

Formatare

Termenul de formatare este destul de amplu, însă în cazul de față trebuie să fie, deoarece această secțiune vă va ghida prin diversele tehnici de formatare a textului, paragrafelor și paginilor. Formatarea tinde să se refere în cea mai mare măsură la lucruri legate de aspect, afișaj, lista de subiecte posibile fiind destul de eclectică: stilul de text, font, dimensiune; alinierea paragrafelor, spațierea dintre linii, indentarea; tipuri speciale de paragrafe; structuri de tip listă; note de subsol, note de margine, etc.

Mare parte a tehnicilor de formatare sunt necesare pentru a diferenția anumite elemente de restul textului. Este deseori necesar să scoateți în evidență cuvinte cheie sau anumite expresii. O listă numerotată sau cu marcatori este de asemenea folosită ca un mod clar și concis de a comunica o problemă importantă. Notele de subsol oferă informații sau clarificări suplimentare fără a întrerupe fluxul principal de text. Așadar, din aceste motive, formatarea este foarte importantă. Totuși, este de asemenea foarte ușor să abuzați de aceste facilități, iar un document prea bine stilizat poate arăta mai rău și fi mai greu de înțeles decât unul fără nicio înfloritură.

Formatarea textului

Împărțirea în silabe

LaTeX împarte cuvintele în silabe oricând este necesar. Dacă algoritmul de împărțire în silabe nu găsește punctele corecte de împărțire în silabe, puteți remedia situația folosind următoarele comenzi pentru a comunica sistemului TeX excepția. Comanda

```
\hyphenation {listă de cuvinte}
```

face cuvintele din argument să fie despărțite în silabe numai în punctele marcate de "-". Argumentul comenzii ar trebui să conțină numai cuvinte formate din litere, sau mai degrabă caractere ce sunt considerate litere de către LaTeX. Se știe că algoritmul de împărțire în silabe nu găsește toate punctele corecte de împărțire în silabe pentru numeroase cuvinte din vocabular. Un fișier cu excepțiile cunoscute din limba engleză este publicat periodic în revista *TUGboat*. (Vezi o listă recentă din 2008:

<http://www.tug.org/TUGboat/Articles/tb29-2/tb92hyf.pdf>)

Indiciile de împărțire în silabe sunt păstrate în memorie pentru limba care este activă când se întâlnește comanda de împărțire în silabe. Asta înseamnă că dacă plasați o comandă de împărțire în silabe în preambulul documentului, va influența împărțirea în silabe pentru limba implicită. Dacă plasați comanda după `\begin{document}` și folosiți un pachet pentru limba națională de genul `babel`, atunci indiciile de împărțire în silabe vor fi active pentru limba activată prin intermediul pachetului. Exemplul următor permite cuvintelor "silabisire" și "Silabisire" să fie împărțite în silabe, dar previne orice împărțire în silabe pentru "FORTRAN", "Fortran" și "fortran". În argument nu sunt permise caractere speciale sau simboluri:

```
usepackage[romanian]{babel}
```

```
\hyphenation{FORTRAN Si-la-bi-si-re}
```

Comanda `\-` inserează o cratimă în locul potrivit într-un cuvânt. Acesta devine de asemenea singurul punct de împărțire în silabe permis pentru acel cuvânt. Comanda `\-` este deosebit de utilă pentru cuvinte ce conțin caractere speciale (spre exemplu, cuvinte cu diacritice), întrucât LaTeX nu împarte automat în silabe cuvinte ce conțin caractere speciale.

```
\begin{minipage}{2in}
Cred că asta este:
su\per\cal\
i\frag\i\lis\tic\
ex\pi\
al\i\do\cious
\end{minipage}
```

**Cred că asta este: supercalifra
listicexpialidocious**

Acest lucru este greu de gestionat pentru multe cuvinte ce conțin o cratimă, de genul "propriu-zis", "după-amiaza". O soluție posibilă ar fi folosirea comenzii `\hyp` din pachetul `hyphenat`. Această comandă pune o cratimă și permite împărțirea automată în silabe a cuvintelor rezultate din cuvântul `compus`. Vom scrie atunci:

```
propriu\hyp{}zis ... după\hyp{}amiaza
```

Mai multe cuvinte pot fi ținute împreună pe o linie cu comanda

`\mbox{text}`

Această comandă face ca argumentul să fie ținut împreună în orice circumstanțe. De exemplu:

Numărul meu de telefon se va schimba în curând. Va fi `\mbox{0116 291 2319}`.

Comanda `\fbox` este similară cu `\mbox`, dar în plus conținutul va fi încadrat într-o căsuță vizibilă.

Pentru a evita complet împărțirea în silabe, penalizarea pentru împărțirea în silabe poate fi setată la o valoare extremă:

`\hyphenpenalty=100000`

Câteodată poate veți dori să schimbați măsura în care LaTeX împarte în silabe modificând valoarea `\tolerance=1000` și `\hyphenpenalty=1000`. Trebuie să experimentați cu aceste valori pentru a obține efectul dorit. O toleranță redusă nu va tolera spații inegale între cuvinte, astfel că împărțirea în silabe se va realiza mai des decât cu o toleranță înaltă.

Citate

Latex tratează ghilimelele duble stânga și drepte ca entități diferite. Pentru ghilimelele simple, `' '` (pe tastaturile americane, acest simbol se găsește pe tasta cu tilda, în general lângă cheia `' 1 '`) reprezintă ghilimelele simple de deschidere, iar `' '` ghilimelele simple de închidere. Pentru a obține ghilimelele duble, pur și simplu dublați simbolurile, iar Latex le va interpreta în mod corespunzător. (Deși, puteți folosi `" "` pentru ghilimele duble drepte). Pe tastaturile englezești, `' '` este la stânga tastei `' 1 '`, pe care se află și `' - '`, iar câteodată `' | '` sau `' | '`. Tasta apostrof (`' '`) este la dreapta tastei `' : / ; '`, pe care se mai află simbolul `@`.

Pentru a ``cita'`
în Latex
Pentru a ```cita''`
în Latex
Pentru a ```cita''`
în Latex

Pentru a `'cita'` în Latex

Pentru a `"cita"` în Latex

Pentru a `"cita"` în Latex

Pentru a „cita”
în Latex

``Vă rugăm să
apăsați tasta
'x'."

„Vă rugăm să
apăsați tasta
<<x>>".

Pentru a „cita” în Latex

“Vă rugăm să apăsați tasta ‘x’.”

„Vă rugăm să apăsați tasta «x»”.

Ghilimelele simple drepte sunt folosite de asemenea ca apostrof în Latex.

Pentru ghilimele stângi așezate în partea inferioară și pentru stilul de citare european, trebuie să folosiți tipul de font T1:

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

Diacritice

Puteți să adăugați diacritice la litere scriind un metacarakter diacritic înaintea literei respective. Iată o listă a metacacterelor diacritice.

| Comandă | Output | Comandă | Output |
|---------------------|--------|-----------------------|--------|
| <code>\`{o}</code> | ò | <code>\´{o}</code> | ó |
| <code>\" {o}</code> | ö | <code>\H{o}</code> | ő |
| <code>\^{o}</code> | ô | <code>\~{o}</code> | õ |
| <code>\v{o}</code> | õ | <code>\={o}</code> | ō |
| <code>\b{o}</code> | Ⓡ | <code>\cdot{o}</code> | ò |
| <code>\d{o}</code> | Ⓡ | <code>\c{o}</code> | ç |
| <code>\r{o}</code> | ô | <code>\t{o}</code> | ōo |
| <code>\i</code> | ı | <code>\^{i}</code> | î |

Pentru a așeza un diacritic pe un 'i' sau 'j', punctul deasupra acestei litere trebuie înlăturat. Acest lucru se realizează scriind `\i` și `\j`. Dacă vreți să vă scrieți toate documentele într-o limbă străină și trebuie să folosiți anumite diacritice de mai multe ori, atunci cu ajutorul configurației corecte, puteți scrie aceste caractere direct în documente.

Recent, XeTeX a inclus posibilitatea de a scrie caractere Unicode direct în fișierul sursă `.tex`. Dacă folosiți XeLaTeX, puteți folosi direct orice diacritice Unicode combinate.

Pentru mai multe informații, vezi capitolul despre [Internaționalizare](#).

Spațiu între cuvinte și fraze

Pentru a obține o margine dreaptă aranjată în output, LaTeX inserează spații de mărimi diferite între cuvinte. Implicit, inserează de asemenea mai mult spațiu la sfârșitul unei fraze. Cu toate acestea, spațiul suplimentar adăugat la sfârșitul frazelor este considerat în general demodat din punct de vedere tipografic în engleza scrisă. (Practica aceasta se întâlnește în design-ul din secolul al XIX-lea și în stilurile de redactare din secolul al XX-lea.) Cele mai multe programe moderne de redactare de text tratează spațiul de la sfârșitul frazei la fel ca spațiul dintre cuvinte. (Vezi, spre exemplu, *Elementele stilului tipografic* de Bringham.) Spațiul adițional de după punctele ce marchează sfârșitul unei fraze pot fi dezactivate cu comanda

```
\frenchspacing
```

care spune LaTeX să nu insereze mai mult spațiu după un punct decât după orice alt caracter. Spațierea franceză poate fi oprită mai târziu în document prin comanda `\nonfrenchspacing`.

Dacă un autor dorește să continue spațierea de tip vechi de la sfârșitul frazei, trebuie să aibă grijă că semnele de punctuație nu sunt interpretate greșit ca sfârșit de frază. TeX presupune că frazele se termină cu puncte, semne de întrebare sau de exclamare. Chiar dacă o literă mare este urmată de punct, acesta nu este interpretat ca sfârșit de frază, deoarece punctele de după litere mari apar în mod normal în abrevieri. Orice excepție de la aceste prezumții trebuie să fie specificată de autor. Un backslash în fața unui spațiu generează un spațiu ce nu va fi mărit. Un caracter tilda `'~'` generează un spațiu care nu poate fi mărit și în plus împiedică apariția unui sfârșit de linie. Comanda `'\@'` în fața unui punct specifică faptul că acest punct termină o frază chiar și atunci când urmează după o literă mare. (Dacă folosiți `\frenchspacing`, niciuna dintre aceste excepții nu trebuie specificată.)

Margini nealiniate și spații între cuvinte

Unele cuvinte foarte lungi, numere și URL-uri nu pot fi împărțite în silabe în mod corespunzător și trec dincolo de marginea din dreapta. O soluție la

această problemă este folosirea mediului `sloppypar`, care spune LaTeX să ajusteze mai puțin strict spațierea dintre cuvinte. Drept urmare, unele spații dintre cuvinte pot fi puțin prea mari, însă cuvintele lungi vor fi plasate cum trebuie.

Acesta este un
paragraf cu
un cuvânt foarte
lung
ABCDEFGHIJKL
MNOPRST;
apoi avem încă un
lucru prost
--- un număr lung
123456789012345
6789.

```
\begin{sloppypar}
Acesta este un
paragraf cu
un cuvânt foarte
lung
ABCDEFGHIJKL
MNOPRST;
apoi avem încă un
lucru prost
--- un număr lung
123456789012345
6789.
\end{sloppypar}
```

Acesta este un paragraf cu un cuvânt foarte lung
MNOPRST; apoi avem încă un lucru prost — un număr
Acesta este un paragraf cu un cuvânt
JKLMNOPRST; apoi avem încă un lucru
1234567890123456789.

Ligaturi

Unele combinații de litere sunt afișate nu doar separat, ci și prin afișarea de simboluri speciale (ca "ff"), numite [ligaturi](#). Ligaturile pot fi interzise prin inserarea `{}` sau, dacă asta nu merge, `{\kern0pt}`, între cele două litere în chestiune. Acest lucru ar putea fi necesar pentru cuvinte compuse din două cuvinte. Iată un exemplu:

`\Large`
Nu shelfful

`ci shelf{}ful`

Nu shelfful

ci shelfful

Unele programe nu reușesc să caute cuvinte în documente ce conțin ligaturi (căutarea "`\finally`" nu va găsi cuvântul "`finally`"). Dacă doriți, pentru mai multă accesibilitate, pentru a dezactiva complet ligaturile pentru întregul document, puteți folosi `\DisableLigatures` din pachetul [microtype](#):

```
\usepackage{microtype}
\DisableLigatures{encoding = *, family = *}
```

Notăți că va dezactiva doar ligaturile de genul `--`, `---`, etc.

Dacă folosiți XeLaTeX și fonturi OpenType, pachetul [fontspec](#) permite dezactivarea ligaturilor standard și activarea ligaturilor *fancy swash*.

Caractere slash

În mod normal, un caracter `/` în LaTeX nu permite caracterele următoare să fie "separate" pe liniile următoare, ceea ce creează erori "overfull" în output (în care literele se întind dincolo de marginea din dreapta). Cuvintele care folosesc caractere slash, cum ar fi "input/output" ar trebui scrise sub forma "input/slash output", care permite liniei să se "împartă" după caracterul slash (dacă este necesar). Utilizarea caracterului `/` în LaTeX ar trebui restrânsă la unități, ca în cazul "mm/year", care nu ar trebui separate pe mai multe linii.

Evidențierea textului

Pentru a scoate în evidență un cuvânt sau o expresie, modul cel mai simplu este comanda `\emph{text}`.

Vreau să
`\emph{sco`
t în
`evidență}`
un cuvânt.

Vreau să *scot în evidență* un cuvânt.

Fonturi

Vezi și capitolul [Fonturi](#).

În LaTeX, sunt multe căi prin care puteți specifica și controla fonturi, iar această secțiune este menită numai să servească ca scurtă introducere a acestui subiect.

Stiluri de fonturi

Există 3 familii principale de fonturi: *roman* (de exemplu, *Times*), *sans serif* (cum este *Arial*) și *monospace* (ca, de exemplu, *Courier*). Puteți de asemenea să specificați stiluri precum *italic* și *bold*.

Tabelul următor prezintă comenzile de care aveți nevoie pentru a accesa stilurile tipice de fonturi:

| Comandă LaTeX | Echivalent cu | Stilul de output | Remărci |
|-------------------------------|--------------------------------|--|---|
| <code>\textnormal{...}</code> | <code>{\normalfont ...}</code> | familia de fonturi implicită pentru document | acesta este fontul implicit sau normal |
| <code>\emph{...}</code> | <code>{\em ...}</code> | <i>emphasis</i> | caractere italice |
| <code>\textrm{...}</code> | <code>{\rmfamily ...}</code> | familia de fonturi <i>roman</i> | |
| <code>\textsf{...}</code> | <code>{\sffamily ...}</code> | familia de fonturi <i>sans serif</i> | |
| <code>\texttt{...}</code> | <code>{\ttfamily ...}</code> | familia de fonturi <i>teletype</i> | acesta este un font de mărime fixă sau <i>monospace</i> |
| <code>\textup{...}</code> | <code>{\upshape ...}</code> | forme verticale | la fel ca <i>typeface</i> normal |
| <code>\textit{...}</code> | <code>{\itshape ...}</code> | forme italice | |
| <code>\textsl{...}</code> | <code>{\slshape ...}</code> | forme oblice | versiune oblică a tipului normal <i>typeface</i> (similar cu, dar ușor diferit de, italice) |
| <code>\textsc{...}</code> | <code>{\scshape ...}</code> | LITERE MARI CU | |

| | FONT DE DIMENSIUNE REDUSĂ | |
|------------------------------|---|---|
| <code>\uppercase{...}</code> | LITERE MARI | Și <code>\lowercase</code> . Sunt câteva capcane, totuși; vezi aici . |
| <code>\textbf{...}</code> | <code>{\bfseries ...}</code> bold | |
| <code>\textmd{...}</code> | <code>{\mdseries ...}</code> grosime medie | font între dimensiunea normală și aldina |

Poate ați observat absența *underline*. Deși acesta este disponibil prin intermediul comenzii `\underline{...}`, textul subliniat în felul acesta nu va fi împărțit în silabe în mod corespunzător. Această funcționalitate trebuie adăugată cu pachetul `ulem`. Scrieți `\usepackage{ulem}` în preambul. Implicit, asta suprascrive comanda `\emph`, generând text subliniat și nu italic. Puțin probabil că veți dori așa ceva, astfel că este de dorit să împiedicați pachetul `ulem` să preia controlul asupra comenzii `\emph`, chemând comanda `underline` dacă și când aveți nevoie.

- Pentru a reface formatarea `em` uzuală, adăugați `\normalem` imediat după începutul mediului *document*. Altfel, adăugați această opțiune când includeți pachetul `ulem`: `\usepackage[normalem]{ulem}`.
- Pentru text subliniat, scrieți `\uline{...}`.
- Pentru text subliniat cu o linie ondulată, puteți utiliza comanda `\uwave{...}`.
- Pentru text tăiat de o linie (*strike-out*), folosiți comanda `\sout{...}`.

Modificarea dimensiunii caracterelor

Pentru a aplica mărimi diferite de fonturi, urmați comenzile din acest tabel:

| Command | Output |
|----------------------------|-------------|
| <code>\tiny</code> | sample text |
| <code>\scriptsize</code> | sample text |
| <code>\footnotesize</code> | sample text |
| <code>\small</code> | sample text |
| <code>\normalsize</code> | sample text |
| <code>\large</code> | sample text |
| <code>\Large</code> | sample text |
| <code>\LARGE</code> | sample text |
| <code>\huge</code> | sample text |
| <code>\Huge</code> | sample text |

Notați că definițiile mărimii fonturilor sunt setate de clasa *document*. În funcție de stilul de document, mărimea efectivă a fontului poate diferi de cele listate mai devreme. Și nu orice clasă de documente are mărimi unice pentru toate cele 10 comenzi de dimensionare a caracterelor.

Mărimi absolute ale caracterelor în clasele de documente *article*, *proc*, *report*, *book* și *letter*

| mărime | 10pt (implicit) | 11pt | 12pt |
|----------------------------|-----------------|----------|----------|
| <code>\tiny</code> | 6.80565 | 7.33325 | 7.33325 |
| <code>\scriptsize</code> | 7.97224 | 8.50012 | 8.50012 |
| <code>\footnotesize</code> | 8.50012 | 9.24994 | 10.00002 |
| <code>\small</code> | 9.24994 | 10.00002 | 10.95003 |
| <code>\normalsize</code> | 10.00002 | 10.95003 | 11.74988 |
| <code>\large</code> | 11.74988 | 11.74988 | 14.09984 |
| <code>\Large</code> | 14.09984 | 14.09984 | 15.84985 |
| <code>\LARGE</code> | 15.84985 | 15.84985 | 19.02350 |
| <code>\huge</code> | 19.02350 | 19.02350 | 22.82086 |
| <code>\Huge</code> | 22.82086 | 22.82086 | 22.82086 |

Mărimi absolute ale caracterelor în clasele de documente *memoir*, *amsart* și *amsbook*

| mărime | 10pt (implicit) | 11pt | 12pt |
|----------------------------|-----------------|----------|----------|
| <code>\tiny</code> | 7.33325 | 7.97224 | 8.50012 |
| <code>\scriptsize</code> | 7.97224 | 8.50012 | 9.24994 |
| <code>\footnotesize</code> | 8.50012 | 9.24994 | 10.00002 |
| <code>\small</code> | 9.24994 | 10.00002 | 10.95003 |

| | | | |
|--------------------------|----------|----------|----------|
| <code>\normalsize</code> | 10.00002 | 10.95003 | 11.74988 |
| <code>\large</code> | 10.95003 | 11.74988 | 14.09984 |
| <code>\Large</code> | 11.74988 | 14.09984 | 15.84985 |
| <code>\LARGE</code> | 14.09984 | 15.84985 | 19.02350 |
| <code>\huge</code> | 15.84985 | 19.02350 | 22.82086 |
| <code>\Huge</code> | 19.02350 | 22.82086 | 22.82086 |

Mărimi absolute ale caracterelor în clasa de documente *slides*

mărime

| | |
|----------------------------|----------|
| <code>\tiny</code> | 17.27505 |
| <code>\scriptsize</code> | 20.73755 |
| <code>\footnotesize</code> | 20.73755 |
| <code>\small</code> | 20.73755 |
| <code>\normalsize</code> | 24.88382 |
| <code>\large</code> | 29.86258 |
| <code>\Large</code> | 35.82510 |
| <code>\LARGE</code> | 43.00012 |
| <code>\huge</code> | 51.60014 |
| <code>\Huge</code> | 51.60014 |

Mărimi absolute ale caracterelor în clasa de documente *beamer*

| mărime | 10pt (implicit) | 11pt | 12pt |
|----------------------------|------------------------|-------------|-------------|
| <code>\tiny</code> | 5.31258 | 6.37509 | 6.37509 |
| <code>\scriptsize</code> | 7.43760 | 8.50012 | 8.50012 |
| <code>\footnotesize</code> | 8.50012 | 9.24994 | 10.00002 |
| <code>\small</code> | 9.24994 | 10.00002 | 10.95003 |
| <code>\normalsize</code> | 10.00002 | 10.95003 | 11.74988 |
| <code>\large</code> | 11.74988 | 11.74988 | 14.09984 |
| <code>\Large</code> | 14.09984 | 14.09984 | 16.24988 |
| <code>\LARGE</code> | 16.24988 | 16.24988 | 19.50362 |
| <code>\huge</code> | 19.50362 | 19.50362 | 23.39682 |
| <code>\Huge</code> | 23.39682 | 23.39682 | 23.39682 |

Ca o notă tehnică, punctele din TeX se bazează pe mărimea standard americană a punctului, pentru care 1 pt este aproximativ 0.351366666666667 mm. Mărimea standard a punctului în cele mai multe programe moderne de calculator (cunoscută sub denumirea de *desktop publishing point* sau

PostScript point) conform căreia 1 pt este aproximativ 0.352777777777778 mm, în vreme ce conform standardului european (*Didot point*) 1 pt este aproximativ 0.375971510382829 mm. (Vezi: [Punctul tipografic.](#))

Chiar dacă puteți schimba ușor outputul fonturilor cu ajutorul acestor comenzi, e mai bine să nu folosiți explicit comenzi de genul acesta, întrucât sunt în opoziție cu ideea de bază din LaTeX, care constă în separarea aspectului logic și vizual ale documentului. Asta înseamnă că dacă folosiți aceeași comandă pentru schimbarea fontului în mai multe locuri pentru a afișa un tip special de informații, ar trebui să utilizați `\newcommand` pentru a defini un "wrapper" logic pentru comanda de schimbare a fontului.

```
\newcommand{\oops}[1]{\textbf{#1}}
```

Nu `\oops{intrați}` în această cameră, este ocupată de `\oops{mașini}` de origine și cu scop necunoscute.

Nu **intrați** în această cameră, este ocupată de **mașini** de origine și cu scop necunoscute.

Această abordare are avantajul că puteți decide mai târziu dacă vreți să folosiți o anume reprezentare vizuală a pericolului alta decât `\textbf`, fără a trebui să căutați prin document, identificând toate aparițiile `\textbf` și determinând pentru fiecare dacă este folosit pentru a indica un pericol sau pentru alt motiv.

Exponent și indice în modul text

Pentru a scrie la exponent în modul text, puteți folosi comanda `\textsuperscript{}`. Asta vă permite, spre exemplu, să scrieți "6th" sub forma "6th".

```
Michelangelo s-a născut la 6th Martie (March  
6\textsuperscript{th}), 1475.
```

Principala utilizare a indicilor în mediul text este tipărirea formulelor chimice. În acest sens, un pachet foarte recomandat este [mhchem](#). Acest pachet este ușor de utilizat și merge cu fonturile implicite de text (în locul fonturilor matematice). Pentru a insera o formulă chimică, utilizați comanda `\ce{}` cu formula echivalentă în modul text, spre exemplu:

% În preambul, adăugați:

Sulfat de amoniu: (NH₄)

```
\usepackage[version=3]{mhchem}
```

```
...
```

```
% În document:
```

```
Sulfat de amoniu:
```

```
\ce{(NH4)2SO4}.
```

Scrierea indicilor în modul text nu este suportată numai de LaTeX; totuși, mai multe pachete permit utilizarea comenzii `\textsubscript{}`. Spre exemplu, [bpchem](#), [KOMA-Script2](#) și [fixltx2e](#) suportă toate această comandă. Dintre acestea, [fixltx2e](#) este poate opțiunea aproape universală de vreme ce este distribuit cu LaTeX și nu necesită niciun pachet adițional pentru a fi implementat.

```
% În  
preambul,  
adăugați:  
\usepackage{fixltx2e}
```

```
...
```

```
% În
```

```
document:
```

```
S-a constatat
```

```
că înălțimea
```

```
\textsubscript
```

```
{mărului}
```

```
este
```

```
diferită de
```

```
înălțimea
```

```
\textsubscript
```

```
{portocalului}
```

```
.
```

S-a constatat că înălțimea mărului es

Dacă nu încărcați un pachet ce suportă comanda `\textsubscript{}`, trebuie să folosiți modul matematic. Acest lucru este ușor de realizat în modul text punând codul între simboluri `$`. În modul matematic scrierea de indici se realizează cu ajutorul unui underscore: `_{}{}`.

Spre exemplu, formula pentru apă se scrie în felul următor:

H_2O este formula pentru apă | H_2O este formula pentru apă

Notăți că în modul matematic textul va apare într-un font potrivit pentru variabilele matematice. În modul matematic, pentru a genera text în stilul "roman", spre exemplu, folosiți comanda `\mathrm`:

Acesta este
text
`\mathrm{roman}`
`\normal`,
iar
`\thisindex`

Acesta este text roman normal, iar

Notăți folosirea comenzii `\<spațiu>` pentru a insera un spațiu în modul matematic.

În mod similar, puteți scrie exponenți în felul următor:

Acesta este text
`\mathrm{roman}`
`\normal`, iar
acesta
`\thisindex`
exponenț

Acesta este text roman normal, ia

Cifre în modul text (numere "în stilul vechi")

Mulți tipografi preferă să folosească cifre poziționate la înălțimea literelor mici, atunci când sunt amestecate cu literele mari, când apar în tabele sau în ecuații, folosind [cifre în modul text](#) în altă parte. LaTeX permite această utilizare prin intermediul comenzii `\oldstylenums`:

`\oldstylenums{1234567890}`

1234567890

Unele fonturi nu au cifre în modul text încorporate; pachetul `textcomp` încercă să remedieze asta generând efectiv cifre în modul text în baza fontului curent selectat. Scrieți `\usepackage{textcomp}` în preambul.

`textcomp` vă permite de asemenea să folosiți puncte zecimale, \$, etc., în `\oldstylenums{}`.

O utilizare comună pentru cifrele în modul text este pentru numerele secțiunilor, paragrafelor și paginilor. Acestea pot fi setate să folosească cifre în modul text scriind ceva cod în preambul:

```
\usepackage{textcomp}

% Includeți totul în \AtBeginDocument{}
\AtBeginDocument
{
  % Faceți \section{} să folosească cifre în modul text
  \let\myTheSection\thesection
  \renewcommand{\thesection}{\oldstylenums{\myTheSection}}

  % Faceți \paragraph{} să folosească cifre în modul text
  \let\myTheParagraph\theparagraph
  \renewcommand{\theparagraph}{\oldstylenums{\myTheParagraph}}

  % Faceți ca numerele de pagini să fie în modul text
  \let\myThePage\thepage
  \renewcommand{\thepage}{\oldstylenums{\myThePage}}
}
```

Dacă veți folosi comenzi adiționale pentru secțiuni sau paragrafe, puteți să adaptați codul de mai înainte pentru a le include și pe acestea.

Simboluri și caractere speciale

Cratime și alte liniuțe

LaTeX cunoaște patru tipuri de liniuțe: [cratimă](#) (-), [en dash](#) (–), [em dash](#) (—), sau [minus](#) (−). Puteți accesa trei dintre ele cu numere diferite de liniuțe consecutive. Al patrulea semn nu este de fapt o liniuță, ci semnul matematic minus:

Cratimă:
într-o după-
amiază

En dash:
paginile 13-
-67

Cratimă: într-o după-amiază

En dash: paginile 13–67

Em dash: da — sau nu?

Minus: 0, 1 și −1

Em dash:
da --- sau
nu?

Minus: \$0\$,
\$1\$ și \$-1\$

Denumirile acestor liniițe sunt: '-'(-) cratimă, '--'(-) en-dash, '---'(—) em-dash și '-'(-) minus. Ele au scopuri diferite:

| Input | Output | Scop |
|-------|--------|---------------------------------|
| — | - | separă cuvintele: după-amiaza |
| -- | - | separă numerele de pagini: 1-10 |
| --- | — | semn de punctuație: da — sau nu |

Utilizați macrocomanda `\hyp{}` din pachetul [hyphenat](#) în loc de cratimă dacă vreți să despărțiți cuvinte compuse pe mai multe linii în LaTeX.

Simbolul Euro €

Când scrieți ceva legat de bani azi, aveți nevoie de [semnul Euro](#). Aveți mai multe opțiuni. Dacă fonturile pe care le utilizați au un simbol pentru Euro și vreți să-l folosiți, mai întâi trebuie să includeți pachetul `textcomp` în preambul: `\usepackage{textcomp}`, apoi puteți insera simbolul Euro cu comanda `\texteuro`. Dacă vreți să folosiți versiunea oficială a simbolul Euro, atunci trebuie să includeți pachetul `eurosym` cu opțiunea `official` în preambul: `\usepackage[official]{eurosym}`, apoi puteți să inserați acest simbol cu comanda `\euro`. În cele din urmă, dacă vreți un simbol Euro care să se potrivească cu stilul curent de font (de exemplu, bold, italic, etc.), însă fontul curent nu oferă acest simbol, puteți din nou să includeți pachetul `eurosym`, însă de data aceasta cu altă opțiune: `\usepackage[gen]{eurosym}` și din nou puteți insera simbolul Euro cu comanda `\euro`.

Puncte de suspensie (...)

O secvență de trei puncte este cunoscută sub denumirea de [puncte de suspensie](#), folosită în general pentru a indica un text omis. Pe o mașină de scris, o virgulă sau un punct ia același spațiu ca orice altă literă. La tipărirea

cărților, în tipografia sau în programele de redactare de text, aceste caractere ocupă numai puțin spațiu și sunt așezate foarte aproape de litera precedentă. De aceea, nu puteți introduce punctele de suspensie tipărind pur și simplu trei puncte, întrucât spațierea dintre ele ar fi prea mică. În schimb, există o comandă specială pentru aceste trei puncte — `\ldots`:

Nu așa ... ci așa:\\
New York, Tokyo,
Budapest, \ldots

Nu așa ... ci așa:
New York, Tokyo, Budapest, ...

Șiruri de caractere predefinite

Sunt câteva comenzi foarte simple în LaTeX pentru afișarea unor șiruri de caractere speciale, predefinite:

| Comandă | Exemplu | Descriere |
|----------------------|-----------------------|-----------------------|
| <code>\today</code> | December 12, 2010 | Data curentă |
| <code>\TeX</code> | \TeX | Programul tău favorit |
| <code>\LaTeX</code> | \LaTeX | Numele jocului |
| <code>\LaTeXe</code> | $\LaTeX 2_{\epsilon}$ | Încarnarea curentă |

Alte simboluri

Latex dispune de foarte multe simboluri. Majoritatea acestora sunt în domeniul matematic, iar capitolele ulterioare vă vor spune cum să le accesați. Pentru simbolurile uzuale, utilizați următoarele comenzi:

| Comandă | Simbol | Comandă | Simbol |
|--------------------------------|--------|----------------------------------|--------------|
| <code>\%</code> | % | <code>\#</code> | # |
| <code>\\$</code> | \$ | <code>\&</code> | & |
| <code>\{</code> | { | <code>\}</code> | } |
| <code>_</code> | - | <code>\S</code> | § |
| <code>\P</code> | ¶ | <code>\textbar</code> | |
| <code>\dag</code> | † | <code>\ddag</code> | ‡ |
| <code>\slash</code> | / | <code>\textbackslash</code> | \ |
| <code>\textless</code> | < | <code>\textgreater</code> | > |
| <code>\textendash</code> | – | <code>\textemdash</code> | — |
| <code>\textregistered</code> | ® | <code>\texttrademark</code> | ™ |
| <code>\textquestiondown</code> | ¿ | <code>\textexclamdown</code> | ¡ |
| <code>\textcircled{a}</code> | Ⓐ | <code>a</code> | ^a |
| <code>\copyright</code> | © | <code>\pounds</code> | £ |

Nemenționată în tabelul de mai sus, tilda (~) se folosește în codul LaTeX pentru a produce [spațiu non-separabil](#). Pentru a afișa semnul tilda, scrieți-l în cadrul unui [text verbatim](#) sau scrieți `\~{}`.

Deșigur, puteți folosi și fontul Postscript ZipfDingbats din pachetul [pifont](#). Scrieți în preambul: `\usepackage{pifont}`, iar apoi folosiți comanda `\ding{number}` pentru a afișa simbolul corespunzător. Iată un tabel cu simbolurile disponibile:

| | | | | | | | | | | | | | | | |
|-----|---|-----|---|-----|---|-----|---|-----|---|-----|---|-----|---|-----|--|
| 32 | | 33 | ✂ | 34 | ✂ | 35 | ✂ | 36 | ✂ | 37 | ☐ | 38 | ⊙ | 39 | |
| 40 | → | 41 | ⊞ | 42 | ☛ | 43 | ☛ | 44 | ☛ | 45 | ↗ | 46 | ↘ | 47 | |
| 48 | ↙ | 49 | ↖ | 50 | ↗ | 51 | ✓ | 52 | ✓ | 53 | ✕ | 54 | ✕ | 55 | |
| 56 | ✕ | 57 | ✕ | 58 | ✕ | 59 | ✕ | 60 | ✕ | 61 | † | 62 | † | 63 | |
| 64 | ✕ | 65 | ✕ | 66 | ✕ | 67 | ✕ | 68 | ✕ | 69 | ✕ | 70 | ✕ | 71 | |
| 72 | ★ | 73 | ☆ | 74 | ☉ | 75 | ☆ | 76 | ★ | 77 | ★ | 78 | ★ | 79 | |
| 80 | ☆ | 81 | ✱ | 82 | ✱ | 83 | ✱ | 84 | ✱ | 85 | ✱ | 86 | ✱ | 87 | |
| 88 | ✱ | 89 | ✱ | 90 | ✱ | 91 | ✱ | 92 | ✱ | 93 | ✱ | 94 | ✱ | 95 | |
| 96 | ☼ | 97 | ☼ | 98 | ☼ | 99 | ✱ | 100 | ✱ | 101 | ✱ | 102 | ✱ | 103 | |
| 104 | ✱ | 105 | ✱ | 106 | ✱ | 107 | ✱ | 108 | ● | 109 | ○ | 110 | ■ | 111 | |
| 112 | □ | 113 | □ | 114 | □ | 115 | ▲ | 116 | ▼ | 117 | ◆ | 118 | ◇ | 119 | |
| 120 | | 121 | | 122 | | 123 | • | 124 | • | 125 | “ | 126 | ” | | |
| | | 161 | ☺ | 162 | ☹ | 163 | ☹ | 164 | ♥ | 165 | ♠ | 166 | ♣ | 167 | |
| 168 | ♣ | 169 | ♦ | 170 | ♥ | 171 | ♠ | 172 | ① | 173 | ② | 174 | ③ | 175 | |
| 176 | ⑤ | 177 | ⑥ | 178 | ⑦ | 179 | ⑧ | 180 | ⑨ | 181 | ⑩ | 182 | ① | 183 | |
| 184 | ② | 185 | ④ | 186 | ⑤ | 187 | ⑥ | 188 | ⑦ | 189 | ⑧ | 190 | ⑨ | 191 | |
| 192 | ① | 193 | ② | 194 | ③ | 195 | ④ | 196 | ⑤ | 197 | ⑥ | 198 | ⑦ | 199 | |
| 200 | ⑨ | 201 | ⑩ | 202 | ① | 203 | ② | 204 | ③ | 205 | ④ | 206 | ⑤ | 207 | |
| 208 | ⑦ | 209 | ⑧ | 210 | ⑨ | 211 | ⑩ | 212 | → | 213 | → | 214 | ↔ | 215 | |
| 216 | ↖ | 217 | → | 218 | ↗ | 219 | → | 220 | → | 221 | → | 222 | → | 223 | |
| 224 | ⇒ | 225 | ⇒ | 226 | ➤ | 227 | ➤ | 228 | ➤ | 229 | ➤ | 230 | ➤ | 231 | |
| 232 | ➤ | 233 | ⇨ | 234 | ⇨ | 235 | ⇨ | 236 | ⇨ | 237 | ⇨ | 238 | ⇨ | 239 | |
| | | 241 | ⇨ | 242 | ↻ | 243 | ⇒ | 244 | ↘ | 245 | ⇒ | 246 | ↗ | 247 | |
| 248 | ⇒ | 249 | ↗ | 250 | → | 251 | ⇒ | 252 | ⇒ | 253 | ⇒ | 254 | ⇒ | | |

Formatarea paragrafelor

Modificarea formatării paragrafelor nu este deseori necesară, mai ales în cazul publicațiilor academice. Totuși, este bine de știut cum se formatează textul în elemente flotante, sau alte tipuri mai exotice de documente.

Alinierea paragrafelor

Paragrafele din Latex sunt de obicei aliniate stânga-dreapta (*justified*). Dacă vreți, dintr-un motiv anume, să modificați alinierea implicită a unui paragraf, aveți la dispoziție trei medii și comenzi Latex echivalente.

| Aliniere | Mediu | Comandă |
|-------------------|------------------------|---------------------------|
| Aliniat la stânga | <code>flushleft</code> | <code>\raggedright</code> |

Aliniat la dreapta `flushright` `\raggedleft`

Centrat `center` `\centering`

Tot textul dintre `\begin` și `\end` în mediul specificat va fi aliniat în mod corespunzător. Comenzile listate mai sus sunt menite a fi utilizate în alte medii. Spre exemplu, pentru coloanele `p` (paragraf) în mediul `tabular`.

Indentarea paragrafelor

Implicit, primul paragraf după un antet urmează standardul anglo-american fără indentare. Indentarea paragrafelor următoare este determinată de un parametru denumit `\parindent`. Lungimea implicită pe care o are această constantă este setată de clasa documentului folosit. Puteți modifica indentarea implicită a paragrafelor cu ajutorul comenzii `\setlength`:

```
\setlength{\parindent}{1cm}
```

Asta va seta indentarea implicită a paragrafelor la 1cm.

Aveți grijă, totuși, dacă decideți să setați indentarea la zero, întrucât asta înseamnă că veți avea nevoie de un spațiu vertical între paragrafe pentru a le delimita. Spațiul dintre paragrafe este controlat de parametrul `\parskip`, care poate fi modificat la fel ca mai devreme. Totuși, acest parametru este utilizat și în altă parte, cum ar fi listele, ceea ce înseamnă că puteți face ca diverse părți ale documentului să arate neîngrijit schimbând aceste setări. Dacă nu vreți să aveți indentare, dar cu un spațiu între paragrafe, folosiți pachetul `parskip`, care face asta, dar și ajustări ale spațierii listelor și altor structuri care folosesc spațierea paragrafelor, astfel încât acestea să nu fie prea diferite. Adăugați la preambul:

```
\usepackage{parskip}
```

Pentru a indenta liniile succesive ale unui paragraf, utilizați comanda TeX `\hangindent`. (Comportamentul implicit constă în aplicarea indentării liniilor după prima linie, care se poate schimba cu comanda `\hangafter`.) Iată un exemplu:

```
\hangindent=0.7cm Acest paragraf are o indentare suplimentară la stânga (vizibilă începând cu linia a doua a paragrafului).
```

Comenzile TeX `\leftskip` și `\rightskip` adaugă spațiu adițional la stânga și la dreapta fiecărei linii, permițând formatarea paragrafelor următoare astfel

încât să difere de marginile generale ale documentului. Acest spațiu se adaugă la indentarea adăugată de `\parindent` și `\hangindent`.

Pentru a schimba indentarea ultimei linii dintr-un paragraf, folosiți comanda TeX `\parfillskip`.

Spațiul alb din LaTeX poate fi făcut flexibil (ceea ce Lamport numește lungimi flexibile — *rubber lengths*). Asta înseamnă că valori de genul `\parskip` pot avea o dimensiune implicită plus o anume extindere minus o anume contracție. Acest lucru este util pe paginile din documente complexe, în care nu orice pagină poate avea un număr exact de linii, astfel că este utilă o anumită ajustare a spațiului vertical. Specificați asta printr-o comandă `\setlength` cum este:

```
\setlength{\parskip}{1cm plus4mm minus3mm}
```

Spațierea liniilor

Pentru a schimba spațiile dintre linii în tot documentul, utilizați comanda `\linespread`, explicată în secțiunea [Spațiere](#) din capitolul [LaTeX personalizat](#).

Pentru a modifica spațierea liniilor în medii specifice, urmați pașii următori:

1. Adăugați `\usepackage{setspace}` în preambulul documentului.
2. Acest pachet oferă posibilitatea de a folosi următoarele medii în document:
 - `doubleSPACE` - toate liniile sunt separate de spații duble.
 - `onehalfspace` - spațierea liniilor este la 1,5 față de spațierea normală.
 - `singlespace` - spațiere normală a liniilor.

După ce declarați pachetul în preambul, utilizarea uneia dintre comenzile `\singlespacing`, `\doublespacing`, sau `\onehalfspacing` va specifica spațierea liniilor pentru toate secțiunile și paragrafele până la folosirea altei comenzi similare.

Vezi secțiunea [Liste personalizate](#) de mai jos pentru a afla cum puteți schimba spațierea liniilor în liste.

Paragrafe speciale

Vezi secțiunea [Paragrafe speciale](#) din capitolul [Formatare avansată](#).

Structuri de liste

Listele apar deseori în documente, mai ales în cele academice, după cum scopul lor este adesea să prezinte informații într-un mod clar și concis. Structurile de liste din LaTeX sunt pur și simplu medii de trei tipuri: `itemize`, `enumerate` și `description`.

Toate listele urmează formatul de bază:

```
\begin{tipul_listei}

  \item Primul item
  \item Al doilea item
  \item Al treilea etc \ldots

\end{tipul_listei}
```

Toate aceste trei tipuri de liste pot avea mai multe paragrafe per item: scrieți pur și simplu paragrafele următoare așa cum le-ați scrie în mod normal, cu o linie liberă între ele. Atâta vreme cât sunt conținute în mediul de listare, vor fi indentate automat pentru a se alinia item-ului corespunzător.

Itemize

Acest mediu este pentru o listă standard cu marcatori:

| | | |
|---|--|---|
| <pre>\begin{itemize} \item Primul item \item Al doilea item \item Al treilea etc \ldots \end{itemize}</pre> | | <ul style="list-style-type: none">• Primul item• Al doilea item• Al treilea etc ... |
|---|--|---|

Enumerate

Mediul `enumerate` este pentru listele numerotate, în care, în mod implicit, fiecare item este numerotat secvențial.

```

\begin{enumerate}
  \item Primul item
  \item Al doilea item
  \item Al treilea etc \ldots
\end{enumerate}

```

1. Primul item
2. Al doilea item
3. Al treilea etc ...

Description

Mediul `description` este puțin diferit. Puteți specifica eticheta pentru item ca argument opțional (deși opțional, ar părea ciudat să nu-l includeți!). Este ideal pentru o serie de definiții, ca în cazul unui glosar.

```

\begin{description}
  \item[Primul:]
  Primul item
  \item[Al doilea:]
  Al doilea item
  \item[Al treilea:]
  Al treilea etc \ldots
\end{description}

```

Primul: Primul item

Al doilea: Al doilea item

Al treilea: Al treilea etc ...

Câteodată s-ar putea să doriți o descriere în care textul să înceapă pe o linie nouă. Acest lucru nu se poate face pur și simplu cu `\`. Trucul constă în utilizarea comenzii `\hfill`.

```

\begin{description}
  \item[Primul] \hfill \
  Primul item
  \item[Al doilea] \hfill \
  Al doilea item
  \item[Al treilea] \hfill \
  Al treilea etc \ldots
\end{description}

```

Primul
Primul item

Al doilea
Al doilea item

Al treilea
Al treilea etc ...

Liste compacte

După cum probabil ați observat, în clasele standard de documente LaTeX, spațierea verticală dintre itemi, ca și spațiul de deasupra și de după liste, sunt

mai mari decât la paragrafe: poate părea nepotrivit dacă descrierile sunt prea scurte. Dacă vrei liste mai compacte, folosește pachetul `mdwlist` (inclusiv în grupul `mdwtools`), care oferă versiuni compacte, "stelate" ale mediilor prezentate mai devreme, cu alte cuvinte, `itemize*`, `enumerate*` și `description*`. Funcționează exact la fel, însă outputul este mai compact.

Ca o alternativă, puteți utiliza clasa de documente `memoir`, cu comanda `\tightlists`.

Liste imbricate

Latex vă va permite să inserați o listă într-un item dintr-o altă listă (până la un nivel de imbricare de 4 -- dacă vă trebuie un nivel de imbricare mai mare, folosește pachetul `easylist`). Pur și simplu începeți mediul corespunzător în locul dorit în lista curentă. Latex va genera automat tipul de numerotare și indentarea corespunzătoare.

```
\begin{enumerate}
\item Primul item
\begin{enumerate}
\item item imbricat 1
\item item imbricat 2
\end{enumerate}
\item Al doilea item
\item Al treilea etc \ldots
\end{enumerate}
```

1. Primul item
 - (a) item imbricat 1
 - (b) item imbricat 2
2. Al doilea item
3. Al treilea etc ...

Liste personalizate

Vezi secțiunea [Liste personalizate](#) din capitolul [Formatare avansată](#).

Liste *inline*

Vezi secțiunea [Liste inline](#) din capitolul [Formatare avansată](#).

Note de subsol

Notele de subsol sunt un mod foarte util de a oferi informații suplimentare cititorului. De obicei, reprezintă informații neesențiale care pot fi așezate la sfârșitul paginii. Asta păstrează corpul principal al textului concis.

Notele de subsol sunt ușor de folosit. Comanda de care aveți nevoie este: `\footnote{text}`. Nu lăsați niciun spațiu între comandă și cuvântul în dreptul căruia vreți să apară marcatorul notei de subsol, altfel LaTeX va procesa acel spațiu, iar rezultatul nu va arăta după cum era de așteptat.

Crearea unei
note de subsol e
ușor de
făcut. `\footnote{`
Un exemplu de
notă de subsol. `}`

Crearea unei note de subsol e ușor de fă

¹Un exemplu de notă de subsol.

Latex se va ocupa de redactarea notelor de subsol la sfârșitul paginii. Fiecare notă de subsol este numerotată secvențial — proces care, după cum probabil ați ghicit deja, este realizat automat.

Puteți modifica marcajul notelor de subsol. Implicit, acestea sunt numerotate secvențial (cu cifre arabe). Totuși, fără a intra în prea multe detalii legate de mecanismele interne ale sistemului LaTeX acum, puteți schimba acest tip de marcaj cu ajutorul următoarei comenzi (care trebuie plasată la începutul documentului, sau măcar înainte de prima comandă ce generează o notă de subsol).

| | |
|---|--|
| <code>\renewcommand{\thefootnote}{\arabic{footnote}}</code> | Cifre arabe, cum sunt 1, 2, 3... |
| <code>\renewcommand{\thefootnote}{\roman{footnote}}</code> | Cifre romane (mici), ca i, ii, iii... |
| <code>\renewcommand{\thefootnote}{\Roman{footnote}}</code> | Cifre romane (mari), precum I, II, III... |
| <code>\renewcommand{\thefootnote}{\alph{footnote}}</code> | Litere mici, cum sunt a, b, c... |
| <code>\renewcommand{\thefootnote}{\Alph{footnote}}</code> | Litere mari, cum ar fi A, B, C... |
| <code>\renewcommand{\thefootnote}{\fnsymbol{footnote}}</code> | O secvență de 9 simboluri (încercați să vedeți cum merge!) |

Pachetul [footmisc](#) oferă multe posibilități de modificare a aspectului notelor de subsol. Acesta poate oferi, spre exemplu, un font diferit pentru notele de subsol.

Probleme și soluții uzuale

- Notele de subsol din păcate nu merg cu tabelele, întrucât se consideră a fi o practică proastă. Puteți depăși această limitare prin mai multe tehnici: puteți utiliza comanda `\footnotemark[123]` în tabel și `\footnotetext[123]{Salut, lume!}` undeva pe pagină. Sau, puteți adăuga `\usepackage{footnote}` și `\makesavenoteenv{tabular}` la preambul, iar apoi puneți mediul `table` într-un mediu `savenotes`, după `\begin{savenotes}`. Notați că ultima variantă nu merge cu pachetele `color` și `colortbl`. Vezi [această pagină FAQ](#) pentru alte abordări.
- Notele de subsol nu merg într-un mediu `minipage` (de fapt, mai multe medii nu oferă suport pentru notele de subsol). Comanda `\makesavenoteenv{nume_mediu}` a pachetului `footnote` ar putea rezolva această problemă pentru cele mai multe medii. Mediul `minipage` include propriile note de subsol, independent de cele ale documentului. Pachetul `mpfnmark` permite o mai mare flexibilitate în gestionarea acestor două seturi de note de subsol.
- Dacă textul din nota de subsol este foarte lung, LaTeX poate împărți nota de subsol pe mai multe pagini. Puteți împiedica asta mărin­d penalitatea din LaTeX pentru o asemenea operație. Pentru a realiza acest lucru, inserați linia următoare în preambulul documentului:

```
\interfootnotelinepenalty=10000
```

- Pentru a face mai multe referințe la aceeași notă de subsol, puteți folosi următoarea sintaxă:

Textul ce are o notă de subsol `\footnote{Aceasta este nota de subsol}` arată așa. Mai târziu, textul ce face o referință la aceeași notă de subsol `\footnotemark[\value{footnote}]` utilizează cealaltă comandă. Dacă vă trebuie suport din pachetul `hyperref`, folosiți în schimb: Textul ce are o notă de subsol `\footnote{Aceasta este nota de subsol}\addtocounter{footnote}{-1}\addtocounter{Hfootnote}{-1}` arată așa. Mai târziu, textul ce face o referință la aceeași notă de subsol `\footnotemark` utilizează cealaltă comandă.

Notați că aceste abordări nu merg dacă sunt alte note de subsol între prima referință și oricare dintre "duplicate".

Note de margine

ului de editare pentru a
notă de margine, folosiți
Pentru formatul de pa- text ce va apare
at în marginea dreaptă, în notă
ormatul de pagină față-
urginea exterioară, iar în
i așezat în marginea cea
o notă de margine

Notele de margine sunt utile în timpul procesului de editare pentru a schimba comentarii între autori. Pentru a insera o notă de margine, folosiți comanda `\marginpar{text ce va apare în notă}`. Pentru formatul de pagină (*layout*) *one-sided* (simplex), textul va fi așezat în marginea dreaptă, începând cu linia unde nota este definită. Pentru formatul de pagină față-verso (*two-sided*) (duplex), textul va fi plasat în marginea exterioară, iar în cazul formatului cu două coloane (*two-column*) va fi așezat în marginea cea mai apropiată.

Pentru a schimba marginea implicită, folosiți comanda `\reversemarginpar` și atunci notele de margine vor fi atunci plasate în partea opusă, care ar trebui să fie marginea interioară pentru formatul de pagină față-verso.

Dacă textul transmis ca argument al comenzii `marginpar` depinde de marginea în care este așezat (să zicem că include o săgeată îndreptată către text sau se referă la o direcție, ca în "după cum se vede la stânga..."), puteți utiliza `\marginpar[left text]{text așezat la dreapta}` pentru a specifica variantele.

Pentru a insera o notă de margine într-o zonă pe care `\marginpar` nu o poate gestiona, cum ar fi notele de subsol sau ecuațiile, folosiți pachetul [marginnote](#).

Vezi și pachetul [mparhack](#).

Această pagină folosește materiale din cartea lui Andy Roberts, [Getting to grips with Latex](#), cu permisiunea autorului.

Formatare avansată

Paragrafe speciale

Dacă ați citit cea mai mare parte sau toate capitolele de până acum, cel mai probabil ați întâlnit formatele din paragrafele următoare. Deși le-ați mai văzut, are sens să le re-introducem aici, pentru a completa informațiile din acest domeniu.

Text verbatim

Sunt mai multe moduri în care puteți introduce text care nu va fi interpretat de compilator. Dacă folosiți mediul verbatim, tot ce este între comenzile `begin` și `end` este procesat așa cum este. Toate spațiile și liniile noi sunt reproduse așa cum sunt, iar textul este afișat într-un font corespunzător de mărime fixă. Orice comandă LaTeX va fi ignorată și tratată ca simplu text. Această situație este ideală pentru codul sursă. Acest mediu a fost folosit într-un exemplu în capitolul dinainte. Iată un exemplu:

```
\begin{verbatim}
Mediul
verbatim
  pur și simplu
reproduce
orice
  caracter
introdus,
inclusiv toate
  spațiile!
\end{verbatim}
}
```

```
Mediul verbatim
  pur și simplu reproduce or:
  caracter introdus,
inclusiv toate spațiile!
```

Notă: îndată ce am intrat în mediul verbatim, singura comandă ce va fi recunoscută este `\end{verbatim}`. Oricare alta va fi afișată, literă cu literă (*verbatim*)! Dacă asta e o problemă, folosiți în schimb pachetul `alltt`, ce oferă un mediu cu același nume:

```
\begin{alltt}
Verbatim
extins cu
abilitatea
de a utiliza
comenzi
normale. De
```

```
Verbatim extins cu abilitatea
de a utiliza comenzi normale.
puteți evidenția cuvintele în
acest mediu, spre exemplu.
```

```
aceea,  
puteți  
\emph{evidenția}  
cuvintele în  
acest mediu,  
spre  
exemplu.  
\end{alltt}
```

Amintiți-vă să adăugați `\usepackage{alltt}` în preambulul documentului ca să-l puteți folosi! În mediul `alltt`, puteți folosi comanda `\normalfont` pentru a reveni la fontul normal. Pentru a scrie ecuații în mediul `alltt`, puteți folosi `\(` și `\)` pentru a le include, în locul caracterului `$`.

Când folosiți `\textbf{}` în mediul `alltt`, notați că fontul standard nu are tipul TT aldin. Fonturile `txtt` au tipul aldin: adăugați pur și simplu `\renewcommand{\ttdefault}{txtt}` după `\usepackage{alltt}`.

Dacă vreți numai să introduceți o expresie scurtă în modul `verbatim`, nu trebuie să folosiți mediul `verbatim` ca atare, ci doar comanda `\verb`:

```
\verb+my text+
```

Primul caracter ce urmează `\verb` este delimitatorul: aici am folosit "+", însă puteți folosi orice caracter, cu excepția "*" și a spațiului; `\verb` va afișa literă cu literă tot textul după primul delimitator și până la următorul. Spre exemplu, codul:

```
\verb|\textbf{Salut, prietene!}|
```

va afișa `\textbf{Salut, prietene!}`, ignorând efectul pe care `\textbf` ar trebui să-l aibă asupra textului.

Pentru mai mult control asupra formatării, totuși, puteți încerca pachetul [fancyvrb](#), care oferă un mediu Verbatim (notați litera mare "V"). Acesta permite trasarea unei linii care să încadreze textul verbatim, modificarea mărimii fontului și chiar efecte tipografice în mediul Verbatim. Poate fi folosit de asemenea în conjuncție cu pachetul [fancybox](#) și poate adăuga referințe la numere de linii (utile pentru grupuri de date sau de cod), putând chiar să includă fișiere externe întregi.

Tipărirea de URL-uri

Oricare dintre pachetele `hyperref` și `url` oferă comanda `\url`, care afișează în mod corespunzător URL-uri, spre exemplu:

```
Mergi la \url{http://www.uni.edu/~numele-meu/cel-mai-bun-website.html}
pentru a vedea site-ul meu Web.
```

va afișa acest URL exact așa cum este scris (similar cu comanda `\verb`), însă comanda `\url` împarte de asemenea în silabe fără a pune o cratimă (în dreptul semnelor de punctuație). Aceasta a fost proiectată pentru URL-uri Web, astfel că le înțelege sintaxa și nu va împărți niciodată în silabe un cuvânt fără semne de punctuație, ci doar în dreptul caracterelor slash și a punctului. Țineți minte, totuși, că spațiile sunt interzise în URL-uri, astfel încât folosirea de spații în argumente `\url` ar putea să nu dea rezultatul dorit, la fel ca utilizarea de caractere invalide pentru URL-uri.

Când utilizați această comandă cu ajutorul pachetului `hyperref`, puteți accesa URL-ul în documentul PDF rezultat, în vreme ce cu pachetul `url`, argumentul comenzii `\url` nu va fi legat de adresa de Web. De asemenea, când folosiți pachetul `hyperref`, pentru a înlătura chenarul (linia dreptunghiulară) din jurul URL-ului, inserați `pdfborder = {0 0 0 0}` în comanda `\hypersetup{}` sau includeți pachetul cu opțiunea `pdfborder = {0 0 0 0}`:

```
\usepackage[pdfborder={0 0 0 0}]{hyperref}
```

Mediul `listing`

Este o extensie a mediului `verbatim` pusă la dispoziție de pachetul `moreverb`. Funcționalitatea suplimentară pe care o oferă constă în adăugarea de numere de linii în text cu ajutorul comenzii:

```
\begin{listing}[pas]{prima linie}. Argumentul obligatoriu prima linie specifică pe ce linie ar trebui să înceapă numărătoarea. Argumentul opțional pas este diferența dintre numerele liniilor consecutive considerate (valoarea implicită este 1, ceea ce înseamnă că fiecare linie va fi numărată).
```

Pentru a utiliza acest mediu, amintiți-vă să adăugați `\usepackage{moreverb}` la preambulul documentului.

Pachetul `listings`

Cu ajutorul pachetului `listings` puteți adăuga text neformatat ca în cazul comenzii `\begin{verbatim}`, însă scopul său principal este includerea de cod sursă din orice limbaj de programare în document. Dacă vreți să includeți pseudocod sau algoritmi, ar fi bine să vă uitați și peste capitolul [Algoritmi și pseudocod](#).

Pentru a utiliza pachetul, trebuie să-l încărcați în preambul:

```
\usepackage{listings}
```

Pachetul `listings` suportă evidențierea cuvintelor cheie, a tipurilor de date, etc. din majoritatea limbajelor de programare și este foarte flexibil. Dacă vreți doar să scrieți cod în document, pachetul vă pune la dispoziție mediul `lstlisting`:

```
\begin{lstlisting}
  Scrieți codul aici...
\end{lstlisting}
```

O altă posibilitate, care este foarte utilă dacă ați creat un program ce cuprinde mai multe fișiere și încă îl mai editați, este să importați cod direct din fișierul sursă. În felul acesta, dacă modificați sursa, trebuie doar să recompilați codul LaTeX și documentul final va fi actualizat. Comanda este:

```
\lstinputlisting{fișier_sursă.py}
```

În acest exemplu avem un fișier sursă Python, însă nu contează: puteți include orice fișier, însă trebuie să scrieți numele complet al fișierului (cu tot cu extensie). Acesta va fi considerat text simplu și va fi evidențiat potrivit setărilor curente, ceea ce înseamnă că nu recunoaște singur limbajul de programare. Puteți specifica limbajul când includeți fișierul în felul următor:

```
\lstinputlisting[language=Python]{fișier_sursă.py}
```

Acest pachet oferă suport pentru următoarele limbaje de programare:

| | | |
|------|--------|--------|
| ABAP | IDL | Plasm |
| ACSL | inform | POV |
| Ada | Java | Prolog |

| | | |
|------------------------|--------------------------|----------|
| Algol | JVMIS | Promela |
| Ant | ksh | Python |
| Assembler ² | Lisp | R |
| Awk | Logo | Reduce |
| bash | make | Rexx |
| Basic ² | Mathematica ¹ | RSL |
| C | Matlab | Ruby |
| C++ | Mercury | S |
| Caml | MetaPost | SAS |
| Clean | Miranda | Scilab |
| Cobol | Mizar | sh |
| Comal | ML | SHELXL |
| csh | Modula-2 | Simula |
| Delphi | MuPAD | SQL |
| Eiffel | NASTRAN | tcl |
| Elan | Oberon-2 | TeX |
| erlang | OCL | VBScript |
| Euphoria | Octave | Verilog |
| Fortran | Oz | VHDL |
| GCL | Pascal | VRML |
| Gnuplot | Perl | XML |
| Haskell | PHP | XSLT |
| HTML | PL/I | |

În cazul unora dintre acestea, se oferă suport pentru mai multe dialecte. Pentru mai multe informații, consultați documentația inclusă în pachet (ar trebui să aibă numele *listings.dvi* sau *listings.pdf*).

Note

1 — Suportă codul din Mathematica numai dacă scrieți în plain text format. Nu puteți include fișiere *.NB cu `\lstinputlisting{...}` ca în cazul oricărui alt limbaj de programare, însă Mathematica poate exporta sub formă de sursă LaTeX stilizată.

2 — Este obligatoriu să specificați dialectul pentru unele limbaje (spre exemplu, `language={ [x86masm]Assembler }`).

Puteți modifica mai mulți parametri ce influențează modul de afișare a codului. Puteți pune codul următor oriunde în document (nu contează dacă este înainte sau după `\begin{document}`), după cum vă este mai la îndemână. Comentariile explică fiecare comandă în parte.

```

\lstset{ %
language=Octave, % alegeți limbajul pentru cod
basicstyle=\footnotesize, % mărimea fonturilor utilizate
pentru cod
numbers=left, % unde să scrie numerele de
linii
numberstyle=\footnotesize, % mărimea fonturilor folosite
pentru numerele de linii
stepnumber=2, % diferența dintre numerele de
linii consecutive % dacă este 1, fiecare linie
va fi numerotată
numbersep=5pt, % cât de departe sunt numerele
de linii față de cod culoarea
backgroundcolor=\color{white}, % fundalului; vă trebuie
\usepackage{color} în preambul
showspaces=false, % evidențiază spațiile cu
underscore % subliniază spațiile din
showstringspaces=false, % șirurile de caractere
showtabs=false, % arată tab-urile din șirurile
de caractere cu underscore
frame=single, % adaugă un cadru (chenar) în
jurul codului
tabsize=2, % setează dimensiunea
implicită pentru tab la 2 spații
captionpos=b, % setează poziția titlurilor
sub figuri sau tabele
breaklines=true, % setează separarea automată a
liniilor
breakatwhitespace=false, % se trece la linia nouă dacă
sunt spații libere?
title=\lstname, % arată numele fișierelor
include cu \lstinputlisting; % încearcă să scrie o notă sau
legendă în loc de titlu
escapeinside={\%*}{*}, % dacă vreți să adăugați un
comentariu în cod
morekeywords={*,...} % dacă vreți să mai adăugați
cuvinte cheie la setul existent
}

```

Linia `escapeinside` necesită o explicație suplimentară. Vă trebuie dacă vreți să adăugați ceva text în cod, care să nu fie afișat. Notați că, implicit,

comentariile ce țin de limbajul de programare vor fi afișate; comanda `escapeinside={A}{B}` definește comentarii doar pentru pachetul `listings`. Întregul cod dintre șirurile de caractere "A" și "B" va fi ignorat. În exemplul anterior, comentariile pentru *Octave* încep cu `%`, și urmează a fi afișate în documentul final dacă nu încep cu `%*`, însă trebuie să vă amintiți să "închideți" comentariul cu altă steluță `"*"`.

Dacă adăugați paragraful dinainte, puteți folosi comanda următoare pentru a modifica setările codului:

```
\lstset{language=C,caption=Text
descriptiv,label=EtichetăDescriptivă}
```

Găsiți mai multe informații în [PDF-ul](#) lui Carsten Heinz și Brooks Moses.

Detalii și documentație pentru pachetul `listings` puteți găsi pe site-ul [CTAN](#).

Pentru un scurt exemplu, vezi [codul de aici](#).

Comentarii ce se întind pe mai multe linii

După cum ați văzut, singurul mod în care LaTeX vă permite să adăugați comentarii este folosind caracterul special `%`, care va transforma în comentariu restul liniei. Această abordare necesită mult timp pentru implementare dacă vreți să inserați comentarii foarte mari sau vreți să comentați o parte a documentului, la care veți reveni mai târziu. Cu ajutorul pachetului `verbatim`, scris în preambul ca de obicei:

```
\usepackage{verbatim}
```

puteți folosi un mediu numit `comment`, care va comenta tot ce cuprinde. Iată un exemplu:

```
Acesta este încă un
exemplu
\begin{comment}
destul de prostesc,
dar util
\end{comment}
de includere a
```

```
Acesta este încă un exemplu de
includere a comentariilor în
document.
```

comentariilor în document.

Notați că asta nu va merge în medii complexe, cum este `math`, de exemplu. Poate vă întrebați, de ce ar trebui să includ un pachet numit `verbatim` pentru a avea posibilitatea de a adăuga comentarii? Răspunsul este imediat: textul comentat nu este interpretat de compilator la fel ca textul `verbatim`, diferența fiind că textul `verbatim` este introdus în document, iar comentariul nu.

Ca o alternativă, puteți defini o comandă `\comentariu{}`, adăugând în preambulul documentului:

```
\newcommand{\comentariu}[1]{}
```

Apoi, pentru a comenta un text, utilizați această comandă:

```
\comentariu{ Acesta este un comentariu lung și care poate să  
se întindă pe mai multe linii, etc. }
```

Pachetul `rotating`

Pachetul `rotating` vă oferă posibilitatea de a roti orice obiect cu un unghi arbitrar. Îndată ce l-ați încărcat cu comanda standard în preambul:

```
\usepackage{rotating}
```

puteți folosi trei medii noi:

```
\begin{sideways}
```

rotește întregul argument cu 90 de grade în sens trigonometric (invers acelor de ceasornic). Mai mult:

```
\begin{turn}{30}
```

va roti argumentul cu 30 de grade. Puteți da orice unghi ca argument, fie că este pozitiv sau negativ. Acesta va lăsa spațiul necesar pentru a evita orice suprapunere de text.

```
\begin{rotate}{30}
```

este la fel ca `turn`, însă nu adaugă spațiu suplimentar.

Dacă vrei ca un element flotant să fie întors cu 90 de grade în sens trigonometric astfel încât să se rotească și titlul sau legenda în mod corespunzător, poți folosi

```
\begin{sidewaysfigure}
```

sau

```
\begin{sidewaystable}
```

Notați, totuși, că figurile sau tabelele rezultate vor fi așezate pe o pagină separată.

Notă: Multe programe de vizualizare pentru DVI nu permit afișarea textului și tabelor rotite. Textul va fi afișat normal. Trebuie să converțiți fișierul DVI în PDF și să-l vizualizați cu un program pentru PDF-uri pentru a vedea efectele de rotire. Aveți grijă totuși că tipărirea acelor fișiere PDF ar putea roti pagina respectivă *din nou în aceeași direcție* în unele situații. Acest comportament poate fi influențat de configurările programului de conversie `dvi2pdf`. Uitați-vă peste manual pentru mai multe informații.

Acest pachet este inclus în grupul 'texlive-latex-recommended' din [Ubuntu](#).

Citarea de text

LaTeX oferă mai multe medii pentru citarea de text, cu mici diferențe și menite pentru tipuri diferite de citate. Toate sunt indentate la una dintre margini, la care puteți adăuga propriile marcaje pentru citate dacă vreți. Mediile oferite sunt:

`quote`

pentru un citat scurt, sau o serie de citate mici, separate de linii libere.

`quotation`

pentru citate mai lungi, de un paragraf sau mai multe, întrucât indentează prima linie a fiecărui paragraf.

`verse`

pentru citate în care liniile libere sunt importante, ca în cazul poeziei. În cadrul acestui mediu, noi strofe sunt create cu o linie liberă, iar liniile noi dintr-o strofă sunt indicate cu ajutorul comenzii pentru o linie nouă, `\`. Dacă o linie ia mai mult spațiu pe o pagină decât

mărimea normală a unei linii, atunci toate liniile următoare sunt indentate până când sunt separate explicit cu comanda `\`.

Rezumate

În publicațiile științifice, se obișnuiește să se înceapă cu un rezumat care oferă cititorului o privire de ansamblu asupra conținutului. LaTeX pune la dispoziție mediul abstract în acest scop. Poate fi accesat în clasele de documente articule și report; nu este disponibil în clasa book, însă este foarte ușor să creezi versiunea proprie dacă vrei.

Liste personalizate

Personalizarea LaTeX depășește domeniul începătorilor. Deși nu este greu de realizat, deoarece începătorii sunt deja încărcăți cu o mulțime de comenzi și medii pe care trebuie să le învețe, trecerea la subiecte mai avansate poate genera confuzie.

Totuși, de vreme ce acest tutorial privește formatarea, voi prezenta totuși un scurt ghid legat de personalizarea listelor. Puteți trece peste această secțiune!

Modificarea spațierii liniilor în liste

În mediile de listare, puteți redefini unele variabile de lungime în LaTeX, spre exemplu:

```
\begin{itemize}
  \setlength{\itemsep}{1pt}
  \setlength{\parskip}{0pt}
  \setlength{\parsep}{0pt}
  \item Primul item
  \item Al doilea item
\end{itemize}
```

Altfel, pentru a crea un aspect unificat în document, puteți defini propriul mediu de enumerare:

```
\newenvironment{enumerare}
{\begin{enumerate}
  \setlength{\itemsep}{1pt}
  \setlength{\parskip}{0pt}
```

```
\setlength{\parsep}{0pt}  
\end{enumerate}}
```

Modificarea listelor numerotate

Lucrul pe care utilizatorii vor să-l schimbe cel mai des la listele numerotate sunt contoarele. De aceea, pentru a merge mai departe, este necesară o scurtă introducere în domeniul *contoarelor* din LaTeX. Pentru orice lucru pe care LaTeX îl numără automat, cum ar fi antetele secțiunilor, figurile și listele cu marcatori, există un contor asociat cu el care controlează numerotarea.

Sunt patru contoare individuale asociate cu listele cu marcatori, fiecare reprezentând unul din cele patru niveluri posibile de imbricare, numite: `enumi`, `enumii`, `enumiii` și `enumiv`. Pentru a reseta oricare dintre aceste contoare în mijlocul unei enumerări simple, folosiți `\setcounter`. Contorul este incrementat de comanda `\item` înainte de a fi afișat. Spre exemplu, pentru a reseta `enumi`, scrieți:

```
\begin{enumerate}  
  \setcounter{enumi}{4}  
  \item Al cincilea element  
\end{enumerate}
```

care va fi afișat sub forma:

5. Al cincilea element

Fiecare contor are de asemenea un format implicit care îi dictează cum este afișat oricând LaTeX are nevoie să-l afișeze. Astfel de formate sunt specificate cu ajutorul unor comenzi LaTeX interne:

| Comandă | Exemplu |
|------------------------|--|
| <code>\arabic</code> | 1, 2, 3 ... |
| <code>\alph</code> | a, b, c ... |
| <code>\Alph</code> | A, B, C ... |
| <code>\roman</code> | i, ii, iii ... |
| <code>\Roman</code> | I, II, III ... |
| <code>\fnsymbol</code> | Făcut pentru note de subsol (vezi mai jos), însă afișează o secvență de simboluri. |

Fiecare entitate pentru un contor păstrează diverse informații despre acesta. Pentru a ajunge la elementul numerotat, folosiți comanda `\the` urmată imediat (adică fără spațiu) de numele contorului, de exemplu, `\theenumi`. Aceasta este deseori denumită *reprezentarea* unui contor.

Pentru a modifica formatarea la un nivel dat:

```
\renewcommand{\reprezentare}{\formatul_comenzii{contor}}
```

Versiunea generică nu este clară, astfel încât câteva exemple vor clarifica lucrurile:

```
% Redefiniți primul nivel
\renewcommand{\theenumi}{\Roman{enumi}}
\renewcommand{\labelenumi}{\theenumi}

% Redefiniți al doilea nivel
\renewcommand{\theenumii}{\Alph{enumii}}
\renewcommand{\labelenumii}{\theenumii}
```

Metoda utilizată mai sus mai întâi modifică în mod explicit formatul folosit de contor. Totuși, elementul ce controlează eticheta (*label*) trebuie actualizat pentru a reflecta schimbarea, lucru pe care-l face a doua linie. Un alt mod de a obține acest rezultat ar fi:

```
\renewcommand{\labelenumi}{\Roman{enumi}}
```

Asta pur și simplu redefinește aspectul etichetei, ceea ce e de ajuns, presupunând că nu aveți de gând să transmiteți referințe la un item specific din listă, caz în care referința va fi afișată în formatul anterior. Această problemă nu apare în primul exemplu.

Notăți că puteți adăuga și alte simboluri, cum ar fi parantezele și punctele, înainte și după contor. Spre exemplu, pentru a crea o listă indexată de litere mici cu paranteze înainte și după litere, puteți da următoarea comandă:

```
\renewcommand{\labelenumi}{(\alph{enumi})}
```

Modificarea listelor cu marcatori

Listele cu marcatori nu sunt atât de complexe întrucât nu trebuie să numere. De aceea, pentru a le personaliza, pur și simplu modificați etichetele. Acest

lucru se poate face manual pentru fiecare intrare cu comanda `\item[etichetă nouă]`, de exemplu `\item[\star]`.

Etichetele listelor cu marcatori se accesează prin intermediul comenzilor `\labelitemi`, `\labelitemii`, `\labelitemiii` și `\labelitemiv`, respectiv pentru cele patru niveluri.

Putem să modificăm aceste etichete în felul următor:

```
\renewcommand{\labelitemi}{\textgreater}
```

Exemplul de mai sus ar seta etichetele pentru primul nivel la un simbol "mai mare decât" ($>$). Desigur, simbolurile textuale din Latex nu sunt foarte atrăgătoare. Puteți folosi unul dintre simbolurile ZapfDingbat, descrise în secțiunea [Alte simboluri](#) din capitolul [Formatare](#). Sau puteți utiliza un simbol matematic:

```
\renewcommand{\labelitemi}{ $\star$ }
```

Puteți crea o listă cu marcatori, fără spații verticale între itemi consecutivi, în felul următor:

```
\begin{itemize}
  \setlength{\itemsep}{0cm}
  \setlength{\parskip}{0cm}
  \item Primul item din listă
  \item Itemul următor
\end{itemize}
```

Detalii ale personalizării listelor

Notați că este necesar ca `\renewcommand` să apară după instrucțiunea `\begin{document}`, astfel încât schimbările realizate să fie luate în considerare. Acest lucru este necesar atât pentru liste numerotate, cât și pentru cele cu marcatori.

Liste *inline*

Listele *inline* sunt un caz special întrucât necesită utilizarea pachetului [paralist](#), care oferă mediul `inparaenum` (cu o specificație opțională de formatare în paranteze drepte):


```

...
\usepackage{paralist}

\begin{document}

\textbf{\itshape
Listele inline} au o
natură secvențială,
la fel ca listele
numerotate,
dar sunt
\begin{inparaenum}
\itshape
a\upshape)
\item formatate în
paragrafele lor;
\item de obicei
etichetate cu litere;
și
\item au în general
prefixul `și' sau
`sau'
înainte de ultimul
item,
\end{inparaenum}
ca în acest
exemplu.
...

```

Listele inline au o natură fel ca listele numerotate, dar sur în paragrafele lor; *b*) de obicei litere; și *c*) au în general prefixi înainte de ultimul item, ca în ac

Pentru a schimba stilurile contoarelor, simbolurile A, a, I, i, și 1 pot fi utilizate în argumentul opțional pentru a produce contoare cu unul din stilurile `\Alph`, `\alph`, `\Roman`, `\roman` și `\arabic`. Spre exemplu:

```
\begin{inparaenum} [(i)]
```

produce etichetele (i), (ii), (iii) ...

Această pagină folosește materiale din cartea lui Andy Roberts, [Getting to grips with LaTeX](#), cu permisiunea autorului.

