\top</code>	\angle	<code>\angle</code>	\surd	<code>\surd</code>
\diamond	<code>\diamondsuit</code>	\heartsuit	<code>\heartsuit</code>	\clubsuit	<code>\clubsuit</code>	\spadesuit	<code>\spadesuit</code>
\neg	<code>\neg</code> or <code>\lnot</code>	\flat	<code>\flat</code>	\natural	<code>\natural</code>	\sharp	<code>\sharp</code>

^aДля отримання цього символу скористайтеся пакетом `latexsym`

Табл. 3.11: Не-математичні символи

Ці символи можна використовувати також у текстовому режимі.

\dagger	<code>\dag</code>	\S	<code>\S</code>	\copyright	<code>\copyright</code>
\ddagger	<code>\ddag</code>	\P	<code>\P</code>	\pounds	<code>\pounds</code>

Табл. 3.12: Розділювачі \mathcal{AMS}

\ulcorner	<code>\ulcorner</code>	\urcorner	<code>\urcorner</code>	\llcorner	<code>\llcorner</code>	\lrcorner	<code>\lrcorner</code>
\lvert	<code>\lvert</code>	\rvert	<code>\rvert</code>	\lVert	<code>\lVert</code>	\rVert	<code>\rVert</code>

Табл. 3.13: Грецькі та іврит літери \mathcal{AMS}

\digamma	<code>\digamma</code>	\varkappa	<code>\varkappa</code>	\beth	<code>\beth</code>	\daleth	<code>\daleth</code>	\gimel	<code>\gimel</code>
------------	-----------------------	-------------	------------------------	---------	--------------------	-----------	----------------------	----------	---------------------

Табл. 3.14: Бінарні відношення \mathcal{AMS}

\triangleleft	<code>\lessdot</code>	\triangleright	<code>\gtrdot</code>	\doteq	<code>\doteqdot or \Doteq</code>
\leqslant	<code>\leqslant</code>	\geqslant	<code>\geqslant</code>	\risingdotseq	<code>\risingdotseq</code>
\leqslantless	<code>\leqslantless</code>	\geqslantgtr	<code>\geqslantgtr</code>	\fallingdotseq	<code>\fallingdotseq</code>
\leqq	<code>\leqq</code>	\geqq	<code>\geqq</code>	\eqcirc	<code>\eqcirc</code>
\lll або \llless	<code>\lll або \llless</code>	\ggg або \gggtr	<code>\ggg або \gggtr</code>	\circ	<code>\circ</code>
\lesssim	<code>\lesssim</code>	\gtrsim	<code>\gtrsim</code>	\triangleq	<code>\triangleq</code>
\lessapprox	<code>\lessapprox</code>	\gtrapprox	<code>\gtrapprox</code>	\bumpeq	<code>\bumpeq</code>
\lessgtr	<code>\lessgtr</code>	\gtrless	<code>\gtrless</code>	\Bumpeq	<code>\Bumpeq</code>
\lesseqgtr	<code>\lesseqgtr</code>	\gtreqless	<code>\gtreqless</code>	\thicksim	<code>\thicksim</code>
\lesseqqgtr	<code>\lesseqqgtr</code>	\gtreqqlless	<code>\gtreqqlless</code>	\thickapprox	<code>\thickapprox</code>
\preccurlyeq	<code>\preccurlyeq</code>	\succcurlyeq	<code>\succcurlyeq</code>	\approxeq	<code>\approxeq</code>
\curlyeqprec	<code>\curlyeqprec</code>	\curlyeqsucc	<code>\curlyeqsucc</code>	\backsim	<code>\backsim</code>
\precsim	<code>\precsim</code>	\succsim	<code>\succsim</code>	\backsimeq	<code>\backsimeq</code>
\precapprox	<code>\precapprox</code>	\succapprox	<code>\succapprox</code>	\vDash	<code>\vDash</code>
\subseteq	<code>\subseteq</code>	\supseteq	<code>\supseteq</code>	\Vdash	<code>\Vdash</code>
\Subset	<code>\Subset</code>	\Supset	<code>\Supset</code>	\Vvdash	<code>\Vvdash</code>
\sqsubset	<code>\sqsubset</code>	\sqsupset	<code>\sqsupset</code>	\backepsilon	<code>\backepsilon</code>
\therefore	<code>\therefore</code>	\because	<code>\because</code>	\varpropto	<code>\varpropto</code>
\shortmid	<code>\shortmid</code>	\shortparallel	<code>\shortparallel</code>	\between	<code>\between</code>
\smallsmile	<code>\smallsmile</code>	\smallfrown	<code>\smallfrown</code>	\pitchfork	<code>\pitchfork</code>
\vartriangleleft	<code>\vartriangleleft</code>	\vartriangleright	<code>\vartriangleright</code>	\blacktriangleleft	<code>\blacktriangleleft</code>
\trianglelefteq	<code>\trianglelefteq</code>	\trianglerighteq	<code>\trianglerighteq</code>	\blacktriangleright	<code>\blacktriangleright</code>

Табл. 3.15: Стрілки \mathcal{AMS}

\dashleftarrow	<code>\dashleftarrow</code>	\dashrightarrow	<code>\dashrightarrow</code>	\multimap	<code>\multimap</code>
\leftleftarrows	<code>\leftleftarrows</code>	\rightrightarrows	<code>\rightrightarrows</code>	\upuparrows	<code>\upuparrows</code>
\leftrightarrows	<code>\leftrightarrows</code>	\rightleftarrows	<code>\rightleftarrows</code>	\downdownarrows	<code>\downdownarrows</code>
\Lleftarrow	<code>\Lleftarrow</code>	\Rrightarrow	<code>\Rrightarrow</code>	\upharpoonleft	<code>\upharpoonleft</code>
\twoheadleftarrow	<code>\twoheadleftarrow</code>	\twoheadrightarrow	<code>\twoheadrightarrow</code>	\upharpoonright	<code>\upharpoonright</code>
\leftarrowtail	<code>\leftarrowtail</code>	\rightarrowtail	<code>\rightarrowtail</code>	\downharpoonleft	<code>\downharpoonleft</code>
\leftrightharpoons	<code>\leftrightharpoons</code>	\rightleftharpoons	<code>\rightleftharpoons</code>	\downharpoonright	<code>\downharpoonright</code>
\Lsh	<code>\Lsh</code>	\Rsh	<code>\Rsh</code>	\rightsquigarrow	<code>\rightsquigarrow</code>
\looparrowleft	<code>\looparrowleft</code>	\looparrowright	<code>\looparrowright</code>	\leftrightsquigarrow	<code>\leftrightsquigarrow</code>
\curvearrowleft	<code>\curvearrowleft</code>	\curvearrowright	<code>\curvearrowright</code>		
\circlearrowleft	<code>\circlearrowleft</code>	\circlearrowright	<code>\circlearrowright</code>		

Табл. 3.16: Заперечення бінарних відношень і стрілок \mathcal{AMS}

\nless	\ngtr	\varsubsetneqq
\lneq	\gneq	\varsupsetneqq
\nleq	\ngeq	\nsubseteqq
\nleqslant	\ngeqslant	\nsupseteqq
\lneqq	\gneqq	\nmid
\lvertneqq	\gvertneqq	\nparallel
\nleqq	\ngeqq	\nshortmid
\lnsim	\gnsim	\nshortparallel
\lnapprox	\gnapprox	\nsim
\nprec	\nsucc	\ncong
\npreceq	\nsucceq	\nvdash
\nprecneqq	\nsuccneqq	\nvDash
\nprecnsim	\succnsim	\nVdash
\nprecnapprox	\succnapprox	\nVDash
\subsetneq	\supsetneq	\ntriangleleft
\varsubsetneq	\varsupsetneq	\ntriangleright
\nsubseteq	\nsupseteq	\ntrianglelefteq
\subsetneqq	\supsetneqq	\ntrianglerighteq
\nleftarrow	\rightarrow	\nleftrightarrow
\nLeftarrow	\Rightarrow	\nLeftrightarrow

Табл. 3.17: Бінарні оператори \mathcal{AMS}

\dotplus	\centerdot	\intercal
\ltimes	\rtimes	\divideontimes
\Cup or \doublecup	\Cap or \doublecap	\smallsetminus
\veebar	\barwedge	\doublebarwedge
\boxplus	\boxminus	\circleddash
\boxtimes	\boxdot	\circledcirc
\leftthreetimes	\rightthreetimes	\circledast
\curlyvee	\curlywedge	

Табл. 3.18: \mathcal{AMS} різне

\hbar	<code>\hbar</code>	\hbar	<code>\hslash</code>	\mathbb{k}	<code>\Bbbk</code>
\square	<code>\square</code>	\blacksquare	<code>\blacksquare</code>	\textcircled{S}	<code>\circledS</code>
\triangle	<code>\vartriangle</code>	\blacktriangle	<code>\blacktriangle</code>	\complement	<code>\complement</code>
∇	<code>\triangledown</code>	\blacktriangledown	<code>\blacktriangledown</code>	\Game	<code>\Game</code>
\diamond	<code>\lozenge</code>	\blacklozenge	<code>\blacklozenge</code>	\star	<code>\bigstar</code>
\sphericalangle	<code>\angle</code>	\sphericalangle	<code>\measuredangle</code>	\sphericalangle	<code>\sphericalangle</code>
\diagup	<code>\diagup</code>	\diagdown	<code>\diagdown</code>	\backprime	<code>\backprime</code>
\nexists	<code>\nexists</code>	\Finv	<code>\Finv</code>	\varnothing	<code>\varnothing</code>
\eth	<code>\eth</code>	\mho	<code>\mho</code>		

Табл. 3.19: Математичний алфавіт

Приклад	Команда	Необхідний пакет
\overline{ABCdef}	<code>\mathrm{ABCdef}</code>	
ABCdef	<code>\mathit{ABCdef}</code>	
\mathnormal{ABCdef}	<code>\mathnormal{ABCdef}</code>	
\mathcal{ABC}	<code>\mathcal{ABC}</code>	
\mathscr{ABC}	<code>\mathcal{ABC}</code>	<code>mathrsfs</code>
\mathscr{ABC}	<code>\mathcal{ABC}</code>	<code>eucal</code> з опцією: <code>mathcal</code> або
	<code>\mathscr{ABC}</code>	<code>eucal</code> з опцією: <code>mathscr</code>
\mathfrak{ABCdef}	<code>\mathfrak{ABCdef}</code>	<code>eufrak</code>
\mathbb{ABC}	<code>\mathbb{ABC}</code>	<code>amsfonts</code> або <code>amssymb</code>

Розділ 4

Корисні можливості

При створенні великих документів, ви можете скористатися деякими корисними можливостями \LaTeX , такими, як генерація покажчиків, робота з бібліографією, та іншими. Набагато детальніший опис можливостей та розширень \LaTeX можна знайти в *LaTeX Manual* [1] та *The LaTeX Companion* [3].

4.1 Включення EPS графіки

\LaTeX надає базові засоби для роботи з плаваючими об'єктами, такими, як зображення чи графіки, за допомогою оточень `figure` та `table`.

Існує також ряд можливостей створювати власне графіку за допомогою базового \LaTeX або спеціальних пакетів. На жаль, більшість користувачів вважають ці способи занадто складним, і тому про них не йтиметься далі в цьому *Вступі*. Ви зможете знайти потрібну інформацію з цього приводу, звернувшись до *The LaTeX Companion* [3] або до *LaTeX Manual* [1].

Набагато простіший спосіб отримання графіки в документі — створити її за допомогою спеціалізованих програмних пакетів¹ і включити готові зображення в документ. Пакети \LaTeX пропонують багато способів це зробити. У цьому *Вступі* ми висвітлимо тільки використання графіки у форматі Encapsulated PostScript (EPS), оскільки він досить простий і широко вживаний. Для того, щоб отримати зображення в форматі EPS, вам потрібен доступ до POSTSCRIPT принтера².

Пакет `graphicx` (автор — D. P. Carlisle) надає гарний набір команд для включення графіки. Він є частиною цілого зібрання пакетів, що назива-

¹Таких, як XFig, CorelDraw!, Freehand, Gnuplot, ...

²Інша можливість отримати PostScript — скористатися програмою GHOSTSCRIPT, що доступна за адресою CTAN:/tex-archive/support/ghostscript. Користувачів Windows та OS/2 може зацікавити GSVIEW.

ється «graphics»³.

Працюючи в системі з доступним для виводу POSTSCRIPT принтером та встановленим пакетом `graphics`, скористайтеся наступною інструкцією для включення зображення в документ:

1. Екпортуйте зображення з вашої графічної програми в формат EPS⁴.
2. Завантажте пакет `graphics` у преамбулі вхідного файлу за допомогою команди

```
\usepackage[драйвер]{graphics}
```

де *драйвер* — назва вашого конвертора «dvi в postscript». Найбільш широко використовувана програма називається `dvips`. Назва драйвера потрібна, оскільки не існує стандарту на те, як включати графіку в TEX. Знаючи назву *драйвера*, пакет `graphics` може вибрати правильний метод включення інформації про графіку в файл `.dvi`, так, щоб драйвер зрозумів це і міг правильно включити файл `.eps`.

3. Скористайтеся командою

```
\includegraphics[ключ=значення, ...]{файл}
```

для включення *файлу* до вашого документа. Необов'язковим параметром може бути список *ключів* і їхніх *значень*, які повинні іти через кому. *Ключі* можуть слугувати для зміни ширини, висоти, та орієнтації включеної графіки. Таблиця 4.1 перелічує найбільш важливі ключі.

Табл. 4.1: Назви ключів пакета `graphics`.

<code>width</code>	масштабує графіку до заданої ширини
<code>height</code>	масштабує графіку до заданої висоти
<code>angle</code>	повертає графіку проти годинникової стрілки
<code>scale</code>	масштабує графіку

³CTAN:/tex-archive/macros/latex/required/graphics

⁴Якщо програма не може експортувати в формат EPS, спробуйте встановити драйвер POSTSCRIPT принтера (наприклад, Apple LaserWriter) і надрукуйте його у файл через цей драйвер. Якщо пощастить, ви отримаєте EPS файл. Пам'ятайте, EPS повинен містити не більше одної сторінки. Драйвери деяких принтерів можна явно настроїти для створення файлів у форматі EPS.

Наступний приклад, сподіваюся, зробить це зрозумілішим:

```
\begin{figure}
\begin{center}
\includegraphics[angle=90, width=0.5\textwidth]{test}
\end{center}
\end{figure}
```

Зображення, що міститься у файлі `test.eps`, *спершу* повернуто на 90 градусів, а *потім* масштабоване до ширини, що дорівнює 0.5 ширини стандартного абзацу. Співвідношення сторін зберігається, оскільки висоту не задано. Ширина і висота можуть бути також задані в абсолютних одиницях. Дивіться таблицю 5.5 на сторінці 77 для більш детальної інформації. Якщо ви бажаєте знати про це більше, читайте [8] і [11].

4.2 Бібліографія

За допомогою оточення `thebibliography` ви можете створити бібліографію. Кожен елемент починається з

```
\bibitem[label]{marker} \bibitem[мітка]{маркер}
```

Маркер потім використовується для посилань на книгу чи статтю:

```
\cite{маркер}
```

Якщо не задати параметр *мітка*, елементи бібліографії будуть пронумеровані автоматично. Параметр після команди `\begin{thebibliography}` встановлює максимальну ширину цього номера. В наступному прикладі {99} повідомляє L^AT_EX, що номер жодного з елементів бібліографії не буде ширшим, ніж число «99».

Паргл~\cite{pa} запропонував,
щоб \ldots

```
\begin{thebibliography}{99}
\bibitem{pa} Н.~Partl:
\emph{German \TeX},
TUGboat Volume~9, Issue~1 (1988)

\end{thebibliography}
```

Паргл [1] запропонував, щоб ...

Бібліографія

[1] Н. Partl: *German T_EX*, TUGboat
Volume 9, Issue 1 (1988) 63

Для роботи з бібліографією більших проектів варто скористатися програмою BibTeX, що входить до більшості дистрибутивів TeX⁵. Він дозволяє підтримувати бібліографічну базу даних і вибирати з неї описи документів, посилання на які є у вашому документі. Візуальна презентація бібліографій, згенерованих BibTeX'ом, базується на концепції стилів, що дозволяє створити бібліографію у відповідності з певним стандартом чи дизайном.

4.3 Показчики

Корисною особливістю багатьох книг є їхні предметні показчики. Показчики можуть бути створені автоматично за допомогою L^AT_EX і супутньої програми `makeindex`⁶. У цьому *Вступі* будуть викладені тільки базові команди для створення показчика. Більш детальний опис можна знайти в *The L^AT_EX Companion* [3].

Для того, щоб L^AT_EX міг створювати показчик, необхідно завантажити пакет `makeidx`:

```
\usepackage{makeidx}
```

та активізувати спеціальні команди індексування, помістивши

```
\makeindex
```

у преамбулу документа.

Вміст показчика створюється командами

```
\index{ключ}
```

де *ключ* — елемент показчика. Ви вводите команди створення показчика у ті місця тексту, куди ці елементи повинні вказувати. Таблиця 4.2 пояснює синтаксис аргумента *ключ* на кількох прикладах.

Коли L^AT_EX обробляє вхідний файл, кожна команда `\index` записує відповідний елемент показчика разом з номером поточної сторінки в спеціальний файл. Файл має те ж ім'я, що і вхідний файл L^AT_EX, але інше розширення (`.idx`). Цей `.idx` файл потім може бути оброблений

⁵ Для роботи з україномовною бібліографією слід скористатися 8-бітним BibTeX'ом `bibtex8`, який також входить до більшості дистрибутивів TeX. (Прим. пер.)

⁶ На системах, що не підтримують довгі імена файлів, програма може називатися `makeidx`.

Табл. 4.2: Приклади синтаксису ключів покажчика

Приклад	Елемент	Пояснення
<code>\index {hello}</code>	hello, 1	Звичайний елемент
<code>\index {hello!Peter}</code>	Peter, 3	Підпорядкований елемент
<code>\index {Sam@\textsl {Sam}}</code>	<i>Sam</i> , 2	Форматований ключ
<code>\index {Lin@\textbf {Lin}}</code>	Lin , 7	Те ж
<code>\index {Jenny textbf}</code>	Jenny, 3	Форматований номер сторінки
<code>\index {Joe textit}</code>	Joe, 5	Те ж

програмою `makeindex`.

```
makeindex назвафайлу
```

Програма `makeindex`⁷ генерує відсортований покажчик з тією ж назвою, але цього разу з розширенням `.ind`. Якщо тепер повторно обробити вхідний файл, цей відсортований покажчик включається в документ у те місце, де \LaTeX знаходить команду

```
\printindex
```

Пакет `showidx`, що входить у \LaTeX 2 ϵ , друкує всі елементи покажчика на лівому полі тексту. Це дуже корисно при перевірці тексту і звірці покажчика.

4.4 Вибагливі колонтитули

Пакет `fancyhdr`⁸ (автор — Piet van Oostrum) надає кілька простих команд, що дозволяють вам налаштувати верхні і нижні колонтитули документа. Якщо ви зараз глянете на верх цієї сторінки, то побачите одне з можливих застосувань цього пакета.

При налаштуванні колонтитулів хитрість полягає у тім, щоб включити туди такі речі, як заголовки поточних розділу чи підрозділу. \LaTeX досягає цього в два етапи. У визначеннях колонтитулів можна використовувати команди `\rightmark` і `\leftmark`, щоб відобразити заголовки поточного розділу чи підрозділу, відповідно. Значення цих двох команд міняються при обробці команд `\chapter` чи `\section`.

⁷Оскільки українські літери пишуться в файл у вигляді команд типу `\IeC {\cyrp }`, для створення покажчиків україномовних документів потрібно користуватися програмою `uamakeindex`. (Прим. пер.)

⁸Доступний з CTAN:/tex-archive/macros/latex/contrib/supported/fancyhdr

```

\documentclass{book}
\usepackage{fancyhdr}
\pagestyle{fancy}
% цим ми забезпечуємо вивід заголовків розділів
% і підрозділів маленькими літерами
\renewcommand{\chaptermark}[1]{\markboth{#1}{}}
\renewcommand{\sectionmark}[1]{\markright{\thesection\ #1}}
\fancyhf{} % вилучаємо поточні установки для колонтитулів
\fancyhead[LE,RO]{\bfseries\thepage}
\fancyhead[LO]{\bfseries\rightmark}
\fancyhead[RE]{\bfseries\leftmark}
\renewcommand{\headrulewidth}{0.5pt}
\renewcommand{\footrulewidth}{0pt}
\addtolength{\headheight}{0.5pt} % залишаємо місце для лінійки
\fancypagestyle{plain}{%
  \fancyhead{} % вилучаємо колонтитули на звичайних сторінках
  \renewcommand{\headrulewidth}{0pt} % і лінійку
}

```

Рис. 4.1: Приклад налаштування fancyhdr

Для максимальної гнучкості команди `\chapter` і подібні не перевизначають `\rightmark` і `\leftmark` самі, а викликають іншу команду, що називається `\chaptermark`, `\sectionmark` чи `\subsectionmark`, відповідальну за перевизначення `\rightmark` і `\leftmark`.

Так що, потрібно змінити вигляд назви розділу у верхньому колонтитулі, просто «оновити» команду `\chaptermark`.

Рисунок 4.1 показує можливий варіант настройки пакета `fancyhdr`, що створює майже такі колонтитули, як у цьому буклеті. У будь-якому випадку я рекомендую вам ознайомитися з документацією пакета за адресою, згаданою у виносці.

4.5 Пакет `verbatim`

Раніше в цьому *Вступі* ви ознайомилися з оточенням `verbatim`. У цьому підрозділі ви дізнаєтеся про пакет `verbatim`. Пакет `verbatim` являє собою повторну реалізацію оточення `verbatim`, що обходить деякі з його обмежень. Саме по собі це не так вже й захоплююче, але, крім того, у нього була додана деяка нова функціональність, і саме з цієї причини я

згадую тут цей пакет. Пакет `verbatim` надає команду

```
\verbatiminput{назвафайлу}
```

яка дозволяє вам включати текстовий файл у ваш документ так, наче його вміст знаходиться усередині оточення `verbatim`.

Оскільки пакет `verbatim` входить у комплект «tools», ви знайдете його установленим на більшості систем. Якщо ви хочете довідатися про нього більше, обов'язково прочитайте [9].

4.6 Звантаження і встановлення пакетів L^AT_EX

Більшість дистрибутивів L^AT_EX містить велику кількість стильових пакетів, але ще більше доступно в мережі. Основним місцем, де їх потрібно шукати в Інтернеті, є CTAN (<http://www.ctan.org/>).

Дистрибутиви таких пакетів, як `geometry`, `hyphenat`, та багатьох інших, типово складаються з двох файлів: файлу з розширенням `.ins`, та іншого — з розширенням `.dtx`. Часто там буде ще й `readme.txt` з коротким описом пакета. Звичайно, його слід читати першим.

У будь-якому випадку, коли ви маєте копію файлів пакета на комп'ютері, вам ще потрібно обробити їх таким чином, щоб (а) ваш дистрибутив T_EX знав про новий стильовий пакет, і (б) отримати документацію. Ось як виконати першу частину:

1. Виконайте L^AT_EX з файлом `.ins`. Це згенерує файл `.sty`.
2. Перемістіть файл `.sty` у місце, де ваш дистрибутив зможе його знайти. Як правило, це у підкаталозі `.../localtexmf/tex/latex` (користувачі Windows чи OS/2 можуть змінити напрям похилих рисок).
3. Поновити базу даних імен файлів вашого дистрибутива. Команда залежить від дистрибутива L^AT_EX, яким ви користуєтесь:
`teTeX`, `fpTeX` — `maktexlsr` або `texhash`;
`web2c` — `maktexlsr`;
`MikTeX` — `initexmf -update-fndb` або скористайтесь GUI.

Тепер з файла `.dtx` можна згенерувати документацію:

1. Виконайте L^AT_EX з файлом `.dtx`. Це згенерує файл `.dvi`. Пам'ятайте, можливо доведеться виконувати L^AT_EX кілька разів, щоб отримати правильні перехресні посилання.
2. Перевірте, чи з'явився серед інших файлів файл `.idx`. Якщо ні — можете перейти до кроку 5.

- Щоб отримати покажчик, виконайте наступне:

```
makeindex -s gind.ist назва
```

де *назва* — назва основного файла (без розширення).

- Виконайте L^AT_EX з файлом `.dtx` ще раз.
- Нарешті, створіть файл `.ps` або `.pdf` для отримання більшого задоволення від читання.

Може трапитися, що ви виявите файл `.glo`. Виконайте наступну команду між кроками 4 і 5:

```
makeindex -s gglo.ist -o назва.gls назва.glo
```

Не забудьте виконати L^AT_EX з файлом `.dtx` ще один раз перед переходом до кроку 5.

Розділ 5

Налаштування L^AT_EX

Документи, створені з використанням вивчених дотепер команд, будуть виглядати прийнятно для широкої аудиторії. Не виглядаючи занадто вибагливо, вони відповідають усім правилам гарного тону в наборі, тому їх легко читати, і на них приємно глянути.

Проте, бувають ситуації, у яких L^AT_EX не надає команду чи оточення, які б задовольняли вашим потребам, або вивід, вироблений певною існуючою командою, не відповідає вашим вимогам.

У цьому розділі я спробую дати деякі поради з приводу того, як навчити L^AT_EX новим речам, і як примусити його давати вивід, що відрізняється від стандартного.

5.1 Нові команди, оточення і пакети

Ви певне помітили, що усі команди, представлені в цій книзі, оточені рамкою і включені в покажчик наприкінці книги. Замість того, щоб прямо використовувати необхідні для цього команди L^AT_EX, я створив пакет, у якому визначив для цього нові команди й оточення. Тепер можна просто писати:

```
\begin{lscommand}  
\ci{dum}  
\end{lscommand}
```



\dum

У цьому прикладі я використовую нове оточення `lscommand`, що малює рамку навколо команди, одночасно з новою командою `\ci`, що виводить назву команди і заносить відповідний елемент у покажчик. Ви можете у цьому переконатися, відшукавши команду `\dum` у покажчику наприкінці книги, де ви знайдете елемент для `\dum`, що вказує на кожну сторінку, де я згадую команду `\dum`.

Якщо я колись вирішу, що мені більше не подобаються команди, набрані в рамці, я просто зміню визначення оточення `\lscmd`. Це набагато простіше, ніж проглядати весь документ, вишукуючи усі місця, де я використав загальні команди \LaTeX для зображення рамки навколо слів.

5.1.1 Нові команди

Щоб створити ваші власні команди, скористайтеся командою

```
\newcommand{назва}[число]{визначення}
```

Звичайно ця команда приймає два аргументи: *назву* команди, яку ви створюєте, і *визначення* команди. Необов'язковий аргумент *число* у квадратних дужках визначає кількість аргументів нової команди (може бути не більше 9). Якщо цей аргумент відсутній, вважається 0, тобто нова команда не має аргументів.

Наступні два приклади повинні допомогти вам отримати уявлення про те, як працює `\newcommand`. Перший приклад визначає нову команду, що називається `\tnss`, що є скороченням від «The Not So Short Introduction to $\LaTeX 2_\epsilon$ ». Така команда знадобиться, якщо вам доведеться багато разів писати назву цієї книги.

```
\newcommand{\tnss}{The not
  so Short Introduction to
  \LaTeXe}
This is ‘‘\tnss’’ \ldots{ }
‘‘\tnss’’
```

```
This is “The not so Short Introduction to
 $\LaTeX 2_\epsilon$ ” ... “The not so Short Introducti-
on to  $\LaTeX 2_\epsilon$ ”
```

Наступний приклад ілюструє, як визначити команду з одним аргументом. Мітка `#1` замінюється на заданий аргумент. Якщо ви хочете використовувати більш одного аргумента, використовуйте `#2`, і так далі.

```
\newcommand{\txsit}[1]
{This is the \emph{#1} Short
  Introduction to \LaTeXe}
% в тілі документа:
\begin{itemize}
\item \txsit{not so}
\item \txsit{very}
\end{itemize}
```

- This is the *not so* Short Introduction to $\LaTeX 2_\epsilon$
- This is the *very* Short Introduction to $\LaTeX 2_\epsilon$

\LaTeX не дозволить вам створити нову команду, яка б замінила вже існуючу. Але є спеціальна команда для випадку, коли ви явно хочете це зробити: `\renewcommand`. Вона має той же синтаксис, що і команда `\newcommand`.

В певних випадках може знадобитися команда `\providecommand`. Вона працює так само, як `\newcommand`, але, якщо команда уже визначена, $\LaTeX 2_\epsilon$ її просто проігнорує.

Потрібно звернути увагу на кілька моментів стосовно пробілів після команд. Деталі дивіться на сторінці 6.

5.1.2 Нові оточення

Аналогічно до команди `\newcommand`, існує команда для створення ваших власних оточень. Команда `\newenvironment` має наступний синтаксис:

```
\newenvironment{назва}[номер]{перед}{після}
```

Як і команду `\newcommand`, `\newenvironment` можна використовувати з необов'язковим аргументом або без нього. Матеріал, вказаний в аргументі *перед*, обробляється до обробки тексту усередині оточення. Матеріал, вказаний в аргументі *після*, обробляється, коли зустрічається команда `\end{назва}`.

Використання команди `\newenvironment` проілюстровано наступним прикладом.

```
\newenvironment{king}
{\rule{1ex}{1ex}%
 \hspace{\stretch{1}}}
{\hspace{\stretch{1}}%
 \rule{1ex}{1ex}}
```

■ Мої покірні піддані... ■

```
\begin{king}
Мої покірні піддані\ldots
\end{king}
```

Аргумент *номер* використовують так же, як і в команді `\newcommand`. \LaTeX контролює, щоб ви не визначили вже існуюче оточення. Якщо потрібно явно перевизначити існуюче оточення, скористайтеся командою `\renewenvironment`, що має такий же синтаксис, як і `\newenvironment`.

Команди, використані у цьому прикладі, будуть пояснені пізніше: опис команди `\rule` дивіться на сторінці 84, команда `\stretch` описана на сторінці 77, інформацію про команду `\hspace` можна знайти на сторінці 77.

5.1.3 Ваш власний пакет

Коли ви визначаєте безліч нових оточень і команд, преамбули ваших документів стають дуже довгими. У цій ситуації хорошою ідеєю може бути створення пакету \LaTeX , що містить визначення усіх ваших команд

і оточень. Потім можна командою `\usepackage` використовувати пакет у ваших документах.

```
% Demo Package by Tobias Oetiker
\ProvidesPackage{demopack}
\newcommand{\tnss}{Не надто короткий вступ до \LaTeXe}
\newcommand{\txsit}[1]{\emph{#1}
                        короткий вступ до \LaTeXe}
\newenvironment{king}{\begin{quote}}{\end{quote}}
```

Рис. 5.1: Приклад пакета

Створення пакета в основному складається з переносу вмісту вашої преамбули в окремий файл з ім'ям, що закінчується на `.sty`. Є тільки одна спеціальна команда, яку ви повинні використовувати

`\ProvidesPackage{назва пакета}`

на самому початку файлу з вашим пакетом. `\ProvidesPackage` указує ЛАТ_ΕX назву пакета, що дозволяє йому видавати зрозуміле повідомлення про помилку, коли ви намагаєтеся підключити пакет двічі. Рис. 5.1 показує маленький приклад пакета, що містить визначені в наведених вище прикладах команди.

5.2 Шрифти та розміри

5.2.1 Команди зміни шрифту

ЛАТ_ΕX вибирає відповідний шрифт і розмір шрифту, ґрунтуючись на логічній структурі документа (розділи, виноски, ...). Іноді буває потрібно змінити шрифт вручну. Для цього ви можете скористатися командами, переліченими в таблицях 5.1 та 5.2. Справжній розмір кожного шрифту — це питання дизайну, і залежить від класу і опцій документа. Таблиця 5.3 показує абсолютні розміри, що відповідають цим командам у стандартних класах документів.

```
{\small The small and
\textbf{bold} Romans ruled}
{\Large all of great big
\textit{Italy}.}
```

The small and **bold** Romans ruled all of great big *Italy*.

Важливою особливістю ЛАТ_ΕX 2_ε є те, що атрибути шрифту незалежні. Це значить, що ви можете давати команди зміни розміру чи навіть

шрифту, зберігаючи при цьому атрибути насиченості чи нахилу, встановлені раніше.

У *математичному режимі* можна використовувати *команди* зміни шрифту, щоб тимчасово вийти з *математичного режиму* і увести нормальний текст. Якщо потрібно переключитися на інший шрифт для набору математики, для цього існує спеціальний набір команд. Дивіться таблицю 5.4.

У зв'язку з командами зміни розміру шрифту помітну роль грають фігурні дужки. Вони використовуються для побудови *груп*. Групи обмежують область дії більшості команд L^AT_EX.

Йому подобаються
`{\LARGE великі і`
`{\small маленькі} літери}`.

Йому подобаються ВЕЛИКІ і маленькі
 ЛІТЕРИ.

Команди, що впливають на розмір шрифту, впливають також на відстань між рядками, але тільки якщо відповідний абзац закінчується усередині області дії команди. Тому закриваюча фігурна дужка `}` не повинна стояти занадто рано. Зверніть увагу на положення команди `\par` у

Табл. 5.1: Шрифти

<code>\textrm{...}</code>	з зарубками	<code>\textsf{...}</code>	без зарубок
<code>\texttt{...}</code>	машинопис		
<code>\textmd{...}</code>	нормальний	<code>\textbf{...}</code>	жирний
<code>\textup{...}</code>	прямий	<code>\textit{...}</code>	<i>курсив</i>
<code>\textsl{...}</code>	<i>похилий</i>	<code>\textsc{...}</code>	КАПІТЕЛЬ
<code>\emph{...}</code>	<i>виділений</i>	<code>\textnormal{...}</code>	звичайний

Табл. 5.2: Розміри шрифтів

<code>\tiny</code>	малесенький	<code>\Large</code>	більший
<code>\scriptsize</code>	дуже маленький	<code>\LARGE</code>	дуже великий
<code>\footnotesize</code>	досить маленький	<code>\huge</code>	ВЕЛИЧЕЗНИЙ
<code>\small</code>	маленький	<code>\Huge</code>	ГІГАНТСЬКИЙ
<code>\normalsize</code>	нормальний		
<code>\large</code>	великий		

Табл. 5.3: Абсолютні розміри шрифтів у стандартних класах

розмір	10pt (за замовч.)	опція 11pt	опція 12pt
<code>\tiny</code>	5pt	6pt	6pt
<code>\scriptsize</code>	7pt	8pt	8pt
<code>\footnotesize</code>	8pt	9pt	10pt
<code>\small</code>	9pt	10pt	11pt
<code>\normalsize</code>	10pt	11pt	12pt
<code>\large</code>	12pt	12pt	14pt
<code>\Large</code>	14pt	14pt	17pt
<code>\LARGE</code>	17pt	17pt	20pt
<code>\huge</code>	20pt	20pt	25pt
<code>\Huge</code>	25pt	25pt	25pt

Табл. 5.4: Математичні шрифти

Команда	Приклад	Вивід
<code>\mathcal{...}</code>	<code>\$\$\mathcal{B}=c\$</code>	$\mathcal{B} = c$
<code>\mathrm{...}</code>	<code>\$\$\mathrm{K}_2\$</code>	K_2
<code>\mathbf{...}</code>	<code>\$\$\sum x=\mathbf{v}\$</code>	$\sum x = \mathbf{v}$
<code>\mathsf{...}</code>	<code>\$\$\mathsf{G\times R}\$</code>	$G \times R$
<code>\mathtt{...}</code>	<code>\$\$\mathtt{L}(b,c)\$</code>	$L(b, c)$
<code>\mathnormal{...}</code>	<code>\$\$\mathnormal{R_{19}}\neq R_{19}\$</code>	$R_{19} \neq R_{19}$
<code>\mathit{...}</code>	<code>\$\$\mathit{ffi}\neq ffi\$</code>	$\mathit{ffi} \neq ffi$

наступних двох прикладах¹:

```
{\Large Не читайте цього! Це
неправда. Вірте мені!}\par}
```

Не читайте цього! Це неправда.
Вірте мені!

```
{\Large Це також неправда.
Але пам'ятайте, я брехун.}\par}
```

Це також неправда. Але пам'ятайте,
я брехун.

Якщо потрібно застосувати команду зміни розміру до цілого абзацу тексту чи навіть більше, варто використовувати синтаксис оточення для команди зміни шрифту.

```
\begin{Large}
Це неправда. Але,
знову ж таки,
що в наші дні\ldots
\end{Large}
```

Це неправда. Але, знову ж та-
ки, що в наші дні. . .

Це дозволить вам уникнути підрахунку безлічі фігурних дужок.

5.2.2 Увага, небезпека

Як відзначено на початку цього розділу, небезпечно загроможувати ваші документи явними командами, на зразок тільки що описаних, тому що це працює всупереч основній ідеї \LaTeX , а саме, відокремлення логічної і візуальної розмітки вашого документа. Це значить, що якщо ви користуєтеся тими самими командами зміни шрифту в різних місцях для набору певного виду інформації, слід використовувати `\newcommand` і визначити команду, що буде «логічною обгорткою» для команди зміни шрифту.

```
\newcommand{\oops}[1]{\textbf{#1}}
Не \oops{заходьте} до цієї кімнати,
в ній знаходиться \oops{машина}
невідомого походження
і призначення.
```

Не **заходьте** до цієї кімнати, в ній знаходиться **машина** невідомого походження і призначення.

Цей підхід має ту перевагу, що ви пізніше можете вирішити, що хочете використовувати інше візуальне представлення небезпеки, ніж `\textbf`, без необхідності пробиратися через весь документ, знаходити усі випадки `\textbf` і визначати у кожному з них, використано `\textbf`, щоб вказати на небезпеку, чи з якогось іншого приводу.

¹Команда `\par` еквівалентна порожньому рядку

5.2.3 Порада

Для завершення нашої подорожі у світ шрифтів та їхніх розмірів, дозвольте дати вам одну пораду:

Пам'ятайте! Чим **БІЛЬШЕ** шрифтів **ВИ** використовуєте у вашому документі, *тим* простіше його читати і тим *гарнішим* він буде.

5.3 Інтервали

5.3.1 Інтервали між рядками

Якщо вам потрібні більші інтервали між рядками, їхнє значення можна змінити, помістивши команду

```
\linespread{коефіцієнт}
```

у преамбулу документа. Для друку «у півтора інтервали» значення *коефіцієнта* повинно бути 1.3, а для друку «у два інтервали» — 1.6. За замовчуванням цей коефіцієнт дорівнює 1.

5.3.2 Форматування абзаців

Два параметри L^AT_EX впливають на формат абзаців. Помістивши визначення типу

```
\setlength{\parindent}{0pt}
\setlength{\parskip}{1ex plus 0.5ex minus 0.2ex}
```

в преамбулу, ви зміните зовнішній вигляд абзаців. Ці два рядки збільшують відстань між абзацами і встановлюють абзацний відступ рівним нулю.

Елементи `plus` та `minus` повідомляють, що T_EX може зменшити або збільшити відстань між абзацами на задану величину у випадку, коли не вдається вмістити абзаци на сторінці.

У континентальній Європі абзаци часто відокремлюють вертикальним відступом і не роблять абзацного відступу. Але майте на увазі, це також впливає на зміст. Відстані між його елементами також збільшуються. Щоб запобігти цьому, ці команди можна перенести з преамбули документа куди-небудь після `\tableofcontents`, або не використовувати їх зовсім, оскільки в професійно виготовлених книгах для відокремлення абзаців використовується абзацний відступ, а не вертикальний відступ.

Табл. 5.5: Одиниці виміру TeX

mm	міліметр $\approx 1/25$ дюйма	□
cm	сантиметр = 10 мм	□
in	дюйм = 25.4 мм	□
pt	пункт $\approx 1/72$ дюйма $\approx \frac{1}{3}$ мм	□
em	приблизно ширина літери «М» поточного шрифту	□
ex	приблизно висота літери «x» поточного шрифту	□

Якщо потрібно зробити абзацний відступ у абзаці, який його не має, вставте команду

```
\indent
```

на початку абзацу². Зрозуміло, що вона матиме ефект тільки якщо `\parindent` не встановлений рівним нулю.

Для створення абзацу без відступу, використовуйте

```
\noindent
```

на початку абзацу. Це може бути зручно, коли документ починається текстом, а не заголовком розділу.

5.3.3 Горизонтальні інтервали

LaTeX автоматично визначає розміри пробілів між словами і реченнями. Щоб додати горизонтальний інтервал, користуйтеся

```
\hspace{довжина}
```

Якщо такий інтервал повинен бути витриманий, навіть якщо він потрапить на початок чи кінець рядка, використовуйте `\hspace*` замість `\hspace`. У найпростішому випадку *довжина* — це просто число та одиниця виміру. Найважливіші одиниці перелічені в таблиці 5.5.

Цей `\hspace{1.5cm}` пробіл
1,5 см.

Цей пробіл 1,5 см.

²Для того, щоб перший абзац розділу був з відступом, користуйтеся пакетом `indentfirst` з комплекту «tools».

Команда

`\stretch{n}`

де n — «фактор пружності», генерує спеціальний «гумовий» пробіл. Він розтягується, заповнюючи все місце, що залишилося в рядку. Якщо в одному рядку трапляться дві команди `\hspace{\stretch{n}}`, вони розтягуються пропорційно своїм факторам пружності.

`x\hspace{\stretch{1}}`
`x\hspace{\stretch{3}}x`

x	x	x
---	---	---

5.3.4 Вертикальні інтервали

Інтервали між абзацами, розділами, підрозділами, ... \LaTeX визначає автоматично. При необхідності, додатковий пробіл *між двома абзацами* можна додати командою:

`\vspace{довжина}`

Звичайно ця команда вставляється між двома порожніми рядками. Якщо цей простір повинен бути збережений навіть угорі чи внизу сторінки, використовуйте варіант команди з зірочкою: `\vspace*` замість `\vspace`.

Команду `\stretch` разом з `\pagebreak` можна застосовувати для набору тексту на останньому рядку сторінки чи для вертикального центрування тексту на сторінці.

Деякий текст\ldots

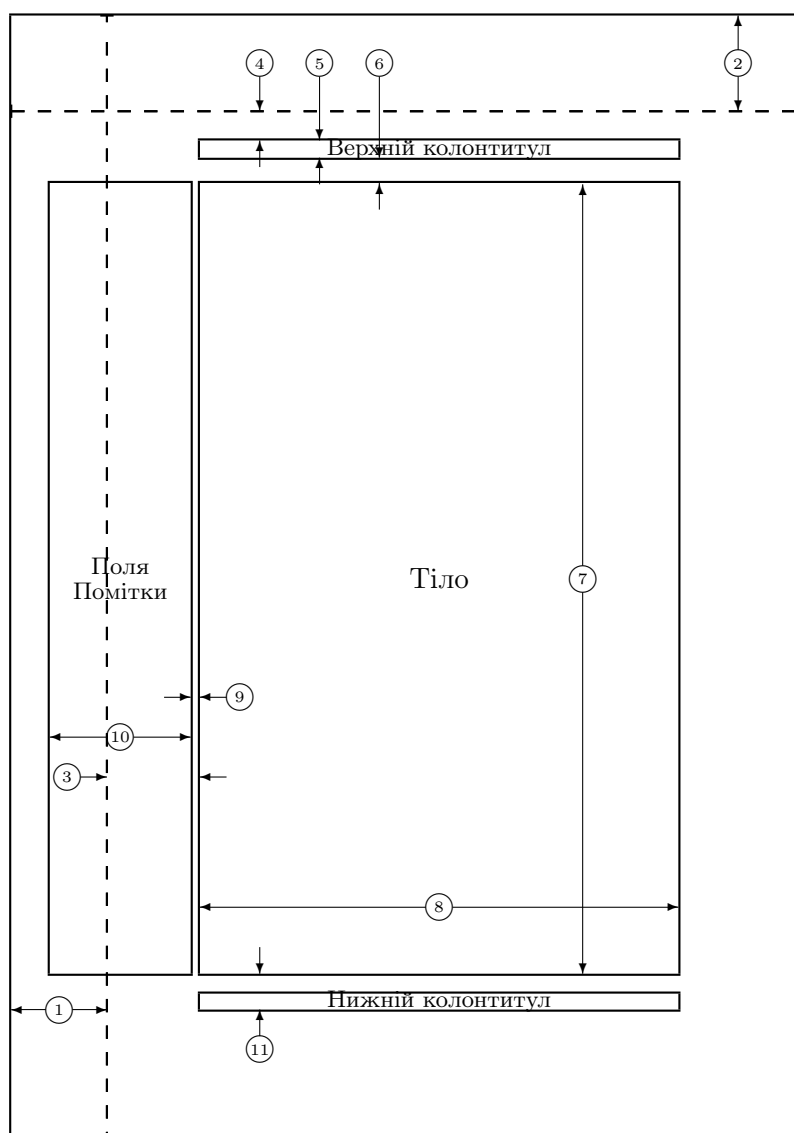
`\vspace{\stretch{1}}`

А іде на останньому рядку сторінки.\pagebreak

Додатковий пробіл між двома рядками *одного* абзацу чи всередині таблиці вказується командою

`\[довжина]`

Командами `\bigskip` та `\smallskip` можна зробити вертикальні відступи задалегідь визначених розмірів без потреби турбуватися про числа.



1	Один дюйм + <code>\hoffset</code>	2	Один дюйм + <code>\voffset</code>
3	<code>\oddsidemargin = 22pt</code> or <code>\evensidemargin</code>	4	<code>\topmargin = 22pt</code>
5	<code>\headheight = 13pt</code>	6	<code>\headsep = 19pt</code>
7	<code>\textheight = 595pt</code>	8	<code>\textwidth = 360pt</code>
9	<code>\marginparsep = 7pt</code>	10	<code>\marginparwidth = 106pt</code> <code>\marginparpush = 5pt</code> (не показано)
11	<code>\footskip = 27pt</code> <code>\hoffset = 0pt</code> <code>\paperwidth = 597pt</code>		<code>\voffset = 0pt</code> <code>\paperheight = 845pt</code>

Рис. 5.2: Параметри макета сторінки

5.4 Макетування сторінки

ЛАТ_ΕX 2_ε дозволяє вказати розмір аркуша у команді `\documentclass`. Потім він автоматично встановлює розміри полів. Але іноді автоматично встановлені значення можуть вас не влаштовувати. Безумовно, їх можна змінити. Рис. 5.2 показує всі параметри, які можна змінити. Ілюстрація була згенерована пакетом `layout` з набору «tools»³.

ЗАЧЕКАЙТЕ! ...перш, ніж ринутися «Давайте зробимо цю завузьку сторінку трошки ширшою», поміркуйте кілька секунд. Як і багато інших речей у ЛАТ_ΕX, існує вагома причина, чому макет сторінки такий, яким він є.

Безумовно, якщо порівняти із сторінкою документа, створеного свіжестановленим MS Word, сторінки документів ЛАТ_ΕX виглядають жалюиво вузькими. Однак, гляньте на вашу улюблену книгу⁴ і порахуйте кількість літер на одному рядку. Ви побачите, що на кожному рядку не більше 66 символів. Тепер зробіть те ж саме зі сторінкою ЛАТ_ΕX. Ви побачите, що тут також близько 66 літер у рядку. Досвід показує, що читання ускладнюється при більшій кількості літер в одному рядку, тому, що очам стає важче переходити від кінця одного рядка до початку наступного. Саме тому газети часто набираються у декілька шпальт.

Тому, збільшуючи ширину вашого тексту, майте на увазі, що ви ускладнюєте життя його читачам. Однак, досить попереджень, я обіцяв розповісти про те, як це зробити. . .

ЛАТ_ΕX надає дві команди для зміни цих параметрів. Їх звичайно використовують у преамбулі документа.

Перша команда призначає фіксоване значення будь-якому параметру

```
\setlength{параметр}{довжина}
```

Інша команда додає довжину до будь-якого параметра

```
\addtolength{параметр}{довжина}
```

Ця друга команда навіть більш корисна, ніж `\setlength`, тому що вона дозволяє робити налаштування відносно існуючих установок. Щоб додати сантиметр до загальної ширини тексту, у преамбулу потрібно помістити наступні команди:

```
\addtolength{\hoffset}{-0.5cm}
\addtolength{\textwidth}{1cm}
```

³CTAN: /tex-archive/macros/latex/required/tools

⁴Я маю на увазі справжню друковану книгу, видану шанованим видавництвом.

У цьому контексті вас може зацікавити пакет `calc`, що дозволяє використовувати арифметичні операції в аргументі `\setlength` і в інших місцях, де ви можете вводити чисельні значення як аргументи функцій.

5.5 Ще про довжини

Скрізь, де це можливо, я уникаю використання абсолютних величин в документах. Я радше спробую прив'язати розміри до ширини чи висоти інших елементів сторінки. Для ширини ілюстрації це може бути `\textwidth`, щоб вона заповнювала сторінку цілком.

Наступні три команди дозволяють визначити ширину, висоту і глибину текстового рядка.

```
\settoheight{команда}{текст}
\settodepth{команда}{текст}
\settowidth{команда}{текст}
```

Наступний приклад показує можливе застосування цих команд.

```
\flushleft
\newenvironment{vardesc}[1]{%
  \settowidth{\parindent}{#1:\ }
  \makebox[0pt][r]{#1:\ }}{}

\begin{displaymath}
a^2+b^2=c^2
\end{displaymath}

\begin{vardesc}{Де}$a$,
$b$ "--- катети прямокутного
трикутника.

$c$ "--- самотня гіпотенуза
цього трикутника.

$d$ "--- взагалі тут не
з'являється. Чи це не дивно?
\end{vardesc}
```

$$a^2 + b^2 = c^2$$

Де: a , b — катети прямокутного трикутника.

c — самотня гіпотенуза цього трикутника.

d — взагалі тут не з'являється. Чи це не дивно?

5.6 Блоки

Л^AT_EX складає сторінки з блоків. Спочатку кожна літера є маленьким блоком, що приклеюється до інших літер, формуючи слово. Слова склеюються з іншими словами спеціальним еластичним клеєм, який дозволяє

серії слів розтягуватися чи стискатися, щоб точно заповнити рядок на сторінці.

Я визнаю, що це досить спрощена версія того, що відбувається насправді, але ідея в тім, що Т_EX завжди працює з блоками і клеєм. Не тільки літера може бути блоком. Ви можете помістити в блок практично все, що завгодно, включно з іншими блоками. Кожен блок потім обробляється ЛАТ_EX так, ніби це окрема літера.

В попередніх розділах ви уже зустрічали деякі блоки, хоча про це і не говорилося. Наприклад, і команда `\includegraphics`, і оточення `tabular`, обидва створюють блок. Це значить, що ви легко можете розмістити поруч дві таблиці чи ілюстрації. Тільки переконайтеся, що їхня загальна ширина не перевищує `\textwidth`.

Ви також можете упакувати будь-який абзац у блок командою

```
\parbox[поз]{ширина}{текст}
```

чи оточенням

```
\begin{minipage}[поз]{ширина} текст \end{minipage}
```

Параметр *поз* може приймати одну з літер *c*, *t* або *b*, контролюючи вертикальне вирівнювання блоку відносно базової лінії оточуючого тексту. *Ширина* приймає аргументом довжину, що визначає ширину блоку. Основна відмінність між `minipage` і `\parbox` у тім, що всередині `\parbox` можна використовувати не усі команди й оточення, в той час як практично все можливо всередині `minipage`.

У той час, як `\parbox` упакує цілий абзац, розбиваючи рядки і таке інше, існує клас блокових команд, що працюють тільки з горизонтально розташованим матеріалом. Одну з них ми вже знаємо. Вона називається `\mbox` і просто упакує послідовність блоків, що можна використовувати для запобігання розриву кількох слів. Оскільки можна розміщувати одні блоки всередині інших, це пакування горизонтальних блоків дає вам надзвичайну гнучкість. У команді

```
\makebox[ширина][поз]{текст}
```

ширина визначає ширину результуючого блока так, як її буде видно ззовні⁵. Крім виразів довжини, тут можна використовувати `\width`, `\height`, `\depth` і `\totalheight`. Вони встановлюються рівними значенням, отриманим виміром параметрів набраного *тексту*. Параметр *поз* приймає

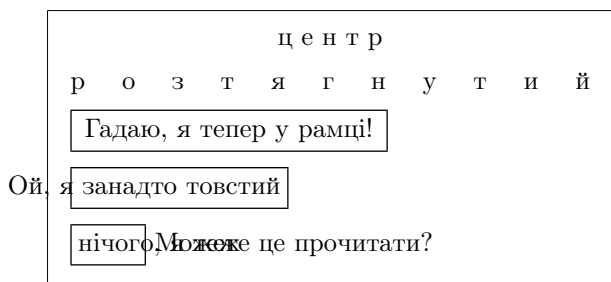
⁵Це означає, що вона може бути меншою, ніж матеріал усередині блоку. Ви можете навіть установити її в `0pt`, так що текст усередині блоку буде набиратися не впливаючи на навколишні блоки.

одне із значень: **c** — центрувати, **l** — вліво, **r** — вправо **s** — рівномірно заповнити блок текстом.

Команда `\framebox` працює точно так само, як `\makebox`, але креслить рамку навколо тексту.

Наступний приклад показує деякі можливості використання команд `\makebox` і `\framebox`.

```
\makebox[\textwidth]{%
  ц е н т р}\par
\makebox[\textwidth][s]{%
  р о з т я г н у т и й}\par
\framebox[1.1\width]{Гадаю,
  я тепер у рамці!}\par
\framebox[0.8\width][r]{Ой,
  я занадто товстий}\par
\framebox[1cm][l]{нічого,
  я теж}
Можете це прочитати?
```

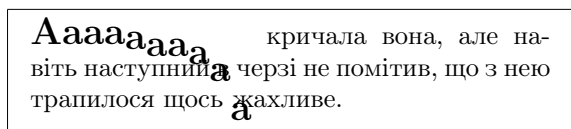


Тепер, коли ми керуємо горизонталлю, очевидний наступний крок — вертикаль⁶. Для L^AT_EX це не проблема. Команда

```
\raisebox{підняття}[глибина][висота]{текст}
```

дозволяє вам встановити вертикальні характеристики блоку. У перших трьох параметрах можна використовувати `\width`, `\height`, `\depth` і `\totalwidth`, щоб мати можливість посилатися на розміри блоку всередині аргументу *текст*.

```
\raisebox{0pt}[0pt][0pt]{\Large%
\textbf{Aaaa}\raisebox{-0.3ex}{a}%
\raisebox{-0.7ex}{aa}%
\raisebox{-1.2ex}{a}%
\raisebox{-2.2ex}{a}%
\raisebox{-4.5ex}{a}}
кричала вона, але навіть наступний
в черзі не помітив, що з нею
трапилося щось жахливе.
```



⁶Повний контроль можна отримати тільки одночасно контролюючи горизонталь і вертикаль ...

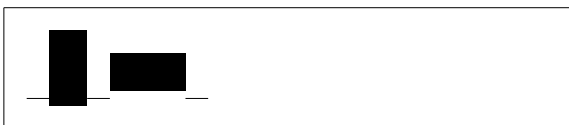
5.7 Лінійки і розпірки

Кілька сторінок тому ви могли помітити команду

```
\rule[підняття]{ширина}{висота}
```

При звичайному використанні вона генерує простий чорний прямокутник.

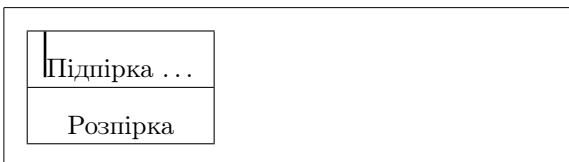
```
\rule{3mm}{.1pt}%
\rule[-1mm]{5mm}{1cm}%
\rule{3mm}{.1pt}%
\rule[1mm]{1cm}{5mm}%
\rule{3mm}{.1pt}
```



Це корисно для малювання вертикальних і горизонтальних ліній. Наприклад, лінія на титульному аркуші намальована командою `\rule`.

Спеціальним випадком є лінійка без ширини, але певної висоти. У професійній типографії її називають *розпіркою*. Її використовують, щоб забезпечити визначену мінімальну висоту елемента сторінки. Ви можете використовувати її в оточенні `tabular`, щоб гарантувати рядку визначену мінімальну висоту.

```
\begin{tabular}{|c|}
\hline
\rule{1pt}{4ex}Підпірка \ldots\
\hline
\rule{0pt}{4ex}Розпірка\
\hline
\end{tabular}
```



Бібліографія

- [1] Leslie Lamport. *L^AT_EX: A Document Preparation System*. Addison-Wesley, Reading, Massachusetts, second edition, 1994, ISBN 0-201-52983-1. [v](#), [1](#), [61](#)
- [2] Donald E. Knuth. *The T_EXbook*, Volume A of *Computers and Typesetting*, Addison-Wesley, Reading, Massachusetts, second edition, 1984, ISBN 0-201-13448-9. [v](#), [1](#)
- [3] Michel Goossens, Frank Mittelbach and Alexander Samarin. *The L^AT_EX Companion*. Addison-Wesley, Reading, Massachusetts, 1994, ISBN 0-201-54199-8. [v](#), [4](#), [12](#), [13](#), [48](#), [61](#), [64](#)
- [4] Кожна установка L^AT_EX повинна містити так званий *L^AT_EX Local Guide*, в якому описані речі, особливі для даної системи. Цей документ повинен міститися у файлі з назвою `local.tex`. Нажаль, деякі лінії системні адміністратори не надають такого документа. У цьому випадку зверніться за порадою до місцевого L^AT_EX 2_ε гуру. [vi](#), [12](#), [24](#)
- [5] L^AT_EX3 Project Team. *L^AT_EX 2_ε for authors*. Входить в дистрибутив L^AT_EX 2_ε як файл `usrguide.tex`.
- [6] L^AT_EX3 Project Team. *L^AT_EX 2_ε for Class and Package writers*. Входить в дистрибутив L^AT_EX 2_ε як `clsguide.tex`.
- [7] L^AT_EX3 Project Team. *L^AT_EX 2_ε Font selection*. Входить в дистрибутив L^AT_EX 2_ε як `fntguide.tex`.
- [8] D. P. Carlisle. *Packages in the 'graphics' bundle*. Входить в набір пакетів «graphics» як файл `grfguide.tex`, доступний із того ж джерела, що і дистрибутив L^AT_EX. [63](#)
- [9] Rainer Schöpf, Bernd Raichle, Chris Rowley. *A New Implementation of L^AT_EX's verbatim Environments*. Входить в набір пакетів «tools» як файл `verbatim.dtx`, доступний із того ж джерела, що і дистрибутив L^AT_EX. [67](#)

-
- [10] Graham Williams. *The TeX Catalogue* — детальний список багатьох пакетів, що мають відношення до $\text{T}_{\text{E}}\text{X}$ and $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$. Доступний за адресою [CTAN:/tex-archive/help/Catalogue/catalogue.html](http://ctan.org/tex-archive/help/Catalogue/catalogue.html).
- [11] Keith Reckdahl. *Using EPS Graphics in L^AT_EX 2_ε Documents*, що пояснює усе і навіть більше, ніж ви захочете знати, про файли EPS і як їх використовувати у документах $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$. Доступний за адресою [CTAN:/tex-archive/info/epslatex.ps](http://ctan.org/tex-archive/info/epslatex.ps). 63
- [12] Андрій Швайка. *TeX & Ukrainian*. На цій сторінці ви знайдете інформацію про $\text{T}_{\text{E}}\text{X}$ і його можливі види, а також як його навчити української мови. Адреса: <http://ph.icmp.lviv.ua/~ashv/tex/TeXandUkrainian.koi.html>. 26

Показчик

Символи

\!, 47
", 21
"', 27
"-, 27
"---, 27
"<, 27
"=", 27
">, 27
"‘, 27
\$, 41
, 44
\(, 41
\), 41
\,, 42, 47
-, 22
—, 22
\-, 21
—, 22
—, 22
..., 23
\:, 47
\;, 47
«, 22
», 22
\@, 28
@, 35
\[, 42
\[, 19, 33, 35, 78
*, 19
#, 5
\$, 5
%, 5
&, 5, 35
^, 5
-, 5

{, 5
}, 5
~, 5
\], 42
^, 44
_, 44
~, 28

А

абзац, 17
вирівнювання, 19
ліворуч, 33
праворуч, 33
відстань між, 76
відступ, 77
кінець, 5
форматування, 76
центрування, 33
акценти, 23, 24
acute, 24
grave, 24
umlaut, 24
математичні, 44
аркуш
А4, 11
А5, 11
В5, 11
executive, 11
legal, 11
letter, 11
розмір, 80
титульний, 11, 29

Б

багатомовність, 24
бібліографія, 63

В

вектори, 45
 вертикальний інтервал, 78
 верхні індекси, 44
 виділення, 31
 вирівнювання абзаців
 ліворуч, 33
 праворуч, 33
 вирівнювання стовпчиків
 ліворуч, 35
 по десятковій крапці, 36
 по центру, 35
 праворуч, 35
 вхідний файл, 8

Г

горизонтальний інтервал, 77
 графіка, 12, 61
 грецькі літери, 44
 групування, 73

Д

два стовпчики, 11
 двосторонній вивід, 11
 дефіс, 22
 довге англійське тире, 22
 довге кириличне тире, 22
 довгі рівняння, 49
 довжина, 77
 дріб, 45
 дужки, 46

Ж

жирні символи, 53

З

заголовок документа, 11
 зміст, 29
 знак мінуса, 22

І

ілюстрації, 38
 індекси
 багаторядкові, 46
 верхні, 44

нижні, 44

інтервал

 вертикальний, 78
 горизонтальний, 77
 міжрядковий, 76
 подвійний, 76

К

квадратний корінь, 44
 квадратні дужки, 6
 клас

 article, 10
 book, 10
 report, 10
 slides, 10

колонтитул

 верхній, 12
 нижній, 12

кольоровий текст, 12

кома, 23

команди, 6

 \!, 47
 \(), 41
 \), 41
 \,, 42, 47
 \-, 21
 \:, 47
 \;, 47
 \@, 28
 \[, 42
 \\, 19, 33, 35, 78
 *, 19
 \], 42
 \addtolength, 80
 \and, 29
 \appendix, 28–30
 \atop, 46
 \author, 29
 \backmatter, 30
 \backslash, 5
 \begin, 32
 \bibitem, 63
 \Big, 47
 \big, 47

`\Bigg`, 47
`\bigg`, 47
`\bigskip`, 78
`\binom`, 46
`\bmod`, 45
`\boldmath`, 53
`\boldsymbol`, 53
`\caption`, 38
`\cdot`, 45
`\cdots`, 47
`\chapter`, 28, 65, 66
`\chaptermark`, 66
`\choose`, 46
`\ci`, 69
`\cite`, 63
`\cleardoublepage`, 39
`\clearpage`, 39
`\cline`, 35
`\date`, 29
`\ddots`, 47
`\depth`, 82, 83
`\displaystyle`, 51
`\documentclass`, 10, 14, 20
`\dq`, 27
`\dum`, 69
`\emph`, 31, 73
`\end`, 32
`\fbox`, 21
`\flqq`, 22
`\footnote`, 31
`\footnotesize`, 73
`\frac`, 45
`\framebox`, 83
`\frenchspacing`, 27, 28
`\frontmatter`, 30
`\frqq`, 22
`\fussy`, 20
`\genfrac`, 46
`\glqq`, 22
`\grqq`, 22
`\height`, 82, 83
`\hline`, 35
`\hspace`, 71, 77
`\Huge`, 73
`\huge`, 73
`\hyphenation`, 20
`\idotsint`, 48
`\iiiint`, 48
`\iiint`, 48
`\iint`, 48
`\include`, 15
`\includegraphics`, 62, 82
`\includeonly`, 15
`\indent`, 77
`\index`, 64
`\input`, 15
`\int`, 46
`\item`, 32
`\label`, 30, 42
`\LARGE`, 73
`\Large`, 73
`\large`, 73
`\LaTeX`, 21
`\LaTeXe`, 21
`\ldots`, 23, 47
`\left`, 46
`\leftmark`, 65, 66
`\linebreak`, 19
`\linespread`, 76
`\listoffigures`, 39
`\listoftables`, 39
`\mainmatter`, 30
`\makebox`, 82, 83
`\makeindex`, 64
`\maketitle`, 29
`\mathbb`, 43
`\mathbf`, 74
`\mathcal`, 74
`\mathit`, 74
`\mathnormal`, 74
`\mathrm`, 51, 74
`\mathsf`, 74
`\mathtt`, 74
`\mbox`, 21, 23, 82
`\multicolumn`, 36
`\newcommand`, 70, 71
`\newenvironment`, 71
`\newline`, 19

<code>\newpage</code> , 19	<code>\sloppy</code> , 20
<code>\newtheorem</code> , 52	<code>\small</code> , 73
<code>\noindent</code> , 77	<code>\smallskip</code> , 78
<code>\nolinebreak</code> , 19	<code>\sqrt</code> , 44
<code>\nonumber</code> , 50	<code>\stackrel</code> , 46
<code>\nopagebreak</code> , 19	<code>\stretch</code> , 71, 78
<code>\normalsize</code> , 73	<code>\subparagraph</code> , 28
<code>\overbrace</code> , 44	<code>\subsection</code> , 28
<code>\overleftarrow</code> , 45	<code>\subsectionmark</code> , 66
<code>\overline</code> , 44	<code>\subsubsection</code> , 28
<code>\overrightarrow</code> , 45	<code>\sum</code> , 46
<code>\pagebreak</code> , 19	<code>\tableofcontents</code> , 29
<code>\pageref</code> , 30	<code>\TeX</code> , 21
<code>\pagestyle</code> , 12	<code>\textbf</code> , 73
<code>\paragraph</code> , 28	<code>\textit</code> , 73
<code>\parbox</code> , 82	<code>\textmd</code> , 73
<code>\parindent</code> , 76	<code>\textnormal</code> , 73
<code>\parskip</code> , 76	<code>\textrm</code> , 51, 73
<code>\part</code> , 28	<code>\textsc</code> , 73
<code>\phantom</code> , 50	<code>\textsf</code> , 73
<code>\pmod</code> , 45	<code>\textsl</code> , 73
<code>\printindex</code> , 65	<code>\textstyle</code> , 51
<code>\providecommand</code> , 71	<code>\texttt</code> , 73
<code>\ProvidesPackage</code> , 72	<code>\textup</code> , 73
<code>\qqquad</code> , 42, 47	<code>\thispagestyle</code> , 12
<code>\quad</code> , 42, 47	<code>\tiny</code> , 73
<code>\raisebox</code> , 83	<code>\title</code> , 29
<code>\ref</code> , 30, 42	<code>\tnss</code> , 70
<code>\renewcommand</code> , 70	<code>\today</code> , 21
<code>\renewenvironment</code> , 71	<code>\totalheight</code> , 82
<code>\right</code> , 46	<code>\totalwidth</code> , 83
<code>\right.</code> , 47	<code>\underbrace</code> , 44
<code>\rightmark</code> , 65, 66	<code>\underline</code> , 31, 44
<code>\rule</code> , 71, 84	<code>\usepackage</code> , 12, 14, 24–26, 72
<code>\scriptscriptstyle</code> , 51	<code>\vdots</code> , 47
<code>\scriptsize</code> , 73	<code>\vec</code> , 45
<code>\scriptstyle</code> , 51	<code>\verb</code> , 34
<code>\section</code> , 28, 65	<code>\verbatiminput</code> , 67
<code>\sectionmark</code> , 66	<code>\vspace</code> , 78
<code>\selectlanguage</code> , 24	<code>\widehat</code> , 44
<code>\setlength</code> , 76, 80	<code>\widetilde</code> , 44
<code>\settodepth</code> , 81	<code>\width</code> , 82, 83
<code>\settoheight</code> , 81	коментарі, 6
<code>\settowidth</code> , 81	коротке типе, 22

крапка, 23

крапка, пробіл після, 28

Л

лапки, 21

лігатура, 23

лінії

горизонтальні, 44

літери

європейські, 24

М

макетування сторінки, 80

математика, 41

математичний

мінус, 22

обмежувач, 47

пробіл, 47

математичні

функції, 45

мова

німецька, 25

українська, 26

Н

напівжирні символи, 43

необов'язкові параметри, 6

нижні індекси, 44

О

обмежувачі, 46

одиниці, 77

односторонній вивід, 11

оператор

інтеграл, 46

суми, 46

опції, 10

оточення, 32

align, 50

array, 48, 49

center, 33

comment, 7

description, 32

displaymath, 42

enumerate, 32

eqnarray, 49

equation, 42, 49

figure, 37, 38, 61

flalign, 50

flushleft, 33

flushright, 33

gather, 50

itemize, 32

lscocommand, 69

math, 41

minipage, 82

multline, 50

quotation, 33

quote, 33

split, 50

table, 37, 38, 61

tabular, 35, 48, 82

thebibliography, 63

verbatim, 34, 66, 67

verse, 33

П

пакет, 7, 12, 69

пакет makeidx, 64

пакети

ae, 26

amsbsy, 53

amsfonts, 43, 60

amsmath, 46–48, 50, 53

amssym, 51

amssymb, 43, 54

babel, 8, 20, 24–28, 41

bm, 53

calc, 81

dcolumn, 36

doc, 13

eucal, 60

eufrak, 60

exscale, 13, 47

fancyhdr, 65, 66

fontenc, 13, 26, 41

geometry, 67

graphicx, 61, 62

hyphenat, 67

ifthen, 13
 indentfirst, 77
 inputenc, 13, 25
 latexsym, 13, 55–57
 layout, 80
 longtabular, 36
 makeidx, 13, 64
 mathrsfs, 60
 showidx, 65
 supertabular, 36
 syntonly, 13, 16
 tabular, 36
 verbatim, 7, 66, 67
 параметр, 6
 переваги L^AT_EX, 4
 перехресні посилання, 30
 плаваючі об'єкти, 37
 поля, 80
 порожні символи, 5
 на початку рядка, 5
 похідна, 44
 правила переносу, 24
 преамбула, 7
 предметний покажчик, 64
 пробіл, 5
 після команди, 6
 програма makeindex, 64

Р

розмір аркуша, 11, 80
 розмір основного шрифту, 11
 розпірка, 84
 розриви рядків, 19
 розширення, 12
 .aux, 15
 .cls, 14
 .dtx, 14, 67, 68
 .dvi, 8, 10, 14, 16, 62, 67
 .eps, 62
 .glo, 68
 .gls, 68
 .idx, 15, 64, 67
 .ilg, 15
 .ind, 15, 65

.ins, 14, 67
 .ist, 68
 .lof, 14
 .log, 14
 .lot, 15
 .pdf, 68
 .ps, 68
 .sty, 14, 67
 .tex, 8, 12
 .toc, 14

С

символ градуса, 22

символи

 зарезервовані, 5

 #, 5

 \$, 5

 %, 5

 &, 5, 35

 ^, 5

 _, 5

 {, 5

 }, 5

 ~, 5

система рівнянь, 49

специфікація розміщення, 38

спеціальні символи, 23

стилі сторінки, 12

стиль сторінки

 empty, 12

 headings, 12

 plain, 12

сторінка

 макетування, 80

стрілки, 45

структура файлу, 7

Т

таблиці, 38

 вирівнювання стовпчиків

 ліворуч, 35

 по десятковій крапці, 36

 по центру, 35

 праворуч, 35

- тильда, 22, 44
тильда (~), 28
типи файлів, 12
тире, 22
 довге
 англійське, 22
 кирилічне, 22
 коротке, 22
титульний аркуш, 29
трикрапки, 23, 47
 вертикальні, 47
 горизонтальні, 47
 діагональні, 47
- Ф**
фігурні дужки, 6, 73
 горизонтальні, 44
формули, 41
функція модуля, 45
- Ш**
шрифт, 72
 без зарубок, 73
 документа, розмір, 11
 жирний, 73
 з зарубками, 73
 капітель, 73
 курсив, 31, 73
 математичний, розмір, 51
 машинопис, 73
 похилий, 73
 прямий, 73
 розмір, 72, 73
- А**
\addtolength, 80
æ, 24
ae, 26
align, 50
amsbsy, 53
amsmath, 43, 60
amsmath, 46–48, 50, 53
amssym, 51
amssymb, 43, 54
\and, 29
\appendix, 28–30
 array, 48, 49
\atop, 46
\author, 29
.aux, 15
- В**
babel, 8, 20, 24–28, 41
\backmatter, 30
\backslash, 5
\begin, 32
\bibitem, 63
\Big, 47
\big, 47
\Bigg, 47
\bigg, 47
\bigskip, 78
\binom, 46
 blackboard bold, 43
 bm, 53
\bmod, 45
\boldmath, 53
\boldsymbol, 53
- С**
calc, 81
\caption, 38
\cdot, 45
\cdots, 47
 center, 33
\chapter, 28, 65, 66
\chaptermark, 66
\choose, 46
\ci, 69
\cite, 63
\cleardoublepage, 39
\clearpage, 39
\cline, 35
.cls, 14
 comment, 7
- D**
\date, 29
dcolumn, 36
\ddots, 47

`\depth`, 82, 83
 description, 32
 `displaymath`, 42
`\displaystyle`, 51
 doc, 13
`\documentclass`, 10, 14, 20
`\dq`, 27
 `.dtx`, 14, 67, 68
`\dum`, 69
 `.dvi`, 8, 10, 14, 16, 62, 67
 dvips, 8

E

`\emph`, 31, 73
 empty, 12
 Encapsulated PostScript, 61
`\end`, 32
 enumerate, 32
 EPS, 61
 `.eps`, 62
 `eqnarray`, 49
 equation, 42, 49
 eucal, 60
 eufrak, 60
 `exscale`, 13, 47

F

`fancyhdr`, 65, 66
`\fbox`, 21
 figure, 37, 38, 61
 `flalign`, 50
`\flqq`, 22
 `flushleft`, 33
 `flushright`, 33
 foiltex, 10
 `fontenc`, 13, 26, 41
`\footnote`, 31
`\footnotesize`, 73
`\frac`, 45
`\framebox`, 83
`\frenchspacing`, 27, 28
`\frontmatter`, 30
`\frqq`, 22
`\fussy`, 20

G

 gather, 50
`\genfrac`, 46
 geometry, 67
 GhostScript, 61
 `.glo`, 68
`\glqq`, 22
 `.gls`, 68
 `graphicx`, 61, 62
`\grqq`, 22

H

`textttheadings`, 12
`\height`, 82, 83
`\hline`, 35
`\hspace`, 71, 77
`\Huge`, 73
`\huge`, 73
 hyphenat, 67
`\hyphenation`, 20

I

 1 без крапки, 24
`\idotsint`, 48
 `.idx`, 15, 64, 67
 ifthen, 13
`\iiiint`, 48
`\iiint`, 48
`\iint`, 48
 `.ilg`, 15
`\include`, 15
`\includegraphics`, 62, 82
`\includeonly`, 15
 `.ind`, 15, 65
`\indent`, 77
 indentfirst, 77
`\index`, 64
`\input`, 15
 inputenc, 13, 25
 `.ins`, 14, 67
`\int`, 46
 `.ist`, 68
`\item`, 32
 itemize, 32

J

j без крапки, 24

К

Knuth, Donald E., 1

L

\label, 30, 42

Lamport, Leslie, 1

\LARGE, 73

\Large, 73

\large, 73

L^AT_EX 2.09, 2L^AT_EX 2_ε, 2

\LaTeX, 21

L^AT_EX3, 2, 4

\LaTeXe, 21

latexsym, 13, 55–57

layout, 80

\ldots, 23, 47

\left, 46

\leftmark, 65, 66

\linebreak, 19

\linespread, 76

\listoffigures, 39

\listoftables, 39

.lof, 14

.log, 14

longtabular, 36

.lot, 15

lscmmand, 69

M

\mainmatter, 30

\makebox, 82, 83

makeidx, 13, 64

\makeindex, 64

\maketitle, 29

math, 41

\mathbb, 43

\mathbf, 74

\mathcal, 74

\mathit, 74

\mathnormal, 74

\mathrm, 51, 74

mathrfs, 60

\mathsf, 74

\mathtt, 74

\mbox, 21, 23, 82

minipage, 82

Mittelbach, Frank, 2

\multicolumn, 36

multline, 50

N

\newcommand, 70, 71

\newenvironment, 71

\newline, 19

\newpage, 19

\newtheorem, 52

\noindent, 77

\nolinebreak, 19

\nonumber, 50

\nopagebreak, 19

\normalsize, 73

O

œ, 24

\overbrace, 44

overfull hbox, 20

\overleftarrow, 45

\overline, 44

\overrightarrow, 45

P

\pagebreak, 19

\pageref, 30

\pagestyle, 12

\paragraph, 28

\parbox, 82

\parindent, 76

\parskip, 76

\part, 28

.pdf, 68

\phantom, 50

plain, 12

\pmod, 45

PostScript, 61

\printindex, 65

\providecommand, 71

- `\ProvidesPackage`, 72
- `.ps`, 68
- Q**
- `\qqquad`, 42, 47
- `\quad`, 42, 47
- quotation, 33
- quote, 33
- R**
- `\raisebox`, 83
- `\ref`, 30, 42
- `\renewcommand`, 70
- `\renewenvironment`, 71
- `\right`, 46
- `\right.`, 47
- `\rightmark`, 65, 66
- `\rule`, 71, 84
- S**
- `\scriptscriptstyle`, 51
- `\scriptsize`, 73
- `\scriptstyle`, 51
- `\section`, 28, 65
- `\sectionmark`, 66
- `\selectlanguage`, 24
- `\setlength`, 76, 80
- `\settodepth`, 81
- `\settoheight`, 81
- `\settowidth`, 81
- showidx, 65
- `\sloppy`, 20
- `\small`, 73
- `\smallskip`, 78
- split, 50
- `\sqrt`, 44
- `\stackrel`, 46
- `\stretch`, 71, 78
- `.sty`, 14, 67
- `\subparagraph`, 28
- `\subsection`, 28
- `\subsectionmark`, 66
- `\subsubsection`, 28
- `\sum`, 46
- supertabular, 36
- syntonly, 13, 16
- T**
- table, 37, 38, 61
- `\tableofcontents`, 29
- tabular, 36
- tabular, 35, 48, 82
- `\TeX`, 21
- `.tex`, 8, 12
- `\textbf`, 73
- `\textit`, 73
- `\textmd`, 73
- `\textnormal`, 73
- `\textrm`, 51, 73
- `\textsc`, 73
- `\textsf`, 73
- `\textsl`, 73
- `\textstyle`, 51
- `\texttt`, 73
- `\textup`, 73
- thebibliography, 63
- `\thispagestyle`, 12
- `\tiny`, 73
- `\title`, 29
- `\tnss`, 70
- `.toc`, 14
- `\today`, 21
- `\totalheight`, 82
- `\totalwidth`, 83
- U**
- `\underbrace`, 44
- underfull hbox, 20
- `\underline`, 31, 44
- URL, 22
- `\usepackage`, 12, 14, 24–26, 72
- V**
- `\vdots`, 47
- `\vec`, 45
- `\verb`, 34
- verbatim, 7, 66, 67
- verbatim, 34, 66, 67
- `\verbatiminput`, 67
- verse, 33

`\vspace`, 78

W

`\widehat`, 44

`\widetilde`, 44

`\width`, 82, 83

www, 22

WYSIWYG, 3, 4

X

xdvi, 8

