Nemzeti Tehetség Program

Az ELTE Eötvös József Collegium
„Szakkollégista tudományos és közéleti műhelykiadványok, rendezvények, utánpótlásnevelés”
című pályázatának (Azonosítószám: NTP-SZKOLL-20-0032)
keretében megrendezett

LaTeX a bölcsészek szolgálatában

című előadássorozat

összefoglaló anyaga

ELTE EÖTVÖS JÓZSEF COLLEGIUM
DR. HORVÁTH LÁSZLÓ
IGAZGATÓ

H-1118 Budapest, Ménesi út 11-13
Tel.: +36 1 460 4481 • Fax.: +36 1 209 2044
E-mail: titkarsag@eotvos.elte.hu • horvathl@eotvos.elte.hu
Honlap: http://www.eotvos.elte.hu/

LaTeX a bölcsészek szolgálatában

ELTE Eötvös József Collegium

NTP-SZKOLL-20 pályázat

Mayer Gyula

2021. tavaszán

Tartalomjegyzék

1. FEJEZET

Bevezetés

A LaTeX egy általános célú dokumentumkészítő rendszer, mely a TeX nevű szedőprogramra épül. Segítségével a legjobb nyomdai minőségben készíthető a rövid feljegyzéstől kezdve a levélen, diplomamunkán, internetes oktatási anyagon, tudományos cikken át egészen a könyvig szinte tetszőleges írásos anyag. A sima TeX-től a LaTeX-et az különbözteti meg, hogy sokkal erősebb és hangsúlyozott benne az a törekvés, hogy a dokumentum szerkezeti, logikai elemei legyenek egységesen megjelölve (tagging, tagolás), az elemekhez kapcsolódó formai/formázási utasítások pedig a dokumentumtól elkülönülő stílusfájlban rögzíttessenek.

A TeX, s vele együtt a LaTeX különleges program. Különlegessé teszi a számítástechnikában szokatlan idős kora (1978-ban született), archaikus felépítése, s ezzel együtt fiatalos alkalmazkodóképessége; minimális hardverigénye, mindemellett nagy tudása; minden számítógéptípuson és minden elterjedt operációs rendszeren való hozzáférhetősége (DOS, Windows, OS/2, MacOS, Linux, UNIX, VMS...), ezek szinte tökéletes kompatibilitása; ingyenessége, szabadon elérhető és felhasználható nyílt forráskódja.

A program kezdetben leginkább a műszaki és természettudományi egyetemek és kutatóintézetek világában terjedt el, azonban a múlt század végére már a bölcsészettudományok művelői is kiterjedten alkalmazták. Évente többezer LaTeX-ben írt könyv jelenik meg, a tudományos folyóiratokban, interneten megjelent cikkek, írások, oktatási anyagok és vele írt diplomamunkák száma milliós nagyságrendű. Egyes tudományokban használata de facto szabvánnyá vált, így sokaknak nélkülözhetetlen segédeszköz.

Magyarországon először 1998-ban jelent meg könyv a LaTeX-ről, Wettl Ferenc, Mayer Gyula és Sudár Csaba LaTeX kezdőknek és haladóknak című műve [14] („zöld könyv"). Ennek, egyes részeiben teljesen átdolgozott, új részeket tartalmazó kiadása [15] is már 17 éves e sorok írásakor. Ez a rövid jegyzet erősen támaszkodik erre a „kék könyvre", két jelentős eltéréssel: 1) a kék könyv írásakor még csak elterjedőben volt az UTF-8 kódolás, ami mára általánosan használtnak tekinthető és tekintendő, 2) a kézikönyvben egy feladat megoldására több különböző megoldást is bemutattunk, hogy a felhasználó azok közül igényeinek megfelelően választhasson, ebben a jegyzetben viszont csak a filológusok speciális igényeihez szerintünk legjobban illő módszereket ismertetjük.

1.1. TeX és LaTeX

TeX, plain TeX • A TeX dokumentumkészítésre, szedésre szolgáló programrendszer. Megalkotásába 1977 májusában kezdett Donald E. Knuth stanfordi matematikus, miután sokat bajlódott „A számítógép-programozás művészete" című művének kiadásával [7, 8]. Olyan programot alkotott, mellyel vágyának megfelelően szép megjelenésű művek nyomtathatók, és mellyel az olyan bonyolult feladatok is megoldhatók, mint a matematikai képletek szedése [9].

A TeX program a nyomdászat több évszázados tudását viszi számítógépre úgy, hogy az arra jellemző gondolkodásmód egy részét is megőrzi, és kibővíti azt a számítástechnika adta új lehetőségekkel. A program lényegében az ólomszedést modellezve működik.

Maga a TEX szó a művészet jelentésű görög τέχνη – nagybetűkkel írva TEXNH – szó első három betűjéből áll. E szó kiolvasva „techné”, így a TEX nem „teksz”-nek, hanem „tech”-nek ejtendő (angol beszélők esetében a [tek] kiejtés is használatos). A TEX márkajel középső, ejtett ‚E’ betűje a TEX tipográfiai képességeit jelzi (és egyúttal megkülönbözteti más, hasonló nevű rendszerektől). Írógépen vagy egy egyszerű szövegfájlba írva a márkajel `TeX`.

A TEX-rendszernek része a METAFONT nevű program, mely a TEX saját betűkészleteinek létrehozására és újak alkotására is alkalmas [10].

A TEX mára nagyon stabil programmá vált, amelyen már csak ritkán változtatnak. A legutolsó verzió száma $3{,}14159$.[1] A TEX-ben szinte minden tipográfiai feladat megoldható, de néha csak igen nagy fáradsággal. Számítástechnikai hasonlattal élve a TEX a nyomdászat assemblere, melyben több száz elemi parancs segítségével bármely szöveg a kívánt formába önthető. A TEX-ben makrókat is írhatunk, használatukkal a bonyolultabb tipográfiai problémák megoldására is könnyen használható parancsokat készíthetünk. Egy-egy jól összeállított makrócsomag a dokumentum-szerkesztés egész folyamatát leegyszerűsítheti.

Több ilyen makrócsomag is készült. Az elsőt maga Knuth készítette el, aminek a „plain TEX” nevet adta. Ennek alapkönyve ma is a TEXBook [9]. Magyarul Bujdosó Gyöngyi és Fazekas Attila könyvében olvashatunk róla [1] (további magyar nyelvű könyvek: az első magyar TEX-disztribúció kézikönyve [3] és Doob egy művének fordítása [2]). Az első időkben még két további megoldás terjedt el széles körben: az egyik az American Mathematical Society ($\mathcal{AMS}$) által támogatott, Michael Spivak által fejlesztett $\mathcal{AMS}$-TEX [13, 1], a másik Leslie Lamport LATEX programcsomagja [11, 12]. Gyakran találkozni azzal a szóhasználattal, mely a plain TEX helyett TEX-et mond. TEX-en az egész rendszert értjük, melynek a plain TEX és a LATEX is része.

LATEX • Lamport az 1980-as évek elején kezdett a TEX-re épített dokumentumkezelő rendszerének megalkotásába. Ha a TEX-et a nyomdászat assemblerének neveztük, a LATEX-et egy magas szintű dokumentum-leíró nyelvnek tekinthetjük. A magas szintű azt jelenti, hogy egy-egy LATEX-parancs mögött igen sok – főként tipográfiai – tudás van, ami lehetővé teszi bonyolult struktúrájú művek egyszerű módon való megírását. A szerzőnek csak saját műve *lejegyzésével* kell törődnie, s a létrejött eredmény nyomdai minőségű lesz.

Elkészült rendszerének Lamport a LATEX nevet adta, első hivatalos verziójának száma 2.09 lett [11]. A LATEX szó „latech”-nek vagy angolosan „létech”-nek ejtendő. Egyszerű szövegfájlban a márkajel `LaTeX`.

A LATEX könnyen megtanulható, sokat tud, ezért igen gyorsan népszerűvé vált. Ma már a legtöbb nagy kiadó foglalkozik LATEX-hel szedett művek kiadásával. Használata a műszaki és tudományos élet bizonyos területein, a műszaki, tudományos szövegszerkesztésben szabvánnyá vált, ismerete ma már az e területen dolgozók számára nehezen nélkülözhető.

Miután Leslie Lamport visszavonult a további fejlesztéstől, egy 1989-es stanfordi TEX-találkozó után Frank Mittelbach, Chris Rowley és Rainer Schöpf megalkották a LATEX3 munkacsoportot, és belekezdtek a LATEX újraírásába és kiterjesztésébe, egy bővített LATEX, a LATEX3 létrehozásába. Később a csoport munkájába sokan bekapcsolódtak, így Johannes Braams, David Carlisle, Michael Downes, Denis Duchier, Alan Jeffrey. A kitűzött cél többek között az volt, hogy az új változat a dokumentum-típusok sokkal szélesebb körét támogassa, egyszerű parancsokkal segítse a tipográfusi és a szerkesztői munkát mind a kinézeti terv elkészítésé-

[1] A TEX verziószáma π-hez konvergál, a METAFONT-é e-hez, az utóbbi verziószáma jelenleg $2{,}7182$.

ben, mind a végső finom igazítások elvégzésében, támogassa az SGML-szabványnak megfelelő dokumentumok lefordíthatóságát[2], kereteket nyújtson az addigra létrejött különféle LaTeX-változatok és az $\mathcal{AMS}$-TeX matematikai tudásának beolvasztására, támogassa a nemzeti nyelvek használatát, és persze változatlanul értse a LaTeX 2.09-ben írt dokumentumokat is.

A LaTeX3-csapat 1994-ben kibocsátotta a LaTeX új verzióját, a LaTeX 2_ε-t, mely a LaTeX3 előzetes verziójának tekinthető [11, 6, 5]. Azóta a 3-as verzió irányába mutató változtatások beépülnek a LaTeX 2_ε-be. 2018-ban UTF-8 vált a szövegfájlok alapértelmezett kódolásává.

A LaTeX 2_ε a régi, LaTeX 2.09-ben írt dokumentumokat is felismeri[3], és automatikusan az ún. *compatibility mode*-ra váltva le is tudja fordítani. Ennek ellenére ma már nem érdemes új dokumentumot LaTeX 2.09-ben írni.

LuaLaTeX • A kék könyv megjelenése óta a `dvi` mint tördelt formátum nagyon erősen visszaszorult, és a ma használatos rendszerekben a tördelőprogram közvetlenül `pdf`-fájlt állít elő. Ilyen tördelőmotor manapság három is van: pdfTeX, XeTeX és LuaTeX. Ezek közül az első 8 bites bemeneti kódolás esetén optimális, a másik kettő natívan UTF-8 alapú. A XeTeX-et kifejezetten bölcsészeti alkalmazás céljából hozták létre a SIL-ben, hogy a világ különböző nyelvein lehetővé tegye bibliafordítások szerkesztését (`http://tug.org/¬ interviews/kew.html`). A LuaTeX alkotói általánosabb célokat tűztek ki maguk elé, elsősorban a forráskód áttekinthetőbbé tétele, a moduláris fölépítés és bővíthetőség, és a könnyebb programozhatóság érdekében. Bölcsészek számára mindkét unicode alapú motor jól használható és stabilan működik. Az utóbbi mellett két érv szól: 1) az elválasztási szabályok egy csoportjának (ssz → sz-sz) kezelésében LuaTeX egy fokkal rugalmasabb (bár még mindig nem optimális), és 2) lua nyelven való programozhatósága és fejlesztői háttere miatt egyértelműen látható, hogy a jövőben növekvő elterjedésére lehet számítani.

Magyarítás • A kék könyv megírásával egyidejűleg Szabó Péter rendkívül sokoldalú és szofisztikált magyarító csomagot készített, melynek használatát a könyv 9.3 szakasza ismerteti. Ez a csomag a babel rendszerre támaszkodik. A XeLaTeX és a LuaLaTeX motorhoz új, ezek lehetőségeit kihasználó lokalizáló csomagok születtek `polyglossia` néven. Ennek keretében e sorok írásakor még csak rudimentális magyarítás áll rendelkezésre, azonban biztosak vagyunk benne, hogy a magyar.ldf csomag fontos képességeinek adaptálására (portolására) és a hivatalos disztribícióba való beépítésére belátható időn belül sor fog kerülni.

Ebben a jegyzetben tehát a LuaLaTeX + polyglossia kombináció használatát javasoljuk és ezt tekintjük alapértelmezettnek.

1.2. Szerző, tipográfus, szedő, szerkesztő

Egy könyv megírásának legfontosabb fázisai már évszázadok óta a következők: a szerző, miután megírta művét, a kéziratot (ellenőrzés, lektorálás után) átadja a kiadónak. Ott egy tipográfus a könyv tartalmát is figyelembe véve minden apró részletre kiterjedően megter-

[2] Az SGML – Standard Generalized Markup Language – a dokumentumleírás nemzetközi szabványos nyelve, száma ISO-8879.

[3] Egy LaTeX 2.09-ben készült mű onnan ismerhető meg, hogy a forráskód első sora a `\documentstyle` paranccsal, míg egy LaTeX 2_ε-ben írt állomány kódja a `\documentclass` paranccsal kezdődik.

vezi a könyvet, azaz elkészíti a könyv oldalainak kinézeti tervét, megtervezi a borítóját stb. Ezután a kézirat eljut a szedőhöz, aki a kinézeti terv alapján kiszedi a művet. Végül a kiszedett mű a nyomdába kerül, ahol kinyomtatják és bekötik. Az egész folyamatot a szerkesztő irányítja.

A számítógépek megjelenése egyrészt megkönnyítette e folyamat több lépését, másrészt azonban kicsit össze is kuszálta a szálakat. Ma már a legtöbb szerző számítógéppel írja művét, számítógéppel végzik a szedést, és a nyomdában is számítógép vezérelte gépek nyomtatnak. A kiváló számítógépprogramok révén a szerző maga is elő tudja állítani a nyomdakész (*camera ready*) eredményt, amit a nyomdában már csak sokszorosítani kell. Ez azt jelenti, hogy a szerző a mű megírásától egészen a szedésig minden munkát kézben tud tartani, el tud végezni, a tipográfusét és a szedőét is. Csakhogy ezekhez a szerző általában nem ért. Így aztán a számítástechnika gyors fejlődése ellenére is sok olyan dokumentum születik (hivatalos levelektől kezdve a cikkeken át egészen a könyvekig), amelyek a tipográfia legelemibb szabályait sem tartják be.

Az okok egyike az, hogy ugyan a mai nagy tudású dokumentumszerkesztő programokkal nemcsak a szöveget lehet begépelni, hanem a tipográfiai munka egy része is elvégezhető, a programokba beépített tipográfiai tudás pedig könnyen mellőzhető.[4] A rossz dokumentumszerkesztői gyakorlatot gyakran a szerkesztőprogram felhasználói felülete inspirálja, mert azon a szerzői és a tipográfusi parancsok ömlesztve szerepelnek, sőt az utóbbiak vannak előtérbe helyezve.

Az igazi megoldás a vizuális megjelenés megtervezésének és a mű megírásának, azaz a tipográfusi és a szerzői munkának világos szétválasztása. Például a tipográfus dolga a cím betűméretének, betűtípusának, a körülötte hagyandó térköz nagyságának meghatározása, míg a szerző dolga csak annyi, hogy megmondja, mi a cím. A LaTeX koncepciójának alapját épp ez az elv képezi: a LaTeX úgynevezett stílusállományaiba beépített tipográfiai tudásnak kell a mű megjelenéséről gondoskodnia. A szerző csak igen ritkán állítgatja a betűméretet, betűtípust, nem méreget térközöket, sorkihagyásokat stb., csak közli a LaTeX-hel a szövegrészek funkcióit: „ez itt a cím”, „ezt a szót ki kell emelni”. Az, hogy mindez vizuálisan hogy jelenik meg, a LaTeX dolga. Tehát a szerző a tipográfusi munka nagy részét a LaTeX-re bízza, amely a szedői munkát a TeX-hel végezteti el. A szerző LaTeX-hel írva mentesül(het) a dokumentum vizuális megjelenésének tervezése, szerkesztése alól, de nyitva áll előtte a lehetőség, hogy szükség esetén a dokumentum végső alakjának legapróbb részleteit is befolyásolja.

1.3. A LaTeX jelene és jövője

TeX-társaságok, TeX Live • A TeX felhasználói és fejlesztői több országban társaságokat hoztak létre. A központi TeX-társaság az amerikai székhelyű TUG (TeX Users Group). Honlapjának címe `http://www.tug.org`. E társaság szerteágazó tevékenységének egyike, hogy minden évben kibocsátja a TeX Live disztribúciót. A TeX Live anyaga a `http://¬ www.tug.org/texlive/` címről ingyen letölthető és másolható.

[4] E programok felhasználóinak gyakori típusa, miután írt néhány sort, tipográfusi munkába kezd. Például veszi a címnek szánt szöveget, betűit megvastagítja, betűméretét megnöveli, beszúr egy-két üres sort, hunyorít, értékel, majd megismétli a fenti lépéseket. A felhasználók másik típusát a különböző fontok túlburjánzó használata, a *„fontitis”* jellemzi. Ez nemcsak a kezdő honlaptervezők betegsége.

LaTeX az Interneten • A TeX-hel és így a LaTeX-hel kapcsolatos anyagok gyűjtőhelye a *Comprehensive TeX Archive Network* (magyarul Átfogó TeX Archívum Hálózat, rövidítve CTAN). Legfontosabb címe: `http://www.ctan.org`, de számtalan más címen is elérhető. A TeX-anyagok leggyakrabban a `tex-archive`, esetleg a `pub/tex` könyvtárban vannak. Az archívum számunkra legfontosabb címei:

```
http://www.ctan.org/tex-archive
ftp://ftp.ctan.org/tex-archive
ftp://ftp.sztaki.hu/pub/tex
http://dante.ctan.org/CTAN
```

Amikor a továbbiakban egy CTAN-on megtalálható állomány vagy könyvtár helyét megadjuk, nem fogjuk leírni az egész címet, hanem a `CTAN` szót a fenti négy, vagy azokkal ekvivalens valamely cím helyettesítésére fogjuk használni. Például a

```
http://www.ctan.org/tex-archive/help/Catalogue/
```

cím helyett csak annyit fogunk írni, hogy `CTAN/help/Catalogue`.

Ezeken a szervereken több gigabyte-nyi anyag van felhalmozva. Ha dokumentációt, program(csomag)ot keresünk, használjuk a `http://www.ctan.org/` fő oldalán felkínált keresőt (*Search*).

A LaTeX honlapjának címe:

```
http://www.latex-project.org/
```

A LaTeX babel rendszerű magyarításának honlapja sok egyéb hasznos információval:

```
http://www.math.bme.hu/latex/
```

A LaTeX kapcsolata más programokkal • A LaTeX-re, illetve LaTeX-ről más nyelvre vagy formátumra konvertálni sokféleképpen lehet.

Fontos megemlíteni az MS Wordben írt fájlok LaTeX-re való konvertálhatóságát. Ezt például a Word `rtf`-formátumán keresztül lehet megvalósítani. A fájlt az MS Wordből nem `doc`, hanem `rtf` formátumban kell menteni, amit azután az rtf2latex2e programmal lehet LaTeX formátumúvá konvertálni.

Több olyan irodai program is létezik, amelynek *export filter*ei közt a LaTeX is szerepel, ami azt jelenti, hogy a dokumentum LaTeX formátumban is elmenthető. A libreoffice *export filter*ként és külön konverterként is használható programja a Writer2LaTeX.

A LaTeX és más formátumú szöveges állományok közti konverzió lehetőségei a `http¬ ://www.tug.org/utilities/texconv/` oldalon vannak összefoglalva.

A LaTeX előnyei • Ahhoz, hogy a LaTeX-et az Olvasó elhelyezhesse a dokumentumkészítő (szövegszerkesztő, dokumentumszerkesztő, kiadványszerkesztő, szedő) programok között, felsoroljuk néhány jellegzetességét. A LaTeX

- képes nyomdai minőségű dokumentum előállítására;
- nyelve egyszerű, bármely szövegszerkesztővel szerkeszthető;
- sok szövegszerkesztő támogatja, többük grafikus felhasználói felülettel rendelkezik (WinEdt,

TeXnicCenter, kile), van köztük WYSIWYG-szerű[5] program is (Textures, LyX, SciWord).

- nyelvének alapelemei a dokumentum logikai struktúrájának leírását szolgálják, ami mentesíti a szerzőt a vizuális szerkesztés gondjától, ugyanakkor – ellentétben a HTML-típusú nyelvekkel – a vizuális megjelenés teljesen szabályozható;
- segítségével a dokumentum olyan összetett részei, mint például az irodalomjegyzék, a tartalomjegyzék, a lábjegyzet, automatikusan készíthetők;
- nyelvével a matematikai formulák könnyen írhatók le, ugyanakkor ezeket a legmagasabb tipográfiai szinten jeleníti meg;
- maga is ingyenes, és ingyenesek a legkülönfélébb tipográfiai feladatok megoldására használható kiegészítő programcsomagok is;
- forráskódja nyílt, korlátozások csak az üzleti célú felhasználásban és a márkanevek használatában vannak (bővebben lásd: LaTeX Project Public License – `http://www.latex-project.org/lppl.txt`);
- rendkívül megbízható, szinte minden számítógéptípus minden fontosabb operációs rendszerén fut, mégis a legmagasabb hordozhatóságot biztosítja;
- kódja más programmal könnyen generálható;
- a vele írt dokumentum változatkezelővel (pl. CVS) tárolható;
- sok nemzeti nyelvet támogat, képes többnyelvű dokumentumok előállítására.

A LaTeX hátrányai • A LaTeX-nek megemlíthetők bizonyos hátrányai is:

- bizonyos feladatok megoldása körülményesebb, nehezebben áttekinthető, mint a szokásos WYSIWYG rendszerekben, azaz pl. képes magazinok tördelésére nem feltétlenül optimális;
- egy dokumentum kinézeti tervének megváltoztatása vagy új terv készítése általában nehéz munka, nemcsak a LaTeX, hanem néha a TeX mély ismeretét is igényli;
- a szükséges ismeretek megszerzése az ingyen rendelkezésre álló dokumentációk ellenére, néha azok nagy száma miatt is nehéz;
- a hibaüzenetek megértése, a hibák kijavítása időnként komoly feladatot jelent.

1.4. A jegyzet jelölései és tagolása

Konvenciók • Először a könyvben használt konvenciókat ismertetjük.

`Írógép típusú betűk:` Mindazt, amit a számítógép előtt ülve be kell gépelni, továbbá az egyszerű programlistákat *írógép típusú* betűkkel szedjük. Használatukat egy példán szemléltetjük: ahhoz, hogy egy szöveg, például ez itt be legyen keretezve, azt kellett begépelnünk, hogy `\framebox{ez}`.

Groteszk betűk: A programok, programcsomagok neveit úgynevezett groteszk betűkkel szed-

[5] WYSIWYG, azaz What-You-See-Is-What-You-Get, magyarul „amit látsz, azt kapod”. E betűszót azokra a programokra használják, amelyek már szerkesztés közben is a dokumentum végső alakját mutatják. A mai WYSIWYG-programok rajongóiétól különböző számítástechnikai kultúrán érlelt nézet szerint e programokra jobban illene a What-You-See-Is-All-You-Get kifejezés: „csak amit látsz, azt kapod”. A LaTeX-re építő, és nem pontosan a végső eredményt mutató programokra (pl. LyX), inkább a What-You-See-Is-What-You-Mean kifejezés illik: „amire gondolsz, azt látod”.

jük. *Groteszk betű* alatt az egyforma vastagságú vonalakból álló, talpatlan betűt értjük, mely a könyvünkben így néz ki.

Dőlt betűk: A LaTeX-parancsok leírásakor gyakori, hogy egy paraméternek csak a típusát adjuk meg, majd részletesen leírjuk, hogy mit lehet helyettesíteni a helyébe. Ezt szemléltetjük a `\framebox` paranccsal: a *`\framebox[sz][pozíció]{szöveg}`* parancs egy *sz* szélességű keretet tesz a *szöveg* köré, és abban a szöveget balra, középre vagy jobbra zárja aszerint, hogy a *pozíció* helyén `l`, `c` vagy `r` betű áll. Például a `\framebox[18mm][r]{ez}` parancsok hatására egy 18 mm széles keretet kapunk, melyben a szöveg jobbra lesz igazítva: ez.

¬: A TeX parancsszavait *nem lehet elválasztani* a forráskódban a sor végén. A `\cleardoublepage` parancs viszont itt nagyon túlfutna a margón, ezért bevezettünk egy mással össze nem téveszthető, a billentyűzetről be nem vihető elválasztójelet parancsszavaknak csak a könyv szövegében való elválasztására, ez a ¬ jel.

A látható szóköz: Ha a forráskódban fontos, hogy egy adott helyen szóköz legyen, ezt a *látható szóköz* jellel jelezzük: ␣. Például ahhoz, hogy a „TeX vagy LaTeX” szöveget megkapjuk, azt kell begépelni, hogy `\TeX\␣vagy␣\LaTeX`.

Programlisták: Sokat segíthet az Olvasónak, hogy a jegyzet példát ad a bemutatott parancsok vagy jelenségek megvilágítására. Ha lehet, a könyvben látszik a példa forráskódja, és mellette jobb oldalon a futtatásának eredménye. A példa kódjának sorai sorszámozva vannak, a sorszámok azonban nem részei a kódnak! Ezt azzal jelezzük, hogy a sorszámokat jóval kisebb karakterekkel szedjük. Tekintsük az alábbi példát:

```
01 Ez egy \emph{egyszerű} példa:
02 $(\sqrt{5}-1)^2=6-2\sqrt5$.
```

Ez egy *egyszerű* példa: $(\sqrt{5}-1)^2=6-2\sqrt{5}$.

`%^` : Egy programrészlet futási eredményét befolyásolhatják a kód elején, a bevezetőjében (preambulumában) kiadott parancsok. Ezeket a kódban úgy fogjuk jelezni, hogy az ilyen sorok elejére a `%^` karaktereket írjuk. A `%` azt jelenti, hogy e sor itt csak megjegyzéssor, a `^` arra utal, hogy a kódsort a program elejére, a preambulumba kell írni. Az `\euro` parancs például csak akkor írja ki az € jelet, ha a preambulumba kerül a `\usepackage{eurosym}` parancs. Ezt így fogjuk kifejezni:

```
01 %^\usepackage{eurosym}
02 A leves csak 1\,\euro\ volt.
```

A leves csak 1 € volt.

∈ *(eleme jel):* A LaTeX olyan nyílt rendszer, amelyhez bárki illeszthet programcsomagokat. Ezek közül több fontosabb programcsomagot részletesen is ismertetünk. Egy ilyen programcsomagot csak úgy használhatunk, ha a `\usepackage` paranccsal betöltjük. Hogy a zavart elkerüljük, a programcsomagok parancsait úgy különböztetjük meg a LaTeX egyszerű parancsaitól, hogy a keretben a sor végére, egy ∈ jel után megadjuk, hogy a parancs melyik programcsomaghoz tartozik. Például az `\ifthenelse` parancs csak az ifthen programcsomag betöltése, azaz a `\usepackage{ifthen}` parancs kiadása után működik. Ezt így jelezzük:

`\ifthenelse``{`*feltétel*`}{`*akkor*`}{`*egyébként*`}` ∈ ifthen

Előfordulhat, hogy egy parancs a LaTeX parancsai között is szerepel, és valamelyik programcsomagban is. Ilyenkor a programcsomag az eredeti parancs képességeit továbbiakkal bővíti. Ennek jelölésére az ∈ jel után nemcsak a programcsomag nevét, hanem a LaTeX

szót is megadjuk.

`\begin{`*tabular*`}[`*poz*`]{`*oszlopok*`}` ∈ LaTeX, array

A táblázatokat a `\begin{tabular}` paranccsal kell kezdeni. Ha azonban betöltjük az array csomagot, kényelmesebben tudjuk használni.

2. FEJEZET

Telepítés, használat

2.1. Telepítés

Mielőtt a LaTeX ismertetésébe kezdenénk, áttekintjük a használandó programokat és az azok telepítésével kapcsolatos legfontosabb információkat. Szükségünk lesz szövegszerkesztőre és olyan megjelenítő programokra, melyekkel a PDF formátumban előállított dokumentumokat megtekinthetjük. Végül szükségünk lesz egy TeX rendszerre. Csak a két legelterjedtebb operációs rendszert (Windows, Linux) tekintjük át, a többiről tájékoztatást kapunk a TeX FAQ-ból [4].

Ahhoz, hogy értsük, mit miért telepítünk, vázlatosan összefoglaljuk, és a fenti sematikus ábrán szemléltetjük a LaTeX működését. Először létre kell hozni egy egyszerű szöveges állományt. Leegyszerűsítve, *szöveges állományon* olyan fájlt értünk, amiben nincs más, csak azok a karakterek, amiket begépeltünk. Adjunk a fájlnak `tex` kiterjesztésű nevet, most legyen ez `file.tex`. Amit mi begépelünk, az valójában egy számítógépprogram. Ez a program azt is tartalmazza, hogy mi a végső dokumentum szövege, de azt is leírja, hogy pontosan hogy nézzen az ki. A LaTeX fordítóprogramja (neve `lualatex`) érti ezt a programot, és elkészíti a tördelt pdf-dokumentumot. Ennek a fájlnak a neve megegyezik a forrásállományéval, csak a kiterjesztése `tex` helyett `pdf` lesz, esetünkben `file.pdf`. Ezt azután megtekinthetjük a képernyőn, és kinyomtathatjuk.

A TeX telepítése • TeX telepítésére Windows operációs rendszer alá a MiKTeX disztribúciót javasoljuk. A `https://miktex.org/` honlapon minden információ megtalálható.

Linux esetében vagy a illető disztribúció saját TeX-csomagját lehet telepíteni, vagy pedig a texlive disztribúció linuxos változatát. Utóbbi általában frissebb, mivel az egyes linux disztribúciók szerzői is ebből dolgoznak. A `https://www.tug.org/texlive/` honlapon minden információ megtalálható.

Szövegszerkesztő • Szükséges még egy megfelelő szövegszerkesztő telepítése. Egyszerű UTF-8 szöveges állományok szerkesztésére szolgáló program (szövegszerkesztő – *text editor*) minden operációs rendszerben rendelkezésre áll, de az ablakkezelő operációs rendszerek visszaszorították használatukat, pl. Windows alatt alaphelyzetben csak egyetlen, igen kis tudású, komolyabb munkára nem alkalmas program fut (Jegyzettömb – NotePad). A Microsoft Word / LibreOffice vagy hasonló dokumentumszerkesztő programok nem e célra valók, azonban kellő óvatossággal TeX-dokumentumok szerkesztésére is használhatóak, a dokumentumot mindig UTF-8 kódolású sima szövegként kell elmenteni.

Windows-felhasználónak a MiKTeX telepítését javasoljuk, ez tartalmaz szövegszerkesztőt is.

Linux-felhasználóknak az Emacs a kézenfekvő javaslat.

Megjelenítő programok telepítése • A pdf-fájlok megjelenítésére és nyomtatására Windows alatt az Acrobat Readert szokás használni, linux esetében az okular és az xpdf a leg-

kedveltebbek.

2.2. Az első mű

2.2.1. Az első mű megírása

Egy egyszerű LaTeX-dokumentumot könnyen létre tudunk hozni. Némi túlzással az írógépelésnél csak annyival kell többet tennünk, hogy a begépelendő szöveg elé írunk néhány sort, és még egyet utána. Hogyan lehet például LaTeX-hel azt a szöveget leírni, hogy „Nem is olyan bonyolult!”. Baloldalt látjuk annak a szövegállománynak a tartalmát, amit be kell gépelnünk, és jobboldalt azt, amit majd kinyomtatás után kapunk. Tájékoztatásul megrajzoltuk az eredmény bal szélének sarkait, e jelek természetesen nem tartoznak a végeredményhez. Ismételten hangsúlyozzuk, hogy a kód előtti sorszámok csak a tájékozódást segítik, nem tartoznak a kódhoz, nem szabad őket begépelni.

```
01 \documentclass{article}
02 \begin{document}
03   Nem is olyan bonyolult!
04 \end{document}
```

Nem is olyan bonyolult!

A dokumentumosztály kiválasztása • A szövegállomány a `\documentclass` paranccsal kezdődik. Utána kapcsos zárójelek közé azt kell beírnunk, hogy milyen dokumentumosztályba tartozik művünk. A LaTeX öt *standard* dokumentumosztályt ismer.

`article`: A kisebb dokumentumokat, feljegyzéseket, üzeneteket, a néhány oldalas cikkeket, tanulmányokat, a rövidebb írásokat mind ebbe az osztályba soroljuk. Az esetek nagy részében tehát a `\documentclass{article}` kezdő parancsot használjuk.

`book`: A könyvek a `book` osztályba kerülnek. Ekkor a szövegállomány első sorába azt írjuk, hogy `\documentclass{book}`.

`report`: Ez a beszámolók, hosszabb tanulmányok készítéséhez használatos dokumentumosztály.

`slides`: Fóliaíráshoz való osztály.

`letter`: Ezt használhatjuk levelek írásához.

Ezen az öt alaposztályon kívül még számtalan osztály, többek között a fentiek továbbfejlesztett változatai érhetőek el.

A preambulum és a dokumentum teste • Első művünk második sorában szerepel a `\begin{document}`, az utolsóban az `\end{document}` parancs. Dokumentumunk teljes szövege e két parancs közé kerül. Ezt a részt nevezzük a *dokumentum testé*nek.

A `\documentclass` és a `\begin{document}` sorok közötti részt, ahová olyan parancsok kerülnek, melyek az egész dokumentumra vonatkoznak, a dokumentum *preambulum*ának nevezzük. Egy LaTeX-dokumentum forrásállománya tehát mindig a következő szerkezetű:

`\documentclass{`*a dokumentumosztály neve*`}`

preambulum
```
\begin{document}
```
a dokumentum teste
```
\end{document}
```

Csomagok • A LaTeX képességei úgynevezett *programcsomag*okkal, röviden csomagokkal (angolul *package*-ekkel) bővíthetők. Egy csomagot a preambulumba írandó `\usepackage` paranccsal töltünk be, melynek argumentumába írjuk a csomag nevét (pl. fontspec), azaz a forrásállományba mindössze egy sort kell beszúrni:

```
\documentclass{article}
\usepackage{fontspec}
\begin{document}
  Nem is olyan bonyolult!
\end{document}
```

2.2.2. Fordítás, megtekintés, nyomtatás

Elkészítettük első művünket, és azt egy `tex` kiterjesztésű állományba mentettük. A példa kedvéért a neve legyen `elsomu.tex`.

Ezután következik a *fordítás*. Miktex esetében a bal felső menű lualatex pontját kell választani, linux esetében pedig akár parancssorból is kiadhatjuk a `> lualatex elsomu` utasítást.

A megtekintéshez és a nyomtatáshoz az elsomu.pdf fájl az Acrobat Reader illetve okular programmal nyitandó meg.

2.2.3. Az első nehézségek

Mivel a (La)TeX dokumentumleíró programnyelv, a programban hibát is követhet el a program szerzője. Erről fordítás közben hibaüzenetet kapunk. A hibaüzenetek tanulmányozására tekintsük az alábbi rövid LaTeX-állományt:

```
\documentclass{article}
\begin{document}

Ezzel baj lesz: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx.

\begin{yyy}
  Ez mi?
\end{yyy}

Itt egy ismeretlen \parancs van.

\end{document}
```

A hibaüzenet lehet csak egy figyelmeztetés, amely olyan hibát jelez, ami nem akadályozza

a TEX-et a továbbfordításban. Ilyen hibaüzenetet kapunk pl. ha a LATEX nem tud megfelelő helyen eltörni egy sort, ezért a sor utolsó szava valamennyire kilóg a szedéstükörből. A fenti programot lefordítva a következő üzenetet kapjuk a képernyőre (több egyéb üzenet után):

```
Overfull \hbox (0.47346pt too wide) in paragraph at lines 4--5
[]\OT1/cmr/m/n/10 Ezzel baj lesz: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxx.
```

Ez azt jelenti, hogy a forrásszövegben a negyedik sortól az ötödikig tartó bekezdés egyik (jelen esetben az egyetlen) sora 0.47346 ponttal túltöltötte (*overfull*) a sort, azaz a sor ennyivel túlcsordult (*overflow*) a megengedett sorszélességen. A LATEX azonban itt nem áll meg, tovább fordít.

A LATEX a forráskód hatodik sorában olyan hibát talál, amivel nem tud mit kezdeni, a `\begin` parancs után olyan szó szerepel, amit nem ismer. Hamarosan megtanuljuk, hogy ide egy létező LATEX-környezet nevét kell írni, de az `yyy` nem az. A hiba súlyossága miatt a LATEX megáll, egy „`! LaTeX Error:`” szöveggel kezdődő üzenetet ad, majd az üzenet végén egy „?” jelenik meg:

```
! LaTeX Error: Environment yyy undefined.

See the LaTeX manual or LaTeX Companion for explanation.
Type  H <return>  for immediate help.
 ...

l.6 \begin{yyy}

?
```

A „?” kiírása után egy egybetűs parancsra vár, hogy mit tegyen. A következő lehetőségeink vannak:

Return folytassa a feldolgozást;
S (*scroll*) írja ki a további hibaüzeneteket, de ne álljon meg;
R (*run*) fusson tovább a fordítás;
Q (*quietly*) hibaüzenetek képernyőre írása nélkül fusson tovább;
I (*insert*) az „i” betű után írt szöveget illessze be a hiba helyére;
E (*edit*) szerkessze a forrásállományt;
1 ...9 a következő 1 vagy...vagy 9 tokent (l. a(z) **??**. oldalt) ugorja át;
H (*help*) írjon ki egy rövid tájékoztatót a hiba mibenlétéről;
X (*exit*) álljon le és lépjen ki;
? az itt szereplő betűparancsokat sorolja fel.

A fenti állománnyal próbáljuk végig mindegyik lehetőséget! Olyan hibaüzenetet is kaphatunk, mely nem a LATEX, hanem az alatta futó TEX hibaüzenete. Ezek is !-jellel kezdődnek, de a `LaTeX Error` üzenet hiányzik. Ilyen üzenetet kapunk a fenti példa lefordítása végén is:

```
! Undefined control sequence.
l.10 Itt egy ismeretlen \parancs
                                  van.
```

```
?
```

Az üzenet azt jelenti, hogy a `\parancs` ismeretlen parancs. Itt jól látszik az is, hogy a (LA)TEX a hibaüzenetbe beírja azt a szöveget, ahol a hibát találta, és a hiba után megtöri ezt a sort.

Néha nehéz megfejteni a hibaüzenetet, különösképp akkor, ha az nem az általunk írt szövegben elkövetett szintaktikai hiba miatt történik.

2.3. A felhasználható karakterkészlet

2.3.1. Alapkészlet

A 10 speciális jel • A forrásállomány begépelésekor alaphelyzetben csak az úgynevezett ASCII karaktereket használhatjuk (l. a szótárat a(z) **??**. oldalon). Ide tartoznak az angol ábécé kis- és nagybetűi, a számok, a szóköz, a sorvége jel, a tabulátor vagy röviden tab karakter, valamint az alábbi írásjelek és speciális karakterek:

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

Ezek a jelek úgy fognak a kész dokumentumban megjelenni, ahogy begépeltük őket. Matematikai formulákban és bizonyos később részletezendő körülmények között[1] ugyanígy használható három további jel:

```
< > |
```

Alapértelmezésben e három jel helyett a (LA)TEX a ¡, ¿, — karaktereket írja ki, ezért megjelenítésükre a `\textless`, `\textgreater`, `\textbar` parancsokat használjuk.

A fent felsorolt karaktereken kívül van még 10, a TEX számára speciális nyomtatható ASCII karakter:

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

Mivel ezeknek (LA)TEX forrásállományokban speciális jelentésük van, megjelenítésük csak parancs segítségével történhet. E tíz jelből hét, úgynevezett *escape* (ejtsd: eszkép) paranccsal – azaz a \ eléírásával – jeleníthető meg. Először a jelet, utána a kiíratót parancsát adjuk meg:

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

A maradék három jel másként jeleníthető meg, mivel a `\\`, `\^`, `\~` parancsoknak más jelentésük van. Több megoldást is adunk:

```
\   $\backslash$   \   \textbackslash    \   \char\string`\\
^   \^{}           ^   \textasciicircum  ^   \char\string`\^
~   \~{}           ~   \textasciitilde   ∼   $\sim$
```

A `\char\string`\<` konstrukció közvetlenül azt a karaktert kéri a betűtípusból, melynek kódja < kódjával egyezik meg. A fenti táblázatban a \ és a ^ karakterre használtuk ezt a megoldást.

[1] Pl. a T1 kódolás használata esetén – lásd a(z) **??**. oldalon.

U Gyakorlat: *A (LA)TEX speciális karaktereinek megjelenítése*
Írjuk le LATEX-ben az alábbi szöveget: 10$-t visszaadott, mert a W&C boltjaiban 50%-kal kevesebbet kellett fizetnie!

A \, {, }, %, $ jelek különösen fontosak, mondhatjuk, hogy ezek a LATEX nyelvének alapjelei.

A \ jel (rep jel): parancskezdő karakter • A \ jelentését az eddigiek alapján is sejthetjük: a LATEX parancsai ezzel a jellel kezdődnek. A ferde törtvonalat *per*nek olvassuk, innen az ötlet, a fordított törtvonalat – pl. egy LATEX-parancs kiolvasásakor – *rep*nek mondjuk.

A LATEX-parancsoknak több típusa van. Az *alfabetikus parancsok* \ jelből és alfabetikus jelekből állnak. E parancsokban megkülönböztetjük a kis- és nagybetűket, tehát `\ibolya` és `\Ibolya` két különböző parancsnév. Az alfabetikus parancsok az első nem-alfabetikus jelig tartanak, pl. egy szám, egy kapcsos zárójel, vagy egy szóköz a parancsszó végét jelzi. A következő három példa mindegyikében `\H` a parancs: `\H{o}`, `\H␣o`, `\H\i`. Nem lehetnek parancsszavak az alábbiak: `\alma2korte`, `\alma␣korte`, `\alma_korte`, `\alma2` (A ␣ jel a begépelt szóközt jelöli).

A *kétjeles parancsok* két jelből, a \ jelből és egy nem-alfabetikus jelből állnak. Pl. a `\'` parancs egy vesszőt tesz az utána következő betűre (`\'o` eredménye ó), míg pl. a `\\` (ejtsd rep-rep) paranccsal kényszeríthetjük a LATEX-et arra, hogy valahol új sort kezdjen. A `\H` parancs alfabetikus, nem számít kétjelesnek. A kétjeles parancsok a második jellel véget érnek, utánuk bármilyen karakter következhet.

A parancsok némelyikéhez hozzá lehet írni egy *-ot. Ezek a *csillagos parancsok*, melyek a csillag nélküli parancs működését kicsit módosítják. Például a `\\` új sort kezd, a `*\\*` ugyancsak, de nem engedi, hogy ott a LATEX új oldalt kezdjen. A `\chapter` új fejezetet kezd egy könyvben, a `\chapter*` ugyancsak, de nem ad a fejezetnek sorszámot.

A % jel: megjegyzések • A % jellel *megjegyzések* iktathatók a forrásállományba. Mindaz, ami egy sorban a % jel és a sor vége között van, csak megjegyzés, beleértve még a sorvége jelet is. Így, ha egy szó végét szóköz kihagyása nélkül követi a % jel, akkor a szó összeolvad a következő sor első szavával. Erre normál szöveg beírásakor szinte soha sincs szükség, de programok írásakor igen. Az alábbi példában a „homo” szó után szóköz kihagyása nélkül következik a %, így az egybeíródik a következő sor első szavával. (Az egyszerűség kedvéért további példáink nagy részének forrásszövegéből kihagyjuk a `\documentclass{article}`, a `\begin{document}` és az `\end{document}` sorokat.)

```
% Az ember tragédiája hetedik szín
A homo%i
usiont hirdette.
```

A homousiont hirdette.

A { és a } jel: blokkok létrehozása • A *kapcsos zárójelek* olyan *blokkok*at hoznak létre a forrásállományon belül, melyeket a LATEX egyetlen egységként tud kezelni. Lássunk egy példát: a szövegbe tetszőleges szélességű és magasságú vonalat húzhatunk a `\rule` paranccsal. E parancsnak két argumentuma van, az első a vonal szélességét, a második a magasságát határozza meg. A `\rule{5mm}{0.25mm}` parancs egy 5 mm széles, 0,25 mm magas vonalat rajzol ki: ▬. Megjegyezzük, hogy itt `0.25mm` helyett `0,25mm` is írható a forrásállományban, mivel itt a (LA)TEX számára egyértelmű a vessző jelentése.

A LaTeX által egységként kezelt blokkokat más módon is kijelölhetjük. A parancsok egy részének például vannak ún. *opcionális*, elhagyható argumentumai, ezeket nem kapcsos, hanem szögletes zárójelek közé kell tenni. Például a `\rule` parancsban rögtön a parancsszó után megadható, hogy a megrajzolt vonal mennyivel kerüljön az alapvonal fölé. Például az előző vonalat 1 mm-rel megemeli a `\rule[1mm]{5mm}{0,25mm}` parancs: ——.

Blokkot jelöl ki az úgynevezett *környezet* is, mely kicsit leegyszerűsítve egy névvel ellátott blokknak tekinthető. A környezet elejét egy `\begin{`*név*`}`, végét egy `\end{`*név*`}` parancs jelöli ki a kapcsos zárójelek helyett. Ezekről a **??**. alszakaszban szólunk részletesebben.

A blokkok tartalmazhatják egymást, de egymásból nem lóghatnak ki. Ez azt jelenti, hogy blokkok így nem kapcsolódhatnak:

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

A karakterek és a parancsszavak a LaTeX tovább már nem bontható blokkjai. Például a `\H` parancs[2] két vesszőt (*hosszú kettős* ékezetet) tesz az argumentumában szereplő betűre. Az „ő” betűt a `\H{o}`, `\H␣{o}`, `\H␣o` parancsok mindegyikével megkaphatjuk. A harmadik esetben a `\H` és az `o` betű közé ékelt szóközre szükség van, mert ez mutatja meg, meddig tart a parancs. Szóköz nélkül a LaTeX azt hinné, hogy a parancs `\Ho`, amit nem találna parancsainak listáján, ezért hibaüzenetet adna. Ha kapcsos zárójelet használunk az argumentum körül, nincs szükség a szóközre a parancs és az argumentum között, de tehetünk is, mint azt a `\H{o}` és a `\H␣{o}` alak mutatja.

A kétjeles parancsoknál nem kell a parancs után szóközt tenni, hisz a LaTeX tudja, hol a parancs vége. Például a `\'` eredményül vesszőt (*hosszú* ékezetet) tesz az argumentumában szereplő betűre. A `\'o`, `\'{o}`, `\'␣o`, `\'␣{o}` parancsok eredménye egyaránt egy ó betű.

U Gyakorlat: *Blokk kijelölése*
A `\textsuperscript` parancs az argumentumába írt szöveget a felső kitevőbe teszi. Vizsgáljuk meg az alábbi parancsokat.

```
12\textsuperscript{h}-kor  1\textsuperscript st
12\textsuperscript h-kor   1\textsuperscript {st}
12\textsuperscripth-kor    1\textsuperscript{st}
```

Ezek közül melyik eredményezi a 12^{h}-kor és a 1^{st} kifejezéseket, és melyik hibás?

A $ jel: matematikai képletek • A $ jel hatására a TeX ún. matematikai módba vált, ami azt jelenti, hogy a következő $ jelig mindent képletnek tekint, az így megkapott képletet pedig egyetlen szóként kezeli. A matematikai képletek kezeléséről szól a **??**. fejezet, most csak annyit mutatunk meg, amennyi matematikai képletekből egy köznapi szöveg megírásához elég.

Matematikai módban a szóközt csak a parancsszó végének jelzésére használjuk, egyébként nem befolyásolja a képlet megjelenését. A $+$, $-$, $/$ és $=$ jeleket értelemszerűen használjuk. Szorzójelként használható a `\cdot` és a `\times` parancs: `$a\cdot b$` eredménye $a \cdot b$, `$a\times b$` eredménye pedig $a \times b$. Törtet a kétargumentumos `\frac` paranccsal jeleníthetünk meg, ennek első argumentuma a számláló, a második a nevező, például az `$\frac{1}{1+x}$` parancs ezt adja: $\frac{1}{1+x}$. Kitevő a „^”, alsó index az „_” karakterrel

[2] H, mint Hungarian umlaut (rövidebb nevén hungarumlaut).

képezhető, például `$a^2$` képe a^2, `$f_1$` képe f_1. Ezzel két további speciális karakter jelentését is megismertük. Gyökvonásra az `\sqrt` parancs használható: `$\sqrt{2}$` képe $\sqrt{2}$.

Irodalom

[1]Gyöngyi Bujdosó és Attila Fazekas. *TEX kezdőlépések*. Budapest: Tertia Kiadó, 1997, 237. old.

[2]Michael Doob. *TEX könnyedén (A gentle introduction to TEX)*. Szeged: Polygon, 1995.

[3]Tibor Fadgyas és Dezső Miklós. *MATEX*. Budapest: Akadémiai Kiadó, 1988.

[4]Robin Fairbairns. *The UK TEX FAQ*. 2004. URL: http://www.tex.ac.uk/.

[5]Michael Goossens, Sebastian Rahtz és Frank Mittelbach. *The LATEX Grapics Companion. Illustrating documents with TEX and PostScript*. Reading/Ma. etc.: Addison-Wesley, 1997.

[6]Michel Goossens, Frank Mittelbach és Alexander Samarin. *The LATEX Companion*. Reading/Ma. etc.: Addison-Wesley, 1994.

[7]Donald E. Knuth. *A számítógépprogramozás művészete*. 1–3. köt. Műszaki Könyvkiadó, 1987–1988.

[8]Donald E. Knuth. *Seminumerical Algorithms*. 2. kiad. 2. köt. The Art of Computer Programming. (Ez már TEX-hel készült). Reading, Massachusetts: Addison-Wesley, 1981. okt.

[9]Donald E. Knuth. *The TEXbook*. Reading/Ma. etc.: Addison-Wesley, 1984.

[10]Donald E. Knuth. *The METAFONTbook*. Reading/Ma. etc.: Addison-Wesley, 1984.

[11]Leslie Lamport. *LATEX. A Document Preparation System*. 2. kiad. Reading/Ma. etc.: Addison-Wesley, 1994.

[12]Frank Mittelbach és tsai. *The LATEX Companion*. 2. kiad. Tools and Techniques for Computer Typesetting. Reading/Ma. etc.: Addison-Wesley, 2004. ISBN: 0-201-36299-6. URL: http://www.latex-project.org/.

[13]Michael Spivak. *The Joy of TEX*. 2. kiad. AMS, 1990.

[14]Ferenc Wettl, Gyula Mayer és Csaba Sudár. *LATEX kezdőknek és haladóknak*. Budapest: Panem, 1998, XVII+409. old.

[15]Ferenc Wettl, Gyula Mayer és Péter Szabó. *LATEX kézikönyv*. Budapest: Panem, 2004, XV+768. old.