Nem olyan rövid bevezető a LaTeX 2ε használatába

avagy LaTeX2e 84 percben

Tobias Oetiker

Hubert Partl, Irene Hyna és Elisabeth Schlegl

2.2. béta változat, 1996. január 25.

Németh László

Bővített magyar változat, 1998.

A fordító előszava

Örvendetes hír, hogy 1998 tavaszán megjelent az első magyar, LaTeX-hel foglalkozó könyv a Panem kiadó gondozásában. A hír persze azoknak jelenthet örömet, akik már ismerték, vagy próbálták megismerni a LaTeX-et; az egész világon, elsősorban egyetemi és akadémiai körökben elterjedt kiadványleíró és tördelő rendszert. A LaTeX-et ma már több száz folyóirat választotta szedőprogramjául szerte a világban. Magyarországon a TipoTeX, Akadémiai és Műszaki kiadó kiadványai között találunk LaTeX-hel készült könyveket (nem beszélve a számítástechnikai, számítástudományi könyvek kiadóiról).

A nyomdai tökéletességre törekvő szedőprogramot művészi érzékkel megáldott természettudósok tervezték és fejlesztik ma is. Kiemelendő közülük Donald E. Knuth matematikus, számos magyar nyelven is megjelent szakkönyv és ismeretterjesztő mű (*A számítógép-programozás művészete I–III*, *Konkrét matematika*, stb.) szerzője, aki a LaTeX lelkét, a TeX-et tervezte, illetve aki a LaTeX kiadványok többségében használt Computer Modern betűcsalád betűit „metszette”.

A most kézben tartott (vagy monitoron bámult) mű mértéktartó bevezető a LaTeX használatába. Őszintén remélem, hogy hasznos útmutatóul szolgál a szakdolgozatok, diplomamunkák, doktori disszertációk elkészítésekor.

A LaTeX helyes és hatékony használatának érdekében két fejezettel lett kibővítve a mű. Az első a helyes magyar tipográfiával ismertet meg, a második a teTeX programcsomag magyarításával és használatával.

Az ötödik fejezetet nem csak a LaTeX felhasználóknak, hanem minden szedésre vállalkozó szerzőnek szíves figyelmébe ajánlom, továbbá a ComputerBooks, Kovásznai, Scientia, stb. „kiadóknak” (ugyanis így néz ki az idézőjel).

Németh László

Köszönetnyilvánítás

Ez a mű a LaTeX 2.09-hez írt német nyelvű ismertetőn alapszik, melynek szerzői:

Hubert Partl <partl@mail.boku.ac.at>
Zentraler Informatikdienst der Universität für Bodenkultur Wien

Irene Hyna <Irene.Hyna@bmwf.ac.at>
Bundesministerium für Wissenschaft und Forschung Wien

Elisabeth Schlegl <nincs e-mail>
Graz

Ha érdeklődnénk a német eredeti után, Jörg Knappen LaTeX 2_ε-hez módosított változata javasolt. Ezt megtalálhatjuk a következő FTP címen:
`CTAN:/tex-archive/info/lkurz`

A szerző a `comp.text.tex` levelezős lista tagjait kérte fel a mű lektorálására. Sokan válaszoltak is a felhívására. A következő személyek segítettek ötleteikkel és javaslataikkal. Az ő segítségüknek köszönhető, hogy a könyv a jelenlegi formájában elkészülhetett. A szerző hálás köszönetét fejezi ki az együttműködőknek. Természetesen a könyvben esetlegesen fellelhető hibák a szerző (és a fordító) lelkiismeretét terhelik.

Rosemary Bailey, David Carlisle, Chris McCormack,
David Dureisseix, Elliot, Robin Fairbairns, Alexandre Guimond,
Cyril Goutte, Neil Hammond, Rasmus Borup Hansen,
Martien Hulsen, Eric Jacoboni, Alan Jeffrey, Byron Jones,
David Jones, Andrzej Kawalec, Christian Kern, Jörg Knappen,
Maik Lehradt, Claus Malten, Hubert Partl, John Refling,
Mike Ressler, Brian Ripley, Young U. Ryu, Chris Rowley,
Craig Schlenter, és Josef Tkadlec.

Előszó

A LaTeX [1] egy kiadványszerkesztő rendszer, mely leginkább matematikai és más tudományos szövegek tipográfiai minőségű szedésére használt. A rendszer lehetővé teszi azonban levelektől kezdve a könyvekig számos egyéb kiadvány kiszedését is. A LaTeX a TeX rendszer [2] egyszerű, és hatékony használatát teszi lehetővé.

Ez a rövid ismertető a LaTeX 2_ε leglényegesebb alkalmazásait mutatja be. A teljes leírás a referenciaművekben található meg [1, 3].

A LaTeX IBM PC-n is hozzáférhető. Sok egyetemi hálózaton installált a rendszer és működésre kész. Helyi útmutatók segíthetik a hozzáférést [4]. Probléma esetén forduljunk ahhoz, akitől ezt a könyvecskét kaptuk. A jelen mű *nem* ad tanácsot az installáláshoz, és a LaTeX rendszer belövéséhez[1], de megtanulható belőle a rendszer által használt nyelvezet.

Az anyag hat fejezetre oszlik:

1. Fejezet A LaTeX 2_ε forrásállományok felépítése. A LaTeX története dióhéjban. – A fejezet elolvasása után már némi képet alkothatunk a rendszerről. A kép még meglehetősen homályos, de jól megalapozza a következő fejezetek anyagát.

2. Fejezet A betűszedés rejtelmei. A LaTeX alapvető parancsai és környezetei. – A fejezet elolvasása után képesek leszünk megírni első LaTeX művünket.

3. Fejezet Matematikai, és egyéb formulák szedése. – Számos példa szemlélteti a LaTeX fő erősségét, valamint összefoglaló táblázatok mutatják a LaTeX rendszeren belül használható matematikai szimbólumokat.

4. Fejezet Specialitások – Egyéb hasznos lehetőségek: PostScript grafikák beillesztése, automatikusan kezelt irodalomjegyzék és tárgymutató, betűkészletek, és pozícionálás.

5. fejezet Tipográfiai szabályok – Amit tudnia kell annak, aki könyv, vagy egyéb kiadványok szedésére vállalkozik.

1. A LaTeX magyarítása egy plussz fejezetben kapott helyet. (*ford.*)

6. fejezet Magyarítás – A teTEX kibocsátás példáján a magyar elválasztásról, stílusfájlról, tárgymutató-készítésről, továbbá egy-két hasznos trükkről esik szó.

Mivel egymásra épülnek, tanácsos sorban olvasni a fejezeteket. Ez a könyv amúgy sem olyan hosszú. A példák gondos tanulmányozást igényelnek, mivel az információk jelentős hányada itt búvik meg.

További LaTeX-hel kapcsolatos anyagok a CTAN FTP archívumokban találhatók nagyobb bőségben. Internet címeik: `ftp.shsu.edu` (USA), `ftp.dante.de` (Németország), `ftp.tex.ac.uk` (Anglia). Javasolt a hozzánk közelebb eső német és az angol FTP helyek használata.

Ha van valami ötletünk e könyvecske bővítésére, változtatására vonatkozóan, értesítsük a szerzőt, és/vagy a fordítót. Ők ugyanis nagyon kiváncsiak a LaTeX kezdők visszajelzéseire, különösen arra, hogy mennyire világosak az itt leírtak, és hogy hogyan lehetne javítani az anyagon.

Tobias Oetiker <oetiker@ee.ethz.ch>
Department of Electrical Engineering,
Swiss Federal Institute of Technology
Németh László <nemethl@sol.cc.u-szeged.hu>

Az aktuális angol nyelvű verzió hozzáférhető a következő címen:
`CTAN:/tex-archive/info/lshort`

Tartalomjegyzék

Táblázatok jegyzéke

1. fejezet

Amit tudnunk kell

A fejezet első része rövid áttekintést ad a LaTeX 2_ε filozófiájáról és történetéről. A második rész a LaTeX forrásállományok felépítését mutatja be. A fejezet áttekintésével képet kapunk a LaTeX működéséről, amelyet majd a további fejezetek finomítanak.

1.1. Nevezetes nevek

1.1.1. TeX

A TeX program Donald E. Knuth alkotása [2]. Szövegek és matematikai formulák szedésére készült.

A TeX-et „teh"-nek ejtjük. Olyan „h"-val, mint ami a magyar ihlet szóban, vagy a német „Ach"-ban, vagy a skót „Loch"-ban van. `ASCII` környezetben `TeX`-ként írjuk.

1.1.2. LaTeX

A LaTeX egy olyan programgyűjtemény, mellyel képessek vagyunk műveinket a legmagasabb tipográfiai követelményeknek is megfelelő minőségben kiszedni, egy előre definiált, professzionális kiadványleíró környezetet használva. A LaTeX-et eredetileg Leslie Lamport írta [1]. A LaTeX a TeX-et használja, mint formázó és szedő alaprendszert.

Jelenleg a LaTeX3 csoport, és vezetőjük, Frank Mittelbach tevékenységének köszönhetően sor került a LaTeX csomag következetes módosítására, a régóta igényelt javításokra és tökéletesítésekre, a több éve megjelent LaTeX 2.09 csomagot követő módosított verziók egységesítésére. Az új verzió a LaTeX 2_ε elnevezést kapta. Ez a könyvecske a LaTeX 2_ε-vel foglalkozik.

A LaTeX-et „lateh"-nek ejtjük. `ASCII` környezetben `LaTeX`-nek írjuk. A LaTeX 2_ε ejtése „lateh kettő e", és `LaTeX2e`-ként hivatkozunk rá.

1.2. Alapok

1.2.1. Szerző, tipográfus, és szedő

Ha publikálni akar, a szerző átadja kéziratát egy kiadóvállalatnak. A cég tipográfusa megállapítja a kész mű felépítését (a hasábok szélességét, a használt betűtípusokat, a fejlécek feletti és alatti kihagyásokat, és a többit). A tipográfus instrukciói a kéziratba kerülnek, s ezek alapján a nyomdász (vagy szedő) kiszedi a művet.

Egy tipográfus megpróbálja kitalálni, mi járhatott a szerző fejében a kézirat megalkotása alatt. Szakmai tudása és a kézirat tartalma alapján meghatározza a fejezetek fejlécének, a hivatkozásoknak, példáknak, formuláknak, és más összetevőknek az alakját.

Egy LaTeX környezetben a LaTeX magára vállalja a tipográfus és a TeX-et használva a szedő szerepét is. Sajnos a LaTeX „csak” egy program, ezért némileg több útmutatásra van szüksége. A szerző plussz információval kell, hogy szolgáljon, leírva készülő művének logikai vázát. Ez a leírás gyakorlatilag a kézirat szövegébe illesztett „LaTeX parancsok”-kal történik.

Ez egészen más megközelítés, mint a WYSIWYG[1], amit a legtöbb modern szövegszerkesztőnél – mint pl. a *Word for Windows* és *WordPerfect* – találunk. Ezekben az alkalmazásokban a felhasználók interaktívan formázzák begépelt szövegeiket, s a szerkesztés alatt is láthatják a végeredményt.

A LaTeX esetében nem látjuk a végeredményt gépelés közben. Megtekinthető azonban, ha lefordítjuk megszerkesztett állományunkat a LaTeX-hel. Így még többször módosíthatjuk a kiadványt, míg jónak nem ítéljük a nyomtatásra.[2]

1.2.2. Tipográfia

A tipográfia mesterség. Kontár szerzők gyakran súlyos tipográfiai hibákat követnek el, úgy vélve, hogy a könyvtervezés csak esztétikai kérdés. „Ha egy könyv művészi, akkor jó a tipográfiája is.” Ám egy könyvet olvasni szokás, nem pedig barangolni benne, mint egy galériában. Az olvashatóság és az áttekinthetőség fontosabb szempont, mint a szép külalak. Például:

- A megfelelő betűméret, a fejezetek és az alfejezetek megfelelő számozása világossá teszi az olvasó számára a kiadvány felépítését.

- A sorokat olyan rövidre kell választani, hogy ne fárassza a szemet, és olyan hosszúra, hogy azért szépen töltse be az oldalakat.

1. Amit látsz, azt kapod. (What you see is what you get.)
2. A LyX program Word for Windows-szerű felülettel, WYGIWYM megközelítéssel (azt kapod, amit szeretnél) lehetővé teszi, hogy szerkesztés közben is lássuk a LaTeX kiadványok egyfajta előnézetét. (*ford.*)

A WYSIWYG rendszerekkel gyakran esztétikus élményt nyújtó kiadványok is előállíthatók, de ezek csekély, vagy következetlen struktúrával rendelkeznek. A LaTeX megakadályozza a szerzőket a formázási hibák elkövetésében, azzal, hogy kikényszeríti a készülő kiadvány *logikai* felépítésének megadását. Ennek köszönhetően tud majd alkalmas tipográfiai beállításokat választani a LaTeX.

1.2.3. Előnyök és hátrányok

Egy gyakran vitatott téma a WYSIWYG világ hívei és a LaTeX-et használók között, hogy mik a „LaTeX előnyei a normál szövegszerkesztőkkel szemben", vagy ennek fordítottja. A legjobb dolog, ha óvatosan kezeljük a témát, bár néha nem menekülhetünk...

A főbb előnyök a normál szövegszerkesztőkkel szemben a következőek:

- Professzionális tipográfia, mellyel a kiadvány úgy néz ki, mintha nyomdában készült volna.
- A matematikai formulák könnyedén megvalósíthatóak.
- A felhasználónak csak egy kevés könnyen érthető parancsot kell megjegyeznie ahhoz, hogy képes legyen megadni a kiadvány logikai szerkezetét. Ha ezt megteszi, utána már szinte sosem kell a kiadvány formázásával foglalkozni.
- Minden összetettebb struktúra – mint a lábjegyzetek, hivatkozások, tartalomjegyzékek és bibliográfiák – könnyedén előállítható.
- A LaTeX képességeit jelentősen megnövelik a szabadon hozzáférhető kiegészítő csomagok. Például beilleszhetünk PostScript grafikákat, vagy létrehozhatunk az adott kívánalmaknak megfelelő bibliográfiákat. Számos csomag ismertetésre kerül a LaTeX kézikönyvben [3].
- A LaTeX szépen tagolt művek írására ösztönöz, mivel csak meghatározott szerkezetű kiadványokat kezel.
- A TeX, a LaTeX 2_ε motorja, portábilis és a legtöbb operációs rendszeren szabadon hozzáférhető.

A LaTeX-nek van némi hátránya is:

- Nagyobb erőforrás-igény (memória, lemezterület, számítási kapacitás) szükségeltetik, mint egy egyszerű szövegszerkesztőhöz. A helyzet azonban javul, ahogy a *Word for Windows 6.0* mérete már meghaladta egy normál LaTeX rendszerét. Processzor-igény szempontjából a LaTeX minden szövegszerkesztőt megelőz, mivel csak a fordításnál igényel nagyobb teljesítményt. A WYSIWYG programok ellenben végig.

- Ámbár az előredefiniált szövegformázó környezetek paraméterezhetők, egy egészen új tipográfájú környezet kialakítása bonyolult, és sok időbe kerül.[3]

1.3. LaTeX forrásállományok

A LaTeX inputja egy sima `ASCII` szöveges állomány. Bármilyen egyszerű szövegszerkesztővel előállíthatjuk.[4] A szerkesztett szöveg együtt tartalmazza a kiadvány szövegét és a LaTeX parancsait.[5]

1.3.1. Szóközök és üres sorok

A helykihagyó karaktereket, mint a szóközt, vagy tabulátor karaktert a LaTeX egységesen szóközként kezeli. Az *egymást követő* helykihagyó karakterek egy szóköznek számítanak. A sor eleji szóközöket nem értelmezi a LaTeX, és egy egyszerű sortörés is csak egy szóköznek felel meg.

A szövegben elhelyezett üres sor jelzi a bekezdés végét. Az egymást követő üres sorok itt is csak egy sornak számítanak. A következő példa szemlélteti ezt. Jobb oldalon a forrás szövege, balra a formázott eredmény látható.

Felesleges a sortörés, és a sok szóköz egy szó után.

Új bekezdést jelöl egy üres sor.

```
Felesleges a sortörés,
és a sok      szóköz
egy szó után.

Új bekezdést jelöl
egy üres sor.
```

1.3.2. Speciális karakterek

A következő jelek foglalt karakterek. Speciális jelentéssel bírnak a LaTeX környezetben, ezért alapesetben nem használhatóak egy betűkészlet elemeként sem. Ha közvetlenül írjuk a szövegbe, normális esetben nem jelennek meg a végeredményben.

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

3. Úgy hírlik, hogy a készülő LaTeX3 rendszer fokozott hangsúlyt fektet ennek módosítására.

4. Sajnos a magyar számítástechnikai köznyelv nem tesz világos különbséget az egyszerű text editorok és a más típust képviselő wordprocessor-ok között. Text editorra példa MS-DOS környezetben: *edit*; UNIX környezetben: *vi*, *pico*, *mcedit*. Wordprocessor-ok a már emlegetett grafikus WYSIWYG rendszerek. (*ford.*)

5. A forrásállomány kiterjesztése `tex`. Fordítása a `latex forras.tex` paranccsal történik. A kapott `dvi` kiterjesztésű állomány többféle segédprogrammal is megtekinthető és nyomtatható. (*ford.*)

Amint a példa mutatja, a foglalt karakterek minden gond nélkül használhatóak, ha egy fordított perjelet[6] teszünk eléjük:

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

Több szimbólum és számos egyéb dolog részben speciális parancsok, részben parancsjelek segítségével érhető el.

1.3.3. LaTeX parancsok

A LaTeX parancsokban nem cserélhetőek fel a kis és a nagybetűk. A parancsokat két csoportra oszthatjuk alakjuk szerint:

- Repjellel (\) kezdődnek, és nevük csak az angol ábécé betűit tartalmazza. A parancsokat egy szóköz, esetleg szám, vagy egyéb jel zárja, de semmiképp sem betű.
- Repjelet és pontosan egy speciális karaktert tartalmaznak.

A LaTeX a parancsokat követő szóközöket nem veszi figyelembe. Ha egy parancs után szóközt szeretnénk, akkor írjuk utána a `{}` jelpárt és végül egy szóközt, vagy speciális helykihagyó parancsot. A csipkés zárójel-pár segítségével a LaTeX nem tünteti el a szóközöket.

Azt olvastam valahol, hogy Knuth a TeX felhasználóit két részre osztja: TeXetségesekre, és TeXetségekre.
A mai dátum: 1998. július 17.

```
Azt olvastam valahol, hogy Knuth
a \TeX{} felhasználóit két részre
osztja: \TeX{}etségesekre, és
\TeX etségekre. \\
A mai dátum: \today
```

Vannak parancsok, amelyek egy, vagy több paramétert várnak. A paramétereket csipkés zárójelben (`{ }`) adjuk meg a parancsok után. Néhány parancsnak fakultatív paraméterek is adhatók, ezeket szögletes zárójelbe (`[ ]`) tesszük a parancs után. A következő példa pár LaTeX parancsot villant fel. Aggodalomra semmi ok, később részletesen ismertetésre kerülnek.[7]

Bízhatsz bennem!

```
\textsl{Bízhatsz} bennem!
```

Kezdj új sort itt jobbra, ha tudsz! Köszönöm!

```
Kezdj új sort itt jobbra,
ha tudsz!\linebreak[3]
Köszönöm!
```

6. A továbbiakban a fordított perjelet *repjel*nek nevezzük, Wettl Ferenc [11] nyomán. (*ford.*)

7. Sajnos megtévesztő az eddig használt LaTeX parancs kifejezés, mivel nem csak a parancsokra, hanem a LaTeX változóira is ilyen formában hivatkozunk. (*ford.*)

1.3.4. Megjegyzések

Ha a LaTeX egy `%` karakterrel találkozik egy forrásállomány feldolgozása közben, az aktuális sor hátralevő részét nem veszi figyelembe. Így olyan megjegyzéseket fűzhetünk a forrásszöveghez, melyek nem kerülnek bele a kész, nyomtatott változatba. Másik tipikus alkalmazása, hogy letiltjuk vele a sorvégi szóköz beillesztését, ami megkönnyíti a hosszabb összefüggő karaktersorozatok írását.

Ez egy példa.

```
Ez egy % hülye
% Jobb: szemléletes <----
példa%
.
```

1.4. A forrásállomány felépítése

Minden LaTeX forrásnak a következővel kell kezdődnie:

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

Ezzel a sorral a kiadvány típusát határozzuk meg. Utána még további parancsokat adhatunk meg, melyek az egész kiadvány stílusát befolyásolják, vagy betölthetünk kiegészítő csomagokat.

A csomagok betöltésére a következő parancs szolgál:

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

Az elsődleges beállítások után a következő paranccsal kezdjük a forrásállomány törzsét:

```
\begin{document}
```

A forrásállomány törzsébe kerül a kiadványunk szövege, némi LaTeX paranccsal megspékelve.[8] A forrásállomány végét a

```
\end{document}
```

parancs jelzi. Ha van még valami hátra, azt a LaTeX már nem veszi figyelembe.

Az 1.1. ábra a minimális LaTeX 2_ε forrásállomány tartalmát mutatja. Valamivel komplikáltabb állományra példa az 1.2. ábra.

8. A `\documentclass` és a `\begin{document}` parancsok közé eső rész a *preambulum*.

```
\documentclass{article}
\begin{document}
A kicsi sz\'ep.
\end{document}
```

1.1. ábra. A minimális LaTeX forrás

```
\documentclass[a4paper,11pt]{article}
\usepackage{magyarul}
\author{H.~Partl}
\title{Minimalizmus}
\begin{document}
\maketitle
\tableofcontents
\section{Bevezetés}
Nos, itt veszi kezdetét kedvenc cikkem.
\begin{verbatim}
\section{Befejezés}
\ldots{} és itt ér véget.
\end{document}
```

1.2. ábra. Példa egy cikk forrására

1.5. A kiadvány tipográfiája

1.5.1. Dokumentumosztályok

A LaTeX elsőként arra kiváncsi, hogy milyen típusú kiadványt szeretnénk írni. Ezt a `\documentclass` paranccsal adjuk meg:

`\documentclass[`*opciók*`]{`*osztály*`}`

Itt az *osztály* határozza meg a kiadvány típusát. Az 1.1. táblázat tartalmazza azt a négy dokumentumosztályt, ami a későbbiekben ismertetésre kerül. A LaTeX 2_ε rendszer további dokumentumosztályokat is tartalmaz, pl. levélíráshoz. Az *opciók* a dokumentumosztály működését módosítják. Itt az egyes paramétereket vesszővel elválasztva adjuk meg. Az 1.2. táblázat mutatja a standard paramétereket.

Példa: Ha egy LaTeX kiadvány forrása a

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

sorral kezdődik, akkor egy *cikk* írásához fogtunk hozzá, ami *tizenegy pontos* alapbetűmérettel, *kétoldalas* formában kerül majd nyomtatásra *A4-es papír*on.

1.5.2. Csomagok

Gyakran kerülünk olyan helyzetbe, hogy a LaTeX kevésnek bizonyul egy-egy konkrét feladat megoldásához. Például ha grafika, színes szöveg, vagy forráskódot tartalmazó állomány beillesztése a célunk, a LaTeX kiegészítésre

1.1. táblázat. Dokumentumosztályok

`article` a tudományos publikációk, előadások, kisebb jelentések, programdokumentációk, meghívók, stb. számára.

`report` a nagyobb jelentések, kisebb könyvek, PhD tézisek, stb. számára, melyek jellemzője, hogy több fejezetre tagolódnak.

`book` az igazi könyvek számára.

`slide` fóliák számára. A dokumentumosztály nagyméretű talpnélküli betűket használ.

1.2. táblázat. A dokumentumosztályok paraméterei

`10pt`, `11pt`, `12pt` A kiadvány betűtípusának alapméretét adhatjuk meg *pont*ban. Ha egy fenti paramétert sem adunk meg, akkor az alapértelmezett `10pt` fog élni.

`a4paper`, `letterpaper`, ... Megadjuk a papír méretét. Az alapértelmezett paraméter a `letterpaper`. Használhatók még a következőek is: `a5paper`, `b5paper`, `executivepaper`, és `legalpaper`.

`fleqn` A matematikai formulákat nem középre, hanem balra zárja a LaTeX.

`leqno` A matematikai formulák számozása a jobb oldal helyett balra kerül.

`titlepage`, `notitlepage` Annak beállítása, hogy a kiadvány címe külön oldalra kerüljön-e vagy nem. Az `article` osztály nem kezd új oldalt a cím után, míg a `report` és a `book` igen.

`twocolumn` A szedés két hasábos formába történik.

`twoside, oneside` Az egy, vagy kétoldalas kiadvány között választhatunk. Az `article` és a `report` alapesetben egyoldalas, míg a `book` kétoldalas.

`openright, openany` Az új fejezetek nyítóoldalának elhelyezkedését állíthatjuk be. Az `openright` paraméterrel az új fejezetek mindig jobb oldalon kezdődnek. Az `article` osztályban – fejezetek híjján – nincs értelmezve ez az opció. A `report` osztályon belül a következő oldalon való fejezetkezdés az alapértelmezett, a `book` osztályon belül a jobb oldali kezdet.

szorul. A kiegészítéseket csomagoknak nevezzük. A csomagok használata a

`\usepackage[`*opciók*`]{`*csomag*`}`

paranccsal történik, ahol a *csomag* a kiegészítő csomag neve, és az *opciók* pedig a csomag speciális paramétereit tartalmazhatja. Jónéhány csomag a LaTeX 2_ε disztribúció része. (Lásd 1.3. táblázat). Másokat külön szerezhetünk be. Valószínű, hogy erről bővebb információt kapunk a helyi hálózatunkon. Az elsődleges információforrás a *LaTeX kézikönyv.* Itt csomagok százainak leírását találjuk meg, illetve annak részletes ismertetését, hogy hogyan írhatunk mi is kiegészítő csomagokat a LaTeX 2_ε rendszerhez.

1.5.3. Oldalstílusok

A LaTeX három előredefiniált fejléc/„lábléc" kombinációt, vagyis oldalstílust tartalmaz. A *stílus* paraméter jelöli ki a

`\pagestyle{`*stílus*`}`

parancsban a megfelelő oldalstílust. Az 1.4. táblázat mutatja a lehetséges oldalstílusokat.

Megváltoztathatjuk csak az aktuális oldal stílusát is a következő parancscsal:

`\thispagestyle{`*stílus*`}`

A *LaTeX kézikönyv* [3] írja le, hogy készíthetünk saját fejléc és „lábléc" stílusokat.

1.6. Nagy művek

Ha nagy szövegdokumentumokon dolgozunk, jogosan merül fel az az igény, hogy a forrásállományunkat több részre osszuk.[9] A LaTeX két parancsa segít ebben. Az

`\include{`*fájlnév*`}`

parancsot a forrásállomány törzsén belül használhatjuk, a *fájlnév* nevű fájl tartalmának beillesztésére. Fontos, hogy a betöltött fájl tartalma mindig új oldalon kezdődik.

9. Ezzel a szerkesztés biztonságosabb, az ellenőrző fordítás pedig gyorsabb lehet. (*ford.*)

1.3. táblázat. Néhány csomag, amit a LaTeX disztribúció is tartalmaz

`doc` LaTeX programok dokumentálását teszi lehetővé.
Leírása a `doc.dtx`[10] állományban és a *LaTeX kézikönyv*ben található.

`exscale` A matematikai betűkészlet jeleinek rugalmas méretezését teszi lehetővé.
Leírása az `exscale.dtx` állományban.

`fontenc` A LaTeX betűkódolásának beállítása.[11]
Leírása a `ltoutenc.dtx` állományban.

`ifthen` Parancsokat nyújt a „ha… akkor… különben…” szerkezet számára. Leírása a `ifthen.dtx` állományban és a *LaTeX kézikönyv*ben.

`latexsym` Szimbólum fontkészlet használatát teszi lehetővé.
Leírása a `latexsym.dtx` állományban és a *LaTeX kézikönyv*ben.

`makeidx` Egyszerű parancsokat tartalmaz tárgymutató készítéséhez.
Leírása ebben a 4.5. pontban és a *LaTeX kézikönyv*ben.

`syntonly` A LaTeX feldolgozza a forrásállományt, anélkül, hogy elvégezné a szöveg tördelését.
Leírása a `syntonly.dtx` állományban és a *LaTeX kézikönyv*ben. Gyors hibaellenőrzésre használható.

`inputenc` Átkódolja a forrásállományt a paraméterként kijelölt kódtáblának megfelelően. Ez lehet ASCII, ISO Latin-1 (nyugat-európai), ISO Latin-2 (közép-európai), 437/850 IBM kódtábla, Apple Macintosh, Next, ANSI-Windows vagy a felhasználó által definiált típus.
Leírása az `inputenc.dtx` állományban.

10. Ez az állomány telepítve kell, hogy legyen egy LaTeX 2_ε rendszerben, és lefordítható `dvi` állományra a `latex doc.dtx` paranccsal. Ez a táblázatban felsorolt összes `dtx` állományra is vonatkozik. [Ez nem mindig teljesül. Például a `teTeX` disztribúció `tetex-texmf-src` részét – mely a `dtx` állományokat is tartalmazza – nem mindig telepítik az egyetemi hálózatokon. Forduljunk a rendszergazdához. (*ford.*)]
11. az `\usepackage[T1]{fontenc}` parancs hatására a `cm` helyett az európai nyelvek számára készült `ec` betűkészleteket használjuk. (*ford.*)

A második parancsot a LaTeX forrás preambulumában használhatjuk. Arra szolgál, hogy kijelöljük a számos beillesztendő, tehát az `\include` parancs paramétereként szereplő fájl közül azt az egyet, amit majd a LaTeX ténylegesen betölt.

Az `\includeonly` paranccsal több ilyen fájlt is kijelölhetünk:

`\includeonly{`*fájlnév*`,`*fájlnév*`,...}`

A listában szereplő fájlok töltődnek majd be ténylegesen a teljes szövegdokumentumba, amennyiben persze szerepeltek az `\include` parancsok valamelyikében is. Jegyezzük meg, hogy a fájlnevek és a vesszők között nem lehet szóköz.

Az `\include` paranccsal beillesztett szöveg új oldalon kezdődik. Ez nem haszontalan, mivel akárhogy is változtatjuk az `\includeonly` parancsban az éppen beillesztendő fájlok listáját, az oldaltörések helye nem változik a betördelt szövegben.

Néha nem kívánatos az új oldalon való kezdés. Ekkor a

`\input{`*fájlnév*`}`

parancs használható. Ez csak egyszerű beillesztést végez.

1.4. táblázat. A LaTeX előredefiniált oldalstílusai

`plain` Az oldalszám a lap aljára kerül, középre pozicionálva. Ez az alapértelmezett oldalstílus.

`headings` Az aktuális fejezet címe és az oldalszám a lap tetejére kerül, míg a lap alja üres marad. (Ez a kiadvány is ezzel az oldalstílussal készült.)

`empty` A fejléc és a lap alja is üres marad.

2. fejezet

Szövegszedés

Az előzőek alapján már lehet némi fogalmunk a LaTeX 2ε forrásállományok felépítéséről. A most következő fejezet anyagával kiegészítve már mindent tudni fogunk egy valódi LaTeX kiadvány elkészítéséhez.

2.1. Sor- és oldaltörés

2.1.1. Sorkizárás

A könyvek gyakran tömbösített, vagy sorkizárt szedést alkalmaznak, tehát a sorok hossza megegyező a szövegben. A LaTeX a sortöréseket és a szóközöket optimalizálva, az egész bekezdés tartalmát figyelembe véve igyekszik elhelyezni. Ha szükséges, automatikusan elválaszt. A bekezdések szedése az adott dokumentumosztálytól függ. Alapesetben a bekezdés első sora beljebb kezdődik, és a bekezdések közti távolság a sorközzel egyezik meg. A 4.2.2. pont részletesebben tárgyalja ezt a témát.

Ha szükséges, adott helyen is sortörésre bírhatjuk a LaTeX-et a

`\\` vagy `\newline`

parancsokkal. Ezzel új sort kezdünk anélkül, hogy új bekezdést nyitnánk. Bővebb változat a

`\\*`

parancs, amivel megtiltjuk a sortörést esetleg közvetlenül követő oldaltörést. Új oldalon folytatja a szedést a LaTeX a

`\newpage`

parancs hatására. A következő négy parancs is a nevéhez hűen működik:

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

vagyis sor- és oldaltörést vált ki, illetve tilt meg. Megadhatunk egy n paramétert is, melynek nulla és négy között változhat az értéke. Amennyiben négynél kisebb az érték, a LaTeX felülbírálhatja kérésünket, ha az kevésbé jó, vagy kicsit rossz, rossz vagy nagyon rossz szövegképet eredményez ($n = 0, 1, 2, 3$).

A LaTeX mindig megpróbálja megkeresni a legjobb sortörést. Ha nem sikerül a magas követelményeknek megfelelő módon eltörni egy sort, akkor azt hagyja kilógni a bekezdés jobb szélén. Fordítás közben ilyenkor kapunk *„overfull hbox”* hibaüzenetet.[1] Leggyakrabban ez akkor következik be, ha a LaTeX nem boldogul egy szó automatikus elválasztásával. Ha kiadjuk a `\sloppy` parancsot, a LaTeX követelményei kisebbek lesznek. Az esetleges problémát úgy oldja meg, hogy nagy mértékben növeli a szóközöket, amíg a végeredmény nem optimális. Ilyenkor az *„underfull hbox”* üzenet figyelmezteti a felhasználót a kevésbé szép, „kihajtott” sorokra. Többnyire azonban elfogadható végeredményt kapunk. A `\fussy` parancs pontosan ellenkező hatású. Arra az esetre való, ha kevésnek találnánk a LaTeX hibaüzeneteit.

2.1.2. Elválasztás

A LaTeX, ha szükséges, elválasztja a szavakat. Az elválasztási eljárás kivételes esetekben nem tud helyesen elválasztani, ilyenkor pár paranccsal segíthetünk ezen. Ilyen hasznos parancs a

`\hyphenation{`*szavak listája*`}`

ahol a paraméterként megadott szavak lehetséges elválasztási helyeit a „`-`” jellel jelöljük. Ezt a parancsot a forrásállomány preambulumában kell elhelyeznünk. A szavak csak betűket tartalmazhatnak. A nagybetűk ekvivalensek a kisbetűkkel. A következő példa alapján az „elválasztás” „Elválasztás”-ként is helyesen lesz elválasztva, míg a „FORTRAN” elválasztása még „Fortran”-ként és „fortran”-ként is meg lesz tiltva. Speciális karakterek és szimbólumok nem használhatók a szavakban.

Példa:

```
\hyphenation{FORTRAN El-vá-lasz-tás}
```

A `\-` paranccsal tetszőleges helyen jelölhetünk elválasztást szavakban. Ilyenkor csak az adott helyen, vagy helyeken fogja elválasztani a megjelölt szót a LaTeX. Különösen akkor jó ez, ha a szavak speciális karaktereket (pl. ékezetes betűket) tartalmaznak, mivel ezeket a LaTeX nem választja el automatikusan.[2]

1. Illetve fel van tüntetve a forrásállomány azon sorának száma is, ahol a hiba történt. Ez nagyon hasznos a javításhoz. (*ford.*)

2. Installálható automatikus (és helyes) magyar elválasztás is a LaTeX-hez. Lásd később. (*ford.*)

– Tyű! – mondta Béla. – Azt a rézfánfütyülőjét!

```
-- Tyű! -- mondta Béla. -- Azt
a réz\-fán\-fü\-tyü\-lő\-jét!
```

Ha szeretnénk, hogy bizonyos szavak ne kerüljenek külön sorba, használjuk az

`\mbox{`*szöveg*`}`

parancsot. A paraméterként megadott szöveg minden esetben egy sorban marad.

A telefonszámom megváltozott. Az új szám 0116 291 2319.

A *fájlnév* paraméternek az állomány nevét kell tartalmaznia.

```
A telefonszámom megváltozott.
Az új szám \mbox{0116 291 2319}.

A \mbox{\emph{fájlnév}}
paraméternek az állomány nevét
kell tartalmaznia.
```

2.2. Speciális karakterek és szimbólumok

2.2.1. Idézőjelek

Ne használjuk a klaviatúra `"` jelét idézőjelnek. A helyesírás rendes nyitó és záró idézőjeleket kíván. A LaTeX esetében két vessző a nyitó idézőjelet, két aposztróf pedig a záró idézőjelet jelöli.

„Nyomja meg az »x« gombot!”

```
,,Nyomja meg az >>x<< gombot!''
```

2.2.2. Kötőjelek

A LaTeXet használva négy hasonló jelet kell tudnunk megkülönböztetni. Ebből hármat különböző számú kötőjelből álló sorozattal kódolunk. A negyedik jel a mínusz jel, amivel a külön fejezetben tárgyalt matematikai környezetben találkozhatunk.

New York-ban egy-két em-
ber – 1848–49 alatt –
yes—or no? (csak angolban!)
0, 1 és -1

```
New York-ban egy-két em-\\
ber -- 1848--49 alatt --\\
yes---or no? (csak angolban!)\\
$0$, $1$ és $-1$
```

A jelek neve: `-` kiskötőjel, `--` nagykötőjel, `---` kvirtmínusz és `$-$` mínuszjel.

2.2.3. Három pont (. . .)

Az írógép esetében egy vessző, vagy egy pont ugyanakkora helyet foglal el a papíron, mint akármelyik betű. A nyomdai termékekben ezek a jelek nagyon közel kerülnek a betűkhöz. Így „három pont” jelet nem kaphatunk három pont leütésével, hiszen a térközök hibásak lennének. A megoldás az

```
\ldots
```

parancs használata.

Meg akarta tenni… Megtette.
New York, Tokió, Budapest, …

```
Meg akarta tenni\ldots{}
Megtette.\\ New York,
Tokió, Budapest, \ldots
```

2.2.4. Ligatúrák

Bizonyos betűkombinációk nem külön betűkként, hanem egyetlen speciális nyomdai jelként kerülnek a kiszedett szövegbe.

ff fi fl ffi ffl helyett ff fi fl ffi ffl

Ezek az úgy nevezett f-ligatúrák. Használatuk kötelező érvényű a magyar nyomdászatban.[3]

2.2.5. Ékezetek és speciális karakterek

A LaTeX sok nyelv ékezetés és speciális karakterét képes megjeleníteni. A 2.1. táblázatban az „o” betű példáján mutatjuk be a lehetséges ékezeteket. Természetesen más betűkkel ugyanúgy működik.

Az i és j betű ékezesítése előtt le kellene szednünk a pontot. Ez megoldható a `\i` és a `\j` parancsokkal.

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

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

3. A német tipográfia a toldalékok és az összetett szavak határán eltekint a ligatúrák használatától. Pl.: Auflage, melynek forrása: `Auf\/lage`.

2.3. Nemzetközi nyelvi támogatás

Ha más nyelveken szeretnénk publikálni, az angoltól eltérő elválasztási szabályok is beállíthatók a LaTeX-en belül.

Számos nyelvhez jól használható Johannes Braams babel csomagja. A csomag használatához LaTeX rendszerünk speciális konfigurációja szükséges. Erről a helyi útmutatók adhatnak tájékoztatást.[4]

Ha rendszerünk megfelelően konfigurált, a babel csomag a

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

paranccsal aktiválható a `\documentclass` parancs után. Az elérhető *nyelvek* listáját a babel csomag dokumentációja ismerteti.

A babel néhány nyelvhez új parancsot is definiál. Nyelvünk számára egy speciális parancsot tartalmazott eddig. Ez pedig az `\ondatemagyar` parancs.

Kelt Szegeden, 1998. július 17-én. `Kelt Szegeden, \ondatemagyar.`

Számos számítógépes környezetben a speciális karakterek közvetlenül vihetők be a billentyűzetről. A LaTeX képes értelmezni ezeket a jeleket is. 1994 decembere óta a LaTeX 2_ε kibocsátásával jónéhány forrás átkódoló bővítés is részét képzi az alap LaTeX rendszernek. Az inputenc csomagot használva jelölhetjük ki azt a kódszabványt, amit a számítógépes környezetünk használ a speciális jelek kódolására. Bár ugyanazt látjuk, a háttérben eltérő kódrendszerek rögzítik a jeleket az eltérő rendszereken. Például MS-DOS környezetben a hosszú ő kódja decimálisan 139 a 852-es kódtáblában, UNIX környezetben 245 az ISO-LATIN 2 kódolásban. Ha lehet, ügyeljünk erre.

4. Illetve a könyv utolsó fejezete. (*ford.*)

2.1. táblázat. Ékezetek és speciális karakterek

ò	\`o	ó	\'o	ô	\^o	õ	\~o
ō	\=o	ȯ	\.o	ö	\"o		
ŏ	\u o	ǒ	\v o	ő	\H o	o̧	\c o
ọ	\d o	o̱	\b o	o͡o	\t oo		
œ	\oe	Œ	\OE	æ	\ae	Æ	\AE
å	\aa	Å	\AA				
ø	\o	Ø	\O	ł	\l	Ł	\L
ı	\i	ȷ	\j	¡	!`	¿	?`

2.4. Szóközök

A szép sorkizárt szedéshez a LaTeX bizonyos határok közt képes változtatni a szóközök méretét. Az angolszász tipográfiának megfelelően a mondatok után kicsivel nagyobb szóközt szed a program. A LaTeX feltételezi, hogy a mondatok ponttal, kérdőjellel, vagy felkiáltójellel végződnek. Kivétel, ha a pont egy nagybetűt követ, mivel ilyen esetekben általában rövidítésről van szó.

Konkrét esetben bármikor módosíthatunk ezen. Nem növelt méretű szóközt kapunk, ha a szóköz elé repjelet helyezünk a forrásban. Ezt eredményezi az is, ha szóköz helyett hullámvonal jelet (`~`) használunk, továbbá a sortörést is meggátoljuk ennél a pontnál („nem törő szóköz"). A `\@` paranccsal nagybetűt követő pontból is mondatzáró pontot csinálhatunk.

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

A mondatok közti nagyobb szóköz kikapcsolható a

```
\frenchspacing
```

paranccsal. Ez jellemző a nem angolszász nyelvekre, így a magyarban is ez a szabály alkalmazandó. Természetesen ha be van állítva a `\frenchspacing`, a `\@` jelölést felesleges használni.

2.5. Címek, fejezetek és kisebb szakaszok

Az olvasó dolgát megkönnyítendő, a művet fejezetekre, kisebb és még kisebb szakaszokra tagolhatjuk. A LaTeX számos paranccsal támogatja a szakaszok megfelelő kialakítását. Lényeges, hogy a parancsok egymásba ágyazásának logikus sorrendjét betartsuk.

Az `article` osztályban a következő szakaszkezdő parancsok használhatók:

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

A `report` és a `book` osztály két további szakaszkezdő paranccsal bővül:

```
\part{...}   \chapter{...}
```

Mivel az `article` dokumentumosztályú forrásokban semmit sem találunk a fejezetekre vonatkozóan, nagyon egyszerűen helyezhetünk el cikkeket egy kiadványba: a `report`, vagy a `book` osztály fejezeteiként. A szakaszok közti helykihagyások, és a szakaszcímek méretezése és számozása teljesen automatikusan történik.[5]

A szakaszkezdő parancsok közül kettő egy kicsit speciális:

- A `\part` parancs nincs hatással a fejezetek számozására.
- Az `\appendix` parancsnak nincs paramétere. Hatására a fejezetek[6] számozása betűkre cserélődik.

A LaTeX a már egyszer lefordított forrásállomány szakaszcímeinek és oldalszámainak ismeretében tartalomjegyzéket képes generálni a

`\tableofcontents`

parancs által jelölt helyre. A helyes tartalomjegyzékhez ezért kétszer fordítsuk le a megváltoztatott forrásállományt. Bizonyos esetekben három fordításra is szükség lehet. A LaTeX figyelmezteti az embert, ha szükséges volna még egy fordítás.

Minden szakaszkezdő parancsnak van egy „csillagos" változata. A „csillagos" parancsok nevét az eredeti nevekből egy `*` hozzátoldásával kapjuk. Az így kapott szakaszcím nem kerül a tartalomjegyzékbe, és nem lesz számozva. Például a `\section{Segítség}` helyett a `\section*{Segítség}` parancsot is használhatjuk.

Alapesetben a tartalomjegyzék címei pontosan megegyeznek a szakaszcímekkel. Néha azonban az extrém hosszú szakaszcímek nem férnek el a tartalomjegyzékbe. Ilyenkor megadhatunk rövidebb címeket is kizárólag a tartalomjegyzék számára a szakaszkezdő parancsok opcionális paramétereként.

```
\chapter[Olvasd el! Izgalmas!]{Ez egy nagyon hosszú
  és különösen unalmas cím}
```

A kiadvány címét a

`\maketitle`

parancs segítségével szúrhatjuk be. A cím tartalmát – a konkrét címet, a szerző(ke)t és a dátumot – előzőleg definiálni kell a

`\title{...}`, `\author{...}` és az opcionális `\date{...}`

parancsokkal. Az `\authors` parancs paramétereként több szerző is megadható a `\and` paranccsal elválasztva.

5. Helyes számozást a magyar stílusfájl haszálatával érhetünk el. Ld. 6. fejezet. (*ford.*)
6. Az `article` stílus esetén a kisebb szakaszok számozása változik meg.

Pár a fenti parancsokból megszemlélhető a 7. oldalon, a 1.2-es ábrán.

A fenti szakaszkezdő parancsokon kívül a LaTeX 2_ε három további parancsot definiált a `book` osztály számára:

`\frontmatter`, `\mainmatter` és `\backmatter`

Ezzel három fő részre bonthatjuk könyvünket. A fejezetcímek tipográfiája és az oldalszámozás más a `\frontmatter` és a `\backmatter` részben. Olyasformán, ami elvárható egy könyvtől.

2.6. Kereszthivatkozások

A könyvek, jelentések és cikkek gyakran tartalmaznak kereszthivatkozásokat ábrákra, táblázatokra és a szövegdokumentum egyéb speciális elemeire. A LaTeX a következő parancsokkal támogatja a kereszthivatkozások egyszerű megvalósítását:

`\label{`*címke*`}`, `\ref{`*címke*`}` és `\pageref{`*címke*`}`,

ahol a *címke* a felhasználó által szabadon választott azonosító. A LaTeX a `\ref` parancs helyére beilleszti annak a szakasznak, ábrának, táblázatnak vagy tételnek a számát, amit megjelöltünk a `\label` paranccsal. A `\pageref` annak az oldalnak a számát illeszti be, ahová a `\label` parancs által jelölt szakasz került. Ebben az esetben is az előző fordítás eredményeit használja a rendszer.

Egy hivatkozás erre a szakaszra így néz ki: „lásd a 2.6. szakaszt a 20. oldalon.”

```
Egy hivatkozás erre a szakaszra
\label{oldal:ez} így néz ki:
,,lásd a \ref{oldal:ez}.~szakaszt
a \pageref{oldal:ez}.~oldalon.''
```

A hivatkozásokhoz tartozó helyes névelő (a/az) automatikus kiválasztására a magyar stílusfájl nyújt további parancsokat (ld. 70. oldal).

2.7. Lábjegyzetek

Lábjegyzetek könnyen szúrhatók a szövegbe a

`\footnote{`*lábjegyzet szövege*`}`

paranccsal.

A lábjegyzetek[a] nem ritkák a LaTeX kiadványokban.

[a] Ez egy lábjegyzet

```
A lábjegyzetek\footnote{Ez egy
  lábjegyzet} nem ritkák a
\LaTeX{} kiadványokban.
```

A lábjegyzetek magyar tipográfiának megfelelő számozására a 69. oldalon találunk útmutatást.

2.8. Idézetek

Gépelt kéziratokban aláhúzás jelzi a kiemelt szavakat. A nyomdai kivitelű munkákban a kiemelés *kurzív* betűtípussal jelenik meg, amit az

`\emph{`*szöveg*`}`

paranccsal érhetünk el LaTeXben.

Ha kiemelt szövegben végzünk kiemelést*, a LaTeX az* alap *betűváltozatot használja a kiemelésre.*

```
\emph{Ha kiemelt szövegben végzünk
  \emph{kiemelést}, a \LaTeX{} az
  \emph{alap} betűváltozatot
  használja a kiemelésre.}
```

2.9. Környezetek

A speciális szövegrészek szedésére a LaTeX számos környezetet biztosít. Használatuk a

`\begin{`*név*`}` *szöveg* `\end{`*név*`}`

szintaxissal történik, ahol a *név* a használni kívánt környezet neve. A környezetek egymásba ágyazhatók, addig, amíg azt az adott környezet támogatja.

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

A következő pontokban a legfontosabb környezetek kerülnek ismertetésre.

2.9.1. Számozott és egyéb felsorolások

Az `itemize` környezettel felsorolásokat készíthetünk. Az `enumerate` környezet számozott felsorolások, míg a `description` környezet leírások, fogalommagyarázatok készítését teszi lehetővé.

1. Kedved szerint keverheted a felsorolás környezeteket:
   - Díszpont helyett lehet
   - gondolatjel is
2. De vigyázat!

   Hibás tipográfia felsorolással sem lesz jobb.

   Helyes tipográfia viszont még szebb is lehet.

```
\begin{enumerate}
\item Kedved szerint keverheted
a felsorolás környezeteket:
\begin{itemize}
\item Díszpont helyett lehet
\item[--] gondolatjel is
\end{itemize}
\item De vigyázat!
\begin{description}
\item[Hibás] tipográfia
  felsorolással sem lesz jobb.
\item[Helyes] tipográfia viszont
  még szebb is lehet.
\end{description}
\end{enumerate}
```

2.9.2. Sorzárás balra, jobbra és középre

A `flushleft` és a `flushright` környezet balra és jobbra zárt bekezdéseket eredményez. A `center` környezet középre pozícionált sorokat eredményez. Ha nem adjuk meg a \\ paranccsal, hogy hol szeretnénk a sortöréseket, akkor a LaTeX automatikusan végzi a sortörést.

Ez a szöveg
balra zárt. A LaTeX nem veszi a sorokat egyforma hosszúra.

```
\begin{flushleft}
Ez a szöveg\\ balra zárt.
A \LaTeX{} nem veszi a sorokat
egyforma hosszúra.
\end{flushleft}
```

Ez a szöveg
jobbra zárt. A LaTeX nem veszi a
sorokat egyforma hosszúra.

```
\begin{flushright}
Ez a szöveg\\ jobbra zárt.
A \LaTeX{} nem veszi a sorokat
egyforma hosszúra.
\end{flushright}
```

A Föld
középpontjában.

```
\begin{center}
A Föld\\
középpontjában.
\end{center}
```

2.9.3. Idézett szövegek és versek

A `quote` környezet szövegrészek; különösen idézetek, példák kiemelését teszi lehetővé.

A sorok hosszára vonatkozó tipográfiai szabály szerint:

> Egy sor se tartalmazzon 66 karakternél többet.
>
> Ezért van olyan nagy margója a LaTeX oldalaknak.

S ezért alkalmaznak több hasábos szedést a lapokban.

```
A sorok hosszára vonatkozó
tipográfiai szabály szerint:
\begin{quote}
Egy sor se tartalmazzon
66~karakternél többet.

Ezért van olyan nagy margója
a \LaTeX{} oldalaknak.
\end{quote}
S ezért alkalmaznak több
hasábos szedést a lapokban.
```

Létezik még két hasonló környezet, a `quotation` és a `verse`. A `quotation` hosszabb, több bekezdésből álló idézetek számára készült, mivel behúzással kezdi a bekezdéseket. A `verse` környezet versek, hosszabb sorokból álló költemények megjelenítésére szolgál, ahol különös jelentősége van az esetleges sortöréseknek. Itt minden sort `\\` paranccsal zárunk, a versszakokat üres sor választja el.

Egy angol verset tudok csak. Humpty Dumpty-ról szól.

> 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.

```
Egy angol verset tudok csak.
Humpty Dumpty-ról szól.
\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}
```

2.9.4. ASCII forrás megjelenítése

A `\begin{verbatim}` és a `\end{verbatim}` közé kerülő szöveg úgy kerül nyomtatásra, amint az a LaTeX forrásállományban szerepelt. Minden sortörés és szóköz is ennek megfelelően fog megjelenni, a LaTeX parancsok sem fognak végrehajtódni ebben a környezetben.

Egy bekezdésen belül ugyanezt a célt szolgálja a

`\verb+`*text*`+`

parancs. A `+` csak egy példa a határolójelre, mivel bármilyen jelet használhatunk, kivéve a betűket, a `*` és szóköz karaktereket. Ebben a leírásban a példák jelentős része a `\verb` parancs segítségével került a szövegbe.

A \verb|\ldots| parancs\ldots

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

```
\begin{verbatim}
A \verb|\ldots| parancs\ldots

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

A␣verbatim␣környezet
,,csillagos''␣változata
kiemeli␣␣a␣␣szóközöket.

```
\begin{verbatim*}
A verbatim környezet
,,csillagos'' változata
kiemeli  a  szóközöket.
\end{verbatim*}
```

A `\verb` parancs csillagos változata ugyanerre használatos:

␣␣␣Még␣ilyet␣:-)␣

```
\verb*|   Még ilyet :-) |
```

A `verbatim` környezet és a `\verb` parancs más parancsok paraméterében nem használható.

2.9.5. Táblázatok

A `tabular` környezettel táblázatokat készíthetünk. A táblázat a beállítástól függően tartalmazhat vízszintes és függőleges vonalakat. A LaTeX az oszlopok szélességét maga állítja be. A

`\begin{tabular}{`*táblázat paraméterei*`}`

parancsból látszik, hogy meg kell adnunk a *táblázat paramétereit* is a környezetben. A paraméterek a táblázat oszlopaira vonatkoznak, illetve a függőleges vonalak beállítására. Egy `l` karakter egy balra zárt tartalmú oszlopot jelöl, egy `r` karakter jobbra zártat, egy `c` karakter középre pozícionáltat. Egy `p{`*szélesség*`}` paraméter sorkizárt tartalmú oszlopot jelöl *szélesség* sorszélességgel, míg egy `|` jel egy függőleges vonalat a táblázatban.

A `tabular` környezetben egy `&` jellel ugrunk a következő oszlop cellájába, `\\` kezd egy új sort, és a `\hline` parancs szúr be egy vízszintes vonalat.

7C0	hexadecimális
3700	oktális
11111000000	bináris
1984	decimális

```
\begin{tabular}{|r|l|}
\hline
7C0 & hexadecimális \\
3700 & oktális \\
11111000000 & bináris \\
\hline \hline
1984 & decimális \\
\hline
\end{tabular}
```

Isten hozott titeket ebben a bekezdésben. Reméljük tetszeni fog a mutatvány.

```
\begin{tabular}{|p{4.7cm}|}
\hline
Isten hozott titeket ebben a
bekezdésben. Reméljük
tetszeni fog a mutatvány.\\
\hline
\end{tabular}
```

A `@{...}` paranccsal megváltoztathatjuk az oszlopválasztó szerkezetet. A parancs megszünteti az oszlopközt két oszlop között, s a megadott paraméterrel helyettesíti. Nagyszerű eszköze ez a decimális igazítás megvalósításának, mint ahogy ez a lenti példából ki fog derülni. Másik hasznos funkciója, hogy leszedhető vele táblázatunk két széléről a térköz:

nincs térköz

```
\begin{tabular}{@{} l @{}}
\hline
nincs térköz\\
\hline
\end{tabular}
```

térközök két oldalt

```
\begin{tabular}{l}
\hline
térközök két oldalt\\
\hline
\end{tabular}
```

Bár létezik a decimális igazításnak egy stílusosabb módja is a dcolumn csomag révén, egyszerűen megvalósítható a `@{}` ismeretében is. Decimális igazítás esetén a tizedesvesszők kerülnek egymás alá az oszlopban. Valójában egy balra és egy jobbra zárt oszlopot fogunk definiálni és egy tizedesvesszőt tartalmazó oszlopközt. Ez a kívánt eredményt fogja szolgáltatni. Ne feledjük el kitenni a `&` jelet az egészrész és a tizedesrész közé a sorokban. Trükkös „oszlopunk”-nak nevet is adhatunk a `\multicolumn` parancs segítségével:

Pi kifejezés	Érték
π	3,1416
π^{π}	36,46
$(\pi^{\pi})^{\pi}$	80662,7

```
\begin{tabular}{c r @{,} l}
Pi kifejezés       &
\multicolumn{2}{c}{Érték} \\
\hline
$\pi$               & 3&1416  \\
$\pi^{\pi}$         & 36&46   \\
$(\pi^{\pi})^{\pi}$ & 80662&7 \\
\end{tabular}
```

2.10. Lebegő dobozok

A publikációk nagy része számos ábrát, táblázatot tartalmaz. Ezek speciális bánásmódot igényelnek, mivel nagy összefüggő, nem törhető szövegrészekként, ún. dobozokként jelentkeznek. A legegyszerűbb az volna, ha mindig új oldalt kezdenénk egy ilyen elemnél, ha az nem fér már el az adott oldalon. Félig üres oldalak jelzik, hogy ez nem a legjobb módszer.

A probléma megoldása az ábrák és táblázatok „lebegtetésében" rejlik. Ha már nem férnek el, az oldal akkor sem marad félig üres, hanem folytatódik az ábra, vagy a táblázat utáni szöveggel. A problémás dobozok pedig a következő oldal(ak)ra kerülnek. A LaTeX két környezetet használ a lebegő dobozok számára. Egyet az ábráknak és egyet a táblázatoknak. Nagyon fontos, hogy megértsük, mi áll a két környezet működésének hátterében. Ha nem tesszük, nagyon sokat fogunk azon bosszankodni, hogy a LaTeX miért nem oda teszi a táblázatainkat és ábráinkat, ahova szeretnénk.

Elsőként vessünk egy pillantást azokra a parancsokra, amivel a LaTeX kezeli a lebegő dobozok problémáját.

Minden „lebegő" ábra és táblázat a `figure` és a `table` környezetbe kerül.[7] Mindkét környezet rendelkezik egy opcionális paraméterrel is:

`\begin{figure}[`*elhelyezés*`]` vagy `\begin{table}[`*elhelyezés*`]`

Az *elhelyezés* paraméter fogja közölni a LaTeX-hel, hogy pontosan hol szeretnénk látni dobozainkat. A paraméter *elhelyezésre vonatkozó parancsjelek* sorozata. Lásd 2.2. táblázat.

7. A lebegő környezetekbe nem csak ábra és táblázat kerülhet. Többnyire azonban a `figure` környezetbe egy `picture` környezetet, vagy EPS képet ágyazunk, a `table` környezetbe pedig egy `tabular` környezetet. Ezért beszélünk csak lebegő ábrákról és táblázatokról. (*ford.*)

2.2. táblázat. Elhelyezésre vonatkozó parancsjelek

Jel	Engedélyezett hely...
`h`	az aktuális *helyen* kerüljön elhelyezésre. Kis dobozok esetén hasznos.
`t`	a lap *tetején*
`b`	a lap *alján*
`p`	egy speciális *oldalon*, ami csak lebegő dobozokat tartalmaz
`!`	mindenképp történjen meg az elhelyezés. (Ne vegyen figyelembe bizonyos belső paramétereket, például az egy oldalon elhelyezhető lebegő dobozok maximális számát.)

Például egy lebegő tábla kezdődhet így is:

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

az elhelyezésre vonatkozó `[!hbp]` paraméter azt jelenti, hogy a LaTeX tegye az aktuális helyre (`h`), vagy a lap aljára (`b`) , vagy egy külön gyűjtőoldalra (`p`) a táblázatunkat, még akkor is , ha nem a legjobbnak találtatott (`!`). Ha hiányzik az *elhelyezés* opcionális paraméter, akkor a `[tbp]` elhelyezés lesz érvényben.

A LaTeX minden lebegő dobozt megpróbál a beállításoknak megfelelően elhelyezni. Ha ez nem jön össze egy doboz esetén az aktuális oldalon, akkor az bekerül az *ábrák*, vagy a *táblázatok* várakozási sorába. Egy új oldal kezdésénél a LaTeX megvizsgálja, lehet-e ez az oldal speciális az elhelyezés szempontjából. Ha nem, akkor a várakozási sorból az első elemet próbálja elhelyezni, természetesen az elhelyezési paraméter figyelembevételével. Siker esetén vizsgálja csak a következőket.[8] Ha egy elemet nem sikerül egyáltalán elhelyezni a a szövegben, akkor az a kiadvány végére kerül, s minden utána következő elem is a várakozási sorból!

> Gyakran csak egyetlen eleme felelős a két várakozási sornak azért, hogy a LaTeX nem úgy helyezi el a lebegő dobozokat, ahogy szeretnénk.

További lehetősége a két lebegő környezetnek, hogy címet adhatunk tartalmuknak a

`\caption{`*cím*`}`

paranccsal. A számozást és az „ábra”, illetve „táblázat” feliratot a LaTeX automatikusan helyezi a cím elé.

A `\tableofcontents` paranccsal analóg működésű a

`\listoffigures` és a `\listoftables`

parancs, mely az ábrák és a táblázatok listáját illeszti be a parancs alkalmazásának helyére. Hosszú címek esetén itt is használhatunk opcionális rövid címeket a listák számára.

```
\caption[Rövid cím]{HHHHoooooszszszszúúúú ccccííííímmmm}
```

Az ábrákra és táblázatokra szintén a `\label` és a `\ref` parancsok segítségével tudunk hivatkozni.

8. A várakozási lista egy FIFO (first in first out) veremnek is tekinthető, magyarul az az elem kerül ki legelőbb a sorból, amelyik legelőször belekerült.

A következő példa egy négyzetet rajzol és szúr be a kiadványba. Jól használható ez a módszer arra, hogy a külön beillesztendő képeink számára helyet hagyjunk a szövegben.

```
A Pop-Art-ra példa a \ref{white}~ábra.
\begin{figure}[!hbp]
\makebox[\textwidth]{\framebox[5cm]{\rule{0pt}{5cm}}}
\caption{5-ször 5 centiméter} \label{white}
\end{figure}
```

A fenti példában[9] a LaTeX *valóban* nagyon keményen (`!`) próbálkozik, hogy az ábrát oda tegye, ahova rögtön kerülhetne (`h`). Ha ez nem sikerülne, a lap alja lenne a következő próba színhelye (`b`). Ha az aktuális oldalon sehogy sem férne el az ábránk, akkor még elférhetne a következő oldalon, esetleg más ábrák és táblázatok társaságában (`p`). Ha nincs elég lebegő elem egy speciális oldalhoz, az új oldalon úgy próbálja elhelyezni a LaTeX az ábránkat, mintha ott fordult volna elő éppen a szövegben.

Fentiek értelmében időnként szükséges lehet a

`\clearpage` vagy a `\cleardoublepage`

parancs használata. A parancsok hatására a LaTeX a várakozási sorban lévő minden elemet elhelyezi a szövegben, és új oldalt kezd. A `\cleardoublepage` esetében az új oldal bal kéz felől fog esni.

Később ismertetésre kerül, hogy illeszthetünk be PostScript ábrákat LaTeX kiadványokba.

2.11. Új parancsok és környezetek definiálása

Az első fejezetben szó volt arról, hogy a kiadvány logikai vázát megadva a LaTeX képes elvégezni a szedést. Ez szép idea, de a gyakorlatban gyakran nem működik, mivel a LaTeX nem tartalmazhat minden olyan speciális környezetet, vagy parancsot, mint amit el tudunk képzelni.

Egy jó megoldás, hogy több LaTeXparancs segítségével szedjük ki, amit elképzeltünk. Ha csak egyszer van erre szükség, nincs is probléma. Ha azonban többször, időben és helyben is sokat veszthetünk, különösen, ha később még változtatni is szeretnénk.

Igazi megoldást a saját parancsok és környezetek definiálása nyújt.

9. feltételezve, hogy az ábrák várakozási sora üres

2.11.1. Új parancsok

Új parancs definiálására a

`\newcommand{`*név*`}[`*szám*`]{`*definíció*`}`

parancs szolgál. Alapesetben a parancs két paramétert vár. A *név* paraméter az új parancs neve lesz, amelyet ténylegesen a *definíció* ír le. A szögletes zárójelbe zárt *szám* paraméter opcionális. Vele olyan új parancsokat gyárthatunk, melyek akár kilenc paramétert is fogadhatnak.

A következő két példa teszi érthetővé a fentieket. Az első egy `\ntrb` nevű parancsot definiál, ami a „Nem túl rövid bevezető a LaTeX 2ε használatába" rövidítése. A parancs nagyon hasznos, ha újra és újra le szeretnénk írni a könyv címét.

„Nem olyan rövid bevezető a LaTeX 2ε használatába" … „Nem olyan rövid bevezető a LaTeX 2ε használatába"

```
\newcommand{\ntrb}
 {Nem olyan rövid bevezető a
 \LaTeXe{} használatába}
% a szövegben:
,,\ntrb'' \ldots{} ,,\ntrb''
```

A következő példa a *szám* paraméter használatát szemlélteti. A `#1` jel a parancsnak megadott paraméter konkrét értékével fog behelyettesítődni az új parancs alkalmazásánál. A további paraméterekre a `#2`-vel és így tovább hivatkozunk.

- *Nem olyan* rövid bevezető a LaTeX 2ε használatába
- *Nagyon* rövid bevezető a LaTeX 2ε használatába

```
\newcommand{\txsit}[1]
 {\emph{#1} rövid bevezető a
 \LaTeXe{} használatába}
% a szövegben:
\begin{itemize}
\item \txsit{Nem olyan}
\item \txsit{Nagyon}
\end{itemize}
```

A LaTeX nem engedi, hogy olyan parancsot hozzunk létre, ami már létezik. Egy már létező parancsot a `\renewcommand` paranccsal tudunk átdefiniálni. A parancs szintaxisa megegyezik a `\newcommand` parancséval. Időnként hasznos lehet a `\providecommand` parancs is. Ez a `\newcommand` parancshoz hasonlóan működik, de ha már létezik a parancs, amit definiálni szeretnénk, akkor nem változtat ezen a LaTeX.

2.11.2. Új környezetek

A `\newcommand` parancshoz hasonlóan épül fel az új környezetet definiáló parancs is. Ezen munkában is számos új környezetet definiált a szerző, a példák, kódrészletek, parancs-ismertetések számára. A `\newenvironment` parancs a következő szintaxist követi:

`\newenvironment{`*név*`}[`*szám*`]{`*előtte*`}{`*utána*`}`

Úgy mint a `\newcommand` esetén, a `\newenvironment` parancs is használható az opcionális paraméterrel, vagy anélkül. Az *előtte* paraméter tartalmazza azokat a formázási parancsokat, amik a `\begin{`*név*`}` kiadásánál lépnek majd működésbe, az új környezetbe kerülő szöveg feldolgozása előtt. Ennek megfelelően az *utána* paraméter tartalma mindig az új környezetet lezáró `\end{`*név*`}` parancs esetén kerül feldolgozásra.

A következő példa illusztrálja a `\newenvironment` parancs használatát.

A fixa ideám…

```
\newenvironment{king}
    {\begin{quote}}{\end{quote}}
% a kiadvány törzsében használd
\begin{king}
A fixa ideám\ldots
\end{king}
```

A *szám* paraméter ugyanúgy használatos, mint a `\newcommand` parancs esetében. A LaTeX ebben az esetben is meggyőződik arról, hogy nem–e egy már létező környezetet próbáltunk felüldefiniálni. Ha ez volna a szándékunk, használjuk a `\renewenvironment` parancsot. A szintaxisa megegyezik a `\newenvironment` parancséval.

3. fejezet

Matematikai szövegek szedése

A fejezet a TEX fő erősségével, a matematikai szövegek szedésével foglalkozik. De vigyázat! Csak a felszínt súrolja az anyag, bár valószínűleg sokaknak ez is elegendő lesz. Ne essünk kétségbe, ha nem találnánk itt megoldást. Kimerítő részletességgel foglalkozik a matematikai szövegszedés rejtelmeivel az AMS-LATEX[1], és még pár csomag.

3.1. Alapok

A LATEX-nek létezik egy speciális matematikai módja. Ha egy bekezdésen belül szeretnénk megjeleníteni matematikai formulákat, akkor azokat a `\(` és `\)`, vagy `$` és `$`, vagy `\begin{math}` és `\end{math}` jelpárok közé zárjuk.

a négyzet meg b négyzet az c négyzet. Vagy valamivel egzaktabban: $c^2 = a^2 + b^2$	`$a$ négyzet meg $b$ négyzet az $c$ négyzet. Vagy valamivel egzaktabban: $c^{2}=a^{2}+b^{2}$`

A TEX-et $\tau\epsilon\chi$-nek ejtjük.

100 m^3 víz

$\heartsuit$-ből jön

```
A \TeX -et $\tau\epsilon\chi$-nek
ejtjük.\\[6pt]
100~m$^{3}$ v\'\i z\\[6pt]
$\heartsuit$-b\H ol jön
```

Nagyobb egyenletek, vagy formulák szedése megkívánja a külön sorba szedést. Ehhez `\[` és `\]`, vagy `\begin{displaymath}` és `\end{displaymath}` közé zárjuk a külön sorba szedendő matematikai szöveget. Az így kapott formulákat nem számozza a LATEX. Az automatikus számozáshoz az `equation` környezetet használjuk.

1. CTAN:/tex-archive/macros/latex/packages/amslatex

a négyzet meg b négyzet az c négyzet. Vagy valamivel egzaktabban:

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

És még egy sor.

```
$a$ négyzet meg $b$ négyzet
az $c$ négyzet. Vagy valamivel
egzaktabban:
\begin{displaymath}
c^{2}=a^{2}+b^{2}
\end{displaymath}
És még egy sor.
```

A `\label` és a `\ref` használatával hivatkozhatunk az egyenletekre.

$$\varepsilon > 0 \tag{3.1}$$

A (3.1)-ből kapjuk...

```
\begin{equation} \label{egy:e}
\varepsilon > 0
\end{equation}
A (\ref{egy:e})-b\H ol
kapjuk\ldots
```

Jegyezzük meg, hogy a két matematikai környezet eltérő stílusú:

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

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

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

Több különbség van a *matematikai* és a *szöveges mód* között. Például a *matematikai mód* esetén:

1. A legtöbb szóköznek és sortörésnek nincs speciális szerepe. A TEX a matematikai kifejezések logikáját követve határozza meg a térközöket, illetve használható egy-két speciális helykihagyó parancs is, mint a `\,`, `\quad` vagy a `\qquad`.

   $$\forall x \in \mathbf{R}: \qquad x^{2} \geq 0 \tag{3.2}$$

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

2. Üres sorok nem használhatók. Egy formula nem állhat több bekezdésből.

3. Minden betű változóként kerül értelmezésre, s ennek megfelelően történik a szedés is. Ha normál szöveget akarunk a formulánkba megjeleníteni (normál betűközökkel és álló betűkkel), akkor a `\textrm{...}` parancs használandó.

$$x^2 \geq 0 \qquad \textrm{ha } x \in \mathbf{R} \tag{3.3}$$

```
\begin{equation}
x^{2} \geq 0\qquad
\textrm{ha } x\in\mathbf{R}
\end{equation}
```

A matematikusok erősen kötődnek az általuk preferált szimbólumokhoz. A hagyományos duplaszárú betűket a `\mathbb` paranccsal hívhatjuk elő, az amsfonts vagy az amssymb csomagokat használva. Az előző példa módosítva:

$$x^2 \geq 0 \qquad \textrm{ha } x \in \mathbb{R}$$

```
\begin{displaymath}
x^{2} \geq 0\qquad
\textrm{ha }  x\in\mathbb{R}
\end{displaymath}
```

3.2. Elemek a matematikai módban

Matematikai módban a parancsok nagy része egyetlen elemre, alapesetben egyetlen karakterre hat. Ha több karakteren szeretnénk a parancsot alkalmazni, ezekből egy elemet képezhetünk[2] csipkés zárójelek segítségével: `{...}`.

$$a^x + y \neq a^{x+y} \tag{3.4}$$

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

3.3. A matematikai formulák leírása

E szakaszban az alapvető formulaleíró parancsokat tekintjük át. A használható szimbólumok részletes listái a 3.10. szakaszban kerülnek ismertetésre, a 42. oldaltól kezdve.

A **görög kisbetűk**et az `\alpha`, `\beta`, `\gamma`, ..., a nagybetűket[3] a `\Gamma`, `\Delta`, ... parancsokkal érhetjük el.

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

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

Kitevők és **indexek** a „`^`” és a „`_`” karakterek segítségével képezhetők.

2. *grouping*
3. A nagy alfa nem definiált a LaTeX 2ε-ben, mivel megegyezik a nagy „A” betűvel. Az új matematikai környezet kiadásával fog ez majd csak megváltozni.

a_{1} $\qquad$ x^{2} $\qquad$ $e^{-\alpha t}$ $\qquad$ a^{3}_{ij}
$e^{x^2} \neq {e^x}^2$

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

Négyzetgyököt az `\sqrt`, n-dik gyököt pedig az `\sqrt[`n`]` paranccsal kaphatunk. A gyökjel méretét a LaTeX határozza meg. Ha csak pusztán egy gyökjelre van szükség, a `\surd` parancs alkalmazandó.

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

```
$\sqrt{x}$ \qquad
$\sqrt{ x^{2}+\sqrt{y} }$
\qquad $\sqrt[3]{2}$\\[3pt]
$\surd[x^2 + y^2]$
```

Az `\overline` és `\underline` parancsokkal **vízszintes vonalak** kaphatók közvetlenül az adott kifejezés felett, vagy alatt.

$\overline{m+n}$

```
$\overline{m+n}$
```

A `\overbrace` és `\underbrace` parancsok hosszú **vízszintes kapcsok**at helyeznek a kifejezés fölé, vagy alá.

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

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

A 42. oldalon találjuk a matematikai módban használható ékezetesítő parancsokat. Az egyszerű kis kalap és hullámvonal **matematikai ékezetek** mellett előállíthatunk nagy hullámvonalat és „nagy kalapot” a `\widetilde` és `\widehat` parancsokkal, amelyek több karaktert is átfoghatnak. A `'` szimbólummal jelölhetjük a derivált kifejezéseket.

$$y=x^{2}\qquad y'=2x\qquad y''=2$$

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

A **vektorok**at gyakran kis nyíl szimbólummal jelöljük. A `\vec` paranccsal állíthatjuk elő a nyíl ékezetet. Ha nagyobb nyílra van szükségünk, mint például az A-ból B-be mutató vektor esetében, érdemes megjegyezni az `\overrightarrow` és `\overleftarrow` parancsokat.

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

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

A logaritmus-szerű függvényeket gyakran álló betűkkel szedjük, nem kurzívval, mint a változókat. Ilyen megfontolásból a LaTeX a legfontosabb függvényneveket már előre definiálva tartalmazza:

`\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`

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

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

A modulo függvényhez két parancs kapcsolódik: A `\bmod` parancs eredménye bináris operátor (például „$a \bmod b$”), a `\pmod` paranccsal pedig az „$x \equiv a \pmod{b}$” kifejezésben is szereplő formulát kapjuk meg.

Törtek a `\frac{...}{...}` paranccsal hozhatók létre. Időnként azonban szerencsésebb az 1/2 alakú jelölés használata.

$1\frac{1}{2}$ óra

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

```
$1\frac{1}{2}$~\'ora
\begin{displaymath}
\frac{ x^{2} }{ k+1 }\qquad
x^{ \frac{2}{k+1} }\qquad
x^{ 1/2 }
\end{displaymath}
```

A `{... \choose ...}` és az `{... \atop ...}` parancs szolgál a binomiális együtthatók leírására. A második parancs ugyanazt eredményezi, mint az első, de zárójelek nélkül.

$$\binom{n}{k}\qquad \genfrac{}{}{0pt}{}{x}{y+2}$$

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

Integráljelet az `\int`, szummajelet a `\sum` paranccsal kaphatunk. A felső és az alsó indexet a `^` és a `_` jellel adhatjuk meg.

$$\sum_{i=1}^{n} \qquad \int_{0}^{\frac{\pi}{2}} \qquad$$

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

A **zárójelek** számos fajtáját kezeli a TEX (például $[\ \langle\ \|\ \ \updownarrow$). A kerek és szögletes zárójelek közvetlenül használhatók, míg a csipkés zárójel a `\{` paranccsal érhető el. A különleges zárójeleket hosszabb parancsokkal (pl. `\updownarrow`) jeleníthetjük meg. A használható zárójelek összefoglaló táblázatát a 44. oldalon találjuk.

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

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

A kezdő zárójel elé helyezett `\left`, és a berekesztő zárójel elé helyezett `\right` parancsok hatására a TEX automatikusan megfelelőre veszi a zárójelek méretét. Minden `\left` parancsot egy `\right` parancsnak kell zárnia. Ha nem akarunk semmit például a jobb oldalon, használjunk „láthatatlan" zárójelet: `\right.`!

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

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

Bizonyos esetekben magunk szeretnénk meghatározni a zárójelek méretét. Erre szolgál a zárójel elé helyezendő `\big`, `\Big`, `\bigg` és `\Bigg` parancs.[4]

$\Big((x+1) (x-1) \Big) ^{2}$

$\big(\Big(\bigg(\Bigg(\quad \big\}\Big\}\bigg\}\Bigg\}\quad \big\|\Big\|\bigg\|\Bigg\|$

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

A matematikai formulákban gyakran szereplő **három pont** megjelenítésére több parancs áll rendelkezésünkre. A már ismert `\ldots` paranccsal alulra kerülnek a pontok, míg a `\cdots` paranccsal középre. A `\vdots` parancs függőlegesen kipontozást tesz lehetővé, a `\ddots` parancs átlósat. A 3.5. szakaszban láthatunk egy további példát is.

$$x_{1},\ldots,x_{n} \qquad x_{1}+\cdots+x_{n}$$

```
\begin{displaymath}
x_{1},\ldots,x_{n} \qquad
x_{1}+\cdots+x_{n}
\end{displaymath}
```

3.4. Helykihagyás matematikai módban

Ha a formulák térközei nem megfelelőek, speciális parancsokkal változtathatunk ezen. A legfontosabbak: a `\,` kicsi térköz, a `\␣` közepes (normál szóköz méretű) térköz, a `\quad` és `\qquad` nagy térközök. A `\!` parancs kicsit csökkent a térközön.

4. A parancsok nem működnek megfelelően, ha `11pt` vagy `12pt` alap betűmérettel dolgozunk. Ilyenkor az `exscale` vagy az `amsmath` csomagok használata segít.

$$0,5 \int\int_{D} g(x,y)\mathrm{d}x\mathrm{d}y$$

helyett

$$0,\!5 \int\!\!\!\int_{D} g(x,y) \, \mathrm{d}x\, \mathrm{d}y$$

```
\newcommand{\ud}{\mathrm{d}}
\begin{displaymath}
0,5 \int\int_{D}
g(x,y)\ud x \ud y
\end{displaymath} helyett
\begin{displaymath}
0,\!5 \int\!\!\!\int_{D}
g(x,y) \, \ud x\, \ud y
\end{displaymath}
```

Jegyezzük meg, hogy a differenciálegyenletekben a „d” álló betű.

3.5. Függőlegesen igazított elemek

Mátrixok szedésére az `array` környezet szolgál. A környezet a `tabular` környezethez hasonló. Itt is a `\\` parancs használatos a sortörésre.

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

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

A következő példa az `array` környezet egy másik alkalmazását mutatja. Csak egy zárójelet kapunk, mivel a `\right` paramétere a „.” láthatatlan zárójel.

$$y = \left\{ \begin{array}{ll} a & \textrm{ha $d>c$}\\ b+x & \textrm{reggel}\\ l & \textrm{este} \end{array} \right.$$

```
\begin{displaymath}
y = \left\{ \begin{array}{ll}
 a & \textrm{ha $d>c$}\\
 b+x & \textrm{reggel}\\
 l & \textrm{este}
  \end{array} \right.
\end{displaymath}
```

Többsoros formulák, vagy egyenletrendszerek számára az `eqnarray` és az `eqnarray*` környezet javasolt az `equation` helyett. Az `eqnarray` megszámoz minden sort. Az `eqnarray*` környezetben nincsenek sorszámok.

Az `eqnarray` és az `eqnarray*` környezet `{rcl}` paraméterű három oszlopos táblázatként funcionál, ahol a középső oszlop az egyenlőségjelet, vagy a nem egyenlő jelet tartalmazza, de bármilyen más jelet is idehelyezhetünk. A `\\` parancs jelöli a sortörést.

$$\begin{eqnarray}
f(x) & = & \cos x \\
f'(x) & = & -\sin x \\
\int_{0}^{x} f(y)\mathrm{d}y & = & \sin x
\end{eqnarray}$$

(3.5) (3.6) (3.7)

```
\begin{eqnarray}
f(x) & = & \cos x      \\
f'(x) & = & -\sin x    \\
\int_{0}^{x} f(y)\mathrm{d}y &
 = & \sin x
\end{eqnarray}
```

Megjegyzendő, hogy a középső oszlop mindkét oldalán elég nagy hely van. Módosíthatunk ezen a `\setlength\arraycolsep{2pt}` paranccsal, ahogy a következő példában látható.

A hosszú egyenletek nem tördelődnek automatikusan. A töréspontot nekünk kell meghatározni, általában a következő módozatok valamelyikével:

$$\begin{eqnarray}
\sin x & = & x - \frac{x^{3}}{3!} + \frac{x^{5}}{5!} - {} \nonumber\\
& & {} - \frac{x^{7}}{7!} + {} \cdots \qquad (3.8)
\end{eqnarray}$$

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

$$\begin{eqnarray}
\lefteqn{ \cos x = 1 - \frac{x^{2}}{2!} + {} } \nonumber\\
& & {} + \frac{x^{4}}{4!} - \frac{x^{6}}{6!} + {} \cdots \qquad (3.9)
\end{eqnarray}$$

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

A LaTeX a `\nonumber` hatására nem számozza az adott sort.

Kicsit nehézkes megoldani az egyenletek törését az ismertetett módszerekkel; az igazi megoldást a jóval hatékonyabb amsmath csomagban találjuk.

3.6. Betűméretezés matematikai módban

A TeX a formula tartalmának megfelelően választja meg a betűméreteket. Például láthattuk, hogy az indexeket automatikusan kisebb méretben szedi. Ha az álló betűk elérése végett a `\textrm` parancsot használjuk, a betűméretezés nem működik. Annak „köszönhető" ez, hogy a `\textrm` tulajdonképpen időleges visszatérés a nem matematikai módba. Használjuk a `\mathrm` parancsot, ha betűméretezést is szeretnénk.[5] Vigyázzunk, mivel a `\mathrm` parancs paraméterében hatástalanok a szóközök és hibajelzést okoznak az ékezetes karakterek. (Nem matematikai módban használjuk inkább a `\textsuperscript` parancsot.)

5. Az AMS-LaTeX használatával méretező `\textrm` parancsot kapunk.

$$12^{\text{h}} \quad 12^{\mathrm{h}} \tag{3.10}$$

```
\begin{equation}
12^\textrm{h} \quad
12^\mathrm{h}
\end{equation}
```

Időnként szükség lehet arra is, hogy mi határozzuk meg a betűméretet. Matematikai módban erre a következő négy parancs szolgál:

`\displaystyle` $\displaystyle(123)$, `\textstyle` $\textstyle(123)$, `\scriptstyle` $\scriptstyle(123)$ és a `\scriptscriptstyle` $\scriptscriptstyle(123)$.

A TEX `\displaystyle` nélkül a szummajel mellé helyezné az indexeket a következő példában.

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

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

A fenti eset példázza, hogy kisebb zárójelekre is szükség lehet, mint amit a standard `\left[ \right]` parancs eredményez. A `\mathop` parancs beállítja a függvénynév és a zárójel közti helyes térközt, azzal, hogy a paraméteréből nagyoperátort csinál. Ezzel a paranccsal írhatunk például kérdőjelet egyenlőségjelek fölé.

3.7. Kifejezések magyarázata

A kifejezésekhez magyarázatot fűzhetünk, ahol megnevezzük a kifejezésben szereplő változókat. A következő példa ezt taglalja:

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

Ahol: a, b – a derékszögű háromszög befogói.

c – a háromszög átfogója.

```
\begin{displaymath}
a^2+b^2=c^2
\end{displaymath}
{\settowidth{\parindent}
   {Where:\ }

\makebox[0pt][r]
 {Ahol:\ }$a$ és $b$ egy
 derékszögű háromszög befogói.

$c$ a háromszög átfogója.}
```

Ha gyakran használunk hasonló szerkezetet, érdemes új környezetet definiálni a `\newenvironment` paranccsal. (Lapozzunk vissza a 29. oldalra.)

3.8. Tételek, szabályok, ...

Ha matematikai tárgyú műveket írunk, valószínűleg szükség lesz „lemma”, „definíció”, „axióma” és egyéb hasonló struktúrákra. A LaTeX a

`\newtheorem{`*azonosító*`}[`*számláló*`]{`*név*`}[`*szakasz*`]`

paranccsal könnyíti meg az ilyen szerkezetek létrehozását.

Az *azonosító* paraméter egy rövid név, amivel hivatkozunk majd a „tételünkre”. A *név* paraméter tartalma fog majd megjelenni a szövegben. A kapcsos zárójeles paraméterek opcionálisak, feladatuk a tételféleség számozásának beállítása. A *számláló* paraméter tartalma egy előzőleg deklarált „tétel”, aminek a számozását követi az új tételünk. A *szakasz* paraméter tartalma egy szakasztípus, ami alapján számozni szeretnénk tételtípusunkat.

Miután deklaráltuk a tételtípusainkat a preambulumban, ezeket a következő paranccsal használhatjuk a szövegen belül:

`\begin{`*azonosító*`}[`*szöveg*`]`
Ez az én izgalmas tételem
`\end{`*azonosító*`}`

Ennyi elmélet elégséges is. A következő két példa remélhetőleg érthetővé teszi az elsőre talán kevésbé világos `\newtheorem` környezetet.

1. Törvény. *Ne bújj el tanúként*

2. Esküdt (A Tizenkét). *Te is lehetsz az! Légy óvatos, és lásd az 1-t.*

3. Törvény. *Nem, nem, nem*

```
% preambulum
\newtheorem{law}{Törvény}
\newtheorem{jury}[law]{Esküdt}
%a szöveg törzsében
\begin{law} \label{law:box}
Ne bújj el tanúként
\end{law}
\begin{jury}[A Tizenkét]
Te is lehetsz az! Légy óvatos,
és lásd az \ref{law:box}-t.
\end{jury}
\begin{law}Nem, nem, nem\end{law}
```

Az „Esküdt” tétel ugyanazt a számlálót használja, mint a „Törvény”. Az eredmény az, hogy a „Esküdt” beilleszkedik a „Törvény”-ek sorába. A kapcsos zárójelbe megadhatjuk a tétel címét, vagy valami hasonlót.

3.8.1. Murphy. *Ami elromolhat, az el is romlik.*

```
\newtheorem{mur}{Murphy}[section]
\begin{mur} Ami elromolhat,
az el is romlik. \end{mur}
```

A „Murphy” tétel számozása az aktuális szakasz számát is magába foglalja. Nem csak a `section`, hanem például a `chapter` és `subsection` paraméterek is gyakran használatosak.

3.9. Félkövér szimbólumok

A félkövér szimbólumok előhívása kicsit bonyolult a LaTeX-ben, lehet, hogy szándékosan, megóvva túlzott használatuktól az amatőr tipográfusokat. A méretváltást nem akadályozó `\mathbf` parancs ugyan félkövér betűket ad, de csak álló betűket. A `\boldmath` parancsot használhatjuk, de csak *nem matematikai módban.* A parancs a szimbólumokkal is működik.

$$\mu, M \qquad \mathbf{M} \qquad \boldsymbol{\mu, M}$$

```
\begin{displaymath}
\mu, M \qquad \mathbf{M} \qquad
\mbox{\boldmath $\mu, M$}
\end{displaymath}
```

A példában a vessző is félkövér. Általában ez nem kívánatos a gyakorlatban.

Az `amsbsy` csomag (az `amsmath` részeként) megkönnyíti a dolgunk. Rendelkezésünkre bocsátja a `\boldsymbol` parancsot, és a „szegény ember félkövér szimbólum-generátorát” a `\pmb`[6] parancsot. Az utóbbit csakis végszükség esetén, a félkövér szimbólum fontkészlet hiányakor használjuk!

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

```
\begin{displaymath}
\mu, M \qquad
\boldsymbol{\mu}, \boldsymbol{M}
\qquad \pmb{\mu}, \pmb{M}
\end{displaymath}
```

6. *poor man's bold*

3.10. A matematikai szimbólumok listái

A következő táblázatok az összes olyan szimbólumot tartalmazzák, amelyhez *matematikai módban* hozzáférhetünk.

A 3.12–3.16. táblázatok[7] szimbólumai az amssymb csomag használatával érhetők el, feltéve, ha az AMS fontok telepítve vannak. Ha az AMS csomag és fontkészlete nincsen még telepítve a rendszerünkön, letölthetjük a következő címről:
`CTAN:/tex-archive/macros/latex/packages/amslatex`

3.1. táblázat. Ékezetek matematikai módban

$\hat{a}$	`\hat{a}`	$\check{a}$	`\check{a}`	$\tilde{a}$	`\tilde{a}`	$\acute{a}$	`\acute{a}`
$\grave{a}$	`\grave{a}`	$\dot{a}$	`\dot{a}`	$\ddot{a}$	`\ddot{a}`	$\breve{a}$	`\breve{a}`
$\bar{a}$	`\bar{a}`	$\vec{a}$	`\vec{a}`	$\widehat{A}$	`\widehat{A}`	$\widetilde{A}$	`\widetilde{A}`

3.2. táblázat. Görög kisbetük

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

3.3. táblázat. Görög nagybetük

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

7. A táblázatok eredetileg David Carlisle `symbols.tex` állományából származtak, majd Josef Tkadlec felvetésére alaposon módosítva lettek.

3.4. táblázat. Relációs jelek

Egyszerűen képezhetők a következő relációk negációi a parancsok elé tett `\not` paranccsal.

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

[a] Elérhető a latexsym csomag használatával

3.5. táblázat. Műveleti jelek

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

3.6. táblázat. Nagyoperátorok

$\sum$	`\sum`	$\bigcup$	`\bigcup`	$\bigvee$	`\bigvee`	$\bigoplus$	`\bigoplus`
$\prod$	`\prod`	$\bigcap$	`\bigcap`	$\bigwedge$	`\bigwedge`	$\bigotimes$	`\bigotimes`
$\coprod$	`\coprod`	$\bigsqcup$	`\bigsqcup`			$\bigodot$	`\bigodot`
$\int$	`\int`	$\oint$	`\oint`			$\biguplus$	`\biguplus`

3.7. táblázat. Nyilak

$\leftarrow$	`\leftarrow` or `\gets`	$\longleftarrow$	`\longleftarrow`	$\uparrow$	`\uparrow`
$\rightarrow$	`\rightarrow` or `\to`	$\longrightarrow$	`\longrightarrow`	$\downarrow$	`\downarrow`
$\leftrightarrow$	`\leftrightarrow`	$\longleftrightarrow$	`\longleftrightarrow`	$\updownarrow$	`\updownarrow`
$\Leftarrow$	`\Leftarrow`	$\Longleftarrow$	`\Longleftarrow`	$\Uparrow$	`\Uparrow`
$\Rightarrow$	`\Rightarrow`	$\Longrightarrow$	`\Longrightarrow`	$\Downarrow$	`\Downarrow`
$\Leftrightarrow$	`\Leftrightarrow`	$\Longleftrightarrow$	`\Longleftrightarrow`	$\Updownarrow$	`\Updownarrow`
$\mapsto$	`\mapsto`	$\longmapsto$	`\longmapsto`	$\nearrow$	`\nearrow`
$\hookleftarrow$	`\hookleftarrow`	$\hookrightarrow$	`\hookrightarrow`	$\searrow$	`\searrow`
$\leftharpoonup$	`\leftharpoonup`	$\rightharpoonup$	`\rightharpoonup`	$\swarrow$	`\swarrow`
$\leftharpoondown$	`\leftharpoondown`	$\rightharpoondown$	`\rightharpoondown`	$\nwarrow$	`\nwarrow`
$\rightleftharpoons$	`\rightleftharpoons`	$\iff$	`\iff` (nagyobb térközök)	$\leadsto$	`\leadsto` [a]

[a] Elérhető a latexsym csomag használatával

3.8. táblázat. Zárójelek

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

3.9. táblázat. Nagy zárójelek

$\lgroup$	`\lgroup`	$\rgroup$	`\rgroup`	$\lmoustache$	`\lmoustache`	$\rmoustache$	`\rmoustache`
$\arrowvert$	`\arrowvert`	$\Arrowvert$	`\Arrowvert`	$\bracevert$	`\bracevert`		

3.10. táblázat. Egyéb szimbólumok

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

[a] Elérhető a latexsym csomag használatával

3.11. táblázat. Nem matematikai szimbólumok

Ezek a szimbólumok szöveges módban is használhatók.

†	\dag	§	\S	©	\copyright
‡	\ddag	¶	\P	£	\pounds

3.12. táblázat. AMS zárójelek

$\ulcorner$	\ulcorner	$\urcorner$	\urcorner	$\llcorner$	\llcorner	$\lrcorner$	\lrcorner

3.13. táblázat. AMS görög és héber betük

$\digamma$	\digamma	$\varkappa$	\varkappa	$\beth$	\beth	$\daleth$	\daleth	$\gimel$	\gimel

3.14. táblázat. AMS Relációs jelek

$\lessdot$	\lessdot	$\gtrdot$	\gtrdot	$\doteqdot$	\doteqdot or \Doteq
$\leqslant$	\leqslant	$\geqslant$	\geqslant	$\risingdotseq$	\risingdotseq
$\eqslantless$	\eqslantless	$\eqslantgtr$	\eqslantgtr	$\fallingdotseq$	\fallingdotseq
$\leqq$	\leqq	$\geqq$	\geqq	$\eqcirc$	\eqcirc
$\lll$	\lll or \llless	$\ggg$	\ggg or \gggtr	$\circeq$	\circeq
$\lesssim$	\lesssim	$\gtrsim$	\gtrsim	$\triangleq$	\triangleq
$\lessapprox$	\lessapprox	$\gtrapprox$	\gtrapprox	$\bumpeq$	\bumpeq
$\lessgtr$	\lessgtr	$\gtrless$	\gtrless	$\Bumpeq$	\Bumpeq
$\lesseqgtr$	\lesseqgtr	$\gtreqless$	\gtreqless	$\thicksim$	\thicksim
$\lesseqqgtr$	\lesseqqgtr	$\gtreqqless$	\gtreqqless	$\thickapprox$	\thickapprox
$\preccurlyeq$	\preccurlyeq	$\succcurlyeq$	\succcurlyeq	$\approxeq$	\approxeq
$\curlyeqprec$	\curlyeqprec	$\curlyeqsucc$	\curlyeqsucc	$\backsim$	\backsim
$\precsim$	\precsim	$\succsim$	\succsim	$\backsimeq$	\backsimeq
$\precapprox$	\precapprox	$\succapprox$	\succapprox	$\vDash$	\vDash
$\subseteqq$	\subseteqq	$\supseteqq$	\supseteqq	$\Vdash$	\Vdash
$\Subset$	\Subset	$\Supset$	\Supset	$\Vvdash$	\Vvdash
$\sqsubset$	\sqsubset	$\sqsupset$	\sqsupset	$\backepsilon$	\backepsilon
$\therefore$	\therefore	$\because$	\because	$\varpropto$	\varpropto
$\shortmid$	\shortmid	$\shortparallel$	\shortparallel	$\between$	\between
$\smallsmile$	\smallsmile	$\smallfrown$	\smallfrown	$\pitchfork$	\pitchfork
$\vartriangleleft$	\vartriangleleft	$\vartriangleright$	\vartriangleright	$\blacktriangleleft$	\blacktriangleleft
$\trianglelefteq$	\trianglelefteq	$\trianglerighteq$	\trianglerighteq	$\blacktriangleright$	\blacktriangleright

3.15. táblázat. AMS Nyilak

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

3.16. táblázat. AMS Negált relációs jelek és nyilak

$\nless$	`\nless`	$\ngtr$	`\ngtr`	$\varsubsetneqq$	`\varsubsetneqq`
$\lneq$	`\lneq`	$\gneq$	`\gneq`	$\varsupsetneqq$	`\varsupsetneqq`
$\nleq$	`\nleq`	$\ngeq$	`\ngeq`	$\nsubseteqq$	`\nsubseteqq`
$\nleqslant$	`\nleqslant`	$\ngeqslant$	`\ngeqslant`	$\nsupseteqq$	`\nsupseteqq`
$\lneqq$	`\lneqq`	$\gneqq$	`\gneqq`	$\nmid$	`\nmid`
$\lvertneqq$	`\lvertneqq`	$\gvertneqq$	`\gvertneqq`	$\nparallel$	`\nparallel`
$\nleqq$	`\nleqq`	$\ngeqq$	`\ngeqq`	$\nshortmid$	`\nshortmid`
$\lnsim$	`\lnsim`	$\gnsim$	`\gnsim`	$\nshortparallel$	`\nshortparallel`
$\lnapprox$	`\lnapprox`	$\gnapprox$	`\gnapprox`	$\nsim$	`\nsim`
$\nprec$	`\nprec`	$\nsucc$	`\nsucc`	$\ncong$	`\ncong`
$\npreceq$	`\npreceq`	$\nsucceq$	`\nsucceq`	$\nvdash$	`\nvdash`
$\precneqq$	`\precneqq`	$\succneqq$	`\succneqq`	$\nvDash$	`\nvDash`
$\precnsim$	`\precnsim`	$\succnsim$	`\succnsim`	$\nVdash$	`\nVdash`
$\precnapprox$	`\precnapprox`	$\succnapprox$	`\succnapprox`	$\nVDash$	`\nVDash`
$\subsetneq$	`\subsetneq`	$\supsetneq$	`\supsetneq`	$\ntriangleleft$	`\ntriangleleft`
$\varsubsetneq$	`\varsubsetneq`	$\varsupsetneq$	`\varsupsetneq`	$\ntriangleright$	`\ntriangleright`
$\nsubseteq$	`\nsubseteq`	$\nsupseteq$	`\nsupseteq`	$\ntrianglelefteq$	`\ntrianglelefteq`
$\subsetneqq$	`\subsetneqq`	$\supsetneqq$	`\supsetneqq`	$\ntrianglerighteq$	`\ntrianglerighteq`
$\nleftarrow$	`\nleftarrow`	$\nrightarrow$	`\nrightarrow`	$\nleftrightarrow$	`\nleftrightarrow`
$\nLeftarrow$	`\nLeftarrow`	$\nRightarrow$	`\nRightarrow`	$\nLeftrightarrow$	`\nLeftrightarrow`

3.17. táblázat. AMS Műveleti jelek

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

3.18. táblázat. Egyéb AMS szimbólumok

$\hbar$	`\hbar`	$\hslash$	`\hslash`	$\Bbbk$	`\Bbbk`
$\square$	`\square`	$\blacksquare$	`\blacksquare`	$\circledS$	`\circledS`
$\vartriangle$	`\vartriangle`	$\blacktriangle$	`\blacktriangle`	$\complement$	`\complement`
$\triangledown$	`\triangledown`	$\blacktriangledown$	`\blacktriangledown`	$\Game$	`\Game`
$\lozenge$	`\lozenge`	$\blacklozenge$	`\blacklozenge`	$\bigstar$	`\bigstar`
$\angle$	`\angle`	$\measuredangle$	`\measuredangle`	$\sphericalangle$	`\sphericalangle`
$\diagup$	`\diagup`	$\diagdown$	`\diagdown`	$\backprime$	`\backprime`
$\nexists$	`\nexists`	$\Finv$	`\Finv`	$\varnothing$	`\varnothing`
$\eth$	`\eth`	$\mho$	`\mho`		

3.19. táblázat. Matematikai stílusok

Példa	Parancs	Szükséges csomag
ABCdef	`\mathrm{ABCdef}`	
ABCdef	`\mathit{ABCdef}`	
$\mathnormal{ABCdef}$	`\mathnormal{ABCdef}`	
$\mathcal{ABC}$	`\mathcal{ABC}`	
$\mathscr{ABC}$	`\mathcal{ABC}`	eucal a mathcal opcióval
vagy	`\mathscr{ABC}`	eucal a mathscr opcióval
$\mathfrak{ABCdef}$	`\mathfrak{ABCdef}`	eufrak
$\mathbb{ABC}$	`\mathbb{ABC}`	amsfonts vagy az amssymb

4. fejezet

Specialitások

Ne olvassuk el ezt a fejezetet! – vagy legalábbis addíg ne, amíg nem rendelkezünk az előző fejezetek anyagán alapuló biztos tudással. Az utolsó fejezet némi nyalánksággal szolgál a LaTeX témaköréből. A teljes tudást azonban – amit az irodalomjegyzék első három kötete foglal magába – semmiképp sem pótolja.

4.1. Betűkészletek és -méretek

A LaTeX automatikusan választja meg a betűméreteket a kiadvány logikai szerkezetének megfelelően. Néha szükség lehet a manuális beállításra. A 4.1. és a 4.2. táblázat tartalmazza az ehhez szükséges parancsokat. A betűtípusok tényleges mérete a függ a típus formájától, valamint a dokumentumosztálytól és ennek beállításaitól.

A kicsi és **kövér** antikok nem halogatták a nagy *döntéseket*.

```
{\small A kicsi és
\textbf{kövér} antikok nem
halogatták {\Large a nagy
\textit{döntéseket}.}}
```

A LaTeX 2_ε lényeges sajátossága, hogy az említett parancsok a betűtípus stílusától függetlenek. Ez azt jelenti, hogy a betűtípus, vagy a betűméretek módosítása mellett továbbra is megmarad a *stílus*, vagyis a kurzív, döntött, félkövér betűváltozatok használata. A LaTeX-hel ma ismerkedőknek ez evidensnek tűnhet, de a LaTeX 2.09 még nem így működött.

Matematikai módban is használhatunk betűkészlet-váltó parancsokat, ha időlegesen visszalépünk a szöveges módba. Ha azonban a formulákban válnak szükségessé más betűkészletek, a 4.3. táblázatban összefoglalt speciális parancsokat használjuk.

A betűméretező parancsokhoz kapcsolódóan fontos szerepe van a csipkés zárójelezésnek. A csipkés zárójelek a zárójelen belülire korlátozzák a legtöbb LaTeX parancs hatását.

4.1. táblázat. Betűkészletek

`\textrm{...}`	antikva	`\textsf{...}`	groteszk
`\texttt{...}`	`írógép`		
`\textmd{...}`	médium[a]	`\textbf{...}`	**félkövér**
`\textup{...}`	álló	`\textit{...}`	*kurzív*
`\textsl{...}`	*döntött*	`\textsc{...}`	KISKAPITÁLIS
`\emph{...}`	*idézett*	`\textnormal{...}`	szöveg

[a] félúton a normál és a félkövér betűváltozat között

4.2. táblázat. Betűméretek

`\tiny`	nagyon apró betűméret	`\Large`	nagyobb
`\scriptsize`	apró betűméret	`\LARGE`	nagy
`\footnotesize`	lábjegyzet méret	`\huge`	hatalmas
`\small`	kisebb betűméret	`\Huge`	óriási
`\normalsize`	normál betűméret		
`\large`	nagy betűméret		

4.3. táblázat. Matematikai betűkészletek

Parancs	*Példa*	*Eredmény*
`\mathcal{...}`	`$\mathcal{B}=c$`	$\mathcal{B}=c$
`\mathrm{...}`	`$\mathrm{K}_2$`	K_2
`\mathbf{...}`	`$\sum x=\mathbf{v}$`	$\sum x=\mathbf{v}$
`\mathsf{...}`	`$\mathsf{G\times R}$`	$\mathsf{G\times R}$
`\mathtt{...}`	`$\mathtt{L}(b,c)$`	$\mathtt{L}(b,c)$
`\mathnormal{...}`	`$\mathnormal{R_{19}}\neq R_{19}$`	$\mathnormal{R_{19}}\neq R_{19}$
`\mathit{...}`	`$\mathit{ffi}\neq ffi$`	$\mathit{ffi}\neq ffi$

Kedveli a nagy és a kicsi betűket.

```
Kedveli a {\LARGE nagy és
a {\small kicsi} betűket}.
```

A betűméretező parancsok megváltoztatják a sorközök nagyságát is, de csak akkor, ha a bekezdés is véget ér a módosított méretű szövegrészben. Ennek megfelelően a `}` zárójelet nem tehetjük közvetlenül a szöveg után. Figyeljük meg a `\par` bekezdés-záró parancs helyzetét a következő két példában.

Ne olvasd el! Nem igaz. Higgy nekem!

```
{\Large Ne olvasd el! Nem igaz.
Higgy nekem! \par}
```

Ez nem igaz. De ne felejtsd el, hogy hazudok!

```
{\Large Ez nem igaz. De ne
felejtsd el, hogy hazudok!}\par
```

A betűkészletek és méretek világában tett utazást fejezzük be egy jótanáccsal:[1]

> **Ne felejtsük!** *Minél* TÖBB betűkészletet használunk a `kiadványunkban`, *az* annál OLVASHATÓBBÁ és *szebbé* válik.

4.2. Térközök

4.2.1. Sorritkítás

Ha nagyobb sorközöket akarunk, használhatjuk a

`\linespread{`*faktor*`}`

parancsot a preambulumban. A `\linespread{1.3}` paranccsal másfeles sortávolság érhető el. Dupla sorközt eredményez a `\linespread{1.6}` parancs. Alapesetben a sortávolság egyszeres, vagyis a *faktor* értéke 1. Szélesebb körben használható sorritkítást az 5.6. szakasz ismertet.

4.2.2. Bekezdés formázás

A LaTeXben két változó szabja meg a bekezdések megjelenését. A behúzás méretét a `\parindent` változó tartalmazza, a `\parskip` pedig a bekezdések közti helykihagyás nagyságát. A preambulumban elhelyezett

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

1. Figyelmeztetés: Ez csak vicc, reméljük senki nem veszi komolyan!

parancsokkal módosíthatjuk a bekezdés megjelenését. Az említett változóknak a `\setlength` paranccsal adhatunk új értéket. A példa két sora hatására megszűnik a bekezdés eleji behúzás és megnövekszik a bekezdések távolsága. (Európában nem ritka az ilyen szedés.) Vigyázzunk, mert a tartalomjegyzék soraira is vonatkozni fog a módosítás, növelve a sorközöket.

Ha behúzást szeretnénk egy különben behúzás nélküli bekezdésben, az

```
\indent
```

parancsot használjuk a bekezdés elején.[2] Természetesen ez csak akkor működik, ha a `\parindent` változó értéke nem zérus.

Ha csak egy bekezdés behúzását szeretnénk letiltani, használjuk a

```
\noindent
```

parancsot a bekezdés előtt. Leggyakrabban mondatba illesztett `displaymath` formulák, és általunk definiált szakaszkezdő sorok esetében használt.

4.2.3. Vízszintes helykihagyás

A LaTeX automatikusan állítja be a szavak és mondatok közti teret. A vízszintes térköz módosítására a

```
\hspace{hossz}
```

parancs használható. Ahhoz, hogy a térközünk ne szóközként viselkedjen, vagyis a sorok végén és a sortörést követő sor elején mindenképp megjelenjen, használjuk a `\hspace*` változatot. A *hossz* a legegyszerűbb esetben egy szám és egy hozzá tartozó mértékegység. A legfontosabb mértékegységek a 4.4. táblázatban vannak felsorolva.

Egy másfél centis térköz.	`Egy\hspace{1.5cm}másfél centis térköz.`

Speciális rugalmas térközt állíthatunk elő a

```
\stretch{n}
```

paranccsal. A térköz addig nyúlik, míg az adott sorban van elég hely. Ha több `\hspace{\stretch{n}}` parancsot használunk egy sorban, a rugalmas térközök arányát az n faktorral szabályozhatjuk.

2. Az összes szakasz első bekezdésének behúzására célszerű a `tools` gyűjtemény indentfirst csomagját alkalmazni.

4.4. táblázat. TEX mértékegységek

`mm`	milliméter ≈ 1/25 hüvelyk
`cm`	centiméter = 10 mm
`in`	hüvelyk ≈ 25.4 mm
`pt`	Pica pont ≈ 1/72 hüvelyk ≈ $\frac{1}{3}$ mm
`em`	az aktuális betűkészlet `M` betűjének szélessége
`ex`	az aktuális betűkészlet `x` betűjének magassága

x x x

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

4.2.4. Függőleges helykihagyás

A bekezdéseket és a kisebb-nagyobb szakaszokat elválasztó térközök nagysága automatikusan kerül megállapításra. Ha szükséges, a

`\vspace{`*hossz*`}`

paranccsal növelhetjük a függőleges térközöket.[3]

A parancsot általában két üres sor közé helyezzük el a LATEX forrásban. Ha szeretnénk, hogy a helykihagyás a lap tetején és az alján is megjelenjen, használjuk a `\vspace*` parancsot.

```
Valami szöveg\ldots

\vspace{\stretch{1}}
Ez az oldal utolsó sorává válik.\pagebreak
```

Egy paragrafuson, vagy táblázaton *belül* a

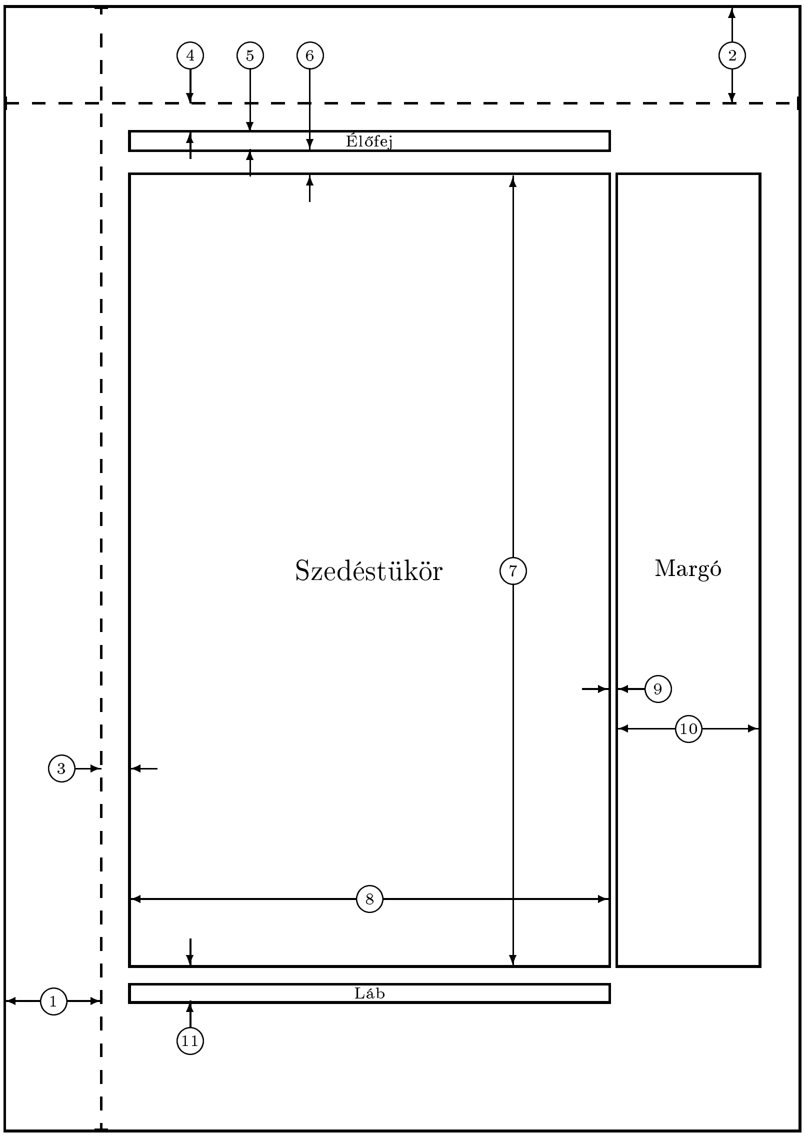
`\\[`*hossz*`]`

paranccsal változtathatunk a következő sorközön.

4.3. Oldalbeállítások

A LATEX 2_ε lehetővé teszi, hogy a `\documentclass` parancs paramétereként beállítsuk a papírméretet. A papírmérettel járó előre megadott értékek

3. Erősen javasolt a `\smallskip`, `\medskip`, `\bigskip` parancsok használata kis térközök esetén. (*ford.*)



1	`egy hüvelyk + \hoffset`	2	`egy hüvelyk + \voffset`
3	`\oddsidemargin = 22pt`	4	`\topmargin = 22pt`
5	`\headheight = 13pt`	6	`\headsep = 19pt`
7	`\textheight = 595pt`	8	`\textwidth = 360pt`
9	`\marginparsep = 7pt`	10	`\marginparwidth = 106pt`
11	`\footskip = 27pt`		`\marginparpush = 5pt (rejtve)`
	`\hoffset = 0pt`		`\voffset = 0pt`
	`\paperwidth = 597pt`		`\paperheight = 845pt`

4.1. ábra. Oldalbeállító paraméterek

többnyire megfelelőek, de időnként szükséges lehet a módosításukra. margók

A 4.1. ábra mutatja a módosítható paramétereket. Az ábrát a layout csomag állította elő.[4]

A LaTeX két parancsa szolgál a paraméterek értékének módosítására. Ezeket általában a preambulum részben helyezzük el.

Az első parancs értéket ad a választott paraméternek:

`\setlength{`*paraméter*`}{`*hossz*`}`

A második parancs segítségével a paraméter értékét növelhetjük *hossz* mérettel.

`\addtolength{`*paraméter*`}{`*hossz*`}`

A második parancs hasznosabb, mint a `\setlength`, mivel relatív módosításokat hajthatunk vele végre. Például a szövegtükör szélessége a következő, preambulumba elhelyezett parancsokkal lett növelve 1 cm-rel ebben a kiadványban:

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

A `\hoffset` és `\voffset` paraméterek a lap bal felső sarkától vett eltolások, melyekhez minden oldalparaméter viszonyítva van. Ezek módosításával egyszerűen beállíthatjuk a szövegtükör elhelyezkedését a lapon.

4.4. Irodalomjegyzék

A `thebibliography` környezettel állíthatunk elő irodalomjegyzéket. Minden irodalom megadása a

`\bibitem{`*címke*`}`

paranccsal kezdődik a környezeten belül.

A *címke* segítségével hivatkozhatunk az egyes irodalmakra.

`\cite{`*címke*`}`

Az irodalmak számozása automatikusan történik. A `thebibliography` környezet még egy példacímkét is igényel, amelyből a környezet megállapítja a számozások maximális szélességét.[5]

4. `CTAN:/tex-archive/macros/latex/packages/tools`
5. 10-nél kevesebb irodalomnál 9, 100-nál 99, 1000-nél 999-et célszerű megadni. A példában 99 szerepel.

Partl [1] javasolta, hogy …

```
Partl~\cite{pa}
javasolta, hogy \ldots

\begin{thebibliography}{99}
\bibitem{pa} H.~Partl:
\emph{German \TeX},
TUGboat Vol.~9, No.~1 ('88)
\end{thebibliography}
```

Irodalomjegyzék

[1] H. Partl: *German TEX*, TUGboat Vol. 9, No. 1 ('88)

4.5. Tárgymutató

A LaTeX-hel és a *makeindex* kiegészítő program[6] segítségével könnyedén állíthatunk elő név- és tárgymutatókat.[7] Itt csak a legalapvetőbb indexelési lehetőségekről esik szó.

A hivatkozások használatának első lépése, hogy betöltsük a makeidx csomagot a preambulumban:

```
\usepackage{makeidx}
```

és utána közvetlenül kiadjuk a

```
\makeindex
```

parancsot, mely egy indexfájl létrehozására szólítja fel a LaTeX-et.

A tárgymutatót az

```
\index{tárgyszó}
```

paranccsal bővíthetjük. A tárgymutatóban a *tárgyszó* és a tárgyszó előfordulásának, pontosabban az előző parancs kiadásának helye fog szerepelni. A *tárgyszó* nem csak egyszerű tárgyszó, hanem összetett kulcs is lehet, melynek szintaxisát a 4.5. táblázat foglalja össze példákkal.

Ha a LaTeX forrásállományt lefordítjuk, minden `\index` parancs beleírja a tárgyszót és a hozzá tartozó oldalszámot egy speciális indexfájlba. A fájlnak

6. Ahol az operációs rendszer csak 8 betűs nevek használatát teszi lehetővé, *makeindx*.
7. Lásd még 6.1.4. szakasz.

4.5. táblázat. Példa indexelésre

Példa	Index	Megjegyzés
`\index{helló}`	helló, 1	Sima tárgyszó
`\index{helló!Péter}`	Péter, 3	Tárgyszó a hellón belül
`\index{Samu@\textsl{Samu}}`	*Samu*, 2	Formázott tárgyszó
`\index{Lina@\textbf{Lina}}`	**Lina**, 7	Ua. mint felül
`\index{Jani|textbf}`	Jani, **3**	Formázott oldalszám
`\index{Józsi|textit}`	Józsi, *5*	Ua. mint felül

ugyanaz a neve, mint a LaTeX forrásállománynak, csak más (`.idx`) a kiterjesztése. Ezt az indexfájlt dolgozza fel a *makeindex* program, ha kiadjuk az operációs rendszerben a

`makeindex` *indexfájl-név*

parancsot.

A `makeindex` program eredménye a tárgyjegyzék, amely nem más, mint a tárgyszavakat és oldalszámaikat rendezve tartalmazó LaTeX forrásállomány, bár nem `.tex`, hanem `.ind` kiterjesztéssel rendelkezik. Ha az eredeti forrásállományunkat mégegyszer lefordítjuk, a tárgyjegyzéket tartalmazó `.ind` állomány a

```
\printindex
```

paranccsal jelölt helyre fog betöltődni.

A showidx csomagot használva a LaTeX minden tárgyszót feltüntet a szöveg bal margóján. Ez nagyon hasznos ellenőrzési lehetőséget nyújt.

4.6. EPS képek beillesztése

A `figure` és a `table` környezet lehetőséget nyújt arra is, hogy lebegő képeket helyezzünk el a kiadványunkban.

Egyszerűbb ábrákat pusztán a LaTeX-hel, illetve kiegészítő csomagjaival is előállíthatunk. Sajnos a legtöbb felhasználó meglehetősen bonyolultnak találja a `picture` és a hasonló környezetek működését, ezért itt egy egyszerűbb lehetőség kerül ismertetésre. Ha valaki többet szeretne tudni a témáról, az az ajánlott könyvekben nézhet utána.

Sokkal járhatóbb útnak igérkezik, hogy valamilyen célszoftverrel[8] előállított képet, rajzot használunk fel. A LaTeX csomagokkal többféle módon

8. mint például *XFig*, *CorelDraw!*, *Freehand*, *Gnuplot*, *XPaint*, *GIMP*, stb.

tehetjük meg ezt. A következő példákban EPS (Encapsulated PostScript) formátumú képek[9] beillesztéséről lesz szó, mivel ez meglehetősen könnyű és általánosan használt. Az EPS képek teljes körű használata vagy PostScript printert igényel, vagy valamilyen PostScript értelmező programét, mint például a *GhostScript* programot, melyet letölthetünk[10] a következő címről:
`CTAN:/tex-archive/support/ghostscript`

A D. P. Carlisle-től származó graphicx csomag hasznos parancsokkal szolgál az EPS beillesztésre. A csomag része a nagyobb graphics gyűjteménynek.[11]

Ahhoz, hogy a graphicx csomaggal, valamint PostScript printerrel, vagy értelmezővel felszerelt rendszeren képet illeszthessünk be LaTeX kiadványokba, lépésről lépésre kövessük az itt leírtakat:

1. Rajzprogramunkból mentsük el az ábránkat EPS formátumban.

2. Töltsük be a graphicx csomagot a LaTeX forrásunk preambulumában az

   `\usepackage[`*driver*`]{graphicx}`

   paranccsal, ahol a *driver* a dvi formátumot PostScript-re konvertáló programunk neve.[12]

3. Használjuk az

   `\includegraphics[`*változó*=*érték*, …`]{`*képfájl*`}`

   parancsot az EPS *képfile* beillesztésére. Lehetőség van egy olyan paraméter megadására is, ahol vesszővel elválasztott *változók*nak adhatunk *érték*et. A változók a beillesztendő kép szélességét, magasságát és elforgatásának mértékét írják le. A 4.6. táblázat tartalmazza a legfontosabb változókat.

A következő példa remélhetőleg mindent tisztáz:

```
\begin{figure}
\begin{center}
\includegraphics[angle=90, width=10cm]{teszt.eps}
\end{center}
\end{figure}
```

9. Megjegyzendő, hogy az említett szoftverek többségével, és a kiváló ImageMagick *convert* programmal más formátumú (`gif`, `jpeg`, stb.) képekből is tudunk EPS formátumú képet előállítani! (*ford.*)
10. A programot a felsorolt programok nagyobb részéhez hasonlóan megtaláljuk az egyetemi számítóközpontokban, és a szabad Linux disztribúciókban is. (*ford.*)
11. `CTAN:/tex-archive/macros/latex/packages/graphics`
12. A legáltalánosabban használt ilyen program a *dvips*.

4.6. táblázat. A graphicx csomag változói

`width`	az ábra méretezése a megadott szélességre
`height`	az ábra méretezése a megadott magasságra
`angle`	forgatás az óramutató járásának irányában

A beillesztett ábra a `teszt.eps` állomány. A képet előbb 90 fokkal elforgattuk és 10 centis szélességre nagyítottuk. Az ábra arányai nem torzulnak, mivel a magasságot nem jelöltük meg külön.

További információt találunk D. P. Carlisle gyűjteményében [8].

5. fejezet

A tipográfia mestersége

A számítógépes szövegszerkesztő programok megjelenése több okból is katasztrófális hatással volt a könyvkiadásra. Lehetővé vált a hozzá nem értők számára, hogy igénytelen és hibás kivitelű nyomdai termékeket állítsanak elő a silány minőségű programokkal. A tipográfia alapjai szerencsére elsajátíthatók, és rendelkezésünkre áll egy igényes szedőprogram is, a LaTeX.[1]

5.1. Amit az elején le kell szögezni

A nyomdászat nemzeti sajátosságai szükségessé teszik, hogy a LaTeX angolszász alapbeállításán változtassunk. A forrásállomány preambulumában, tehát a `\begin{document}` előtti részben már meg kell tennünk a következő módosításokat: (A magyar stílusfájl a következő beállításokat automatikusan elvégzi, továbbá a lábjegyzetek számozását is magyarítja, csak az `\usepackage{magyarul}` parancsot kell kiadnunk. Csak akkor rágjuk át magunkat a következőkön, ha nem áll rendelkezésünkre az említett stílusfájl.)

- Kapcsoljuk ki a `\frenchspacing` paranccsal a mondatvégi pontot követő nagyobb térközt.
- Használjuk az indentfirst csomagot, hogy a szakaszok első bekezdése is behúzással kezdődjék.
- A magyar elválasztást feltétlenül állítsuk be. A babel csomag `magyar` opcióval való betöltésével ez automatikusan megtörténik.[2]
- Egyéb praktikus beállításokról se feledkezzünk meg:

---

1. A kereskedelmi programok között a *QuarkXPress* program neve említendő meg. A LaTeX mellett szól, hogy szabad program, továbbá nem képesújságok, hanem matematikai, fizikai, vegyészeti, zenei, és egyéb tudományos kiadványok szedésére lett kiélezve.
2. Ha a rendszerbeállítás megtörtént. A magyar elválasztás teTeX környezetben való beállítását és a magyar stílusfájl használatát a következő fejezet ismerteti.

- Töltsük be a nemzeti karakterek használatát támogató `inputenc` csomagot a megfelelő (`latin2`, `cp852`) paraméterrel.
- A szabványos európai betűkódolást a `fontenc` csomag `T1` opciójával állíthatjuk be. Ha ezt megtettük, a belső idézőjeleket elérhetjük a `>>` és a `<<` jelpárokkal is.
- A papírméretet célszerű európai szabványnak megfelelő méretre, például A4-esre állítani.

A fenti módosítások kivitelezése (fordított sorrendben):

```
\usepackage[T1]{fontenc}
\usepackage[latin2]{inputenc}
\usepackage[magyar]{babel}
\usepackage{indentfirst}
\frenchspacing
```

5.2. Általános alapelvek

A tipográfia Paul Renner és Kner Imre meghatározásai nyomán egyszerre „mechanizált grafika”, és „szétszedhető betűk segítségével sokszorosított írás”. Az *írás* szó nyilvánvalóvá teszi, hogy a *helyesírás* is a tipográfia része.

Kevésbé ismert, hogy a helyesírás szabályai alapvető tipográfiai szabályokat is rögzítenek, például a később részletesebben ismertetésre kerülő írásjelhasználatot. Ha tehát valaki az idézőjelek helyett más nemzetek idézőjeleit, vagy egyéb jeleit használja, a magyar helyesírás szabályai ellen vét.

A következő szabályokat igyekezzünk betartani:

- Írjunk helyesen! Ha bizonytalanok vagyunk, forgassuk A magyar helyesírás szabályait.[3]
- Használjuk mindig a megfelelő írásjeleket! A következő példa a legtipikusabb hibákat és javításukat mutatja:

Legyen „idézet” az "idézet",
Béla bá’ a Béla bá′,
Vége… a vége...,
1994–98 az 1994-98,
5×6 az 5x6,
0,25 a 0.25 helyett!

```
Legyen
,,idézet'' az "idézet", \\
Béla bá' a Béla bá$'$, \\
Vége$\ldots$ a vége..., \\
1994--98 az 1994-98, \\
5$\times$6 az 5x6, \\
0,25 a 0.25 helyett!
```

3. Van rá remény, hogy a kiváló *Helyes-e?* magyar helyesírás-ellenőrző program szabad verziója elérhető lesz a LaTeX-es közösség számára is, a Morphologic cég jóvoltából.

- Használjuk a rendelkezésre álló betűváltozatokat! A kurzívot szövegközi kiemelésre; a kiskapitálist címekhez, vagy szerzők nevének kiemelésére; a félkövéret címekben.
- Ne használjuk egyszerre több hasonló betűtípus betűváltozatait! (A címekben, és a kiemelt szövegrészekben használjunk nyugodtan más betűtípusokat, de ezek legyenek díszbetűk, illetve groteszk (talpnélküli) betűtípusok, ahogy ebben a könyvben is látható.)
- A betűméret könyvekben 9-10 pontos; dolgozatokban 11-12 pontos.
- A sorokban 40 betűnél több és 80 betűnél kevesebb legyen.
- A sorritkítás a betűméretnél 10-20%-kal legyen nagyobb.

5.3. Írásjelek

Az írásjelek használatának alapjait „A magyar helyesírás szabályai”-ban találjuk meg. A nyomdai gyakorlat még további tipográfiai előírásokat is figyelembe vesz. Ha komolyabb mű szedésére vállalkoznánk, tanuljuk meg a szabályokat!

5.3.1. Szóköz

Fontos *írásjel* a szóköz, mellyel a szövegkészítéssel ismerkedők gyakran hadilábon állnak.

> „Szóközt hagyunk a szavak, valamint az írásjellel lezárt mondatok és tagmondatok között, a zárójelek és az idézőjelek közé foglalt közlési egységek előtt és után, a gondolatjel előtt és után stb. Nincs viszont szóköz a pont, a kérdőjel, a felkiáltójel, a vessző, a kettőspont, a pontosvessző előtt, továbbá a kötőjel és a nagykötőjel két oldalán; a kezdő zárójel és idézőjel hozzátapad az utána következő, a berekesztő zárójel és idézőjel pedig az előtte álló szóhoz stb.” (AMHSZ **239.**)

5.3.2. Idézőjelek

A kezdő idézőjel alul, a berekesztő idézőjel felül helyezkedik el, és mindkettő 99 alakú. Ha az idézet maga is tartalmaz idézetet, a belülre kerülő idézőjelek hegyével befelé forduló, >> és << alakú jelek. (Speciális eset az idézett párbeszédek idézőjel nélküli, csak gondolatjelekkel és sortörésekkel tagolt, többnyire szépirodalmi leírása. AMHSZ **258.**) Figyelmeztetés! A nemzetek idézőjelei meglehetősen nagy eltéréseket mutatnak:

magyar:	„nn »nn« nn”
angol:	‘nn “nn” nn’
amerikai:	“nn ‘nn’ nn”
német:	„nn ‚nn‘ nn“
francia:	« nn « nn » nn »

```
\begin{tabular}{rl}
magyar: & ,,nn >>nn<< nn''\\
angol: & `nn ``nn'' nn' \\
amerikai: & ``nn `nn' nn'' \\
német: & ,,nn ,nn` nn`` \\
francia: & << nn << nn >> nn >>
\end{tabular}
```

Ha nem használjuk a T1 kódolást, a belső idézőjelek a `\guillemotright` és a `\guillemotleft` parancsokkal érhetők el.

5.3.3. (Kis)kötőjel

Kötőjelet szavak és szórészek között az alábbi esetekben alkalmazunk:

- Mássalhangzó halmozódásnál (*sakk-kör*, *ott-tartózkodás*, *Széll-lel*).
- Szóismétléseknél, mellérendelő összetételeknél és ikerszóknál, valamint bizonyos alárendelő összetételeknél (*egy-egy*, *örökkön-örökké*, *süt-főz*, *szoba-konyhás*, *dimbes-dombos*, *meg-megállt*, *könyvritkaság-gyűjtemény*, *hidegvíz-csap*); továbbá szokatlan, költői szóösszetételekben (*gyémánt-hitemet*).
- Egymást követő azonos elő- vagy utótagú összetett szavak esetén (*gyors- és gépíró*; *tej-, zöldség- és gyümölcsfelhozatal*; *gépgyártó, -szerelő, és -javító üzem*).
- Kétezeren felüli számokban. az ezresek után (*kétezer-egy*, *hatvanötezer-ötszázharminchat*).
- Tulajdonnevekhez kapcsolt főnevek és belőlük képzett melléknevek esetén (*Afrika-kutató, József Attila-díjas*).
- Kettős családnevekben (*Konkoly-Thege*).
- Többelemű földrajzi nevek több típusában (*Arany-patak*, *Holt-Tisza*, *Arany-patak-völgy*).
- Az *-e* kérdőszócska esetén (*Szabad-e?*).
- Hozzávetőlegességet, vagylagosságot kifejező szókapcsolatokban (*egy-két* (ember), *nyolc-tíz* (napra), *5-6* (darab), *10-12* (éves), *5-10%*).

A legutolsó szabályt vö. a következő szakasz 3. pontjával!

5.3.4. Nagykötőjel

A nagykötőjel és a gondolatjel alakja megegyező, a LaTeX-ben a `--` jelsorozat kódolja. Nagykötőjelet a szavak közé a következő esetekben teszünk:

- Két vagy több nép kapcsolatának kifejezésekor (*angol–magyar* (szótár), *francia–spanyol* (határ), *olasz–német* (mérkőzés); *Osztrák–Magyar Monarchia*, *Cseh–Morva-dombság*, *Magyar–Finn Baráti Társaság*).
- Két vagy több tulajdonnév kapcsolatának kifejezésekor (*Marx–Engels–Lenin*. Tipográfiai előírás, hogy keresztnévvel is megadott szerzőpáros esetén a nagykötőjel két oldalán kis teret (*spácium*) hagyunk:

  Marx Károly – Engels Frigyes `Marx Károly\,--\,Engels Frigyes`
- A valamitől valameddig viszony érzékeltetésekor (*Budapest–Bécs*, *Duna–Majna–Rajna-csatorna*, *Duna–Tisza köze*, *1983–1984. évi*, *a 128–9. lapon*, *az I–IV. osztályban*, *Ménesi út 11–13.*; *az első–negyedik osztályban*, *kelet–nyugat irányban*, *tavasz–nyár folyamán*, *június–július hónapban*). Bonyolult esetben itt is spáciumot kell használni: *i. e. 753 – i. sz. 456.*
- Géptípusok betű- vagy szó- és számjelzése között (*Apollo–13*, *T–34*).
- Szakpárok jelölésére (*matematika–fizika tanár*).

5.3.5. Egyéb jelek

10%-os, 0 °C-on fagy, 100°-on forr, a ∼ szép, a 6. §-ban, $a \cdot b$

```
10\%-os, 0~$^\circ$C-on fagy
100$^\circ$-on forr, a $\sim$
szép, a 6.~\S-ban, $a\cdot b$
```

5.4. Betűtípusok

Nagyon lényeges, hogy elkerüljük a betűkeverés során gyakran előforduló hibát; a hasonló, de nem ugyanolyan típusú betűk keverését![4] A tervezett műhöz kiválasztanunk egy betűcsaládot, melyhez végig ragaszkodnunk kell. *A betűcsalád egy betűtípus változatait foglalja magába.* Például a Computer Modern betűcsalád tartalmazza a Computer Modern betűtípus különböző méretű álló, *döntött*, *kurzív*, **félkövér**, KISKAPITÁLIS, és ςπεϑιαλις (speciális) betűváltozatait.[5] A választott betűtípus legyen jól olvasható, lehetőleg klasszikus (barokk, reneszánsz, vagy klasszicista).

Az 5.1. táblázat rövid összefoglalása a Unix-szerű operációs rendszereken általánosan használt LaTeX keretrendszer, a teTeX programgyűjtemény

4. Hasonló betűtípusról beszélünk már akkor is, ha mindkét betűtípus talpas betű!
5. Fontos megjegyezni, hogy hiányos betűcsalád esetén a programok maguk generálnak le hamis betűváltozatokat, melyek meg (sem) közelítik az igaziakat! Tehát igyekezzünk teljes betűcsaládokat használni!

5.1. táblázat. A teTEX standard betűcsaládjai

név	kód	osztály	csomag	mat. csomag
Avantgarde	`pag`	konstruktivista	avantgar	
Bookman	`pbk`	eklektikus	bookman	
Charter	`bch`	újságbetű	charter	
Computer Modern	`cmr`	eklektikus		
Courier	`pcr`	írógépbetű	courier	
Helvetica	`phv`	groteszk	helvetic	
N. C. Schoolbook	`pnc`	újságbetű	newcent	
Palatino	`ppl`	reneszánsz	palatino	palatcm
Times	`ptm`	barokk	times	mathptm
Utopia	`put`	eklektikus	utopia	

standard betűcsaládjainak. A táblázatban szereplő csomagneveket a preambulumban kiadva elérhetővé válnak az adott betűcsalád betűi. (A Courier írógépbetű a `\texttt` paranccsal hívható elő, az Avantgarde és a Helvetica talpnélküli betűk pedig a `\textsf` paramétereként.[6])

A LaTeX betűcsaládja alapbeállításban a Computer Modern. A matematikai környezet különösen ragaszkodik ehhez a betűcsaládhoz. A palatcm csomag egy csaknem teljes megoldás a Palatino betűcsalád matematikai környezetben való használatára, de sajnos a görög betűk Palatino változatai nem állnak rendelkezésre. Jelenleg a mathptm csomag az egyetlen igazi alternatíva, mely a Times betűcsalád teljes körű használatát biztosítja.

Ha kellő tudással vagyunk felvértezve, a

`\fontfamily{`*betűcsalád-kód*`} \selectfont`

parancspár segítségével több betűcsaládot is használhatunk egy kiadványon belül.

5.5. Hasábok

Ha apró betűvel sűrű szöveget szeretnénk szedni, ajánlatos több hasábos szedést alkalmazni. Egyszerűen megtehetjük ezt a multicol csomagban leírt

6. A táblázatban nem szerepel a chancery csomag, mellyel a Zapf Chancery betűtípus médium kurzív betűváltozatát érhetjük el. A `\textmd` és a `\textit` parancsok egymásba ágyazásával hívhatjuk elő a kalligrafikus díszbetűket. Természetesen csak címként használandó!

`multicols` környezet segítségével. Használata:

\begin{multicols}{*hasábszám*}

Példával:

A környezet maga választja meg a hasábok magasságát az optimális térkitöltés érdekében. Hosszabb szöveg esetén legalább negyven betű legyen egy sorban!

```
\begin{multicols}{2}
A környezet maga választja meg
a hasábok magasságát az optimális
térkitöltés érdekében. Hosszabb
szöveg esetén legalább negyven
betű legyen egy sorban!
\end{multicols}
```

A `multicols` környezet további lehetőségeit a csomag dokumentációja, illetve az ajánlott irodalmak taglalják.

5.6. Sorritkítás

A sorközöket többnyire a betűméret 10-20%-ára vesszük. A LaTeX esetében 20% körüli az alapbeállítás: A 10, 11, 12 pontos betűméretek esetén 12, 13,6, 14,5 pontos a sorritkítás.

Előfordulhat, hogy ennél sűrűbbre szeretnénk venni a sorokat. Ezt a `\baselinestretch` változó értékének módosításával tehetjük meg. A változtatás életbeléptetéséhez még egy méretmódosító parancs kiadására is szükség van:

Ha a faktort 0-ra állítjuk, tömör, ún. kompressz szedést kapunk.

Ha a faktort 0-ra állítjuk, tömör, ún. kompressz szedést kapunk.

```
\newcommand{\szoveg}{Ha a faktort
0-ra állítjuk, tömör, ún.
kompressz szedést kapunk. \par}
\renewcommand{\baselinestretch}{0}
\normalfont \szoveg
\renewcommand{\baselinestretch}{2}
\large \szoveg
```

Alapesetben a faktor értéke `1`, ez az érték tartozik a 20%-os sorritkításhoz. A `0.96` értékkel 15%-os, a `0.92` értékkel 10%-os sorritkítást kapunk. Így 5-10%-os papírmegtakarítást érhetünk el az íráskép kismértékű változása mellett. (A dolgozatok „hígításához” használt értékek: `1.24` – másfeles sortáv, `1.66` – dupla sortáv.)

6. fejezet

A magyar teTEX környezet

Bár a fejezet elsősorban a UNIX operációs rendszereken elterjedt teTEX kibocsátás magyarítását tárgyalja, nem kell kétségbe esnünk más platformok alatt sem, mivel lényegi különbségek nincsenek. (Erről bővebben az adott TEX megvalósítás dokumentációja tájékoztat.)

6.1. Magyarítás

A LaTEX rendszer teljes magyarítása megoldható, de kiegészítésekre van szükségünk. A fejezetben ismertetett anyagok hozzáférhetők az Interneten.[1]

6.1.1. A magyar elválasztás beállítása

Az említett címről letöltött telepítőprogram automatikusan elvégzi a szükséges módosításokat. Röviden a telepítőprogram működéséről: A magyar elválasztási szabályokat tartalmazó `huhyph.tex` állományt a teTEX rendszer `hyphen` alkönyvtárába másolja, kibővíti a `language.dat` állományt a `magyar huhyph.tex` sorral, meghívja a teTEX fájladatbázisát frissítő texhash programot, és lefordítja az `initex 'lplain \dump'` hívással a LaTEX forrását a gyorsabban betöltődő `lplain.fmt` állományra, melyet `latex.fmt` néven a `web2c` alkönyvtárban helyez el. Ezek után már csak nem szabad elfeledkeznünk a LaTEX forrásállományunkban a `\usepackage[magyar]{babel}` parancs kiadásáról.

6.1.2. A „magyarabb” babel

Feltétlenül érdemes használni a babel csomag magyar vonatkozású részének, a `magyar.ldf` állomány – Bérces Józsefnek köszönhető – javítását és bővítését, mely a fejezet és a szakaszcímek, valamint a tételek, ábrafeliratok

1. `http://sol.cc.u-szeged.hu/~nemethl`

számozását teszi helyre. Továbbá lehetővé teszi a oldal- és egyéb hivatkozások névelőjének automatikus helyes kiválasztását („a” vagy „az” eldöntése). Fordított aposztrófot közvetlenül a kettős többjegyű mássalhangzók elé téve ezek helyes elválasztását állíthatjuk be. Az előző pontban ismertetett telepítőprogram ezt is telepíti automatikusan.

A következő parancsokkal rakhatunk helyes névelőt a szavak és számok elé, illetve a hivatkozásokhoz:

`\az{`*szó*`}`, `\azr{`*címke*`}`, `\azp{`*címke*`}`, `\azc{`*címke*`}`

A parancsoknak létezik nagybetűt kirakó változata is, itt a parancsnevekben a kis „a” betű helyett „A” betű szerepel. A parancsok csillagos változata nem teszi ki a paramétert, csak a névelőt. A 2.6. szakaszban szereplő példa javított változata mutatja a leírtakat:

Egy névelős hivatkozás arra a szakaszra így néz ki : „A 2.6. szakaszban a 20. oldalon.”

```
Egy névelős hivatkozás arra a
szakaszra így néz ki :
,,\Azr{oldal:ez}.~szakaszban
\azp{oldal:ez}.~oldalon.''
```

6.1.3. A magyar stílusfájl

A magyar stílusfájl használata egyszerűvé teszi a LaTeX forrásállományok szerkesztését. Az `\usepackage{magyarul}` paranccsal a már ismertetett beállítások (`fontenc`, `inputenc`, `babel`, `indentfirst`, `frenchspacing`) automatikusan végrehajtódnak. A szakaszcímek és -számok közti teret szóköznyi méretűre csökkenti.[2]

A másik módosítás a lábjegyzetek számozása. Az előző lábjegyzet rossz, mivel a magyar tipográfiai szabályok szerint csak a lábjegyzet-hivatkozások számai kerülnek kitevőbe. A magyar stílusfájl használatával helyes lábjegyzetszámozásokat kapunk.[3]

Ha az eredeti lábjegyzet-számozásokat és címeket is használni szeretnénk (pl. egy többnyelvű kiadványban), akkor a magyar stílusfájlt a `gyenge` opcióval töltsük be (`\usepackage[gyenge]{magyarul}`). Ekkor a magyar szakaszcímek és lábjegyzetek a `\masection`, `\masubsection`, `\mafootnote` parancsokkal érhetők el.

6.1.4. Magyar szószedetek készítése

A szószedetek készítésénél felmerülő probléma, hogy a *makeindex* program alapesetben csak az angol, illetve a német helyesírási szabályok alapján képes csak sorbarendezni a tárgyszavakat. Egy lehetséges megoldást jelent a *mami* program,[4] mellyel e könyv tárgymutatója is készült. Használata a

[2]Az eredeti hibás változat ebben a szakaszban látható.

3. Lásd például ezt.

4. A LaTeX kezdőknek és haladóknak egy másik megoldást ismertet.

makeindex programéval megegyező, igényli azonban a makeindex programot is, és az inputenc csomag használatát. A *mami* program opcionálisan megcímezi az azonos betűvel kezdődő szócsoportokat is. Bővebb leírása a program dokumentációjában.

6.2. Tippek és trükkök

6.2.1. LaTeX forrás MS-DOS és UNIX/Linux alatt

Fordítási hibát okozhat az MS-DOS környezetből származó forrásállomány a teTeX környezetben, mivel a sorok vége másképp van kódolva.[5] A *tr* szűrő használatával kitörölhetjük a hibáért felelős vezérlőjelet:

```
tr -d "\r" < dos.tex > unix.tex
```

Ha az MS-DOS szövegek 852-es kódtáblával leírt magyar ékezeteit nem az inputenc csomag használatával szeretnénk értelmezni, végleges konverziót a következő egyszerű módon hajthatunk végre:

```
tr `cat 852` `cat iso2` < dos.tex > unix.tex
```
[6]

Ahol a 852 állomány tartalma:

```
\240\265\202\220\241\326\242\340\224\231\213\212\243\351\201\232\373\353
```

Az iso2-é pedig:

```
\341\301\351\311\355\315\363\323\366\326\365\325\372\332\374\334\373\333
```

A módszer előnye, hogy a visszakonvertálás magától értetődő:

```
tr `cat iso2` `cat 852` < unix.tex > dos.tex
```

Bonyolultabb esetekhez használjunk komolyabb konvertáló programokat, melyek szabadon hozzáférhetők az Interneten.

6.2.2. A LyX program magyar beállításai

A LaTeX néha nem is csekély kényelmetlensége, hogy kis dolgokat lassabban lehet benne megcsinálni, mint a WYSIWYG rendszerekkel. A megoldás a LyX beszerzése.[7] A LyX gazdag dokumentációval és példagyűjteménnyel rendelkezik. A WYSIWYG rendszerekből megszokott kényelmi funkciók mellett

5. DOS alatt a soremelés vezérlőjel mellett még egy „kocsi vissza” vezérlőjelet is találunk minden sor végén.
6. A burkot nem ismerők számára: az aposztrófok fordítottak (előhívásuk AltGr-7), szerepük pedig a burokutasítások egymásba ágyazott kiértékelésénél van.
7. Linux operációs rendszer alá pl.:
ftp://jate.u-szeged.hu/pub/linux/RedHat/contrib/i386/
A lyx mellett az xforms csomagra is szükség van a telepítéshez!

lehetőség van LATEX parancsok kiadására, új parancsok írására is szerkesztés közben.

A magyarítás a következő módon történik: Írjuk be a „Layout/LaTeX Preamble. . . ” menüpont kiválasztása után megjelenő ablakba:

```
\usepackage[LyX]{magyarul}
```

(A magyar stílusfájl `LyX` opciója azt ellensúlyozza, hogy a LYX felesleges szóközt illeszt be a lábjegyzetek elé.)

Végül mentsük el alapbeállításnak a beírtakat a „Layout/Save layout as default” menüponttal.

6.2.3. LATEX kiadványok nyomtatása

Több lehetőség közül is választhatunk. A `dvilj`, `dvilj2p`, stb. konverterek előnye, hogy (a HP lézernyomtatóig) hordozható formátumot állítunk elő, melyet a célgépen csak a nyomtatóportra kell másolnunk egy cp, vagy copy paranccsal.[8] Figyelmeztetés! A kellemetlen tapasztalatok elkerülése végett egy üres oldalt hagyjunk a kiadványunk végén, ha ezt a módszert választanánk.

A másik módszer a *dvips* program használata. A kapott PostScript állomány a *GhostView* (*gv*) program segítségével nem csak PostScript, és nem is csak lézernyomtatókon nyomtatható kényelmesen. Használatára példa:

```
dvips -Z -f -tA4 file.dvi > file.ps
```

A „Z” opcióval kisebb PostScript állományokat kapunk a bittérképes betűkészletek tömörítése miatt. A „t” opcióval a *letter* alapbeállítás helyett A4-es papírméretet állítunk be. (Az „f” opció ahhoz szükséges, hogy az outputot fájlba irányíthassuk.) Ha PostScript formátumú betűkészleteket (fontokat) használnánk (a Computer Modern kivételével az 5.1. táblázatban szereplő összes betűcsalád az), akkor hasznos lehet a „V” opció kiadása. Ekkor a PostScript fontok bittérképes változatai a PostScript állományhoz lesznek csatolva, így a fájl olyan helyen is kinyomtatható lesz, ahol nincsenek meg a PostScript fontok. Hátránya nyilvánvalóan az, hogy csak olyan felbontásban nyomtatható ki a kiadvány, mellyel a generált bittérképes font rendelkezik (teTEX környezetben ez 600 dpi alapesetben).

6.2.4. Hogy készült a fordítás?

Röviden: Linux operációs rendszer alatt, magyar teTEX környezetben, a *Midnight Commander mcedit*[9] nevű szövegszerkesztőjével. A forrásállomá-

8. Linux: `cp x.lj /dev/lp1`, MS-DOS: `copy x.lj lpt0:`. Az egyszerűség már nem igaz a Windows 95-re, ahol a nyomtatókezelésen belül pár dolgot módosítanunk kell az alapbeállításon.

9. Beállítás az *mc*-ben: „Options/Display bits. . . ”: ISO 8859-1, és „Full 8 bits input”. Valamint „Options/Configuration”: „use internal edit”.

nyok X11 alatt kerültek lefordításra és megtekintésre, két gomb lenyomásával (a *Midnight Commander*-ben „F2” és „t”). Mivel a szerkesztés sima virtuális terminálon történt, a fordítás és nézegetés közben észlelt hibák párhuzamosan javíthatók voltak, csak a Ctrl-Alt-Fn és a Ctrl-Alt-F7 gombokkal kellett váltani. Ez meglehetősen nagy könnyítés volt a MS-DOS után, ahol nem győzte az ember papírra jegyezni a javításra váró hibákat.

Irodalomjegyzék

[1] Leslie Lamport. *LaTeX: A Document Preparation System.* Addison-Wesley, Reading, Massachusetts, second edition, 1994, ISBN 0-201-52983-1.

[2] Donald E. Knuth. *The TeXbook,* Volume A of *Computers and Typesetting*, Addison-Wesley Publishing Company (1984), ISBN 0-201-13448-9.

[3] Michel Goossens, Frank Mittelbach and Alexander Samarin. *The LaTeX Companion.* Addison-Wesley, Reading, Massachusetts, 1994, ISBN 0-201-54199-8.

[4] Minden LaTeX telepítésnek tartalmaznia kell egy úgy nevezett *LaTeX helyi útmutatót*, amely ismerteti a helyi rendszert. Az útmutatót a `local.tex` állomány tartalmazza. Sajnos ezt néhány lusta rendszeroperátor nem teszi elérhetővé. Ebben az esetben kérjük egy helyi LaTeX szakértő segítségét.

[5] LaTeX3 Project Team. *LaTeX 2ε for authors.* A LaTeX 2ε disztribúció része `usrguide.tex` néven.

[6] LaTeX3 Project Team. *LaTeX 2ε for Class and Package writers.* A LaTeX 2ε disztribúció része `clsguide.tex` néven.

[7] LaTeX3 Project Team. *LaTeX 2ε Font selection.* A LaTeX 2ε disztribúció része `fntguide.tex` néven.

[8] D. P. Carlisle. *A „graphics" gyűjtemény csomagja.* A „graphics" gyűjtemény része a `grfguide.tex` állomány a LaTeX disztribúcióban.

[9] *A magyar helyesírás szabályai. Tizenegyedik kiadás*, Akadémiai Kiadó, Budapest, 1986.

[10] Virágvölgyi Péter: *A tipográfia mestersége – számítógéppel.* Tölgyfa Kiadó, Budapest, 1996.

[11] Wettl Ferenc et al.: *LaTeX kezdőknek és haladóknak.* Panem Kiadó, Budapest, 1998.

Tárgymutató

Q

R

S