

L^AT_EX nemcsak matematika szakosoknak

Kovács Zoltán

Blahota István, Nagy Károly és Toledo Rodolfo közreműködésével

ver.: 2005. szeptember 14.

Tartalomjegyzék

Bevezetés	1
1. Minden kezdet...	4
2. Egy dokumentum részei	6
3. Betűk, szavak, mondatok, bekezdések	9
4. I ♥ math	14
5. Táblázatok	20
6. Bibliográfia készítése	23
7. Ami a kiadványszerkesztéshez még szükséges lehet	26

Bevezetés

Jelen segédanyag célja, hogy rövid és mértéktartó bevezetést nyújtson a L^AT_EX-be: az érdeklődő kapjon némi képet a L^AT_EX legfontosabb lehetőségeiről s remélhetőleg kapjon kedvet a L^AT_EX használatához, a matematika szakos hallgató kapjon segítséget a (matematika) szakdolgozata gépeléséhez.

Magyar nyelven kiváló referencia a következő könyv: *Wettl – Mayer – Sudár: L^AT_EX kezdőknek és haladóknak* (Panem).

Szövegszerkesztés, DTP

A *szövegszerkesztő* kifejezés tulajdonképpen legalább két fontos fogalmat takar, amelyek azonban a gyakorlatban nem vegytisztán jelentkeznek: text editor, word processzor. A text editorokkal csak karakterekből álló álló (ASCII) szöveges fájlokat hozunk létre. A word processzossal egy lépcsővel fentebb léphetünk, s a szövegfájlt formázhatjuk.

A nagyobb teljesítményű személyi számítógépek azt is lehetővé teszik, hogy a nyomdai előkészítés művelete is (pl. szakszerűen tipografált oldalak tördelése, grafikai elemek elhelyezése, szöveg és kép együttes kezelése, színbontás) a nyomdákba kerüljön. Nyomdai előkészítés személyi számítógépen – lényegében erre használják a *desktop publishing* (DTP) kifejezést.

WYSIWYG

WYSIWYG – what You See Is What You Get, azt kapod amit látsz. Az elnevezés találó, s az irodai alkalmazásokat nagymértékben felhasználóbarátta tette: a gépíró a számítógépe képernyőjén pontosan (?) azt látja, amit a printere fog nyomtatni. Fontos következmény, hogy a szövegírás és a formázás így összemosódik.

Mi a L^AT_EX?

A L^AT_EX¹ alapja a T_EX, a L^AT_EX egy ehhez készített makró csomag. A T_EX egy szöveg- és kiadványszerkesztésre, nyomdai előkészítésre alkalmas programrendszer, mely megalkotója Donald E. Knuth, *A számítógép-programozás művészete* c. világszerte kiadott mű szerzője. A program a tipográfia sok évszázados mesterségét viszi számítógépre, természetesen kibővítve a számítógép adta új lehetőségekkel. A T_EX-re sokan úgy tekintenek, mint a tipográfia assemblerére, melyben minden tipográfiai feladat megoldható, de gyakran csak nagyon fáradságos programozói munkával. A T_EX-ben makrók is írhatók, ami lehetővé teszi, hogy a T_EX-re egy magasabb szintű, barátságos programnyelv épüljön, mint például a L^AT_EX.

A L^AT_EX nem texteditor, erre egy tőle független texteditort kell alkalmazni. Jelen sorokat Linux operációs rendszer alatt, Emacs szerkesztővel írom, de kezdő L^AT_EX-esként egyszerűen Norton Editort használtam. A gépelésnél írom a szövegemet, s közben utasításokat helyezek el a szövegben a L^AT_EX számára. A gépelés befejezésekor ugyanis az egész szöveget át fogom adni a L^AT_EX-nek, aki lényegét tekintve a tipográfusom. Bizonyos szövegszerkesztők (így az Emacs is) speciálisan fel vannak készítve a T_EX, L^AT_EX használatára.

A személyi számítógépek és (irodai) szövegszerkesztők elterjedésével teljesen általánossá vált, hogy a szerzők a műveiket a megírástól a nyomdai előkészítésig maguk készítik. Csakhogy az irodai szövegszerkesztők (beleértve a piacvezető szoftvert is) a tipográfiát nagymértékben a szerzőre bízzák (vagy csodás felületükkel csábítják a szerzőt arra, hogy a beépített tipográfiai tudást mellőzze). A szerző az esetek többségében a tipográfiahoz nem ért, legfeljebb a szépérzékére támaszkodik. (Egy kis döntés itt, aláhúzás amott, betűméret váltás, margóállítás, kövérítés, árnyékolás, használjunk minél több betűcsaládot, s legtöbbször valami tipográfiai főrmérvény az eredmény.)

A L^AT_EX használatával a tipográfiai és a szerzői munka kettéválik, a dokumentum (grafikai) tervezését, megjelenítését, tördelését a szerző átadja a L^AT_EX-nek. A szerző írja a művét, közben üzeneteket hagy a L^AT_EX-nek, hogy éppen milyen tipográfiai problémát kell megoldani (itt egy cím van, itt egy kiemelés, itt új bekezdés, stb.) Ha a tipográfus (L^AT_EX) a feladatot rosszul oldotta meg, akkor nyitva áll a lehetőség, hogy a szerző legapróbb részletekbe is beleszóljon. (Ezt azonban sokszor nem olyan könnyű megtenni, s csak a gyakorlottaknak, tipográfiailag képzeteknek ajánlható.)

Integrált környezetek

L^AT_EX dokumentumok kezelésére több (ingyenesen használható) integrált környezetet fejlesztettek ki, messzemenőig figyelembe véve a L^AT_EX egyéni sajátosságait. Ezek nem csupán kényelmes szövegszerkesztőt biztosítanak (például a T_EX-parancsokat gyakran ikonok segítségével érhetjük el) hanem a L^AT_EX forrást futtathatjuk, megnézhetjük, konvertálhatjuk a szerkesztőből.

A WinShell egy Windows (3.x, 95, 98, NT, ...) alatt futó integrált környezet Letöltés, további információk: <http://www.winshell.de>.

A Kile szintén egy integrált L^AT_EX környezetet biztosít, Linux felhasználók számára. Letöltés, további információk: <http://kile.sourceforge.net>.

Használjam-e a L^AT_EX-et?

A L^AT_EX-nek vannak előnyei és hátrányai. Először is világos, hogy az irodai szövegszerkesztésben a piacvezető szoftvernek nem versenytársa, az előzőekből kiderül, hogy nem is erre fejlesztették (fejlesztik), bár erre is alkalmas. Ha olyan szoftvert keresünk, mellyel elsősorban szöveges természetű és inkább hosszabb, mint rövidebb munkánkat szépen akarjuk kinyomtatni, esetleg profi nyomdai előkészítését

¹ Amit ma L^AT_EX-nek mondunk, az tulajdonképpen az ún. L^AT_EX_{2_ε} verzió. A régebbi, ma már nem támogatott verzió a L^AT_EX 2.09.

elvégezni, akkor már szóba jön a \LaTeX . Ha ráadásul a munka sok képletet is tartalmaz, netán matematikai természetű, akkor szinte szóba sem jön más. (Úgy tűnik, hogy az utóbbi állítás a színvonalas matematikai könyvek, folyóiratok kiadóinak is véleménye világszerte.) Ha a művünkben a grafikus elemek vannak túlsúlyban, s speciális grafikus effektusokat akarunk alkalmazni (pl. képeskönyv, reklámanyag, speciális tördelési igények), akkor a \LaTeX már kevésbé javasolható.

Néhány lehetőséget villantok fel, egyikéről-másikáról lesz szó később. Alkalmas ábrák kezelésére, vonalas jellegű ábrát, pl. műszaki rajzot lehet vele (ill. kiegészítéseivel) készíteni, be lehet illeszteni más programmal készített ábrát, fényképet. Lehet táblázatokat szerkeszteni. Automatikusan tudunk bibliográfiát, tartalomjegyzéket indexet készíteni. Több tucat nyelv elválasztási szabályát képes elég jó hatékonysággal alkalmazni (a magyart is). Használhatjuk a legáltalánosabban elterjedt fontcsaládokat, ha akarunk, saját fontcsaládot tervezhetünk. A sok egyéb mellett néhány extra: kottaszerkesztésre, sakkállások szedésére, hieroglif írásra alkalmas kiegészítése is van.

A \TeX és a \LaTeX legfontosabb előnye talán az, hogy szabad szoftver, forráskódja szabadon elérhető, ennek minden előnyével együtt. (Pl. sokan fejlesztik.)

A \LaTeX hátránya (mások szerint előnye), hogy nem WYSIWYG rendszer, ami bizonyos esetekben a forrás áttekinthetőségét nehezíti (pl. táblázatoknál). A végeredményt csak akkor látjuk, miután a tipográfus (\LaTeX) visszaadja a munkát egy ún. *device independent* (dvi) fájl formájában, s ezt egy újabb szoftver segítségével a képernyőn megnézzük, vagy nyomtathatóra konvertáljuk s kinyomtatjuk. Nagy előny, hogy a DTP-ben szabványnak tekinthető PostScript állománnyá történő konverzió is lehetséges, illetve a grafikus elemeket illetően a \LaTeX „PostScript barát”.

A bűvös hármas

Az előző bekezdést egy kicsit részletesebben magyarázva: minden \TeX -el készült munka (legalább) három lépésben készül el:

1. Tetszőleges texteditorral begépeljük a forrásfájlt. Ennek kiterjesztése `tex`, pl.: `file.tex`.
2. A \LaTeX fordítóprogramjával lefordítjuk a fájlt. A fordítóprogram több fájlt generál, pl.: `file.log`, `file.dvi`.
3. A `file.dvi` fájlt megnézhetjük a képernyőn (egy `dvi` néző programmal), nyomtathatjuk (egy `dvi` nyomtató programmal), konvertálhatjuk (egy `dvi` konvertáló programmal, pl. PostScript állománnyá.)

\TeX a neten

A \TeX -hez kapcsolódó anyagok gyűjtőhelyét CTAN-nak (Comprehensive \TeX Archive Network)-nek hívják. CTAN ftp szerverek világszerte működnek, én az `ftp.dante.de` szerveret szoktam használni. Ezek a szerverek a forrásfájlok mellett dokumentációkat is találunk. Több GB anyagról van szó. A \LaTeX legfrissebb verzióját a `macros/latex/base/` könyvtárban találjuk.

A magyar \LaTeX honlapon (www.math.bme.hu/latex/) hasznos információkat kap az ember, többek között a telepítéshez.

1. Minden kezdet...

1.1. Hello world!

Már a bevezetőben is említettem, hogy egy \LaTeX forrás a művünkből és a \LaTeX -nek adott instrukciókból áll. A következő minta tartalmazza azt a kötelező minimumot, mellyel a szokásos *Hello world!* szöveg létrehozható. Tehát a szövegszerkesztőnkkel (text editor) létrehozunk egy *tex* kiterjesztésű fájlt (legyen mondjuk *hello.tex*) az alábbi szöveggel:

```
\documentclass{article}
\begin{document}
Hello world!
\end{document}
```

A fájl első, második, és negyedik sora a \LaTeX -nek adott instrukció, a harmadik sor a művünk. Megfigyelhetjük, hogy a \LaTeX -nek szánt instrukciók a „backslash” karakterrel kezdődnek.

Minden \LaTeX fájl a `\documentclass{a dokumentumosztály neve}` sorral kezdődik.² A dokumentumosztály megadásával rögzítjük a művünkre vonatkozó tipográfiai elveket, paramétereket.

A `\documentclass{article}` és a `\begin{document}` közötti rész az ún. *preambulum*, jelenleg ide nem írtam semmit. A művünk mindig a `\begin{document}` és az `\end{document}` közé kerül.

1.2. Magyarul

Most az előbbi üzenetet írjuk le magyarul is. Lesz némi változás:

```
\documentclass{article}
\usepackage{t1enc}
\usepackage[latin2]{inputenc}
\usepackage[magyar]{babel}
\begin{document}
\frenchspacing
Szervusz világ.
\end{document}
```

A *preambulum* bővült, a \LaTeX képességeit *programcsomagokkal* bővítettük. A `\usepackage{t1enc}` sorral a \LaTeX belső kódkészletét bővítjük egy 256 betűs ábécével, amely majdnem minden európai nyelv ábécéjét tartalmazza. A `\usepackage[latin2]{inputenc}` sor hatására a forrásfájlba a magyar billentyűzetről bevitt ékezetes betűket értetjük meg a \LaTeX -el. Végezetül a `\usepackage[magyar]{babel}` sor sok magyar sajtóságot rögzít, például az elválasztást.

Egy további sor is ajánlott a magyar nyelvű dokumentumok *preambulumában*:

```
\usepackage{indentfirst}
```

Ezzel a sorral a fejezetek első bekezdését is behúzza a \LaTeX , míg alapértelmezésben a fejezet első bekezdése nem behúzással kezdődik. Jelen dokumentum szerkesztésekor is ezt a csomagot használtam.

A `\frenchspacing` parancs azt jelzi a \LaTeX -nek, hogy a szóköz és a mondatköz ugyanolyan távolság legyen. Alapértelmezésben ez utóbbi nagyobb.

1.3. A dokumentumkezdő parancsok

```
\documentumclass[opciók]{osztály}
\usepackage[opciók]{csomag}
```

A legfontosabb dokumentumosztályok, s javasolt hatáskörük:

article Kisebb dokumentumok, cikkek, rövidebb tanulmányok.

book Könyvek, nagyobb munkák.

²Ez a mondat többszörösen sem igaz, egyrészt csak bizonyos, de ma már általános verziótól kezdve létezik ez a parancs, másrészt itt nem részletezett dolgok állhatnak a forrás elején.

report Beszámolók, hosszabb tanulmányok.

slides Írásvetítő fóliára íráshoz.

letter Levélíráshoz.

A különböző dokumentumosztályok használatában sok különbség van, pl. mások lehetnek a L^AT_EX-nek számtal instrukciók.

Néhány opció a dokumentumosztályokhoz:

10pt, 11pt, 12pt A normál szöveg méretét határozza meg, közülük persze csak egy adható meg. Alapértelmezés a 10pt. (Később majd látjuk, hogy nem csak ez a háromféle betűméret lehetséges.)

letterpaper, a4paper, a5paper, b5paper A papírméretet adja meg. Alapértelmezés a letterpaper, ami Magyarországon nem használatos, tehát ezt feltétlenül be kell állítani pl. a4-re.

oneside, twoside Beállíthatjuk, hogy a dokumentumot egyoldalas vagy kétoldalas nyomtatásra szánjuk. **Article** és **report** osztályban az alapértelmezés oneside, **book** osztályban twoside.

landscape A lapokat landscape (fekvő) helyzetbe hozza.

Ha több opciót adunk meg, akkor ezeket vesszővel elválasztva kell megadni.

1.4. Parancsok deklarációk, környezetek

Itt az ideje áttekinteni a L^AT_EX-nek adott instrukciók szerkezetét. Ezek lehetnek parancsok, deklarációk vagy környezetek. A jegyzet további részében az olvasó több konkrét példát talál mindegyikre.

A parancs általános alakja a következő:

```
\parancs[opcionális argumentum]{kötelező argumentum}
```

vagy argumentum nélkül:

```
\parancs
```

A környezetek szerkezete:

```
\begin{környezetnév}[opc]{köt}
```

```
...
```

```
\end{környezetnév}
```

Gyakori az argumentumok nélküli környezet is.

A deklarációk olyan argumentum nélküli parancsok, melyek bizonyos paraméterek értékét átállítják, de nem nyomtatnak ki semmit. Hatásuk mindig egy *blokkra* korlátozódik. Blokkot kapcsos zárójelpárral (`{}`) vagy `\begin{név}... \end{név}` környezettel hozhatunk létre. Pl.:

```
{\deklaráció }
```

Lehetőség van új parancsok, környezetek létrehozására (`\newcommand`, `\newenvironment`), sőt meglévőek átdefiniálására (`\renewcommand`). A jegyzet olvasása közben erre talál az olvasó néhány konkrét példát, használatuk szabályaira ezen rövid munkában nem térek ki, csakúgy mint sok más fontos T_EX-objektum, pl. a *számlálók* használatára.

1.5. Rejtett szöveg

Ha `%` karaktert gépelünk, a forrásszöveg sora további részét a L^AT_EX nem fordítja, így lehetőségünk van olyan megjegyzések elhelyezésére, amelyek csak a forrásban látszanak.

2. Egy dokumentum részei

2.1. Cím, szerző, dátum

A preambulumban adjuk meg a megfelelő információkat:

```
\author{Safarevics}
\title{Algebra}
\date{2000}
```

Majd a `\begin{document}` után hívjuk ezt az információt elő: `\maketitle`. A dátum megadása opcionális.

A címlap megjelenése dokumentumosztályonként változhat. Az `article` osztálynál nem kerül külön lapra, ezen a `titlepage` opció változtat. Saját címlapot tudunk szerkeszteni a `titlepage` környezettel.

2.2. Fejezetek, kisebb szakaszok

A \LaTeX az `article` dokumentumosztályban az alábbi parancsokkal támogatja a szakaszok kialakítását:

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

Ehhez a `book` és `report` dokumentumosztályban még két további parancs társul:

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

A továbbiakban az előzőek mindegyikét szakasznak fogom hívni. A szakaszok közti helykihagyások, a szakaszcímek betűváltoztatának kialakítása, valamint a számozás automatikusan történik. Például a jelen szakaszt a következőképpen nyitottam meg a forrásfájlban:

```
\section{Egy dokumentum részei}
```

A számozást úgy lehet letiltani, hogy a parancs után `*`-ot gépelünk:

```
\section*{Egy dokumentum részei}
```

2.3. Tartalomjegyzék

Ennek készítését a \LaTeX automatikusan elvállalja, a megfelelő helyen ki kell adni a `\tableofcontents` parancsot. A `tocdepth` számláló beállításával lehet szabályozni, hogy a szakaszok milyen mélységig kerüljenek a tartalomjegyzékbe. Pl. `\setcounter{tocdepth}{1}` beállítás (`article` dokumentumosztályban) csak a számozott `section`-öket veszi fel.

Ha a nem sorszámozott szakaszokat is fel akarunk venni, akkor ezt megtehetjük a következő módon:

```
\section*{Egy dokumentum részei}
\addcontentsline{toc}{section}{Egy dokumentum részei}
```

A második argumentumban határoztuk meg, hogy milyen szinten vegye fel a sort a tartalomjegyzékbe.

Korrekt tartalomjegyzéket a kész mű után háromszori futtatással fogunk kapni. (Gondolkozzunk rajta, miért.)

2.4. Hosszú művek

Hosszú művek írása nehézkessé válik, ha az egészet egy fájlba írjuk. A fő fájlban lehetőség van további fájlok behívására. Például a jelen szakaszom a `tagolas.tex` fájlban van, melyet a fő fájlomban úgy hívtam be, hogy `\include{tagolas}`. Tehettem volna úgy is, hogy `\input{tagolas.tex}`. A két megoldás között a(z egyik) különbség, hogy az `include` parancsral behozott fájl új oldalon fog kezdődni.

2.5. Matematikai szöveg tagolása: newtheorem

A matematikai szövegeket egy fejezeten belül is kisebb részekre, tételekre, definíciókra, bizonyításokra stb. szokták osztani. Ezek egységes tipográfiai kezeléséhez, számozásához nyújt segítséget a `\newtheorem` parancsa.

Tekintsük az alábbi példát. Tegyük fel, hogy a preambulumban begépetük, hogy `\newtheorem{sttl}{Lemma}`. Majd a szövegben valamikor:

```
1. Lemma. Minden négyzet téglalap.           \begin{sttl}
                                                    Minden négyzet téglalap.
                                                    \end{sttl}
```

A használat a következő volt: a `\newtheorem` parancs egy tételszerű környezetet definiál. Az első argumentumában definiáltam a tételszerű környezet nevét: `sttl` (ez egy saját magunk által meghatározott fantáziánév), a második argumentumban pedig, azt definiáltam, hogy mit írjon ki a szerkesztés során a környezet címének: **Lemma**. Amikor a szövegben behívtam ezt a környezetet, kiírta a címet, generált hozzá egy számot, s a tétel „testét” tipográfiailag is elkülönítette a folyó szövegtől.

A helyzet majdnem ilyen egyszerű, mint leírtam, a probléma csak az, hogy a tételszerű környezetek tipográfiai kezelése a különböző nyelvekben más és más. Például az előző tétel angol szövegben a következőképpen nézett volna ki.

Lemma 1 *Every square is a rectangle.*

Az adott nyelvek tipográfiai szabályait a `babel` csomag tudja (amit a magyar szövegekhez `magyar` opcióval be is hívunk mindig). Az angol verziót ezen csomag behívása nélkül kapjuk meg. (Mint ez a dokumentum mutatja, a nyelvek között menet közben lehet váltani. Ehhez annyit kell tenni, hogy a `babel-t` [`magyar`, `english`] opcióval kell behívni, s a szöveg közben a `\selectlanguage{english}`, illetve `\selectlanguage{magyar}` parancsokkal válthatunk a nyelvek között.)

A `\newtheorem` paranccsal bonyolultabb dolgokat is meg tudunk oldani. Használatára három lehetőség is van:

```
\newtheorem{név}{cím}
\newtheorem{név}{cím}[számláló]
\newtheorem{név}[együtnév]{cím}
```

Az első parancs használatát már láttuk. A `számláló` opcionális argumentumban előírhatjuk, hogy a számozásban egy más számláló aktuális értéke is megjelenjen. Például, ha az előző példában a `sttl` környezetet az alábbi módon hozzuk létre: `\newtheorem{sttl}{Lemma}[subsection]`, akkor eredménye:

2.5.1. Lemma. *Minden négyzet téglalap.*

Az `együtnév` opcionális argumentum egy másik tételszerű környezet neve. Ha ezt megadjuk, akkor ezeknek a környezeteknek azonos számlálója lesz. Ha pl. a következőt adjuk meg:

```
\newtheorem{thm}{Tétel}
\newtheorem{sttl}[thm]{Lemma}
```

akkor a sorszámozás a következőképpen alakul: 1. Tétel, 2. Lemma, 3. Lemma, 4. Tétel. Míg az opcionális argumentum nélkül: 1. Tétel, 1. Lemma, 2. Lemma, 2. Tétel.

Végezetül megemlítjük, hogy a tételszerű környezetnek lehet opcionális argumentuma is:

1. Tétel (Az algebra alaptétele). *Minden komplex együtthatós polinomnak van komplex gyöke.*

Mely tételt így gépeltem (miután a preambulumban `\newtheorem{thm}{Tétel}`-el létrehoztam a `thm` tételszerű környezetet.

```
\begin{thm}[Az algebra alaptétele]
  Minden komplex együtthatós polinomnak van komplex gyöke.
\end{thm}
```

Bizonyítások

A bizonyítás új bekezdésben, behúzás nélkül, a *Bizonyítás:* szóval szokott kezdődni, s valamilyen „bizonyítás vége” jellel zárul. Ehhez létre kell hozni egy új környezetet. (Vannak csomagok, melyekbe ez be van építve.) Ezt a preambulumban lehet elvégezni, pl. így:

```
\newenvironment{proof}{\par\noindent{\itshape Bizonyítás:}\ }{\rule{1ex}{1ex}}
```

A fenti sor a gyakorlatban a következő eredményt adja:

Bizonyítás: Ez azért van így, mert...■

Ezt így gépeltem:

```
\begin{proof}
  Ez azért van így, mert\ldots
\end{proof}
```

Elemezzük a parancsot! `proof` az új környezet neve, erre a névre kell hivatkozni majd a szövegben: `\begin{proof}`, `\end{proof}`. A `{\par\noindent{\itshape Bizonyítás:}\ }` rész modja meg, hogy a környezet meghívásakor, tehát a `\begin{proof}` leírásakor mi történjen: kezdődjön új bekezdés, behúzás nélkül, jelenjen meg „Bizonyítás:” szöveg (kurzív betűtípussal), majd egy szököz következzen. A

```
{\rule{1ex}{1ex}}
```

rész azt mondja meg, hogy mi történjen az `\end{proof}` kiadásakor: Rajzoljon a L^AT_EX egy 1ex szélességű és vastagságú vonalat, ahol 1ex az x betű magasságának mértéke.

3. Betűk, szavak, mondatok, bekezdések

3.1. Betűváltozatok

A \LaTeX -el számtalan ismert fontcsalád használható (pl. Times, Helvetica, Garamond, Bookman, Palatino). Az alapértelmezés egy saját fonttípus.³

Ha egy dokumentumban egy egységes fonttípust akarunk használni, akkor a legegyszerűbb a preambulumban erre áttérni, pl.: `\usepackage{times}`. (Ha többféle fonttípust akarunk használni, akkor ez finomabb megoldást igényel. Egyébként egy fonttípuson belül annyiféle betűváltozat van, hogy erre csak ritkán van szükség.)

Betűváltozat	Ezt kapom	Ezt gépelem	
		parancs	deklaráció
alak	álló <i>döntött</i> <i>kurzív</i> KISKAPITÁLIS	<code>\textup{álló}</code> <code>\textsl{döntött}</code> <code>\textit{kurzív}</code> <code>\textsc{kiskapitális}</code>	<code>{\upshape álló}</code> <code>{\slshape döntött}</code> <code>{\itshape kurzív}</code> <code>{\scshape kiskapitális}</code>
vastagság	normál félkövér	<code>\textmd{normál}</code> <code>\textbf{félkövér}</code>	<code>{\mdseries normál}</code> <code>{\bfseries félkövér}</code>
család	antikva groteszk írógép	<code>\textrm{antikva}</code> <code>\textsf{groteszk}</code> <code>\texttt{írógép}</code>	<code>{\rmfamily antikva}</code> <code>{\sfamily groteszk}</code> <code>{\ttfamily írógép}</code>
méret	legkisebb script lábjegyzet kicsi normál nagy nagyobb nagyon nagy óriás behemót		<code>{\tiny legkisebb}</code> <code>{\scriptsize script}</code> <code>{\footnotesize lábjegyzet}</code> <code>{\small kicsi}</code> <code>{\normalsize normál}</code> <code>{\large nagy}</code> <code>{\Large nagyobb}</code> <code>{\LARGE nagyon nagy}</code> <code>{\huge óriás}</code> <code>{\Huge behemót}</code>

1. táblázat. A betűk négy jellemzője

Az alapértelmezés a legtöbb dokumentumosztályban álló, normál vastagság, antikva. (Ez parancs ill. deklaráció nélkül is elérhető.) A méret relatív, abszolút értéke függ a dokumentumosztály opciójától. A fentieket egymásba is lehet ágyazni. Például, `\textbf{\textsf{\large félkövér groteszk}}` eredménye: **félkövér groteszk**. (Itt a `\textup` kiírása felesleges, mert ez most alapértelmezés.) Előfordulhat, hogy valamely négy jellemzővel megadott font a rendszerünkön nem létezik. Ekkor a \LaTeX ezt automatikusan helyettesíti egy más típussal.

3.2. Speciális karakterek

A klaviatúrán elérhető jelek általában úgy fognak megjelenni, ahogyan gépeltük, kivéve a következőket:
 \backslash , $\{$, $\}$, $\%$, \sim , $\$$, $_$, $\^$, $\&$, $\#$,
amelyeket a következőképpen gépeltem:

`\backslash$, \{, \}, \%, \~{ }, \$, _, \^{\}, \&, \#.`

Parancssal adhatók meg a klaviatúrán nem hozzáférhető egyéb jelek, pl:

`\dag` `\ddag` `\S` `\copyright` `\pounds` `\textvisiblespace`

³Amelyet pl. az \mathfrak{A} jellegzetes farkáról vagy éppen a \mathcal{Z} kunkoráról azonnal fel lehet ismerni.

3.3. Kiemelés

Álló szövegben kiemelésre *kurzív* betűket használunk, míg *kurzív környezetben* álló a szokásos. Ritkábban használjuk az aláhúzást, a CSUPA NAGYBETŰKKEL ÍRÁST, bekeretezést. Írógép környezetben nemigen szoktunk *kiemelni*, de lehet. Nemecek nevét a *Pál utcai fiúk*ban pedig így írták le (először): nemecek ernő.

Az előző bekezdést így gépeltem:

```
Álló szövegben kiemelésre \emph{kurzív}
betűket használunk, míg {\itshape kurzív környezetben \emph{álló} a
szokásos.} Ritkábban használjuk az \underline{aláhúzást}, a
\MakeUppercase{csupa nagybetűkkel írást}, \framebox{bekeretezést}. {\ttfamily
Írógép környezetben nemigen szoktunk \emph{kiemelni}, de lehet.}
Nemecek nevét a \emph{Pál utcai fiúk}ban pedig így írták le
(először): \MakeLowercase{Nemecek Ernő}.
```

A tanulság az, hogy az `\emph{}` parancs kiemel, nekünk a környezettel nem kell foglalkoznunk. Sőt a „*Pál utcai fiúk*ban” leírását tanulmányozva megfigyelhetjük a L^AT_EX ún. *kurzív kiegyenlítését*, a *k*-t nem dönti rá a *b*-re: *kb*, nem pedig *kb*, amit a `{\itshape Pál utcai fiúk}`ban adna. Különböző csomagokkal több más kiemelési lehetőség is elérhető, mint pl. a *ritkítás*, de ezek tárgyalásával nem foglalkozunk.

3.4. Bekezdések igazítása (kizárása)

A szokásos dokumentumosztályokban a sorok a bal margónál kezdődnek és a jobb margónál végződnek. (A bekezdés első és utolsó sora lehet kivétel.) Ha egy sor a jobb margón túllóg, akkor azt nevezzük *túlcsordulásnak* – erről később külön is lesz szó.

A bekezdések zárásának jellege környezettel vagy deklarációval megváltoztatható.

```
\begin{flushleft} \end{flushleft} \raggedright balra zár
\begin{flushright} \end{flushright} \raggedleft jobbra zár
\begin{center} \end{center} \centering középre zár
```

Ezt a szöveget balra zártuk. Ezt a szöveget balra zártuk, zárhattuk volna jobbra is.

```
\begin{flushleft}
Ezt a szöveget balra zártuk...
\end{flushleft}
```

Ezt a szöveget jobbra zártuk. Ezt a szöveget most éppen jobbra zárjuk.

```
\begin{flushright}
Ezt a szöveget jobbra zártuk...
\end{flushright}
```

Ez a szöveg középre van zárva. Se jobbra, se balra, éppen középre.

```
\begin{center}
Ez a szöveg középre van zárva...
\end{center}
```

3.5. Speciális formájú bekezdések

Minden indoklás nélkül három példa:

Arisztotelész a lehető leghívebben fordítva, a következőket mondja:

„A szubsztanciához hozzátartozik az is, hogy semmi velük ellentétes nincsen. Hiszen mi lenne egy elsődleges szubsztanciának az ellentéte?”

Arisztotelész ... a következőket mondja:

```
\begin{quotation}
  „A szubsztanciához ...”
\end{quotation}
```

Arisztotelész a lehető leghívebben fordítva, a következőket mondja:

„A szubsztanciához hozzátartozik az is, hogy semmi velük ellentétes nincsen. Hiszen mi lenne egy elsődleges szubsztanciának az ellentéte?”

Arisztotelész ... a következőket mondja:

```
\begin{quote}
  „A szubsztanciához ...”
\end{quote}
```

Felhőbe hanyatlott a drégeli rom,
Rá visszasüt a nap, ádáz tusa napja
Szemközt vele nyájas, szép zöld hegy-orom,
Tetején lobogós hadi kopja.

```
\begin{verse}
Felhőbe hanyatlott a drégeli rom,\
...
\end{verse}
```

3.6. Listák

A \LaTeX standard listáit az alábbi környezetekkel lehet elérni: `itemize`, `enumerate`, `description`.

- Első elem,
- Második elem.

```
\begin{itemize}
  \item Első elem,
  \item Második elem.
\end{itemize}
```

1. Első elem,
2. Második elem.

```
\begin{enumerate}
  \item Első elem,
  \item Második elem.
\end{enumerate}
```

infámia becstelenség, alávalóság, szégyen

infámis gyalázatos, aljas, becstelen, elvetemült

```
\begin{description}
  \item[infámia] becstelenség, ...
  \item[infámis] gyalázatos, ...
\end{description}
```

A listák 4 mélységig egymásba ágyazhatók.

3.7. A gépelés néhány szabálya

- ◇ A forrásban mindegy, hogy hány szóközt gépelünk, az eredmény egyetlen szóköz. Hasonló vonatkozik az új bekezdésre: mindegy, hogy hányszor nyomjuk meg az `Enter` gombot, az eredmény annyit tesz, mint egyszer.

Látsz-e előttem és utánam különbséget? Látsz-eelőttemésutánamkülönbséget?

◇ Az egymás mellé írt három pont szedésére az `\dots` való:

Egy gondolat bánt engemet... Egy gondolat bánt engemet\dots

◇ Ha a pont nem a mondat végét jelzi, akkor ezt jelezni kell, mert egyébként utána túl nagy lesz a szóköz (ha nem használjuk a `\franchspacing` deklarációt):

Pl. elmehetnénk a Tátrába! Pl.\ elmehetnénk a Tátrába!

Egyébként a magyar nyelvű szövegek gépelésére ajánlott a `\franchspacing` deklaráció, ekkor a szóköz és a mondatköz nem különbözik.

◇ A mondat végét akkor is jelezni kell, ha a pont előtt nagybetű van:

Jött a NATO. Jött a NATO\@.

◇ Névrövidítések után tildét teszünk, hogy ott semmiképpen ne legyen sortörés.

I. Newton I.~Newton

◇ Ügyeljünk a kötőjel, gondolatjel és a nagy gondolatjel megkülönböztetésére!

– Kötőjeles szóösszetételekben:

látod-e, Árpád-ház látod-e, Árpád-ház

– Gondolatjel, mely körül a magyarban mindig van szóköz:

– No fiúk – mondta Richárd, --\No\fiúk\--\mondta\Richárd,

– A nagy gondolatjelet pl. az angol használja, szóköz nélkül:

“That’s quite enough—I hope I shan’t grow any more—...” ‘‘That’s quite enough---I hope I shan’t grow any more---\dots’’

◇ A magyar idézőjel formája: 99 fent is, lent is, az angolé 6,9 vagy 66, 99, mindkettő fenn:

„magyar”, “english” ,,magyar’’, ‘‘english’’

◇ A vesszőt nem előzi meg szóköz. (Ez nem `LATEX` szabály, de gyakran megszegik.)

3.8. Ha nem vagyunk elégedettek az eredménnyel

◇ *Ha rossz az automatikus elválasztás, vagy nem akarunk elválasztást egy bizonyos helyen.*

Az első megoldás az, hogy az előfordulás helyén bejelöljük a szóba a helyes elválasztást egy olyan elválasztójellel, mely csak a forrásban látszik:

Nem volt nála kar\-ó\-ra, mikor felrepült a ka\-ró\-ra.

Második megoldásként (mely pl. az előző példában nem lenne alkalmazható) a preambulumban – a `\begin{document}` előtt – megadjuk, hogy egy szónak mi a helyes elválasztása:

`\hyphenation{ke-rék-pár-út}`.

Az `\mbox` parancs argumentumába írt szavak nem lesznek elválasztva: `\mbox{szarvas}`. Ha egy egész mondatrész kerül `\mbox`-ba, akkor a szavak között sortörés sem lesz.

A hosszú többjegyű mássalhangzók elválasztására egyszerű megoldás, ha a kettőzött mássalhangzó elé fordított aposztrófot teszünk:

fröccsen fröccsen fröccsen fröccsen fröccsen fröccsen fröccsen	<code>frö'ccsen</code>	<code>frö'ccsen</code>	<code>frö'ccsen</code>
csen fröccsen fröccsen fröccsen	<code>frö'ccsen</code>	<code>frö'ccsen</code>	<code>frö'ccsen</code>
	<code>frö'ccsen</code>	<code>frö'ccsen</code>	<code>frö'ccsen</code>

◇ *Ha rossz helyen történik a sortörés, vagy túlsordulás van.*

A sorok törése kétféle filozófia alapján lehetséges: vagy megengedünk túl nagy szóközöket, s cserében nem lesz túlsordulásunk, vagy nem engedünk meg túl nagy szóközöket de esetleg túlsordulásunk lesz. A \LaTeX alapértelmezésben ez utóbbit csinálja, aminek következtében az optimális végeredmény érdekében néha be kell avatkoznunk. Azonban fogadjuk meg azt a tanácsot, hogy a sortörésekbe csak az utolsó simításként szólunk bele. Ha beavatkozunk, három dolgot tehetünk:

- **Kikényszeríthetünk** egy sortörést a `\newline` vagy a vele ekvivalens `\\` paranccsal. Megjegyzendő, ha sortörést akarunk kikényszeríteni, ugyanakkor lapdobást megtiltani, akkor a `*` használandó. Az előző parancsokkal megtört sor *nem lesz igazítva* (nyomdai nyelven: *kizárva*) a margóhoz.
Ha a `\linebreak` parancsot használjuk, akkor a sortörést szintén kikényszerítjük, de a sor *igazítva lesz* a margóhoz.
- **Megtilthatjuk** a sortörést a `\nolinebreak` paranccsal.
- **Tanácsot adhatunk** a \LaTeX -nek, hogy itt jó lenne egy sortörés, de ha tud jobbat, akkor csinálja azt. Erre `\linebreak[n]` és `\nolinebreak[n]` parancs való, ahol n lehetséges értéke 0-tól 4-ig terjed, s ez a kérésünk erősségét fejezi ki.

◇ *A bekezdések behúzása.*

A bekezdések elválasztására két megoldás van: a behúzás, a másik a sorköz növelése a bekezdések között. A \LaTeX megoldása dokumentumosztályonként különbözhet. A bekezdés általában behúzással kezdődik a fejezetcímek utáni első bekezdés azonban nem.

Ha a \LaTeX megoldásán változtatni akarunk: a behúzást az `\indent` paranccsal kényszeríthetjük ki, míg a `\noindent` paranccsal letilthatjuk azt.

A bekezdések közötti sorköz megváltoztatására egy példa: `\setlength{\parskip}{2cm}`

Az `indentfirst` csomag betöltése után a fejezet első bekezdése is behúzással kezdődik.

◇ *Szándékosan akarunk egynél több szóközt hagyni.*

A `_` parancs egy szóköznyi hely beiktatását jelenti. Ez önmagában hatástalan a sor elején. Ekkor `\mbox{_}` a megoldás. (Ennek lényege, hogy a sor elején lenni kell valaminek, az `\mbox{_}` paranccsal egy láthatatlan, kiterjedés nélküli objektumot tettünk be.)

A `` parancs láthatatlan tintával írja le a „valami” szót. (Ez minden méretet tekintve egy akkora üres helyet hagy, mint amit a „valami” leírása jelentene.)

◇ *A felsorolásjel nem megfelelő.*

Ha `\item[ezt akarom]` módon adjuk ki az `\item` parancsot, akkor a felsorolásjel helyén „ezt akarom” lesz. A magyarban például nem szokásos a \bullet , inkább $-$. Ezt azonban nem célszerű minden egyes `\item`-nél beállítani, hanem egyszer tudatni a \LaTeX -el a preambulumban:

```
\renewcommand{\labelitemi}{\textbf{-}}
```

Jelen felsorolásnál például:

```
\renewcommand{\labelitemi}{\$ \diamondsuit \$}.
```

4. I ♥ math

4.1. Az első képlet

Az első képlet lehetne az, hogy $a+b=c$, amit pusztán a klaviatúráról begépeltem, de ez nem jól néz ki. Miért? A \TeX két alaplomban „járatható”: szöveg módban és matematikai módban.

Ha matematikai képletet akarunk gépelni, akkor a \TeX -et át kell állítani matematikai módba.

Ennek két fő oka van. Egyrészt egy sor tipográfiai kérdés máshogyan kezelendő a matematikai szövegek esetén. Hasonlítsuk össze az „első képletet” ezzel: $a + b = c$, amelyet így gépeltem: $\$a+b=c\$$. Itt az első dollár jellel áttértem matematikai módba, a másodikkal befejeztem azt és visszatértem szöveg módba.

A másik kényszerítő ok áttérésre, hogy a matematikai szimbólumok a szöveg módban nem is érhetők el. Például a fejezet címét így gépeltem: I $\$ \heartsuit \$$ math. A \heartsuit utasítást a \LaTeX szöveg módban meg sem értette volna.⁴

A továbbhaladás előtt egy fontos figyelmeztetés. A klaviatúráról bevitt sima szóköz szerepe a matematikai módban kizárólag a forrásszöveg tagolása, matematikai módban a szóközt külön ki kell kényszeríteni, ha szükségünk van rá. Erre pl. az alábbi parancsok alkalmasak: \backslashquad , \backslashquadr , \backslashenskip . (Próbáljuk ki!)

4.2. A képlet helye

A matematikai képletek elhelyezése kétféle módon történhet: a folyó szöveg sorában: pl. $a^2 + b^2 = c^2$ vagy a folyó szövegtől elválasztva (azt fogom rá mondani, hogy „kirakatban”). Pl.:

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

Ha a folyó szövegben térünk át matematikai módra, akkor erre a következő lehetőségek vannak: $\$képlet\$$, $\backslash(képlet\backslash)$, $\backslashbegin\{math\}$ képlet $\backslashend\{math\}$.

Ha egy egysoros képletet akarunk kirakatba tenni, akkor erre a legegyszerűbb lehetőség: $\backslash[képlet\backslash]$. Ha többsoros képletet, egyenletet teszünk a kirakatba, akkor ez a megoldás nem jó, később ezt is megtárgyaljuk.

4.3. Most jöjjenek végre a képletek

A matematikai képletekben használt jelek, betűfajták, elhelyezési lehetőségek széles skálájából csak egy csipetnyi ízelítőt nyújtunk. A képletek igényes gépeléséhez általában több, itt nem részletezett csomagot használunk. Emellett egy (vastagabb) referencia-anyagra is szükségünk lehet.

Az alábbi példákban nem mindig utalok a matematikai módra való áttérésre.

Halmazok

$x \in H$	$A \cup B$	$A \cap B$	$A \setminus B$	$A \times B$	$A \subset B$	$A \subseteq B$
$x \in H$	$A \cup B$	$A \cap B$	$A \setminus B$	$A \times B$	$A \subset B$	$A \subseteq B$

Továbbá: \emptyset : \backslashemptyset . Ha az utolsó két esetben $\backslashsub\dots$ helyett $\backsup\dots$ -t gépelünk akkor fordított irányú lesz a tartalmazás. S ha már itt tartunk, a három pont kódja matematikai módban \backldots a szövegmódú \backdots helyett.

Logikai jelek

$p \vee q$	$p \wedge q$	$p \Rightarrow q$	$p \Leftarrow q$	$p \Leftrightarrow q$	$\neg q$
$p \vee q$	$p \wedge q$	$p \Rightarrow q$	$p \Leftarrow q$	$p \Leftrightarrow q$	$\neg q$

Továbbá a kvantorok: $\forall \backforall$, $\exists \backexists$.

⁴ Bár az eléggé kétséges, hogy a szív matematikai szimbólum-e.

Műveleti jelek

$a + b$	$a - b$	a/b	$a \cdot b$	$a \times b$	$a \pm b$	$a \mp b$	$a \circ b$	$a \oplus b$
<code>a+b</code>	<code>a-b</code>	<code>a/b</code>	<code>a\cdot b</code>	<code>a\times b</code>	<code>a\pm b</code>	<code>a\mp b</code>	<code>a\circ b</code>	<code>a\oplus b</code>

Kisebb, nagyobb

A klaviatúráról begépett =, <, > jelek működnek matematikai módban is. Továbbá:

$$a \leq b, a \geq b. \quad \text{\code{\$a\le b}, \code{\$a\ge b}.$$

Mindent tagadunk

A `\reláció` tagadása általában: `\not\reláció`, kivétel két önállóan is létező alak, a `\notin` és a `\ne`, mely az első esetben szebb eredményt ad, mint a `\not\in`.

$$\not\subset, \not\in, \neq, \neq. \quad \text{\code{\$}\not\subset}, \code{\$}\not\in, \\ \code{\$}\notin, \code{\$}\not<, \code{\$}\not=, \code{\$}\ne.$$

Fenn és lenn

Az alapkonstrukció:

$$a^n, a_m, a^{nm}, a_{nm} \quad \text{\code{\$}a^n}, \code{\$}a_m, \code{\$}a^{nm}, \code{\$}a_{nm}$$

Illetve lenn-fenn variálva:

$$a_n^m, a_n^m, a^{a^a}. \quad \text{\code{\$}a_n^m}, \code{\$}a_{n^m}, \code{\$}a^{a^a}.$$

Valamit valamire

`\stackrel{f}{\rel{amit}}{\rel{amire}}`.

$$A \xrightarrow{f} B \quad \text{\code{\$}\stackrel{f}{\rel{A}}{\rel{B}}}$$

Vízszintes vonalak

A `\overline` és `\underline` rugalmas hosszúságú vízszintes vonalat húznak:

$$\overline{m+n}, \underline{a+b+c} \quad \text{\code{\$}\overline{m+n}}, \\ \code{\$}\underline{a+b+c}$$

Törtek, binomiális együtthatók

`\frac{számláló}{nevező}`, `n\choose k`. Pl.

$$\frac{1}{1 + \frac{1}{n}} \quad \text{\code{\$}\frac{1}{1+\frac{1}{n}}}$$

$$\binom{n}{k} \quad \text{\code{\$}n\choose k}$$

Gyökvonás

`\sqrt[hanyadik gyöke]{minek}`. Az opcionális argument nélkül négyzetgyököt kapunk:

$$\sqrt{2}, \sqrt[3]{2} \qquad \text{\$}\sqrt{2}\$, \text{\$}\sqrt[3]{2}\$$$

4.4. Zárójelek

A zárójelek lehetnek fix méretűek, vagy rugalmas méretűek, melyeket a \LaTeX automatikusan a benne foglalt képlet magasságához igazít. Alapszabály, hogy a zárójeleknek párban kell mindig állni (kezdő-záró), kivéve a `|` zárójelet. Ha valamely oknál fogva a kezdő vagy záró alak hiányzik, akkor be kell iktatnunk egy láthatatlan zárójelet, melynek parancsa: `\left`. (láthatatlan nyitó zárójel), `\right`. (láthatatlan záró zárójel).

Fix méretű zárójelek

A leggyakrabban használt zárójeleket a klaviatúráról visszük be. A speciálisabb zárójelet pedig paranccsal érjük el.

<code>(a)</code>	<code>[a]</code>	<code> a </code>	<code>{a}</code>	<code>\langle a \rangle</code>
<code>(a)</code>	<code>[a]</code>	<code> a </code>	<code>\{a\}</code>	<code>\langle a \rangle</code>

A fenti alapméreten kívül még négy fix méret van, de ezeket csak igen ritkán kell használnunk. Először mindig a rugalmas zárójellel próbálkozzunk, ha ez nem ad jó eredményt, csak akkor próbáljuk ki valamely fix méretet, melyek tehát a következők:

$$|, |, |, |, | \qquad \text{\$}\big|, \text{\$}\Big|, \text{\$}\bigg|, \text{\$}\Bigg|$$

A rugalmas méretű zárójelek

A konstrukció: `\leftzárójel... \rightzárójel`.

$$\left(\frac{a}{b}\right), \left(\sqrt{\frac{a}{a+\frac{1}{a}}}\right) \qquad \text{\$}\left(\frac{a}{b}\right)\$, \text{\$}\left(\sqrt{\frac{a}{a+\frac{1}{a}}}\right)\$.$$

4.5. Az analízis elemei

Függvények

A leképezések megadásakor gyakran használjuk a kettőspontot. Például $f: X \rightarrow Y$. Ezt a kettőspontot azonban *nem* a klaviatúráról visszük be, erre a `\colon` parancs való, mert ekkor a jobb oldali térköz a kettőspont után kicsit nagyobb lesz, mint a bal oldalon.

$$f: X \rightarrow Y \qquad \text{\$}f\colon X\to Y\$$$

A normál kettőspontot használhatjuk viszont másra:

$$a : b, a := x^2. \qquad \text{\$}a:b\$, \text{\$}a:=x^2\$.$$

Hasznos még a következőt tudni: $a \mapsto b$, melyet így gépeltem: `a\mapsto b`.

Szumma és integrál

A két jel használata eléggé hasonló: $\sum_{i=1}^n a_i^2$, $\int_a^b f$. Mindkét jel megjelenítése kicsit különbözik a folyó szövegben és a kirakatban. A szumma pl. így néz ki folyó szövegben: $\sum_{i=1}^n a_i^2$, így pedig a kirakatban: $\sum_{i=1}^n a_i^2$. Ha ránézünk erre a sorra, akkor megértjük, hogy miért van ez így. Nézzünk két példát kirakatban:

$$\sum_{i=1}^n a_i^2, \int_a^b f \qquad \backslash\text{sum}_{i=1}^n a_i^2, \backslash\text{int}_a^b f$$

Az integrállal kapcsolatban több finomságot kell megemlíteni. Ha a határokat nem szeretjük az előbbi módon, akkor megtehetjük a következőt: $\int_a^b f$, mely a kirakatban a következőképpen néz ki:

$$\int_a^b f$$

A másik finomság az eddig szándékosan mellőzött dx . Ezt praktikus az előtte szereplő függvénytől kissé elválasztani egy \backslash , beiktatásával: $\int_a^b f \backslash, dx$:

$$\int_a^b f dx \text{ helyett } \int_a^b f \backslash, dx$$

A kettős integrálban a két integráljelet pedig közelebb kell egymáshoz hoznunk. Ez így érhető el: $\int \int z \backslash, dx \backslash, dy$.

$$\int \int z dx dy \text{ helyett } \int \int z \backslash, dx \backslash, dy$$

Határérték

A kulcsszavak: \lim , \limsup , \liminf . Az alábbi példa kirakatban készült, a folyó szövegben ez is egy kicsit máshogyan néz ki.

$$\lim_{n \rightarrow \infty} \frac{1}{n} = 0 \qquad \backslash[\backslash\lim_{n \rightarrow \infty} \frac{1}{n} = 0 \backslash]$$

4.6. Függvénynevek

Figyeljük meg, hogy az előző példában a \lim fontjai nem a matematikai fontok, bár matematikai módban voltunk. Ez általában így van a speciális, „nevet kapott” függvényeknél is. Tehát $\sin x$, nem pedig $\sin x$, amit a $\$ \sin x \$$ gépelése adna. Az általános megoldás: $\$ \mathop{\text{trm}}{\text{függvénynév}} \$$, amire gyakran használt függvénynevek esetén érdemes is megtanítani a L^AT_EX-et a preambulumban:

$\backslash\text{newcommand}\{\text{tg}\}\{\mathop{\text{trm}}{\text{tg}}\}$

A leggyakoribb függvénynevek szerencsére be vannak építve, de sajnos nem mindig a magyar használatnak megfelelően. (Az utóbbiakat ki is hagyom most a felsorolásból.)

log	lg	ln	exp	sin	cos	det	min	max
$\backslash\text{log}$	$\backslash\text{lg}$	$\backslash\text{ln}$	$\backslash\text{exp}$	$\backslash\text{sin}$	$\backslash\text{cos}$	$\backslash\text{det}$	$\backslash\text{min}$	$\backslash\text{max}$

A \min és a \max a \lim -hez hasonlóan használható.

4.7. Szöveg matematikai módban

Sokszor szükség van arra, hogy egy képletben szöveget is elhelyezzünk, tehát matematikai módban ideiglenesen szövegmódra váltsunk. Ennek semmi akadálya a korábban megismert parancsokkal. Pl.

$$|x| = x \text{ ha } x \geq 0$$

$$|x| = -x \text{ ha } x < 0$$

Figyeljük meg a szóközöket a *ha* körül!

4.8. Kirakatban

Számozott egyenlet

Az `equation` környezet hatása azonos a `\[\]` parancs-párral, de egy automatikusan generált címkét is kapunk.

$$3x + 5y = 2 \quad (1)$$

```
\begin{equation}
3x+5y=2
\end{equation}
```

Többsoros formulák

A matematikai formulák sokszor nem férnek el egy sorban. Az optimális az lenne, ha a \LaTeX a matematikai formulákat is törné, mint a szöveget. Erre azonban jelenleg a \LaTeX nincs felkészítve. (Ez egyúttal azt is jelzi, hogy a probléma általában nem is olyan könnyű.)

A többsoros formulák kirakatban való elhelyezésére az `eqnarray`, vagy az `eqnarray*` környezetet használhatjuk. A csillagos változat nem generál címkét a sorokhoz. A használatuk lényege, hogy minden sorban ki kell jelölni egy igazítási pontot, melyet két `&` közé teszünk: `&igazítási pont&`. Ezek az igazítási pontok a szedésnél egymás alá fognak kerülni. A sorok végét `\` jelzi:

$$\begin{array}{rcl} 3x + 4y & = & 25 \\ -2x + 8y & = & 42 \end{array}$$

```
\begin{eqnarray*}
3x+4y&=&25\\
-2x+8y&=&42
\end{eqnarray*}
```

A többsoros formulák szedésének rengeteg fortélyja van, nagyon sok probléma megoldásához speciális csomagok használhatók. A következő példák csak néhány problémát illusztrálnak.

Többsoros képletek első sora sokszor gondot okoz:

$$\begin{array}{l} \sin^4 x + \cos^4 x = \\ = \sin^4 x + (1 - \sin^2 x)^2 = \\ = \sin^4 x + 1 - 2 \sin^2 x + \sin^4 x. \end{array}$$

```
\begin{eqnarray*}
\lefteqn{\sin^4 x + \cos^4 x =} \\
&&= \sin^4 x + (1 - \sin^2 x)^2 = \\
&&= \sin^4 x + 1 - 2 \sin^2 x + \sin^4 x.
\end{eqnarray*}
```

Ha a három egyenlőség jelet igazítottam volna, akkor az eredmény csúnya lenne. Ezért a `\lefteqn` paranccsal az első sort balra zártam, az igazítási pontot pedig a további sorok elejében határoztam meg.

Ha egy sor `+`-al vagy `-` műveleti jellel kezdődik, akkor probléma az, hogy ezen műveleti jelek mindkét oldalán szokott állni valami. Hasonlítsuk össze:

$$\begin{array}{r} y = a + b + c \\ \quad + d + e + f \end{array}$$

$$\begin{array}{r} y = a + b + c \\ \quad + d + e + f \end{array}$$

A különbség egészen apró, mégis a másodikat tartják helyesnek, ahol a második sorban `\mbox{+d}`-t gépeltem, hogy legyen a `+` mindkét oldalán valami.

4.9. Betűváltozatok

A matematikai módban is több betűváltozat van. A szövegmódú `\textváltozat` parancs helyett `\mathváltozat` az elérésük. Pl: $\mathbf{a^2 + b^2 = c^2}$ (gépelése tehát: `\mathbf{a^2+b^2=c^2}`).

A gyakran használt *kalligrafikus* betűk a `\mathcal` parancs segítségével érhetőek el: `\mathcal{A B C D E}` eredménye: $\mathcal{A B C D E}$. Kalligrafikus változata csak nagybetűknek van.

A matematikai jeleknek egyetlen extra változata a félkövér. Erre valószínűleg kizárólag akkor van szükségünk, ha félkövér szövegben matematikai formulát akarunk elhelyezni. Még *szöveg módban* át kell váltanunk a `\boldmath` deklarációval: $\mathbf{a^2 + b^2 = c^2}$ (gépelése: `\boldmath $a^2+b^2=c^2$`).

4.10. Táblázatok

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

2. táblázat. Görög kisbetűk.

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

3. táblázat. Görög nagybetűk.

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

4. táblázat. Ékezetek matematikai módban.

5. Táblázatok

Táblázatok készítésére a L^AT_EX-nek két beépített környezete van. Az egyik a `tabular`, a másik az `array`. Utóbbi csak matematikai módban használható. Ezek egyszerűbb táblázatok szerkesztéséhez elegendők is. (A lehetőségeket különböző csomagok – a legfontosabb talán az `array` – lényegesen kibővítik.)

5.1. A legegyszerűbb táblázat

Minden táblázat szerkesztéséhez meg kell adnunk, hogy hány oszlopunk lesz, s az egyes oszlopokban az elemek (sorok) hogyan legyenek egymáshoz igazítva.

Ezeket az információkat a `tabular` vagy `array` parancs kötelező argumentumában kell megadni. Tekintsük az alábbi táblázatot:

4	9	2	<code>\begin{array}{ccc}</code>
3	5	7	<code>4&9&2\\</code>
8	1	6	<code>3&5&7\\</code>
			<code>8&1&6</code>
			<code>\end{array}</code>

A `\begin{array}` parancs `{ccc}` argumentumával megszabtuk a táblázat oszlopai számát (a 3 db betű 3 oszlopra utal), s az oszlopokban elvégzett igazítás típusát (`c` – *center* középre igazítást írtunk elő mindhárom oszlopban.) Az elemeket soronként gépeltük, a sorban az oszlopokat `&` választja el. A sor végét `\\` zárja. A `{ccc}`-t gépelhettük volna így is: `{*3c}`, vagyis háromszol ismételje a `c`-t. Az igazítás típusa lehet még: `l` – *left*, *bal*, `r` – *right*, *jobb*. (Ezek a betűk az *igazító parancsok*.)

Még egy egyszerű példa, más igazítás típusokkal:

Bevétel:	12e Ft	<code>\begin{tabular}{lr}</code>
Kiadás:	15e Ft	<code>Bevétel:& 12e Ft\\</code>
Haszon:	–3e Ft	<code>Kiadás:& 15e Ft\\</code>
		<code>Haszon:& \$-3e Ft</code>
		<code>\end{tabular}</code>

Ennyi a mátrixok szerkesztéséhez már elég is. Az első példánk köré már csak zárójelet kell tenni:

$\left(\begin{array}{ccc} 4 & 9 & 2 \\ 3 & 5 & 7 \\ 8 & 1 & 6 \end{array} \right)$	<code>\left(</code>
	<code>\begin{array}{ccc}</code>
	<code>4&9&2\\</code>
	<code>3&5&7\\</code>
	<code>8&1&6</code>
	<code>\end{array}</code>
	<code>\right)</code>

Vagy egy másik gyakran használatos matematikai típus:

$ x = \begin{cases} x & \text{ha } x \geq 0 \\ -x & \text{ha } x < 0. \end{cases}$	<code> x =\left{\</code>
	<code>\begin{array}{rl}</code>
	<code>x&\textrm{ha } x\ge 0\\</code>
	<code>-x&\textrm{ha }x<0.</code>
	<code>\end{array}</code>
	<code>\right.</code>

5.2. Vonalak a táblázatban

Ha a táblázat oszlopaival vonalakkal el akarjuk egymástól választani, akkor ezt a `|` karakterrel az oszlopszerkezetet megadó argumentumban kell jelölni. A sorok közé vonalat húzhatunk, ha a sortörés (`\\`) után beiktatjuk a `\hline` parancsot. (Ez nem fog külön sornak számítani.)

4	9	2
3	5	7
8	1	6

```
\begin{array}{|c|c|c|}
\hline
4&9&2\\ \hline
3&5&7\\ \hline
8&1&6\\ \hline
\end{array}
```

Bevétel:	12e Ft
Kiadás:	15e Ft
Haszon:	-3e Ft

```
\begin{tabular}{l|r}
Bevétel:& 12e Ft\\ \hline
Kiadás:& 15e Ft\\ \hline\hline
Haszon:& $-3e Ft
\end{tabular}
```

5.3. Parancsok az argumentumban

Beszúrás az oszlopok közé

A `@{karakter}` parancs két oszlop közé beszúrja a karakter-t. Speciálisan `@{}` 0 szélességűre csökkenti az oszlopok közötti távolságot:

Indulás	8:00
Érkezés	10:15
Menetidő	2:15

```
\begin{tabular}{@{}l|r@{:}l@{}}
Indulás&8&00\\
Érkezés&10&15\\
\hline
Menetidő& 2&15
\end{tabular}
```

(Figyeljük meg, hogy most a vízszintes vonal nem lóg túl az oszlopokon, mint az előző példában.)

Adott szélességű oszlop

Lehetőség van arra, hogy valamelyik (akár mindegyik) oszlop adott szélességű legyen. Ilyenkor a `LATEX` az oszlopba írt szöveget szükség esetén töri. Ezt a `p{távolság}` paranccsal érhetjük el.

szabályos test	csúcsok száma	lapok száma
szabályos tetraéder	4	4

```
\begin{tabular}{|p{1.5cm}|p{1cm}|p{1cm}|}
\hline
szabályos test&csúcsok száma&lapok száma\\
...
\end{tabular}
```

5.4. Összevont oszlopok

A `\multicolumn` paranccsal egy sorban több oszlop összefogható egyetlen oszloppá. (Például több oszlophoz azonos felirat tartozik.) Használata: `\multicolumn{n}{igazítás}{szöveg}`

Határidő		Vételi ajánlat	
		min.	max.
EURÓ	Július	264,30	264,30

```

\begin{tabular}{clrr}
\hline
\multicolumn{2}{c}{Határidő}&
\multicolumn{2}{c}{Vételi ajánlat}\\
\multicolumn{2}{c}{}&min.&max.\\
\hline
EURÓ &Július&264,30&264,30\\
\hline
\end{tabular}

```

Speciális használata a `\multicolumn` parancsnak, ha egy oszlopban egy elemre az igazítást másként akarjuk véghezvinni, mint általában az oszlopban. Az előző táblázatban pl. a „min.” és a „max.” jobb lenne, ha középre lenne igazítva. A megoldás, hogy `\multicolumn{1}{c}{min.}`-t és `\multicolumn{1}{c}{max.}`-t gépelek:

Határidő		Vételi ajánlat	
		min.	max.
EURÓ	Július	264,30	264,30

A vízszintes vonal hossza

A `\cline` parancs argumentumába kötőjellel elválasztva a kezdő és záró oszlop sorszámát kell beírni.

Határidő		Vételi ajánlat	
		min.	max.
EURÓ	Július	264,30	264,30

```

\begin{tabular}{clrr}
\hline
\multicolumn{2}{c}{Határidő}&
\multicolumn{2}{c}{Vételi ajánlat}\\
\cline{3-4}...
\end{tabular}

```

Függőleges vonal az összevont oszlopokban

Helyezzünk el függőleges vonalakat az előző tőzsdei táblázatban! Ha `\begin{tabular}{|clrr|}` módon teszem, ez az eredmény:

Határidő		Vételi ajánlat	
		min.	max.
EURÓ	Július	264,30	264,30

Mindez azért történt így, mert a `\multicolumn` parancs egyben felmentést adott a függőleges vonalak húzása alól is. Ha a három függőleges vonalat végig akarom húzni, akkor ezt a `\multicolumn` argumentumában kell kérnem:

Határidő		Vételi ajánlat	
		min.	max.
EURÓ	Július	264,30	264,30

```

\begin{tabular}{|clrr|}
\hline
\multicolumn{2}{|c|}{Határidő}&
\multicolumn{2}{|c|}{Vételi ajánlat}\\
\cline{3-4}
\multicolumn{2}{|c|}{}&
\multicolumn{1}{c}{min.}&
\multicolumn{1}{c}{max.}\\
\hline
EURÓ &Július&264,30&264,30\\
\hline
\end{tabular}

```

A `|` karaktert néha elvileg több helyen is el lehet helyezni. Általános elvként azt kell megjegyezni, hogy minden oszlop (kivéve az elsőt) az igazító paranccsal kezdődik (l, r, c, p) s a függőleges vonal ezután következik.

6. Bibliográfia készítése

A bibliográfia készítéséhez egy külön programot kell segítségül hívnunk a \LaTeX mellett, a \BibTeX -et. Mit kínál a \LaTeX és a \BibTeX a bibliográfia készítéssel kapcsolatban?

- Lehetőség van egy folyamatosan bővíthető és több műben is felhasználható adatbázis létrehozására.
- A műben elhelyezett hivatkozási kulcsok alapján ebből az adatbázisból a \BibTeX kiválogatja az adott műben éppen használt bibliográfiai tételeket,
- ezeket rendezi (rendezheti több szempont alapján is), címkével látja el, azaz megcsinálja az irodalomjegyzéket;
- majd a címkeket (már a \LaTeX) berakja a szövegbe a megfelelő hivatkozások helyére.

Aki szerkesztett már bibliográfiát, az különösen meg tudja becsülni a \BibTeX -et, hiszen kézzel nagyon nehéz a tételeket egységes tipográfiai stílusban rendezni.

6.1. A bibliográfiai adatbázis létrehozása

Az adatbázisunk (nevezzük ezt a továbbiakban \BibTeX fájlnak) az éppen szerkesztett fájlról független közönséges szöveges, `bib` kiterjesztésű fájl. Ennek szerkesztését a \TeX -re felkészített szövegszerkesztők rendszerint megkönnyítik. A bibliográfiai tételeket adatbáziselemekben kell elhelyezni. Egy adatbáziselem a következő szerkezetű:

```
@elemtípus{kulcs,           @book{radoorban,
mezőnév={szöveg},         author={Ferenc Radó and Béla Orbán},
mezőnév={szöveg},         title={A geometria mai szemmel},
mezőnév={szöveg},         publisher={Dacia Könyvkiadó},
mezőnév= szám              year= 1981
}                             }
```

Ebben a példában az *elemtípus* `book`, azaz könyv. A *kulcs* egy olyan fantázianév, mely segítségével az adatbáziselemre tudunk majd hivatkozni a szövegben. A többi mezőnév és szövegelem magáért beszél. Arra kell még ügyelni, hogy az évszámon kívül a „szöveg”-et zárójelbe kell tenni. Egyes elemtípusoknál más-más mezőnevek szerepelhetnek, kötelezően, vagy opcionálisan.

article – Folyóiratcikk, újságcikk.

kötelező: author, title, journal, year

opcionális: volume, number, pages, month, note

book – Könyv.

kötelező: author vagy editor, title, publisher, year

opcionális: volume vagy number, series, address, edition, month, note

inbook – Egy könyv része.

kötelező: author vagy editor, title, chapter és/vagy pages, publisher, year

opcionális: volume vagy number, series, type, address, edition, month, note

booklet – Nyomtatott munka, kiadó nélkül.

kötelező: title

opcionális: author, howpublished, address, month, year, note

incollection – Saját címmel rendelkező könyvrész.

kötelező: author, title, booktitle, publisher, year

opcionális: editor, volume vagy number, series, type, chapter, pages, address, edition, month, note

proceedings – Konferencia közlemény.

kötelező: title, year

opcionális: editor, volume vagy number, series, address, publisher, month, organization, note

inproceedings – Konferencia közleményben megjelent cikk.

kötelező: author, title, booktitle, year

opcionális: editor, volume vagy number, series, pages, address, month, organization, publisher, note

manual – Technikai dokumentáció.

kötelező: title

opcionális: author, organization, address, edition, month, year, note

mastersthesis – Diplomamunka, szakdolgozat.

kötelező: author, title, school, year

opcionális: type, address, month, note

misc – Nem besorolható.

kötelező: –

opcionális: author, title, school, year

Az előző mezőnevek egy kivétellel informatívak. Az egy kivétel a *type* mezőnév, mely módosítja a bibliográfiai elem típusát. Pl. ha a *mastersthesis* elemtípusnál nem adjuk meg, akkor a megfelelő bibliográfiai tételnél az irodalomjegyzékbe a BibTeX automatikusan generál egy *Master's thesis* szöveget. Ha megadjuk, hogy `\type={Szakdolgozat}`, akkor ezt a szöveget kapjuk.

A nevek megadása

A legjobb, ha mindig *teljes nevet* veszünk be az adatbázisba, az esetleges rövidítést pedig a BibTeX elvégzi, ha a bibliográfia stílusa ezt kéri. Ha valami oknál fogva egy szerző keresztneve csak rövidítve ismert, vagy egyáltalán nem ismert, akkor ezt még a BibTeX tudja kezelni, csak a bibliográfiánk esetleg nem lesz teljesen egységes.

A neveket az angol szokás szerinti sorrendben kell megadni. A magyar nevek beírásánál egyedül az „ifj.” típusú rövidítésekre kell vigyázni. Az alábbi példák mindegyike helyes:

```
author={Esti, Kornél}
author={Kornél Esti}
author={Esti, ifj., Kornél}
author={Ludwig van Beethoven}
author={van Beethoven, Ludwig}
author={Johann Sebastian Bach}
author={Bach, Johann Sebastian}
```

Mint azt a fejezet elején láttuk, a társszerzőket *and* -el kell elválasztani. (Arra külön visszatérünk, hogyan lehet ezt a magyar szövegben *és*-re, vagy vesszőre cserélni.)

A BibTeX fájl nevét a `\bibliography{fájlnév}` paranccsal kell tudatni a műben. (Kiterjesztés nélkül.)

A bibliográfia stílusai

A bibliográfia stílusát a `\bibliographystyle{stílusnév}` paranccsal adjuk meg, szintén a fő `tex` fájlban. Az „alap” BibTeX négy standard stílust tartalmaz:

- `plain` – Az elemek ABC szerint rendezettek, a címkék számok, a nevek teljes alakban szerepelnek.
- `unsrt` – Az elemek a szövegbeli hivatkozások sorrendjében rendezettek, a címkék számok.
- `alpha` – Az elemek ABC szerint rendezettek, a címkék a szerzők nevéből és a kiadvány kiadási évéből képezett képezett rövidítések.
- `abbrv` – Olyan mint a `plain` stílus, de a szerzők nevei (és még néhány mező) rövidítettek.

A CTAN szervereken sok más egyéb stílus van, sőt van lehetőség saját bibliográfia stílus készítésére is.

6.2. A kulcsok elhelyezése a szövegben, a bibliográfia generálása

A BibTeX fájlban minden tételhez rendeltünk egy kulcsot. A szövegben erre a kulcsra kell hivatkozni a `\cite{szöveg}{kulcs}` paranccsal. Pl.:

A [8, 25. oldalon] találhatójuk. . . A `\cite[25.\ oldalon]{radoorban}` találhatójuk

Itt a `radoorban` kulccsal címkézett tétel az irodalomjegyzékben történetesen a 8. elem.

Ha azt akarjuk, hogy az irodalomjegyzékben olyan mű is szerepeljen, amelyre nem történt hivatkozás, akkor ezt a `\nocite{kulcslista}` paranccsal lehet elérni. Ha az adatbázisunk mindegyik tételét ki akarjuk írni, akkor ez a következőképpen módosul: `\nocite{*}`.

A kész mű esetén a bibliográfia készítése a következőképpen zajlik:

1. `LATEX`. Ekkor a műben levő `\cite` hivatkozások belekerülnek egy `.aux` fájlba.
2. BibTeX. A BibTeX az `aux` fájl alapján kikeresi az adatbázisból a tételeket, létrehozza a bibliográfiát, mely egy `.bbl` fájlba kerül.
3. `LATEX`. A `LATEX` beolvassa a `.bbl` fájlt, de a címkék még nem kerülnek a szövegbe.
4. `LATEX`. A címkék bekerülnek a szövegbe.

6.3. Ha be kell avatkozni

A fentebbi eljárás mutatja, ha be kell avatkozni a bibliográfiába, akkor ezt a `.bbl` fájlban tudjuk megtenni. (Csak az utolsó BibTeX futtatás után, mert egyébként a fájl felülíródik.) Ha beletekintünk a `bbl` fájlba, akkor a tételeket a következő formában találjuk:⁵

```
\bibitem{radoorban}
F.~Radó and B.~Orbán.
\newblock {\em A geometria mai szemmel}.
\newblock Dacia Könyvkiadó, 1981.
```

Tehát egy tétel egy `bibitem`-ben van, melynek tetszőleges adatát módosíthatjuk. Felsorolok néhány feltételezhető esetet:

- ◇ Az *and* szócskát *és*-re vagy vesszőre kell cserélni.
- ◇ Magyar szöveggörnyezetben a magyar és mondjuk angol neveket tartalmazó irodalomjegyzékben a magyar neveket magyar alakjukban, az angol neveket angol alakjukban szokás írni.
- ◇ Problémát okozhat még az ékezetes betűk nem megfelelő lexikografikus rendezése.

⁵Tehát a következő sorokat a BibTeX hozta létre, nem én gépeltem be. Láthatjuk, hogy rövidített névalak van, tehát a bibliográfiám stílusa `abbrv` volt.

7. Ami a kiadványszerkesztéshez még szükséges lehet

7.1. Lábjegyzet, széljegyzet, végjegyzet

Kiadványban gyakran alkalmaznak lábjegyzetet⁶, széljegyzetet⁷ és végjegyzetet⁹⁹.

A lap alján elkülönítve megjelenő szöveget lábjegyzetnek nevezzük és a következő paranccsal hívjuk meg:

```
\footnote[num]{szöveg}
```

A `\footnote` parancs eggyel megnöveli a footnote számláló értékét, a *szöveget* a lábjegyzetbe írja, elé egy jelet tesz, amely a szövegben is megjelenik a parancs kiadásának helyén. A *num* megadása opcionális. A *num* megadásakor az automatikus sorszámozás felfüggesztődik, ilyenkor a footnote számláló helyén a *num* jelenik meg mind a szövegben mind a lábjegyzetben.

A lábjegyzetek a könyvekben és a cikkekben minden fejezetben újra sorszámozódnak. Ugyanez elérhető szakaszonként is, ha kiadjuk a

```
\setcounter{footnote}{0}
```

parancsot. A lábjegyzet távolságát a szövegtől az

```
\addtolength{\skip\footins}{távolság}
```

paranccsal lehet állítani. A lábjegyzet jeleit is megváltoztathatjuk a

```
\renewcommand{\thefootnote}{\alph{footnote}}
```

paranccsal. Az `\alph` kicserélhető az `\Alph`, `\roman`, `\Roman`, `\arabic`, `\fnsymbol` parancsok bármelyikére.

Széljegyzetet a

```
\marginpar{szöveg}
```

paranccsal készítünk. Egy oldalas szedésnél mindig a jobb szélre, kétoldalas szedésnél mindig a külső *széljegyzet* margóra kerül a szöveg.

Végjegyzetnél az endnotes csomag betöltése után a lábjegyzetnél megismert parancsokhoz hasonló parancsok használhatóak, csak a foot helyett end szerepel. A végjegyzetet a `\theendnotes` parancs jeleníti meg.

7.2. Hasábozás

Nagyon ritkán többhasábos szöveget is szedünk. Alapértelmezésben a dokumentum egyhasábos, mint ezt eddig is tapasztaltuk. Ez megváltoztatható a dokumentum elején a preambulumban kéthasábosra a

```
\documentclass[twocolumn,a4paper]{article}
```

paranccsal, vagy a dokumentum közepén is a

```
\onecolumn  
\twocolumn[szöveg]
```

paranccsal, a *szöveg* opcionális, a kéthasábos szedésnél a lap teljes szélességében jelenik meg. Mindkét parancs kiadásával új oldal kezdődik, így az oldal közepén ezekkel nem változtatható meg a hasábok száma. Erre a célra be kell tölteni a `fix2col` csomagot majd a

```
\begin{multicols}{n}[bevezető]  
szöveg  
\end{multicols}
```

parancsot használjuk, ahol $n < 10$. Ez n hasábra szedi a *szöveget*, ha a *bevezető* meg van adva akkor a *bevezető* szöveg a hasábok fölé kerül.

⁶A lap alján elkülönítve megjelenő szöveg

⁷A lap szélén, margón megjelenő szöveg

⁹⁹A szakasz, a fejezet végén megjelenő szöveg

⁷Ez a láb jegyzet helye

7.3. Képbeillesztés

Gyakran előfordul, hogy valamilyen ábrát szeretnénk beilleszteni a dokumentumba. A betöltéshez a `\graphicx` csomag `\includegraphics{}` parancsát használjuk. Ahhoz, hogy a tördeléssel a \LaTeX sikeresen boldoguljon, szükség van a kép geometriai méreteire. Ezt az információt a PostScript formátumú képek általában tartalmazzák, míg például a `jpg` formátumú képek nem. Egy `jpg` kép méreteinek megállapításához egy \LaTeX -től független segédprogramot használhatunk, az *ebb*-t. A végeredményhez a dokumentumot fordítsuk `pdf` formátumba!

A továbbiakban egy `tuli.jpg` nevű fájl beillesztésével foglalkozunk. Először parancssorból futtassuk az *ebb* programot:

```
ebb tuli.jpg
```

Az eredmény egy `tuli.bb` nevezetű fájl, amelyet rögtön nyissunk is meg egy egyszerű szövegszerkesztővel (akár a *Kile*-vel). A következőt kell látnunk (persze a dátum biztosan nem ez lesz.):

```
%Title: ./tuli.jpg
%Creator: ebb Version 0.5.2
%BoundingBox: 0 0 369 276
%CreationDate: Tue Sep 13 21:53:50 2005
```

Nekünk ebből csak a `BoundingBox` utáni négy számra van szükségünk. A `tex` dokumentumba visszatérve gépeljük be a következőt:

```
\includegraphics[bb=0 0 369 276,scale=0.25]{tuli.jpg}
```

A `scale=0.25` azt jelenti, hogy a kép eredeti méretét negyedére kicsinyítettük.



A következőt kapjuk:

Az `\includegraphics` parancs segítségével betett kép pontosan oda kerül, ahol a parancsot kiadtuk. Ha középre szerkesztve, feliratozva, számozva akarjuk a képet betenni, akkor `figure` környezetet használjuk:

```
\begin{figure}[h]
\centering
\includegraphics[bb=0 0 369 276,scale=0.25]{tuli.jpg}
\caption{Két tulipán.}
\end{figure}
```



1. ábra. Két tulipán.

A `[h]` a *here* angol szót rövidíti, azaz a kép ott legyen, ahol a parancsot kiadtuk. Ezt persze nem mindig tudja a \LaTeX teljesíteni, mert esetleg már nincs annyi hely a lapon. További opciók: `[t]` – top, azaz a kép a lap tetejére; `[b]` – bottom, a lap aljára; `[p]` – page, külön oldalra kerüljön.

A képiünket a megfelelő parancsok segítségével transzformálhatjuk. Az előbb már használt skálázáson túl például lehetőség van forgatásra. A használatot az alábbiakból értelemszerűen le tudjuk olvasni:

```

\begin{figure}[h]
\centering
\includegraphics[bb=0 0 369 276,scale=0.25, angle=45]{tuli.jpg}
\caption{Két tulipán -- forgatva}
\end{figure}

```



2. ábra. Két tulipán – forgatva

Előírt magasság és szélesség:

```

\begin{figure}[h]
\centering
\includegraphics[bb=0 0 369 276,scale=0.25, width=5cm, height=2cm]{tuli.jpg}
\caption{Két tulipán -- előírt magasság és szélesség}
\end{figure}

```



3. ábra. Két tulipán – előírt magasság és szélesség