

Nulladik laborgyakorlat
Adminisztratív és egyéb kérdések

Dokumentumszerkesztés \LaTeX használatával – 3 kredit

Kurzusok:

- LA – \LaTeX – L terem
- WO – M\$ Word – J terem

Mindenki ellenőrizze, hogy a megfelelő kurzusra jelentkezett-e, egyébként nem kaphat jegyet a félév végén!!!

Akinek nem jutott hely, ne csüggedjen, a tárgy minden félévben indul!

Elérhetőség: Kontra Gergely, I.B.113, Tel: 3227, vagy a kgergely@mcl.hu e-mail címen.

Tegezzetek! Én is mindenkit letegezek könnyörtelen módon!

Ne sunyuljatok! Ne írjatok fel mást magatok helyett (kari fegyelmi ügy)!

Az órák 12:15-kor kezdődnek és 13:45-kor érnek véget (nincs szünet). A késést öt percig tolerálom (12:20-ig be kell érni). Aki 12:20 után, de 12:30 előtt ér be, az „Késsett” minősítést kap. Aki 12:30 után érkezik, az adminisztratív módon a „Hiányzott” bejegyzést kapja, de az órára beülhet.

A félév során 6 darab kis házi feladatot kell megoldani, amelyeket a következő gyakorlat napja előtti munkanap, reggel 8 óráig kell feltölteni a <http://mcl.hu/~kgergely/hf> címen. Csak az időben leadott házi feladatokat tudom kijavítani. Aki a határidő után, de az óra kezdetéig elküldi a házi feladatát, az „Késsett” minősítést kap, azonban dolgozatát nem ellenőrzöm. Aki az óra kezdetéig nem küldte el az anyagát, az a „Hiányzott” bejegyzést kapja a feladatára.

A bejegyzések online ellenőrizhetők a

<http://mcl.hu/~kgergely/hianyzas.pdf> címen.

Követelményrendszer. Aláírást az kaphat, aki

- Az órák maximum 30 %-áról hiányzik (maximum 4 óra), a többin akár időben, akár késve, de ott van,
- A kisHF-ok legalább 70 %-át beadja (minimum 4 db) időben, vagy késve.

A félév végi jegyet a nagy házi feladatra adom. A feladatot a 10. hét környékén osztom ki, és a 12. hét környékén kell beadni a fenti webcímen. A nagy házi feladat leadásával késni nem lehet! Aki késik, az nem kap jegyet. Ha a jegy kétes, a késések és hiányzások számát figyelembe veszem. Egyéb esetben nincs jelentősége a késéseknek és hiányzásoknak.

Jegyzet. Két elektronikus: <http://mcl.hu/~jenedyg/kinezet.pdf>
– az alapvető tipográfiai szabályszerűségek összefoglalása, és
<http://mcl.hu/~jenedyg/foliak.pdf> – az órák fóliáit tartalmazza.

További nyomtatott irodalmak (az első kettő tipográfiai, a másik kettő $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$):

- Virágvölgyi Péter, *A tipográfia mestersége*, Osiris, 2001.
- Gyurgyák János, *Szerkesztők és szerzők kézikönyve*, Osiris, 1996.
- Leslie Lamport, *L^AT_EX*, 2nd Ed., Addison-Wesley, 1994.
- Wettl, Mayer, Sudár, *L^AT_EX kezdőknek és haladóknak*, Panem, 1998.

Motivációk. A tárgy azért indult, mert a hallgatói dolgozatok (diploma, TDK, önlab beszámoló stb.) minősége meglehetősen alacsony volt. Szükséges beszélni az alapvető tipográfiai szabályokról, mert azokat sehol sem oktatják (középiskolában kellene). Másrészt a szövegtördelő programokat sem tudják használni a hallgatók (tartalomjegyzék kézzel stb.) A tárgy célja, hogy

- betekintést nyújtson azokba az alapvető tipográfiai szabályokba, melyek ismerete elvárható egy diplomás embertől,
- megtanítsa a $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\varepsilon}$, vagy $\text{M}\$$ Word használatát, elsősorban a beszámolók írására fókuszálva,
- a tartalom legyen az elsődleges és ne a forma.

Mi a \LaTeX ? Egy dokumentumtördelő scriptnyelv. Parancsokkal szabályozhatjuk a dokumentumunk megjelenését, az egyes formai elemek (cím, lábjegyzet, képlet) beszúrása is parancsokkal történik. A forrást le kell fordítani és a lefordított fájl jeleníthető meg. Mint amikor C-ben programozunk és az eredményt csak fordítás és futtatás után tudjuk ellenőrizni.

Ki használ \LaTeX -et? Aki tudományos művet (cikket) ír sok képlettel (komoly kutatócsoportok kizárólag ezt használják). Aki könyvet ír. Aki nem akar a formázással bajlódni. Aki nem akar szívni a diploma leadásakor (könnyű az utólagos átformázás, a lényeg a tartalom).

Miért használjak \LaTeX -et? Stabil, megbízható. Portábilis a kimenete és a forrás is (tényleg). Átjárható különböző operációs rendszerek között. Egyszerű. Forrásunk szerkesztéséhez őskori gépet is használhatunk (akár 286-ost). Nem kell törődni a formázással.

A tárgy oktatásának módja. Számítógépes tanórák keretében, heti két órában. A gyakorlati elsajátítást segíti, hogy a tanult ismeretek azonnal kipróbálhatóak a számítógépen.

UNIX alapú oktatás (ural2), mert

- voltak problémák Windows alatt,
- kevesebb csicsa, fapados felület, de nagy megbízhatóság,
- a begépelés kényszere miatt az utasítások jobban rögzülnek,
- a legtöbbször Windows op. rendszere van, így a portabilitás érzékelhető.

A félév elején a \LaTeX alapjaival foglalkozunk, a tartalmi feltöltéshez szükséges utasításokat vesszük át. A formázással kapcsolatos utasítások a félév végére kerülnek. Fontos, hogy

- a tartalommal foglalkozzunk elsősorban és nem a formázással,
- tanuljuk meg, hogy a dokumentum nem attól lesz szép, ha átformázzuk,
- a formázással kapcsolatos kérdésekkel várjunk a félév második feléig,
- a házi feladatokban kerüljük a formázási parancsokat (pl. `\\`).

A \LaTeX elsajátításához javasolt az otthoni telepítés is. Mivel a \LaTeX ingyenes program, szabadon terjeszthető/telepíthető. A \LaTeX teljes működőképességéhez szükséges fájlokat disztribúciókban szállítják. A \LaTeX forrás szerkesztéséhez szövegszerkesztő programok használata javasolt.

Windows rendszerekben a Mik \TeX disztribúciót (<http://www.miktex.org>), szövegszerkesztéshez a \TeX nicCenter (<http://www.toolscenter.org>), vagy WinEdt (<http://www.winedt.com>) alkalmazásokat javaslom (az utóbbi nem ingyenes).

POSIX rendszerek alatt (Mac OS X, Linux, *BSD stb.) a Te \TeX disztribúciót (<http://www.tug.org/teTeX>), szövegszerkesztéshez az Emacs (<http://www.gnu.org/software/emacs/emacs.html>), vagy Nirvana Editor (<http://www.nedit.org>) programokat javaslom.

A \LaTeX -ről dióhéjban. A \LaTeX (ejtsd: lateh) gyakorlatilag egy scriptnyelv. Olyasmi, mint a HTML, csak portábilis. Leslie Lamport írta a \TeX dokumentumformázó nyelv kiegészítéseként, hogy a dokumentumok készítése könnyedén megoldható legyen. Alapelv: a tartalom a lényeg és nem a forma. Ebből következően a komplex formázási feladatok megoldása macerás lehet \LaTeX -ben. Jelenlegi verziója a $\text{\LaTeX} 2_{\epsilon}$, mi is ezt tanuljuk.

\LaTeX vs. \TeX . A félévben csak \LaTeX -hel foglalkozunk. A \TeX szintén tördelőnyelv, de sokkal összetettebb, bonyolultabb. A \TeX -et Donald E. Knuth írta, amikor összeveszett a kiadóval a *Programozás művészete* című könyvének tördelése kapcsán. A saját programjával tördelte művét, amelyet így formázott állapotban tudott leadni.

Donald E. Knuth, *The \TeX book*, Addison-Wesley, 1996.

Grafikus felületek a \LaTeX használatához. Bár az órákon GUI-t nem használunk, de léteznek \LaTeX -hez is. Mégsem az őskorban vagyunk...

Windows alatt a Scientific Word (<http://www.sciword.demon.co.uk>) egy olyan alkalmazás, amely a \LaTeX és a \TeX motorját használja a begépelte információ megjelenítéséhez. Szintén hasznos lehet a Word2TeX alkalmazás (<http://www.sciencesoftware.com/Word2TeX.asp>), amely a Microsoft Wordbe beépülve lehetővé teszi a \LaTeX formátumba mentést. Segítségével a Wordben megírt dokumentumainkat ki tudjuk menteni \LaTeX fájlként, így a megszokott (?) programot használva is előállíthatjuk a \LaTeX forrást.

POSIX rendszerek alatt a LyX (<http://www.lyx.org>) program biztosít grafikus felületet a \LaTeX használatához. Kezelése egyszerű, kimenete saját formátum, de tud \LaTeX -be is menteni. Az Abiword, mint sok platformon futó alkalmazás (<http://www.abisource.org>) szintén tud \LaTeX formátumba menteni.

Az órák menete. X terminált kell indítani, majd bejelentkezni az ural2-re. Az ural2-n egy terminált kell indítani, majd onnan az alkalmazásokat:

\$ nedit filename.tex &

→ a szerkesztéshez, mindig nyitva hagyjuk

\$ xdvi filename.dvi &

→ a megjelenítéshez, mindig újraolvassa a fájlt

\$ latex filename

→ a fordításhoz. Ha változott a forrás, le kell futtatni.

begépelés a billentyűzeten

↓ [szerkesztőprogram (nedit)] ↓

filename.tex

↓ [L^AT_EX (latex)] ↓

filename.dvi

↓ [megjelenítőprogram (xdvi)] ↓

látható kimenet a képernyőn

A dvi (device independent) fájlból lehet tetszőleges outputot készíteni (PDF, PS, fax, nyomtatóformátumok stb.), de erről részletesen majd a félév végén.

Első laborgyakorlat

A \LaTeX használatának alapjai

Mit kell elindítani?

- xterm (Itt fogunk figyelmesen fordítani: latex filename) → a fordítás során megjelenő hibaüzeneteket érdemes elolvasni...
- nedit (Nirvana Editor: a szövegszerkesztéshez), advanced usereknek vi, vagy emacs, vagy ami tetszik.
- xdvi (DVI fájl megjelenítő: itt látjuk majd ami elkészült)
- gv (GhostView): <http://www.mcl.hu/~jeneyg/foliak.ps>, vagy
acroread (Acrobat Reader): <http://www.mcl.hu/~jeneyg/foliak.pdf>

Minden \LaTeX file-ban megtalálhatjuk az alábbi három fontos sort:

```
\documentclass{article}
```

```
\begin{document}
```

```
...Where do you want to go today?...
```

```
\end{document}
```

A forrásfájl felépítését tekintve két fő részből áll:

- Preambulum (a `\begin{document}` előtt)
- Dokumentumtörzs (a `\begin{document}` után)

Ha magyarul akarunk írni (ékezetes betűk), akkor egy külön csomagot be kell töltenünk:

```
\documentclass{article}
\usepackage[latin2]{inputenc}

\begin{document}
... Ide jön a szöveg ...
\end{document}
```

Szükséges még a `\usepackage{t1enc}` csomag is, mellyel Seyðisfjörður, Đurđevac, Þórshöfn típusú szavak is leírhatók, továbbá néhány olyan dolog is működni fog, amelyek a csomag nélkül nem mennének.

A dokumentumosztályok típusai a következők (egyelőre maradjunk az első három egyikénél):

- article – cikk (általában ezt használjuk)
- report – beszámoló („könyvesebb”, mint az előző)
- book – könyv (ez már komoly dolog)
- letter – levél (hivatalos levélforma)
- slide – fólia (előadásokhoz)

A dokumentumosztályokon belül léteznek opciók is, melyeket később fogunk részletezni. Az `a4paper` opció a standard A4-es kimenethez szükséges, használjuk mi is! A `babel` csomaggal magyaráthatjuk a \LaTeX et, erről később részletesen szó lesz, azért használjuk!

Példa:

```
\documentclass[a4paper]{report}
\usepackage{t1enc}
\usepackage[latin2]{inputenc}
\usepackage[magyar]{babel}

\begin{document}
... Ide jön a szöveg ...
\end{document}
```

Speciális karakterek: `\`, `{`, `}`, `%`, `~`, `$`, `_`, `^`, `&`, `#`

Így írjuk őket: `\` = `\backslash`, `{` = `\{`, `}` = `\}`, `%` = `\%`, `~` = `\~{}`,
`$` = `\$`, `_` = `_`, `^` = `\^{}`, `&` = `\&`, `#` = `\#` A „fok” jelölésére speciális utasítás `\circ`: Odakint -50°C van.

A `\`, `{`, `}`, `%` jelek feltétlenül fontosak a $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ ben:

- Visszaper (vagy másnéven rep) jel: parancsok kezdete
- A kapcsos zárójelek blokkosítanak, például parancsok argumentumait is közéjük kell zárni
- A százalék kommentez, akár sorvége jelet is!

Kommentek. A százalék jellel (%) a sor hátralévő részét kikommentezhetjük. Még az újsor karaktert is figyelmen kívül hagyja a \LaTeX ! Sor elején lévő szóközök nem számítanak. A sor vége jel – ha csak egy van belőle – szóközként funkcionál.

Így is írhatnám a szöveget, ha akarnám:

```
Így_is_írhatnám_%_de_nem_írom
a_ sz%
ö%szöljünk_itt_egy_kicsit
v%
%
eget,
ha_akarnám:%lám_itt_nem_volt_szóköz,_de_volt_újsor_karakter
```

Szóközök, térközök: \LaTeX alatt nem baj, ha több space-t ütött le az ember:
három_holló_=_három_holló három holló = három holló

A dokumentumokat bekezdésekkel struktúráljuk. Új bekezdés: két újsor karakter
(a tex file-ban egy üres sor)

Nem törhető szóköz: tilde (~): II.~Rákóczi~Ferenc eredménye:
II. Rákóczi Ferenc, II. Rákóczi Ferenc, II. Rákóczi Ferenc, II. Rákóczi Ferenc,
II. Rákóczi Ferenc, II. Rákóczi Ferenc

Blokkosítás: `\mbox{valami}`: `\mbox{II. Rákóczi Ferenc}` eredménye:
II. Rákóczi Ferenc, II. Rákóczi Ferenc, II. Rákóczi Ferenc, II. Rákóczi Ferenc,
II. Rákóczi Ferenc, II. Rákóczi Ferenc

Számok: IOO („kisel, nagyó, nagyó”) \neq 100 (száz),

Rövid számok kiírása.

Szebb így: Kétszer kettő néha öt, mint 2×2 néha 5!

Magyarban tizedesvessző, angolban tizedespont: 3,14, or 3.14. A helyiértékek jelölése magyarban térköz (pl. \,), angolban vesszővel történik: 21 345,6 Ft, illetve 21,345.6 Euro, respectively.

Mértékegység, pénznemek: nem törhető szóközzel: 1~cm

1 cm, 1 cm, 1 cm, 1 cm, 1 cm, 1 cm, 1 cm, 1 cm, 1 cm, 1 cm, 1 cm, 1 cm,
1 cm, 1 cm, 1 cm, 1 cm, 1 cm, 1 cm, 1 cm, 1 cm

Nyelvtan!!! No comment.

Itt ér véget a mai gyakorlat.

A hallgatók házi feladata a következő:

Letölteni a webről egy kb. 10 000 karakteres szöveget, amiben vannak különleges karakterek is, majd $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ formátumba hozni. A házi feladatban a bruttó (utasításokkal együtt számolt) karakterszám kb. 10 000 legyen.

A házi feladatokat a <http://mcl.hu/~kgergely/hf> címen található feltöltő scripttel kell beadni. A sikeres feltöltésről a megadott e-mail címre értesítést küld a script. A feladat leadásának határideje a következő gyakorlat napja előtti munkanap, reggel 8 óra.

Második laborgyakorlat

A szöveg begépelése

Írásjelek: pont, vessző közvetlenül az őt megelőző szóhoz zárva, utána szóköz.
Egyébként elég hülyén néz ki,ugye?Na.

Pontok halmozása: `\dots`, azaz `...`, ami nem azonos a három ponttal (`...`), sem a „három pont és köztük szóközök”-kel (`. . .`)!!!

Zárójel szorosan zárja a környezetét (nincs szóköz), de kívülről szóközzel!
Fordítva(ordítva)elég ocsmány.

Idézőjel: magyarban „ilyesmi” (`, , ilyesmi' '`, két vesszővel nyitjuk és két aposztróffal zárjuk), brit angolban looks like ‘this’ (``this'`, fordított és sima aposztrófok), in the US it “differs” (`` `differs' '`, az előbbi duplán) a bit. A hüvelyk (`"`, bár más tördelőkben ezt használtuk, itt nem jó!) nem idézőjel!!! Az idézőjelek helyes megjelenítéséhez szükséges a `tlenc` csomag!!!

Kötőjel, gondolatjel, minuszjel, hosszú gondolatjel esete (aki nem tudja miről van szó: <http://mcl.hu/~jenedy/kinezet.pdf>):

Kötőjel (-): egy minusz szimbólum leütésével előidézhető. Van-e kedved kipróbálni 5-6-szor?

Gondolatjel (—): két minusz szimbólum – mindkét oldalán szóközzel, vagy tartományt jelölve szóköz nélkül: 100–150 oldalas HF-eket fogadok el.

Hosszú gondolatjel (—, magyarban nem használjuk): angolban kedvelt, magyarban nem nyert. Három minusz szimbólum.

Minuszjel (—): \$-\$, odakint -5°C van, odakint -5°C van.

Párbeszédet gondolatjellel kezdünk --\, szöveg alakban (nincs szóköz a --\, és a szöveg között!!!).

Példa:

Hirtelen_feltépte_az_ajtót,_majd_így_kiáltott:

--\, Gyere_elő_a_vackodból,_te_vén_szipirtyó_--_sziszegte
dühösen_--,_ha_nem_jössz_elő,_megbúbollak\dots

Még_10-12_alkalommal_kiáltott_a_sötétbe_(mert_nem_látta,_hogyan
hol_lehet_az_öregasszony_a_4--8_méteres_távolságban).

Végül_,, visszavonulót''_fújt,_mert_nagyon_fázott_a
\$-5^\circ\$C-os_hidegben.

Elválasztási galibák. Bár a `babel` csomag az esetek többségében jól választ el, van amit nem tudhat, pl. karóra (karó-ra, vagy kar-óra).

Lokális megkerülő megoldás. Puha elválasztójel alkalmazása, amely `\-` alakban adott. Pl. `maka\ - róni`, makaróni, makaróni, makaróni, makaróni, makaróni, makaróni, makaróni

Globális megkerülő megoldás. A dokumentum tetszőleges részén kiadjuk a `\hyphenation{el-vá-lasz-ta-ni-mer-kell-neki-tyuhajja}` parancsot, és jól fog elválasztanimerkellnekityuhajja elválasztanimerkellnekityuhajja

Figyelem! Ha nincs betöltve a `t1enc` csomag, a `\hyphenation` nem fog működni ékezetes karakteres szavakkal!

Többnyelvű doksik esetén a `\usepackage[english,magyar]{babel}` paranccsal indulunk, és a `\selectlanguage{nyelv}` paranccsal lehet kiválasztani az aktuális nyelvet.

Rövidebb, bekezdésen belüli idegen nyelvű sorokra a `\foreignlanguage{nyelv}{szöveg}` parancsot érdemes használni.

Létezik még a `foreignlangauge`-nek környezet változata is, azaz a `\begin{foreignlanguage}{nyelv} szöveg \end{foreignlanguage}` utasítások közé is lehet idegen nyelvű szöveget írni.

Ligatúrák. A \LaTeX automatikusan kicseréli az ff, fl, fi és ffi betűkapcsolatok esetén a duplabetűket ligatúrákra. Ha nem akarjuk (miért nem?), akkor pl. `ef{ } féle` elkerülő manőverekre van szükség.

Speciális karakterek: Ha magyaron kívül más nyelveken is akarunk szavakat írni, szükség lehet ezen utasításokra: `\'e = é`, `\`e = è`, `\~n = ñ`, `\"o = ö`, `\H{o} = ő`, `\~o = õ`, `\^u = û`, `\o = ø`, `\ss = ß` stb.

Ha i-re, vagy j-re teszünk ékezetet, akkor előbb le kell vennünk a pontot az `\i` és `\j` utasításokkal, amivel `ı` és `ĵ` betűket kapunk. Így már lehet `ı́`ópofa ékezetet tenni rájuk. De az í-t is így írjuk: `\' \i`.

A \LaTeX tartalmaz néhány beépített utasítást, mellyel a \LaTeX logókat jeleníthetjük meg: `\TeX`, `\LaTeX` és `\LaTeXe`, azaz \TeX , \LaTeX és $\text{\LaTeX} 2_{\epsilon}$.

Mondatközök (ha nem akarjuk, a `\frenchspacing` paranccsal, vagy a `frenchspacing` csomag betöltésével kikapcsolhatjuk).

Alapértelmezésben a \LaTeX a mondatok közé nagyobb térközt tesz, mint a szavak közé. Ettől olvashatóbbá válik a nyomtatott szöveg.
mint a szavak közé. Ettől olvashatóbbá válik a nyomtatott szöveg.

Honnan tudja a \LaTeX , hogy hol van a mondat vége? Ha kis betűt, vagy számot pont követ, akkor az mondatvég (nagy betű utáni pont általában rövidítés).
Vannak azonban kivételek:

A 2. ábrán látható...Nem mondatvég, ezért A 2.~ábrán látható, vagy
A 2.\ ábrán látható alakban kell írni.

Kedvencem a BME. Mondatvég, ezért `Kedvencem a BME\@.` alakban kell írni.

Tipikus hibák szóköz–mondatköz témakörben. A kisbetűs rövidítések után elfeljtünk _t írni, ha azok nem mondat végére kerültek (pl., ill., ún., stb., vö. esetén a forrásba `pl. _ ill. _ ún. _ stb. _ vö. _`-t kell írni).

Sorszámnevek után pont szerepel, de hatásáról elfelejtkezünk (`2.~helyezett`, vagy `2._helyezett` a helyes és nem a `2._helyezett`, az utóbbiban mondatvég lesz a pont után).

Nagybetűs rövidítések után nem tesszük ki a \@-t.

Tessék odafigyelni, vagy betölteni a `frenchspacing` csomagot!

Kiemelés. A szövegben kiemelést az `\emph{szöveg}` paranccsal lehet elérni. Hatására (alapbeállításként) *dőlt betűvel* jelenik meg a szöveg.

Ha azonban a környezetében dőlt betűkkel szedtük a szöveget, akkor a kiemelés felállítja a kiemelt szavakat. Azaz az `\emph{ }` utasítás egymásba is ágyazható.
`\emph{Akár hiszed \emph{akár} nem}`

Lábjegyzetbe* a `\footnote{szöveg}` parancs használatával írunk. Hatására a szöveg a lap alján, apróbb betűvel jelenik meg.

A `\footnote` parancsot közvetlenül – szóköz nélkül – ahhoz a szóhoz, vagy mondathoz kötjük, amelyről írni akarunk. Például[†] így.
`Például\footnote{elegem van a példákból}` így.

*azért lábjegyzet, mert a lap alján jelenik meg apróbb betűvel

[†]elegem van a példákból

Itt ér véget a mai gyakorlat.

A hallgatók házi feladata a következő:

A múlt heti házit kiegészíteni a mai órán tanultakkal (elválasztási galibák, többnyelvűség kezelése, kiemelés, szóköz–mondatköz megfelelő használata). Ha nincs a szövegben, akkor töltsetek le olyan szöveget, amiben van! A házi feladatban a bruttó (utasításokkal együtt számolt) karakterszám 10 000–15 000 legyen.

A házi feladatokat a <http://mcl.hu/~kgergely/hf> címen található feltöltő scripttel kell beadni. A sikeres feltöltésről a megadott e-mail címre értesítést küld a script. A feladat leadásának határideje a következő gyakorlat napja előtti munkanap, reggel 8 óra.

Harmadik laborgyakorlat
Környezetek,
a dokumentum struktúrálása

A dokumentum struktúrálása (fejezetek, alfejezetek):

```
\part{Rész cím} -> csak book és report osztályokban  
\chapter{Alrész cím} -> csak book és report osztályokban  
\section{Fejezet címe}  
\subsection{Alfejezet címe}  
\subsubsection{Al-alfejezet címe}  
\paragraph{Még kisebb egység címe}  
\subparagraph{A legkisebb egység címe}
```

Ugyanezek csillaggal (`\section*{Hello}` stb.): számozatlan fejezetek

A `\tableofcontents` utasítás automatikusan generált tartalomjegyzéket illeszt be. Csak a számozott fejezeteket tartalmazza.

Dokumentum címe, szerző stb.

A `\title{...}`, `\author{...}`, `\date{...}` utasításokkal megadhatjuk a dokumentum címét, a szerzőt és a dátumot. A `\date{}` utasítás beillesztésével érhetjük el, hogy ne szerepeljen dátum a címben. A `\date{\today}` utasítás olyan, mintha nem is lenne: ha elmarad a `\date`, akkor a rendszeridő alapján a mai dátumot állítja be a \LaTeX .

Több szerző esetén az `\and` paranccsal sorolhatjuk fel őket. A cím beillesztése a `\maketitle` paranccsal lehetséges.

A dokumentum bármely részén ki lehet iratni az aktuális dátumot, a `\today` paranccsal. Ma pl. 2004. szeptember 3. van.

Egy példa:

```
\documentclass[a4paper]{book}
\usepackage[latin2]{inputenc}
\usepackage{t1enc}
\usepackage[magyar]{babel}

\begin{document}
\title{Az aranyhalak nemi élete \&\& és egyéb cigerettatípusok}
\author{Dr.~Kiss Géza \&\& egyetemi tanár \and
Kovács Alajos \&\& nyugdíjas alezredes}
\maketitle
\section{Bevezetés}
Itt jön a szöveg maga\&dots
\end{document}
```


Környezetek

\LaTeX alatt a `valami` környezetet `\begin{valami}` utasítással nyitjuk meg, és `\end{valami}` utasítással zárjuk.

Eddig is használtuk a `document` környezetet.

Környezetek egymásba is ágyazhatóak, de csak teljes átfedéssel. Azaz a `\begin{aaa} ... \begin{bbb} ... \end{bbb} ... \end{aaa}` előfordulhat, de `\begin{aaa} ... \begin{bbb} ... \end{aaa} ... \end{bbb}` nem létezhet.

Az utasítások `\command{param}` alakúak. A \LaTeX mindenképpen vár paramétert, ezért kell trükköznünk a paraméter nélküli utasításoknál (pl. `\LaTeX` és `\dots`). A szögletes zárójelekbe írt paraméterek opcionálisak (lásd pl. a `\documentclass[a4paper]{book}` parancsot).

Verbatim. Ha programkódot kell megjeleníteni egy doksiban és abban sok, a \LaTeX által vezérlőkarakternek használt karakter található, akkor ne menjen el a kedvünk, létezik megoldás!

A `\verb+szöveg\@%+` utasítással a `szöveg\@%` verbatim módban jelenik meg, azaz minden karakter megjelenik. Jelen esetben a + jellel jelöltük a verbatim széleit, de tetszőleges más karakter is használható (kivéve a betűket a szóközt és a csillagot)

Ha hosszabb – több soros – verbatimot szeretnénk, akkor célszerű a `\begin{verbatim} szöveg \end{verbatim}` környezetet használni.

A fenti utasítás és környezet csillagos változata megjeleníti a szóközöket is, pl. `\verb*+Helló világ!+ hatására Helló_világ!` kimenetet láthatjuk.

Felsorolások, számozott,
számozatlan listák

Három típusú felsorolás létezik $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ alatt: a számozatlan lista (`itemize`), a számozott lista (`enumerate`) és a leíró lista (`description`). Mindhárom környezetként létezik.

```
\begin{itemize}
\item Első elem
\item Második elem
\item Harmadik elem
\end{itemize}
```

Ugyanez `itemize` helyett `enumerate`-tel számozottá válik.

Felsorolások, listák 2. Ha a felsorolás egy pontját ki akarjuk emelni, akkor használhatjuk az `\item` parancs opcionális paraméterét (figyelem! Erre nagyon ritkán van szükség!). Például:

```
\begin{itemize}
\item Első elem
\item[$\circ$] Második, kiemelt elem
\end{itemize}
```

A `description` környezetben a leírt kifejezést az opcionális paraméterben adjuk meg, pl.

```
\begin{description}
\item[a butaságok] listába szedve sem válnak tudományossá,
\item[a jó gondolatok] viszont szebben mutatnak rendszerezve
\end{description}
```

Felsorolások, listák egymásba is ágyazhatóak.

```
\begin{enumerate}
\item Első külső elem
\item Második külső elem
\begin{enumerate}
\item Első belső elem
\item Második belső elem
\begin{description}
\item[a butaságok] listába szedve sem válnak tudományossá,
\item[a jó gondolatok] viszont szebben mutatnak rendszerezve
\end{description}
\item Harmadik belső elem
\end{enumerate}
\item Harmadik külső elem
\end{enumerate}
```

Idézetek, versek környezetei

A \LaTeX három környezetet támogat az idézetek beszúrásához. Ezek a `quote`, a `quotation` és a `verse` környezetek.

A `quote` környezetben nem kezdődnek behúzással a bekezdések. A `quotation` környezetben a bekezdések behúzással kezdődnek, míg a `verse` környezetben minden sor eleje azonos helyen kezdődik, a sorok végét `\\` karakterekkel jelöljük.

```
\begin{verse}
Tűz, te gyönyörű \\
lobogó, csillag erejű \\
Te fűtsd a mozdonyt halálra \\
Hajszold, hogy fekete magánya \\
Ne legyen néki teher.
\end{verse}
```

Példa a `quote` és `quotation` környezetek használatára. Próbáljuk ki mindkettőt hosszú sorokkal rendelkező szövegrészleten!

```
\begin{quote}
```

```
Ha elég hosszúak a sorok, akkor megláthatjuk, miféle különbséget rejt e két környezet, persze figyelniük kell arra is, hogy egy új bekezdést is kezdjünk.
```

```
Ha elég hosszúak a sorok, akkor megláthatjuk, miféle különbséget rejt e két környezet, persze figyelniük kell arra is, hogy egy új bekezdést is kezdjünk.
```

```
\end{quote}
```

Tárgymutató (Index)

Sokszor szükséges tárgymutató egy dokumentumhoz, megkönnyíti az olvasó munkáját, ha szöveges kulcsra akar rákeresni.

Használatához a preambulumban a `\usepackage{makeidx}` és `\makeindex` parancsokat kell kiadni. A tárgymutató tartalmát a dokumentum testében lévő `\index{kulcs}` kulcsok fogják alkotni. A `\printindex` utasítással a dokumentum tetszőleges helyén kinyomtathatjuk a tárgymutatót. Az `\index{kulcs!alkulcs}` utasítással albejegyzést is lehet csinálni.

A `\makeindex` utasítás hatására az összes kulcs kiíródik a `valami.idx` file-ba, ha a `valami.tex` file-ba dolgozunk. A `\printindex` parancs a `valami.ind` file tartalmát iratja ki. A `valami.ind` a `valami.idx` file sorba rendezett változata, amit pl. a `makeindex valami` utasítással érhetünk el. Ez utóbbit a terminálban kell futtatni.

Itt ér véget a mai gyakorlat.

A hallgatók házi feladata a következő:

A tanult környezetekkel rendelkező dokumentum letöltése a webről és tördelése a tanult szabályok alapján. A tárgymutató nem kötelező része a házi feladatnak! Aki akarja, beleteheti. A házi feladatban a bruttó (utasításokkal együtt számolt) karakterszám 7 000–10 000 legyen.

A házi feladatokat a <http://mcl.hu/~kgergely/hf> címen található feltöltő scripttel kell beadni. A sikeres feltöltésről a megadott e-mail címre értesítést küld a script. A feladat leadásának határideje a következő gyakorlat napja előtti munkanap, reggel 8 óra.

Negyedik laborgyakorlat

Irodalomjegyzék

Az irodalomjegyzék beillesztése a `thebibliography` környezettel lehetséges. A környezetnek van egy argumentuma, amely segíti a \LaTeX -et az irodalomjegyzék tördelésében. Egyelőre elég annyit tudni, hogy ha tíznél kevesebb művet sorolunk fel, akkor egy 9-est, ha száznál kevesebbet, de tíznél többet, akkor 99-et stb. kell ide írni.

A hivatkozott irodalmakat a `\bibitem{cimke}` paranccsal soroljuk fel (a `\cite{cimke}` utasítással tudunk majd rá hivatkozni). Az irodalomjegyzék automatikusan generálódik. Alapbeállításként a sorszámok növekvő arab számok szögletes zárójelek között.

A `\bibitem[Jópofa]{cimke}` hatására azonban a tartalomjegyzék vonatkozó elemének `Jópofa` lesz a „száma”, és ha hivatkozunk rá a szövegben, akkor is a `[Jópofa]` jelenik meg.

A szabályosan szedett irodalmi hivatkozások alakja a következő (forrás: IEEE):

- Könyv: szerző, *cím dőlt betűkkel*, kiadó neve, kiadó városa, kiadás éve.
- Újságcikk: szerző, „cím idézőjelek között”, *újság neve dőlt betűkkel*, évfolyam, szám, oldalak, kiadás éve
- Diplomadolgozat/disszertáció: szerző, *cím dőlt betűkkel*, oktatási intézmény, város, végzés éve.
- elektronikus forrás: teljes URL (átviteli protokoll megjelölésével), megtekintés időpontja percre pontosan

Példa:

...itt pedig hivatkozok a könyvre~\cite{konyv}.

...

```
\begin{thebibliography}{9}
```

```
\bibitem{konyv}
```

```
Virágvölgyi Péter,  
{\it A tipográfia mestersége},  
Osiris, Budapest, 2001.
```

```
\bibitem{cikk}
```

```
Claude E.\ Shannon,  
,,A Mathematical Theory of Communication'',  
{\it Bell System Technical Journal}, vol.~27, no.~10,  
pp.~379--423 és pp.~623--656, 1948.
```

```
\bibitem{url}
```

```
{\tt http://www.karinty.hu/~{}NIKITA/szedes/Bevezet.htm},  
2002.~január~9., 19:14
```

```
\end{thebibliography}
```

A `\cite` utasításnak létezik opcionális paramétere is, amellyel például az oldal-számot (vagy egyéb kiegészítő információt) adhatjuk meg. A kimeneten az iro-dalom címkéje mellett vesszővel jelenik meg a szögletes zárójelen belül [1, 160–165. oldal].

Példa:

```
...zárójelen belül~\cite[160--165.\ oldal]{konyv}.
```

```
...
```

```
\begin{thebibliography}{9}
```

```
\bibitem{konyv}
```

```
Virágvölgyi Péter,
```

```
{\it A tipográfia mestersége},
```

```
Osiris, Budapest, 2001.
```

```
...
```

```
\end{thebibliography}
```

Hivatkozás egyéb objektumokra

A \LaTeX automatikus metódust kínál a hivatkozások terén. Minden számozott objektumra lehet hivatkozni. Módja egyszerű: a hivatkozni kívánt objektumot megcímkezzük a `\label{cimke}` utasítással, majd a hivatkozás helyén a `\ref{cimke}` utasítással hívjuk elő a számot.

Példa:

```
\section{Bélabácsi}
```

```
\label{sec:bela}
```

```
Itt kezdem a fejezetet\dots
```

```
...
```

```
Később (vagy akár hamarabb is) elkezdek hivatkozni
```

```
a~\ref{sec:bela}~fejezetre.
```

Mindenre hivatkozhatunk, ami számozott (lábjegyzetre, listaelemre, fejezetre, képletre, táblázatra stb.). Oldalra a `\pageref{cimke}` utasítással hivatkozhatunk. Példa:

Bővebben a Bélabácsi fejezetben, a `\pageref{sec:bela}` oldalon...

Nem kötelező, de gyakran követett konvenció, hogy a címkében megjelöljük az objektum típusát. Például fejezeteket, alfejezeteket `sec:xxx`, táblázatokat `tab:xxx`, ábrákat `fig:xxx`, képleteket `eq:xxx` címkével stb. látjuk el.

Irodalomjegyzék beli elemre a `\cite{cimke}` utasítással hivatkozunk, a `\ref{}` nem használható! Oka: létezik egy Bi \TeX kiterjesztés, mellyel intelligensebb módon lehet irodalomjegyzéket beilleszteni nagyobb adatbázisokból (mi nem tanuljuk).

Az `\az{valami}` és `\Az{valami}` utasításokkal a `valami` elé határozott névelőt tehetünk, a különbség: az első kisbetűvel írja a névelőt, a második nagybetűvel kezdi (mondat elején jó). Az `\aref{}` és `\Aref{}` utasításokkal névelővel hivatkozhatunk objektumokra (tehát gyakorlatilag `\aref{cimke} = \az{\ref{cimke}}`).

Az `\acite{}`, `\Acite{}`, `\apageref{}` és `\Apageref{}` utasítások hasonlóan az előzőek kiegészítései. A korábbi példák tehát helyesen:

Később (vagy akár hamarabb is) elkezdek hivatkozni `\aref{sec:bela}~fejezetre`.

Bővebben a Bélabácsi fejezetben, `\apageref{sec:bela}` oldalon...

Új utasítások és környezetek létrehozása

Új utasítás definiálása. A `\newcommand` paranccsal történik. Az első argumentumba kell beírni az utasítás azonosítóját, a második argumentumban pedig a végrehajtandó parancsokat kell leírni.

Például a `\newcommand{\ft}{\emph{FONTOS!!!}}` hatására minden `\ft` utasítás a *FONTOS!!!*-t eredményezi. Ha már létező parancsot akarunk felüldefiniálni, akkor a `\renewcommand` parancsot kell alkalmaznunk, például így: `\renewcommand{\ft}{\emph{nem is olyan fontos\dots}}`, hatására az `\ft` utasítás már a *nem is olyan fontos...*-t eredményezi.

Ha egy esetleg létező utasítást akarunk definiálni, de csak akkor, ha nem létezik, akkor a `\providecommand` utasításra van szükségünk.

A definiált parancsok argumentumokat is fogadhatnak, ekkor a `\newcommand{\prncs}[argsz][default]{def}` alakot kell alkalmazni. Az utasítással a `\prncs` parancsot definiáljuk, melynek `argsz` számú argumentuma van. Az első argumentum opcionálissá tehető. Az első argumentum alapértelmezett beállítását a `default` írja le. A parancs definíciója a `def` argumentumban található. Az argumentumokat a `#1`, `#2` stb. alakban hívhatjuk elő.

Ha a parancs argumentuma nem lehet több egy bekezdésnél akkor a csillagos verziókat használjuk (`\newcommand*`, `\renewcommand*` stb.).

Példa: automatikus hivatkozás fejezetre és használata:

```
\newcommand*{\secref}[1]{\aref{sec:#1}.~fejezet}  
\newcommand*{\Secref}[1]{\Aref{sec:#1}.~fejezet}  
\Secref{bela}ben olvashatunk arról, hogy...
```

Hasonlóképpen hozhatunk létre környezeteket a

```
\newenvironment{körny}[argsz][default]{nyitódef}{záródef},
```

illetve a

```
\renewenvironment{körny}[argsz][default]{nyitódef}{záródef}
```

parancsokkal. Hatásukra a `körny` nevű környezetet `argsz` számú argumentummal definiáljuk. A csillagos verziókat érdemes használni, ha az argumentumok nem lehetnek hosszabbak egy bekezdésnél.

Példa:

```
\newenvironment*{bela}%  
{Nyitottam egy környezetet, azt írtam bele, hogy}%  
{Elég már ebből a környezetből, bezárom itt.}  
\begin{bela}  
Kacsacsőr  
\end{bela}
```

Másik példa, mellyel verseskötetet lehet egyszerűen szedni:

```
\newenvironment*{vers}[2][Petőfi Sándor]%  
{\begin{center} #1: {\sc #2} \end{center} \begin{verse}}%  
{\end{verse}}  
  
\begin{vers}[Arany János]{Toldi}  
Ég a napmelegtől a kopár szík sarja \\  
...  
\end{vers}  
  
\begin{vers}{Anyám tyúkja}  
Ej, mi a kő! Tyúkanyó, kend \\  
...  
\end{vers}
```

Ha az egyik parancsargumentumot a záródefinícióban akarjuk alkalmazni, akkor trükközni kell. A `\newsavebox{\valami}` utasítással létre kell hozni egy string tárolására alkalmas `\valami` változót, melynek a `\sbox{\valami}{érték}` utasítással adhatunk értéket. A későbbiekben a `\usebox{\valami}` utasítással irathatjuk ki a `\valami` értékét.

Az előbbi környezet a költő nevének aláírásával:

```
\newsavebox{\kolto}
\newenvironment*{vers}[2][Petőfi Sándor]%
{\begin{center}{\sc #2}\end{center}\sbox{\kolto}{#1}\begin{verse}}%
{\end{verse}\begin{flushright}\usebox{\kolto}\end{flushright}}
```

A környezetet természetesen ugyanúgy használhatjuk, mint az előbb.

Itt ér véget a mai gyakorlat.

A hallgatók házi feladata a következő:

Irodalomjegyzékkel és arra hivatkozásokkal rendelkező mű letöltése az Internetről és tördelésük a tanult szabályok alapján. Új utasítás(ok) és környezet(ek) definiálása. Micimackósok kíméljenek! A házi feladatban a bruttó (utasításokkal együtt számolt) karakterszám 7 000–10 000 legyen.

A házi feladatokat a <http://mcl.hu/~kgergely/hf> címen található feltöltő scripttel kell beadni. A sikeres feltöltésről a megadott e-mail címre értesítést küld a script. A feladat leadásának határideje a következő gyakorlat napja előtti munkanap, reggel 8 óra.

Ötödik laborgyakorlat

Táblázatok

Táblázatok. A \LaTeX alatt a `tabular` környezettel illeszthetünk be táblázatot. A környezet argumentumában adjuk meg, hogy milyen oszlopai lesznek a táblázatnak, illetve azt, hogy mely oszlopok közé szeretnénk függőleges vonalat húzni (`|`). Az oszlopok négy típusba sorolhatók: `left` (balra igazított), `right` (jobbra igazított), `center` (középre igazított) és `p{szélesség}`: adott szélességű.

Példa két oszlopú, három csíkos táblázatra:

```
\begin{tabular}{|l|r|}
balra igazított oszlop & jobbra igazított oszlop \\
akkor látszik & ha több elemű \\
\end{tabular}
```

Táblázatok 2. A táblázatban a sorokat `\\` jellel választjuk el egymástól, a soron belüli elemeket pedig az `&` jellel. Amelyik sor elé vízszintes vonalat szeretnénk húzni, ott a `\hline` parancsot kell kiadunk. Így ha az első sorban és az utolsóban szerepel a `\hline` utasítás, akkor keretezett lesz a táblázat.

Példa kétszer kettes keretezett táblázatra:

```
\begin{tabular}{|r|l|}  
\hline  
Egy & megérett a meggy \\  
Kettő & csipkebokor vessző \\  
\hline  
\end{tabular}
```


Táblázatok 3. Dupla vonalakat a `|` és a `\hline` duplázásával lehet elérni. Ha több, azonos rendű oszlopunk van, akkor a deklaráció lerövidíthető a `*` operátorral: a `*{3}{r|l|c|}` ekvivalens az `r|l|c|r|l|c|r|l|c|` definícióval.

Példa nyolc vonatos vasúti menetrendre:

```
\begin{tabular}{l||*8{c|}}
A vonat száma:                & 437 & ... \\
\hline\hline
Indul Bp.\ Keleti-pu.-ról      & 8:00 & ... \\
Érkezik Hatvanba              & 8:52 & ... \\
Indul Hatvanból               & 8:55 & ... \\
Érkezik Miskolc Tiszai-pu.-ra & 10:22 & ... \\
\hline
\end{tabular}
```

Táblázatok 4. Az @ paraméterrel a táblázat oszlopai közötti elválasztó átírható; a @{} nullára csökkenti az oszlopok közötti távolságot, a @{:} kettőspontot tesz két oszlop elemei közé. A \multicolumn{n}{oszldef}{szöveg} paranccsal n cellát összevonva oszldef módon írjuk a szöveget.

Példa: menetrend szebben (a kettőspontok egymás alá kerülnek):

```
\begin{tabular}{l||*8{r@{:}l|}}
A vonat száma: & \multicolumn{2}{c|}{437} & ... \\
\hline\hline
Indul Bp.\ Keleti-pu.-ról & 8&00 & ... \\
Érkezik Hatvanba & 8&52 & ... \\
Indul Hatvanból & 8&55 & ... \\
Érkezik Miskolc Tiszai-pu.-ra & 10&22 & ... \\
\hline
\end{tabular}
```

Táblázatok 5. A `\multicolumn{1}{oszldef}{szöveg}`-gel egyetlen cellában írhatjuk felül az oszlopokra vonatkozó definíciót. Az `@{ }` nagyon hasznos, mert ismétlődő mintákat könnyedén beilleszthetünk az oszlopok elé és után. A tradicionális nyomdászatban számok írásakor az azonos helyiértékeknek egymás alá kell kerülniük. Ez csak akkor probléma, ha tizedestörteket kell egymás alá írni. A megoldás: `r@{,}l`

Példa táblázatra számokkal:

```
\begin{tabular}{c|r@{,}l}
Kifejezés          & \multicolumn{2}{c}{Érték} \\
\hline
 $\pi$           & 3&1415927          & \\
 $\pi^{\pi}$        & 36&46216          & \\
 $\pi^{\pi^{\pi}}$  & 80662&666          & \\
\end{tabular}
```

Táblázatok 6. Arra kell ügyelni, hogy az @ tényleg eltünteti mindent: ha az utolsó oszlop után, vagy az első oszlop előtt szerepel, megszünteti a távolságot a táblázat széle és a szöveg között -> ezt manuálisan kell korrigálni.

Egy másik példa áruházi pénztárgéphez

```
\begin{tabular}{l|r@{,}l@{\,Ft\_\_\}}
\multicolumn{1}{c|}{Áru_\neve}_\&_\multicolumn{2}{c}{Érték}_\ \\
\hline
Krumpli_\&_\_24&40_\ \\
Kenyér_\&_\_106&20_\ \\
Tej_\&_\_110&80_\ \\
...
\end{tabular}
```


A táblázatok elég bután néznek ki szövegben, ezért célszerű „úsztatni” őket. Erre szolgál a `table` környezet. A `table` környezettel „be kell burkolni” a `tabular` környezetet. A `table` környezetnek van egy opcionális paramétere, mellyel befolyásolhatjuk a táblázat helyét, lehetséges értékei `here` (ide vele), `top` (lap tetejére), `bottom` (lap aljára), `page` (külön lapra). Az opcionális paraméterben ezek tetszőleges sorrendben felsorolhatóak, mellyel a prioritásukat határozzuk meg. A \LaTeX kiválasztja melyik a legjobb és elhelyezi a táblázatot. Alapértelmezés: `[tbp]`. Ha azt szeretnénk, hogy mindenképpen adott helyre tegye, akkor egy felkiáltójellel lehet szigorítani az utasítást: `[!h]` jelentése: mindenképpen ide tedd.

A `table` környezet belsejében a `\caption{szöveg}` paranccsal adhatunk feliratot a táblázathoz, illetve itt is lehet a `\label{cimke}` parancsot használni. A `\caption` pozíciójától függően a táblázat felett, vagy alatt jelenik meg a felirat.

Példa úsztatott táblázatra:

```
\begin{table}[htb]
\caption{Nagy bölcsességek}
\label{tab:fontos}
\center
\begin{tabular}{c|r@{,}l}
Kifejezés & \multicolumn{2}{c}{Érték} \\ \hline
 $\pi$  & 3.1415927 & \\
 $\pi^{\pi}$  & 36.46216 & \\
 $\pi^{\pi^{\pi}}$  & 80662.666 & \\
\end{tabular}
\end{table}
```

Az előbbi egy új `tabl` környezet definiálásával:

```
\newenvironment*{tabl}[3]% #1: caption, #2: cimke, #3: oszlopdef
{\begin{table}[htb]%
\caption{#1} \label{tab:#2} \center%
\begin{tabular}{#3}}
{\end{tabular}\end{table}}
...
\begin{tabl}{Nagy bölcsességek}{fontos}{c|r@{,}l}
Kifejezés          & \multicolumn{2}{c}{Érték} \\
\hline
 $\pi$           & 3&1415927          & \\
 $\pi^{\pi}$       & 36&46216          & \\
 $\pi^{\pi^{\pi}}$  & 80662&666          & \\
\end{tabl}
```


Képek, ábrák beillesztése

Ábrák. A \LaTeX -nek van beépített parancskészlete ábrák készítéséhez, de mivel nem vagyunk mazochisták, ezeket nem tanuljuk. Helyette mindenkinek javaslom a grafikus felülettel rendelkező programok használatát, majd EPS (Encapsulated PostScript) formátumba mentést.

A `figure` környezet szintén egy úszó objektum, melybe grafikus file-okat lehet beilleszteni. A beillesztés az `\includegraphics{filename}` utasítással történik és a `graphics` csomagot szükséges betölteni a használatához. A `\label{cimke}`, `\caption{felirat}` ugyanúgy használható, de itt csak az ábra alá lehet írni.

Ha az ábra nem megfelelő szögben áll, vagy nem megfelelő a mérete, a `\rotatebox{fok}{mit}` és `\resizebox{vízszintes}{függőleges}{mit}` parancsokkal módosíthatunk rajtuk. Az utóbbinál a két paraméter egyikénél használható a felkiáltójel, jelentése: az arányokat tartsd.

A <http://www.mcl.hu/~joneyg/epsfiles> címről próbaképpen pár EPS file letölthető. Tetszőleges grafikus formátum pl. a UNIX-os `convert` paranccsal EPS-sé alakítható (`convert filename.jpg filename.eps`).

Példa a `figure` környezet alkalmazására:

```
\usepackage{graphics}
...
\begin{figure} [bp]
\center
\resizebox{80mm}{!}{
\rotatebox{-90}{
\includegraphics{maki.eps}}}
\caption{A \LaTeX\ órák mestere}
\label{fig:maki}
\end{figure}
```

A hivatkozások egyszerű kezeléséhez az alábbi utasítások definiálását javaslom (ezekkel jóval kevesebbet kell gépelni):

```
\newcommand{\secref}[1]{\aref{sec:#1}~fejezet}  
\newcommand{\figref}[1]{\aref{fig:#1}~ábra}  
\newcommand{\figreff}[1]{\aref{fig:#1}~ábrá}  
\newcommand{\tabref}[1]{\aref{tab:#1}~táblázat}  
\newcommand{\Secref}[1]{\Aref{sec:#1}~fejezet}  
\newcommand{\Figref}[1]{\Aref{fig:#1}~ábra}  
\newcommand{\Figreff}[1]{\Aref{fig:#1}~ábrá}  
\newcommand{\Tabref}[1]{\Aref{tab:#1}~táblázat}
```

...

```
\Figreff{maki}n egy majmot láthatunk. Szép \figref{maki}.
```

Itt ér véget a mai gyakorlat.

A hallgatók házi feladata a következő:

Táblázatokkal és ábrákkal rendelkező dokumentum letöltése a webről és tördelése \LaTeX -hel. Legalább egy olyan táblázat is legyen, amelyikben tizedes törtek is vannak. Hivatkozás táblázatokra és ábrákra a folyó szövegben. A házi feladatban a bruttó (utasításokkal együtt számolt) karakterszám 7 000–10 000 legyen.

A házi feladatokat a <http://mcl.hu/~kgergely/hf> címen található feltöltő scripttel kell beadni. A sikeres feltöltésről a megadott e-mail címre értesítést küld a script. A feladat leadásának határideje a következő gyakorlat napja előtti munkanap, reggel 8 óra.

Az oktató házi feladata a következő:

A matematikai szimbólumokat tartalmazó fénymásolatok sokszorosítása, szétosztása.

Hatodik gyakorlat

A matematikai szedés

Matematikai szedés. A \LaTeX egyik legnagyobb előnye más szövegszerkesztőkhöz képest a szép matematikai szedés. Három matematikai módot különböztetünk meg egymástól:

- soron belüli matematikai mód (`$` dollárjelek `$`, vagy `\(\)` ilyenek közé kerül a matematikai kifejezés, vagy a `math` környezetbe),
- külön sorba kerülő, de nem számozott egyenlet (`displaymath` környezetbe, vagy `\[ilyenek \]` közé írjuk a matematikai kifejezést),
- számozott, külön sorba írt egyenlet (`equation` környezet). Az utóbbi címkézhető (`\label{cimke}`-vel) és hivatkozható.

Az anyagban szereplő utasításokat matematikai módban kell kiadni!

Példa:

Mindegy, hogy $a^2 + b^2 = c^2$, vagy $\backslash(a^2 + b^2 = c^2\backslash)$, vagy

```
\begin{math}
```

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

```
\end{math},
```

ugyanazt láthatjuk a kimeneten.

Viszont a

```
\[ x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \]
```

 és a

```
\begin{displaymath}
```

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

```
\end{displaymath}
```

is ugyanaz.

Az `equation` környezet számozott, tehát címkézhető; figyeljük meg a hivatkozás formáját – zárójelben szerepel, tehát `(\ref{eq:valami})` jellegű –, de az `\aref({eq:valami})` utasításnál az argumentumot kell bezárni kerek zárójellel!

...mindenki tudja, hogy a π közelíthető `\aref({eq:pi})` kifejezéssel. Mégis megadjuk az egyenletet

```
\begin{equation}
\label{eq:pi}
\pi = \lim_{n \to \infty} \sqrt{\sum_{k=1}^n \frac{6}{k^2}}.
\end{equation}
```

Kiemelt képletek esetében az írásjelek (pont, vessző) mindig a környezeten belülre, a formula végére kerülnek.

Matek módban abszolút nem számít, hogy hány szóközt ütött az ember: $\$ab_ = a_ _ _ b\$,$ ami így néz ki: $ab = ab$. Matek módban nem szabad üres sort írni (nincs új bekezdés).

A pont alapértelmezése: tizedespont (amerikai program), a vessző: felsorolás, melynek következtében $\$3,14\$,$ így néz ki: 3,14. Megoldás: tizedes törteknél negatív térköz ($\backslash!$), mellyel: $\$3, \backslash!14\$, = 3,14$.

A kapcsolósárójelekkel blokkosíthatunk ($\{ \dots \}$), amire gyakran van szükség ($\$x^y + z \backslash neq x^{\{y + z\}}\$$).

Relációjelek. Az $=, <, >$ ugyanúgy használható matek módban. A nem egyenlő, kisebb egyenlő és nagyobb egyenlő jelek a $\backslash ne, \backslash le, \backslash ge$ parancsokkal hívhatóak elő.

Összeadás, kivonás, osztás a megszokott $+$, $-$, $/$ jelekkel lehetséges. Vegyük észre, hogy matek módban másképp mutatnak ezek a jelek: $(2+3)/5=3-2$ simán és $(2 + 3)/5 = 3 - 2$ matek módban.

A szorzásjelet nem $*$ -gal jelöljük! Mérnöki gyakorlatban a csillag jelentése: konvolúció. Helyette nem írunk semmit, vagy középmagas pontot (\cdot), vagy \times -t (\times) teszünk.

Alsó index az $_{\text{valami}}$, felső index a $^{\text{valami}}$ utasításokkal lehetséges. Vigyázat! Egy matematikai elemnek csak egy alsó és egy felső indexe lehet. Indexek egymásba ágyazása blokkosítással lehetséges (lásd $\pi^{\pi^{\pi}}$). Erre is figyelni kell: $x^{y^z} \neq (x^y)^z$: $x^{y^z} \neq (x^y)^z$ (vegyük észre a z -k különbözőségét).

Görög betűk. Szimplán α , β ..., nagy betűk nagy kezdőbetűvel, pl. Φ , Ω ...

Gyökjel az `\sqrt{valami}` utasítással lehetséges. A \LaTeX – kevésbé töketlenül, mint a $\text{M}\$ \text{Word}$ – többet is egymásba tud ágyazni:

```
\begin{equation}
\label{eq:emeletes}
\lim_{n \to \infty} \left( \sqrt{ \frac{1}{\frac{1}{n} + \sqrt{ \frac{1}{\frac{1}{n} + \sqrt{ \dots }}}}} \right) = ?
\end{equation}
```

Ha a gyökjel kitevőjét meg akarjuk változtatni, akkor az opcionális paraméterrel tehetjük meg: $\$ \sqrt[3]{2} \approx 1, \!26 \$$: $\sqrt[3]{2} \approx 1,26$.

Nagy törtjel a `\frac{számláló}{nevező}` utasítással lehetséges (lásd a korábbi példákon). Kombinatorikai kifejezésekben hasznos lehet az „enalattaká”, amit `{n \choose k}` utasítással jelenítünk meg. Ha zavar minket a zárójel, akkor lehet `{n \atop k}` is.

Zárójelek. A hagyományos kerek és szögletes szimplán, a kapcsos `\{, \}` módon. Ha méretre nem stimmel, akkor célszerű használni a `\left\{` és a `\right]` parancsokat. Jelentése: a `\left` és a `\right` közötti tartalom magasságához igazítsd a zárójelek méretét.

Ha valahova nem akarunk valódi zárójelet (csak a határt akarjuk jelölni), akkor `\left.-t`, vagy `\right.-t` kell írni. Figyelem! A `\left` és a `\right` egy sorban kell szerepeljen! (később lesz jelentősége).

Normaérték jele `\|`, az abszolútértéké `|`: $\sqrt{(a - b)^2} = |a - b|$
Ha két vektor (sík) párhuzamosságát akarjuk jelölni, akkor a `\parallel` parancsra van szükség: `e \parallel f`.

Ha valami miatt mégse jó a keretezés (amit a `\left` és `\right` utasításokkal csináltunk), akkor létezik négy méretmódosító utasítás: `\big(`, `\Big(`, `\bigg(`, `\Bigg(`

Szumma, produktum és integrál a `\sum`, `\prod`, `\int` utasításokkal lehetséges. Az alsó és felső index az operátor alá és fölé kerül. Deriváláshoz a „d” operátort a `\mathrm{d}` paranccsal állíthatjuk elő. Ha vesszőt használunk, akkor a `'` is tökéletes:

$$f'(x) = \frac{\mathrm{d} f(x)}{\mathrm{d} x}$$

Ne feledjük, hogy az integrálásban is szerepel a „d” operátor! Célszerű létrehozni egy új utasítást: `\newcommand*{\dd}{\mathrm{d}}...`

Függvények. A $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ beépített függvényekkel rendelkezik, amelyek megdöbbentő módon `\arccos`, `\sin`, `\tanh`, `\exp`, `\ln` stb. alakúak.

Szöveg bevitele a `\textnormal{szöveg}`, vagy a `\textrm{szöveg}` paranccsal lehetséges: `\[x \geq 0, \textrm{ ha } x \text{ nemnegatív} \]` A `\textnormal` parancs azonban nem követi a környezeti beállításokat ezért az esetek többségében a `\textrm` parancsot érdemes használni.

Kettőspont általában relációjelekben szerepel (pl. `:=`), ezért ha nem relációjelként szeretnénk használni, akkor a `\colon` parancs kell, pl. `$f\colon x \mapsto x^2$`.

Pontok halmozása – ugyanúgy, mint a szövegben – a `\ldots` paranccsal lehetséges az alapvonalra. Vigyázat, az angol kultúrájú országokban középre rakják

a pontokat, ha műveleteket hagyunk ki vele: $x_1 + \cdots + x_n$. Létezik függőleges (\vdots) és ferde (\ddots) pontozás is.

Itt ér véget a mai gyakorlat.

A hallgatók házi feladata a következő:

Matematikai képletekkel telezsúfolt jegyzet-részlet \LaTeX -ben való tördelése. A még nem tanult eszközöket (mátrixok feltöltése, egymás alá rendezett egyenletek) próbáljátok elkerülni. A házi feladatban a bruttó (utasításokkal együtt számolt) karakterszám 7 000–10 000 legyen.

A házi feladatokat a <http://mcl.hu/~kgergely/hf> címen található feltöltő scripttel kell beadni. A sikeres feltöltésről a megadott e-mail címre értesítést küld a script. A feladat leadásának határideje a következő gyakorlat napja előtti munkanap, reggel 8 óra.

Hetedik gyakorlat
Matematikai szedés II

Az `eqnarray` környezet alkalmas igazított egyenletek bevitelére. Gyakorlatilag egy `{rcl}` típusú tömb, melyben az oszlopokat `&`-tel, a sorokat `\\`-pel választjuk el. Csillagos változata számozatlan egyenleteket szűr be, csillag nélküli változatában a `\nonumber` paranccsal lehet kikapcsolni a számozást egy sorban.

Példa:

```
\begin{eqnarray}
1^3 + 2^3 & = & (1 + 2)^2 \nonumber \\
1^3 + 2^3 + 3^3 & = & (1 + 2 + 3)^2 \\
& \vdots & \nonumber \\
\sum_{i=1}^n i^3 & = & \left( \sum_{i=1}^n i \right)^2
\end{eqnarray}
```

Ha túl hosszú az egyenletünk, a \LaTeX nem fogja automatikusan törni, mert nem tudja, hogy hol törheti az egyenletet. Túl hosszú egyenleteknél is az `eqnarray` környezetet kell használni. Hasznos lehet a `\lefteqn{}` parancs, mellyel az argumentum hosszát virtuálisan 0pt-re állíthatjuk. Ha az egyenlőségjel két oldalán lévő távolság nem tetszik, akkor használjuk a `\setlength\arraycolsep{2pt}` parancsot.

```
\setlength\arraycolsep{2pt} \begin{eqnarray*}
\lefteqn{f(x) = f(x)\Bigg|_{x=x_0} + (x - x_0) f'(x)\Bigg|_{x=x_0}}
\\ \&\&\{ + \frac{(x - x_0)^2}{2} f''(x)\Bigg|_{x=x_0} +
\ldots + \frac{(x - x_0)^n}{n!} f^{(n)}(x)\Bigg|_{x=x_0} + \ldots,
\end{eqnarray*} vagy akár \begin{eqnarray}
f(x) \&= \& f(x)\Bigg|_{x=x_0} + (x - x_0) f'(x)\Bigg|_{x=x_0}
\nonumber + \frac{(x - x_0)^2}{2} f''(x)\Bigg|_{x=x_0} \\ \&\&\{ +
\ldots + \frac{(x - x_0)^n}{n!} f^{(n)}(x)\Bigg|_{x=x_0} + \ldots
\end{eqnarray}
```

Tömbök matematikai módban. Gyakran van rájuk szükség (pl. mátrix ábrázolása), az `array` környezet biztosítja nekünk. Használata ugyanaz, mint a `tabular` környezeté, de matek módban csak ez megy.

```

\begin{equation}
|x| = \left\{ \begin{array}{rl}
x & \text{ha } \$x \geq 0$, \\
-x & \text{ha } \$x < 0$.}
\end{array} \right.
\end{equation}

```

Az egységmátrixot pedig az

```

\[\mathbf{I} = \left( \begin{array}{cccc}
1 & & & \\
0 & 1 & & \\
\vdots & & \ddots & \\
0 & 0 & & 1
\end{array} \right) \]

```

képlet írja le.

Emlékezzünk arra, hogy kétféle utasítás létezik: általános érvényű (pl. `\bf`) és argumentumot fogadó (pl. `\textbf`). Az általános érvényű parancsok hatása is korlátozható, ha blokkosítjuk a dokumentumunkat a `{ }` jelekkel (csak `{\bf ezt}` vastagon). A továbbiakban a könnyebb érthetőség kedvéért az általános érvényű utasításokat `{\utasítás }` alakban írjuk, míg az argumentumot fogadó utasítások alakja `\utasítás{ }` lesz.

Beavatkozás matek módban. A `\mathit{ }`, `\mathrm{ }`, `\mathbf{ }`, `\mathsf{ }`, `\mathtt{ }`, `\mathcal{ }` és `\mathnormal{ }` parancsokkal lehet betűtípust váltani matek módban. A leggyakrabban a `\mathbf{ }`-re van szükség, mivel ezzel lehet kövér betűt írni (vektorok és márixok esetében lehet fontos). Példa: `\mathbf y = \mathbf{H} \cdot \mathbf x`

További betűtípust módosító parancsok (zárójelben a szükséges csomag):
`\mathscr{}` (`[mathscr]{eucal}`), `\mathfrak{}` (`eufrak`) és `\mathbb{}`
(`amssymb`). Tessék őket kipróbálni!

A matematikai képletek „kivastagítása” a `{\boldmath }` utasítással lehetséges, de ezt text módban kell kiadni! Például Pithagoras mondá:
`{\bf\boldmath Kedvenc tételelem az $a^2 + b^2 = c^2$ }.}`

Matematikai méretek módosítása az alábbi parancsok egyikével lehetséges: `{\displaystyle }`, `{\textstyle }`, `{\scriptstyle }` és `{\scriptscriptstyle }`. A parancs kiadása után az őt követő szövegre a környezet végéig hatással lesz a parancs.

Példa: `$a_{\scriptscriptstyle \text{kicsi index}}^{\text{normál index}}$`

Térközök matematikai képletekben. Gyakran kell „odébb tenni” a magyarázó szöveget, illetve a feleslegesen összefolyó elemek „szétválasztásakor” is szükség lehet rájuk. A már ismert `\,` ($\!|$) mellett használható a `\:` ($\!|$) és a `\;` ($\!|$). A szintén ismert `_` utasítás szóközt generál, a `\quad` ($\!|$) és `\qquad` ($\!|$) nagyobb térközök beszúrására alkalmas. A `\quad` az aktuális betűtípus „M” betűjének szélessége. A `\!` utasítással lehet negatív térközt beilleszteni ($-\frac{3}{18}$ quad, $\!|$).

A `` utasítással a `valami` nem jelenik meg, hanem csak a helyét hagyja ki a \LaTeX . Tipikus alkalmazása az alábbi:

`\$^{12}_ {6}\$C, {}^{12}_6C.`

Ha valakinek ez még mindig kevés, akkor gányolhat a `\hspace{hossz}` utasítással, ahol a `hossz` tetszőleges mértékegységben megadható és lehet negatív is. A `\hspace{hossz}` matek módban tényleg gányolás, kerüljük a használatát!!!

Új matematikai operátor a `\mathop{}` paranccsal hozható létre. Ha utána írjuk a `\limits` parancsot, akkor az alsó és felső index alá, illetve felé kerül (ez a default). Ha a `\nolimits` parancsot írjuk utána, akkor alsó és felső indexbe kerülnek az indexek. A gyakran használt operátorokat célszerű `\newcommand`-dal definiálni.

Kétváltozós műveleti jelet a `\mathbin{formula}` paranccsal definiálhatunk, új relációjelet a `\mathrel{formula}` hoz létre. Ha szeretnénk valamit írni az – akár létező, akár most definiált – relációjel felé, a `\stackrel{amit}{amire}` parancsra van szükség.

Ha automatizálni akarjuk a matek mód előhívását, akkor az `\ensuremath{arg}` parancsot alkalmazhatjuk. Jelentése: ha matek módban vagy, csak hajtsd végre az `arg`-ot, ha nem, akkor kapcsolj matek módba és utána hajtsd végre az `arg`-ot.

Példák:

```
\providecommand*{\log}{\mathop{\log}\nolimits}
\newcommand*{\R}{\ensuremath{\mathbb{R}}}
```

...

Jelölje \R a valós számok halmazát, legyen továbbá $a, b \in \R$, ezzel:

```
\[ \log_2 \left( a \mathbin{\circ} b \right)
      \stackrel{!}{\mathrel{\rho}} c \]
```

Matematikai opciók. A dokumentumok elején a `\documentclass` opcionális paramétereivel meghatározhatjuk a külön sorba írt egyenletek általános képét: a `leqno` opciójával az egyenletek számozása bal oldalra kerül. A `fleqn` opcióval pedig az egyenletek nem középre rendezve, hanem balra zárva jelennek meg. Ha nem tetszik a matematikai képletek behúzásának mértéke, akkor a `\setlength\mathindent{távolság}` paranccsal felülírhatjuk azt.

Ami kimaradt, de lehet rá szükség: AMS \LaTeX csomagok (AMS = American Math Society), pl. `amsmath`, `amsfonts`, `amstext` stb. Erről külön tárgyat indíthatnánk —> érdeklődők utánanézhhetnek a <http://www.ams.org/tex/amslatex.html> címen.

Itt ér véget a mai gyakorlat.

A hallgatók házi feladata a következő:

Brutális képleteket tartalmazó jegyzetrészlet tördelése \LaTeX -hel. Különösen érdekel az órán tanultak (matematikai tömbök, egymás alá rendezett egyenletek) megfelelő alkalmazása. A házi feladatban a bruttó (utasításokkal együtt számolt) karakterszám 7 000–10 000 legyen.

A házi feladatokat a <http://mcl.hu/~kgergely/hf> címen található feltöltő scripttel kell beadni. A sikeres feltöltésről a megadott e-mail címre értesítést küld a script. A feladat leadásának határideje a következő gyakorlat napja előtti munkanap, reggel 8 óra.

Nyolcadik gyakorlat

Tételek a szövegben

Tételek, lemmák gyakran szerepelnek a szövegben. A tételek kezeléséhez az alábbi dolgokra van szükség:

1. A preambulumban definiálni kell egy új tételkörnyezetet a

`\newtheorem{körny_neve}{megjelenítendő_szöveg}` paranccsal.

Két helyre illeszthetünk be opcionális paramétert, melyek jelentése a következő példákból remélhetőleg kiviláglik.

2. A dokumentum törzsében `\begin{körny_neve}` és

`\end{körny_neve}` parancsok közé kerül a tétel szövege.

A bizonyításokra nincs külön környezet, de definiálhatunk egyet.

A szükséges definíciók:

```
% Preambulumba!!!  
\newtheorem{ttl}{tétel} % tétel beillesztésére szolgáló parancs  
\newtheorem{lem}{lemma} % lemma beillesztésére szolgáló parancs  
\newtheorem{pld}{példa}[section]  
% a példák számozása a fejezetszám alapján  
\newtheorem{fel}{feladat}[chapter]  
% a feladatok számozása a chapter száma alapján  
\newtheorem{prb}[pld]{probléma}  
% a példák számlálóját használjuk a problémákhoz is  
\newenvironment*{proof}{\noindent \emph{Bizonyítás:} }%  
{\hfill $\Box$ \par} % latexsym csomag legyen betöltve!
```


A dokumentumban így használhatjuk a definíciókat:

```
\begin{ttl} Kedvenc tételelem \end{ttl}
\begin{proof} Triviális. \end{proof}
\begin{lem} Azért kimondok egy még gyengébb tételt \end{lem}
\begin{proof} Hogyezeddignemjutottazeszembe\dots \end{proof}
\begin{pld} Lám milyen szépen örökli a fejezetszámot! \end{pld}
\begin{fel} Lám milyen szépen örökli a chapter számot! \end{fel}
\begin{prb} Ez meg úgy számozódik, mint a példa! \end{prb}
```

További lehetőségeként lásd a `theorem`, `ntheorem` és `amsthm` csomagokat.

A szöveg formázása

Sorok törése. Ha olyan helyre akarunk sortörést, ahova a \LaTeX magától nem tesz, akkor a `\`, `*`, `\[hossz]`, vagy `*[hossz]` utasítások egyikére van szükség. A csillagos változat nem engedi a lap aljára a sortörést. A `hossz` opcionális paraméterrel megadhatjuk, hogy mekkora helyet hagyjon ki a \LaTeX .

Léteznek beépített távolsággal sort ugró parancsok is, ezek a `\smallskip`, `\medskip` és `\bigskip`. **FIGYELEM!** A manuális sortörésre igen ritkán van szükség, csak akkor használjuk, ha tényleg kell.

Egy bekezdésként (egy üres sor után) írt sort a `\noindent` paranccsal tehetünk nem bekezdéssé, illetve az `\indent` parancs teszi a sort bekezdéssé, ha azt nem úgy írtuk. E két parancsra általában új utasítások definiálásakor lehet szükség (lásd a `proof` környezetnél)!

Sorkizárás, túllógó sorok. Ha fordítás során sok olyan sorral találkozunk, melyeket a \LaTeX nem tudta eltörni, akkor be kell avatkoznunk. A \LaTeX -ben rengeteg regiszter segíti a szedés minőségét, a regiszterek értékei pedig állíthatóak (erről később lesz szó). A \leftthyphenmin és \rightthyphenmin változókkal meghatározhatjuk a minimálisan szükséges karakterszámot az elválasztójel előtt, illetve utána. Az alábbi beállítások javasoltak magyar nyelv használata esetén:

```
\setlength\leftthyphenmin{2} % legalább két karakter a kötőjel előtt  
\setlength\rightthyphenmin{2}% legalább két karakter a kötőjel után  
\setlength\hyphenpenalty{0} % nem baj, ha elválasztunk a sor végén  
\setlength\uchyph{-1} % ne válasszon el nagybetűvel kezdődő szavakat
```

Előfordul azonban, hogy csak körülményes módon szabadulhatunk meg a kilógó soroktól (pl. manuális sortörés). Megoldások a következő oldalon.

A `{\sloppy}` utasítással nagyobb szóközöket is megengedünk a szavak között, mint például a fóliák szövege esetében. A `{\fussy}` utasítással kapcsolhatunk vissza az alapértelmezett üzemmódra. Azonos módon működnek a `sloppypar` és `fussypar` környezetek is.

Keskeny helyeken célszerű a tömbös (justified) szedés helyett balra, vagy jobbra (esetleg középre) zárt szedést alkalmazni, mellyel elkerülhetjük a sorok egységnyi szélességbe kényszerítését. A balra zárt szedést a `flushleft` környezet, vagy a `{\raggedright}` parancs aktivizálja. A szöveg jobbra zárásához a `flushright` környezetet, vagy a `{\raggedleft}` parancsot kell használni. Középre a `center` környezettel, vagy a `{\center}` utasítással lehet zárni a sorokat.

Szedés, lapdobás, fejezetek elhelyezése. A szedés típusa a `\documentclass` parancs opcionális paraméterével állítható. Egyoldalas szedéshez a `oneside` opciót, kétoldalas szedéshez a `twoside` opciót kell beírni. Az utóbbi esetben a lapok megjelenése különböző lesz a bal és jobb oldalon.

Ha kétoldalas a szedés, és `book`, vagy `report` osztályban dolgozunk, a `\documentclass` parancs opcionális paraméterével állítható, hogy új `chapter` csak jobb oldalon (`openright`), vagy tetszőleges helyen (`openany`) kezdődhet.

A `\clearpage` utasítással lehet lapot dobni, jelentése: kezdj egy új oldalt. A `\cleardoublepage` is új oldalt kezd, de úgy, hogy az jobb oldali lap legyen.

Többhasábos szedés. A `\documentclass` opcionális paraméterében állítható. A `onecolumn` (alapértelmezett) opció egyhasábos művet állít elő, a `twocolumn` kéthasábosat. A kéthasábos műben a `\onecolumn` és a `\twocolumn[szöveg]` utasításokkal válthatunk az egy- és kéthasábos üzemmód között. A `szöveg` a két hasáb fölé kerülő bevezető szövege.

Ha két hasáb kevés lenne, akkor a `multicol` csomagot töltsük be. A `multicols` környezet argumentuma meghatározza a hasábok számát (tíznél kisebb), illetve itt is megvan a bevezető szöveg lehetősége. Ha sok hasáb van, célszerű a `\raggedright`-tal balra igazítani. Példa:

```
\usepackage{multicol}
...
\begin{multicols}{3}[Egye fene, három oszlopba szedem
szövegemet:] Hogyha nékem sok pénzem lesz...
\end{multicols}
```

Betűtípusok változtatása: a dokumentum szintjén történik (a nyomdások szerint: egy betűcsaláddal kell szedni az egész művet, de a betűcsalád lehetséges alakjait is használni kell). Az Adobe által ingyenesen elérhetővé tett fontok a megfelelő csomag betöltésével használhatóak (pl. `times`, `avant`, `avantgar`, `bookman`, `chancery`, `charter`, `courier` stb., lásd: `ls -l /usr/share/texmf/tex/latex/psnfss/*.sty`).

Betűméret megváltoztatása. A `\documentclass` opcionális paraméterében megadható az alapértelmezett betű mérete. Lehetséges értékek `10pt`, `11pt` és `12pt` (alapértelmezés: `10pt`).

Médium (papír) megváltoztatása. Szintén opcionális paraméterrel állítható, lehetséges értékei: `letterpaper`, `legalpaper`, `executivepaper`, `a4paper`, `a5paper` és `b5paper` (alapértelmezés: `letter`, ezért írtuk be az `a4paper`-t mindig). A `landscape` opció elforgatja a lapot 90° -kal.

Betűk alakja. Három jellemző: alak, vastagság, és család.

Alakjukat tekintve a betűk lehetnek állók (`upright`, `\textup{}`, vagy `{\up }`), döntöttek (`slanted`, `\textsl{}`, vagy `{\sl }`), kurzívak, vagy dőltek (`italics`, `\textit{}`, vagy `{\it }`) és kiskapitálisok (`small caps`, `\textsc{}`, vagy `{\sc }`)

Vastagságukat tekintve lehetnek normál (`medium`, `\textmd{}`, vagy `{\md }`), illetve félkövér (`boldface`, `\textbf{}`, vagy `{\bf }`) típusúak.

Végül családjukat tekintve lehetnek antikva (`roman`, `\textrm{}`, vagy `{\rm }`), groteszk (`sans serif`, `\textsf{}`, vagy `{\sf }`) és írógép (`typewriter`, `\texttt{}`, vagy `{\tt }`) családba tartozók.

A betűk alakját, vastagságát és családját változtató parancsok egymásba ágyazhatók.

Betűk mérete. Ha szövegen belül változtatni akarunk a betűk méretén a `{\tiny }`, `{\scriptsize }`, `{\footnotesize }`, `{\small }`, `{\normalsize }`, `{\large }`, `{\Large }`, `{\LARGE }`, `{\huge }` és `{\Huge }` parancsokat használhatjuk.

Betűközök mérete a `letterspace` csomag betöltése után a `\mbox{\letterspace to hossz{szöveg}}` paranccsal lehetséges. `\mbox{\letterspace to 0.75\naturalwidth{PÉLDA}}` eredménye:
PÉLDA

Sorközök távolsága a `doublespace` csomag `spacing` környezetével változtatható, ahogy az alábbi példa mutatja:

```
\usepackage{doublespace}
...
\begin{spacing}{1.5} ...másfeles sorközű szöveg... \end{spacing}
```

Itt ér véget a mai gyakorlat.

A hallgatók házi feladata a következő:

Ez már a nagy házi feladat! Egy tetszőleges saját, vagy internetről letöltött szöveget kell formázni, amely az alábbi követelményeket teljesíti: van benne folyó szöveg, van benne legalább egy táblázat, van benne legalább egy kép/ábra, van benne legalább egy képlet és tartalmaz irodalomjegyzéket, illetve azokra hivatkozást. A házi feladatban a bruttó (utasításokkal együtt számolt) karakterszám 25 000–30 000 legyen.

A házi feladatokat a <http://mcl.hu/~kgergely/hf> címen található feltöltő scripttel kell beadni. A sikeres feltöltésről a megadott e-mail címre értesítést küld a script. A feladat leadásának határideje a 12. oktatási hét péntek, reggel 8:00.

Kilencedik laborgyakorlat
Vízszintes és függőleges térközök

Hosszméreték. \LaTeX -ben létezik rugalmatlan és rugalmas hossz méret. Az előbbi egyszerűen `szám mértékegység` alakban (egybeírva!) kell megadni. A használható mértékegységek a 127. fólián láthatóak.

A rugalmas méreteket `szám1[me1] plusz szám2[me2]`, vagy `szám1[me1] plusz szám2[me2] minusz szám3[me3]` alakban lehet megadni, ahol `[me x]` egy mértékegység. Az előbbi esetben a hossz `szám1[me1]` és `szám1[me1]+szám2[me2]` közötti tetszőleges értéket vehet fel, az utóbbi esetben az intervallum határai `szám1[me1]+szám2[me2]` és `szám1[me1]-szám3[me3]`.

A `\setlength{\hp}{hossz}` paranccsal állíthatjuk be egy már létező `\hp` hosszúságparancs méretét. A `\addtolength{\hp}{hossz}` utasítással hossz mértékben növelhetjük `\hp` méretét. A `\the\hp` utasítás írja ki a `\hp` aktuális értékét. A `\newlength{\hp}` hoz létre egy új `\hp` hosszúságot.

A `\settowidth{\hp}{szöveg}` a szöveg szélességére állítja `\hp` értékét, a `\settoheight{\hp}{szöveg}` utasítás szöveg alapvonalától – a sor alapvonalától – mért távolságot adja vissza, míg a `\settodepth{\hp}{szöveg}` az alapvonalától mért mélységre állítja `\hp` értékét.

A `\hspace{hossz}` és `\vspace{hossz}` parancsokkal vízszintes és függőleges térközt lehet kihagyni. Csillagos változataik sor-, illetve oldaltörés esetén is kihagyják a megfelelő méretű helyet. Példa:

```
\newlength{\gyalu}
```

```
A ,,gyulai gyalu'' szélessége \settowidth{\gyalu}{gyulai gyalu}
\the\gyalu\ (ide ,, \hspace*{\gyalu}'' pont beférne), magassága
\settoheight{\gyalu}{gyulai gyalu} \the\gyalu, mélysége pedig
\settodepth{\gyalu}{gyulai gyalu} \the\gyalu.
```

Végtelenül rugalmas térközök. A `\vfill` és a `\hfill` utasításokkal illeszthetünk be vízszintes és függőleges végtelenül rugalmas térközt a dokumentumunkba. Ha például külön sorba írjuk, hogy bal széle `\hfill` jobb széle, akkor ezt láthatjuk:

bal széle

jobb széle

hiszen a `\hfill` rugalmasan kitöltötte a teret. A `\hfill` parancs ekvivalens a `\hspace{0pt plus 1fill}` paranccsal.

Léteznek még `\hrulefill` és `\dotfill` parancsok is, melyekkel végtelenül rugalmas vízszintes vonalat, illetve pontsorozatot lehet beilleszteni. Aláírandó dokumentumokat ezért célszerű ilyesformán befejezni:

```
\vfill Budapest, 2002.\ november 19. \hfill \dotfill
```

Ha több `\hfill`, vagy `\vfill` követi egymást, akkor azok száma aránylik a távolságokhoz. Ha nem akarunk sokat ismételni, akkor a `\stretch{szám}` mértéket is alkalmazhatjuk a `\hspace{}`, vagy `\vspace{}` argumentumaként:

```
eleje \hspace{\stretch{4}} közepe \hspace{\stretch{1}}  
vége \\ ugyanaz, mint az \\  
eleje \hfill\hfill\hfill\hfill közepe \hfill vége \\  
és az \\ eleje \hspace{0pt plus 4fill} közepe  
\hspace{0pt plus 1fill} vége \\ ugye?
```


Mértékegységek összehasonlító táblázata

jele	jelentése	A méret nagysága más egységben							
		pt	pc	dd	cc	mm	in	bp	sp
pt	pont	1	0,0833	0,9346	0,07788	0,3515	0,0138	0,996	65536
pc	pica	12	1	11,215	0,93457	4,2175	0,166	11,955	786432
dd	Didôt pont	1,07	0,0892	1	0,08333	0,3761	0,01481	1,066	70124
cc	cicero	12,84	1,07	12	1	4,5128	0,17767	12,792	841489
mm	miliméter	2,845	0,2371	2,6591	0,22159	1	0,03937	2,835	186468
in	hüvelyk	72,27	6,0225	67,5415	5,62846	25,4	1	72	4736287
bp	nagy pont	1,004	0,0836	0,938	0,07817	0,3528	0,01389	1	65782
cm	centiméter	28,45	2,371	26,591	2,2159	10	0,3937	28,346	1864680
sp	skálázott pont (1/65536 pt)								
ex	az aktuálisan használt betűtípusban „x” magassága, most éppen 7.22783pt								
em	az aktuálisan használt betűtípusban „M” szélessége, most éppen 13.82pt								
mu	matematikai mértékegység, csak matek módban létezik! (1/18 em)								
fill	végtelenül rugalmas mértékegység								

Címoldalt a `titlepage` környezettel definiálhatunk. A címoldal tartalmát a \LaTeX azonnal megjeleníti, az oldalszámlálót pedig 1-re állítja. A címoldal környezetben akár több oldalnyi információt is megjeleníthetünk. Példa:

```
\begin{titlepage} { \center
\resizebox{11cm}{!}{\includegraphics{kisbme.eps}} \\
{\large \bf Budapest University of Technology and Economics}
\\ \vfill {\Large Valami jópofa cím kell ide} \\[20pt]
{\Large Nomeg egy szerző} \vfill {\bf Konzulens}: név \vfill
{\Large Budapest, 2002.} \clearpage \mbox{} \vfill \noindent
{\large Mégegyszer a szerző: cím}} \\[10pt] Copyright
\copyright 2002. All rights reserved ... \\[12pt] This
document was typeset in \LaTeXe. \vfill \end{titlepage}
```

Számlálók

A számlálók egész számok, melyeknek nincs mértékegysége. Sok beépített számlálót használtunk eddig is – bár nem vettük őket észre –, pl. a `section` számláló tárolja a fejezetszámot, a `table` számláló pedig a legutóbbi táblázat sorszámát.

A `\newcounter{szml}[ős]` paranccsal hozhatunk létre egy `szml` számlálót, melynek őse `ős` lesz. Ha az `ős` értékét léptetjük (lásd később), az `szml` kinullázódik. A `\setcounter{szml}{szám}` utasítással írhatjuk felül egy számláló értékét. Az `\addtocounter{szml}{szám}` segítségével egy tetszőleges szám-ot adhatunk `szml`-hez. A `\theszml` kiírja `szml` aktuális értékét, értékátadáshoz pedig a `\value{szml}` utasítást kell használni. Példa:

```
\newcounter{szaml} \setcounter{szaml}{\value{page}} \theszaml
```

Beépített számlálók a $\text{\LaTeX} 2_{\epsilon}$ -ben

neve	értéke
part	rész sorszáma
chapter	fejezet sorszáma
section	szakasz sorszáma
subsection	alszakasz sorszáma
subsubsection	al-alszakasz sorszáma
paragraph	paragrafus sorszáma
subparagraph	alparagrafus sorszáma
page	oldalszám
equation	egyenlet sorszáma
figure	ábra sorszáma
table	táblázat sorszáma
numi	lista sorszáma az első szinten
numii	lista sorszáma a második szinten
numiii	lista sorszáma a harmadik szinten
numiv	lista sorszáma a negyedik szinten
footnote	lábjegyzet sorszáma
secnumdepth	fejezet számozás „mélysége”
tocdepth	a tartalomjegyzék „mélysége”

Számlálót a `\stepcounter{szml}`, vagy a `\refstepcounter{szml}` parancsokkal lehet léptetni. Az utóbbi annyival tesz többet, hogy a legközelebbi `\label` parancs `szml` értékét fogja hozzárendelni a címkéhez.

A `\theszml` utasítás automatikusan jön létre, ha definiáljuk a `szml` számlálót. A számlálók értékét azonban többféle módon is kiirathatjuk. Ha nem tetszik a kapott `\theszml` utasítás, akkor át kell definiálni.

```
\newcounter{szaml}  
\renewcommand*{\theszaml}{\Roman{szaml}}
```

Számlálók megjelenítési lehetőségei $\text{\LaTeX} 2_{\epsilon}$ -ben

<code>\arabic{szml}</code>	arab számokkal	1, 2, 3...
<code>\roman{szml}</code>	kisbetűs római számokkal	i, ii, iii...
<code>\Roman{szml}</code>	nagybetűs római számokkal	I, II, III...
<code>\Alph{szml}</code>	nagybetűkkel	A, B, C...
<code>\alph{szml}</code>	kisbetűkkel	a, b, c...
<code>\fnsymbol{szml}</code>	lábjegyzet szimbólumokkal	*, †, ‡...

Itt ér véget a mai gyakorlat.

A hallgatók házi feladata a következő:

Ez már a nagy házi feladat! Egy tetszőleges saját, vagy internetről letöltött szöveget kell formázni, amely az alábbi követelményeket teljesíti: van benne folyó szöveg, van benne legalább egy táblázat, van benne legalább egy kép/ábra, van benne legalább egy képlet és tartalmaz irodalomjegyzéket, illetve azokra hivatkozást. A házi feladatban a bruttó (utasításokkal együtt számolt) karakterszám 25 000–30 000 legyen.

A házi feladatokat a <http://mcl.hu/~kgergely/hf> címen található feltöltő scripttel kell beadni. A sikeres feltöltésről a megadott e-mail címre értesítést küld a script. A feladat leadásának határideje a 12. oktatási hét péntek, reggel 8:00.

Az oktató házi feladata a következő:

A szedéstükör és a listák formai elemeit tartalmazó, valamint a pifont csomag szimbólumait tartalmazó fénymásolatok elkészítése, szétosztása.

Tizedik laborgyakorlat

Listák formázása

Felsorolások (`itemize` környezet). Négy mélységűek lehetnek. Az első szinten `\labelitemi`, a másodikon `\labelitemii`, a harmadikon `\labelitemiii`, a negyediken `\labelitemiv` adja meg a használt felsorolásjel kinézetét. Mindegyikük a `\renewcommand` paranccsal átdefiniálható. Az `\item` opcionális paraméterével pedig egyetlen helyen módosíthatjuk azt.

```
\usepackage{pifont}
...
\renewcommand{\labelitemi}{\ding{49}}
\begin{itemize}
\item Jópofa ez a töltőtoll a sor elején\dotsc
\item[$\circ$] ide most üres karika kellett,
\item {[}de akkor sem kell zavarba esni, ha
,,['' az első karakter]. \end{itemize}
```

Leíró listák (`description` környezet). Hat mélységűek lehetnek. Az `\item[szöveg]` parancs segítségével a szöveg címkével jelenik meg egy elem. Ha más kinézetre törekszünk a `\descriptionlabel` paranccsal módosíthatjuk a címkék megjelenését, de ne felejtsük el a szükséges térközt kihagyni (`\labelsep`)! Példa:

```
\begin{description}
\item[Rock'n'roll] a 60-as években kialakult zenei irányzat.
Meghatározó képviselői közé tartozik...
\end{description}
\renewcommand*{\descriptionlabel}[1]{\hspace{\labelsep} \sc #1:}
\begin{description}
\item[Rock'n'roll] a 60-as években kialakult zenei irányzat.
Meghatározó képviselői közé tartozik...
\end{description}
```

Számozott listák (`enumerate` környezet). Négy mélységig ágyazhatóak egymásba. A négy szintre – a felsorolásokhoz hasonlóan – `i`, `ii`, `iii`, `iv` ragokkal hivatkozunk. `enumi` az első szint számlálója, a már ismert módon, `\theenumi` utasítással irathatjuk ki értékét. A `\labelenumi` parancs mondja meg, hogy hogyan kell megjelennie a számlálónak a címkében. Példa:

Azt mondják a magyar nyelvű művek szedésekor antikva szöveg elé kurzív betűket kell írni felsorolásban. Hát legyen:

```
\renewcommand*{\theenumi}{\alph{enumi}}
\renewcommand*{\labelenumi}{\it \theenumi)}
\begin{enumerate} \item Az első szinten még arab számokat
láthatunk, \begin{enumerate} \item de a másodikon már
megjelennek a dőlt betűk.
\item A nyomdászok biztosan jobban tudják, hogy miért
\end{enumerate} \item esztétikusabb így. \end{enumerate}
```

A `\p@enumii` a második szintű listaelemre történő hivatkozás esetén a hivatkozás alakját állítja be. Az „@” jelet tartalmazó parancsok ún. belső L^AT_EX parancsok, melyek átdefiniálásához egy speciális üzemmódba kell váltani, amely a `\makeatletter` és `\makeatother` parancsokkal lehetséges. Példa:

```
\renewcommand*{\theenumi}{\arabic{enumi}}
\renewcommand*{\labelenumi}{\theenumi.\sim\S}
\renewcommand*{\theenumii}{\roman{enumii}}
\renewcommand*{\labelenumii}{\theenumi--\theenumii.}
\makeatletter \renewcommand*{\p@enumii}{\theenumi--}
\makeatother \begin{enumerate} \item Az első paragrafus
\begin{enumerate} \item az első paragrafus első bekezdése
\item az első paragrafus második bekezdése \label{fontos}
\end{enumerate} \item második paragrafus. \end{enumerate}
Lásd \ref{fontos}. Jó, mi?
```

Az `enumerate` csomag. Segítségével a \LaTeX kitalálja, hogy hogyan is akarjuk szervezni a listákat. az `1`, `i`, `I` a, `A` karaktereket keresi az `enumerate` környezet opcionális argumentumában: ha talál, akkor a számozás a talált értéknek megfelelően `arabic`, `roman`, `Roman`, `alph`, vagy `Alph` lesz. Ha több ilyen is van, akkor a nem kívánatosakat kapcsos zárójelbe kell rakni, hogy elkerüljük a félreértést. Példa:

```
\usepackage{enumerate}
...
\begin{enumerate}[Most Kovácsék {az} I.~emeleten {laknak},]
\item noha zavarja őket a másodikon lakó.
\item noha zavarja őket a harmadikon lakó.
\item noha zavarja őket a negyediken lakó.
\end{enumerate}
```

A `list` környezet. $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ alatt a `list` környezetből származtatják az eddig megismert listákat, de még a korábban tanult `quote`, `quotation` és `verse` környezeteknek is a `list` az alapja. Két paramétere van, az első az alapértelmezett címke, a második a beállításokhoz használható (lásd a mellékelt lapot). Példa:

```
\newenvironment*{idezet}
  {\begin{list}{}{\setlength{\rightmargin}{\leftmargin}}%
  \item[, , \ignorespaces] {\unskip''\end{list}}
...
```

Egy Közgázon oktató magas beosztású személy a következőt állította munkahelyéről: `\begin{idezet}`

Mi itt a Közgázon mindenhez értünk, csak a gazdálkodáshoz és a szervezéshez nem.

```
\end{idezet} No comment.
```


Egy másik példa:

```
\newenvironment*{Bevásárlólista}
  {\begin{list}{\$ \Rightarrow$ \hfill} % ha rövidebb lenne,
   \setlength{\rightmargin}{0pt} % mint a hely
   \setlength{\leftmargin}{\parindent} % igazodva a
   \setlength{\itemindent}{\parindent} % környezethez
   \setlength{\listparindent}{\parindent}
   \setlength{\labelwidth}{.5\parindent}
   \setlength{\labelsep}{.5\parindent}
   \setlength{\topsep}{1ex plus 0.5ex} % az x méretéhez
   \setlength{\partopsep}{0pt} % igazodva
   \setlength{\parsep}{0.5ex plus 0.25ex minus 0.25ex}
   \setlength{\itemsep}{0.2ex plus 0.1ex minus 0.1ex}}
{\item[] A bontott csirkét el ne felejtse!\end{list}}
```

Dokumentumok felszabdalása,
egyéb file-ok csatolása

Hosszabb művek felszabdalása. Ha nagy művet írunk, áttekinthetőbb a dokumentum, ha külön file-okban tároljuk az egyes fejezeteket. A dokumentumok felszabdalását az `\include{}` parancs támogatja. Argumentumában a file nevét kell megadni, akár a `.tex` kiterjesztés nélkül. Az `\includeonly{}` paranccsal megmondhatjuk a \LaTeX -nek, hogy mely file-okat akarjuk valóban betölteni (akkor hasznos, ha egy file-on dolgozunk és a többi fordítására nem akarunk időt pazarolni).

Az `\include{}` parancs hatására új oldal kezdődik (mivel alapvetően `chapter`-ök beillesztésére tervezték). Ha ez nem tetszik, akkor az `\input{}` felhasználásával oldaltörés nélkül is beilleszthetünk file-okat.

```
\documentclass[a4paper,10pt]{report} ... \begin{document}
... \includeonly{bevezeto} % most csak ezen dolgozunk
\include{bevezeto} \include{modell.tex} ...
\include{befejezes} ... \end{document}
```

Egyéb file-ok csatolása a \LaTeX forráshoz. Több file-t is lehet egyetlen \LaTeX forrásba fűzni. A `\documentclass` utasítás előtt a `filecontents`, környezet jelöli a csatolt file-t. Egyetlen argumentuma a file nevét tartalmazza, a környezetbe pedig a file tartalmát kell írni. Akár több file-t is csatolhatunk a módszerrel.

```
\begin{filecontents}{maki.eps}
%!PS-Adobe-2.0 EPSF-1.2
%%Title: Maki majom ...
\end{filecontents}
\begin{filecontents}{magyar.ldf}
\ProvidesFile{magyar.ldf}
  [1996/12/23 v1.3h Magyar support from the babel system] ...
\end{filecontents}
\documentclass[a4paper,10pt]{report} ...
```

Interaktivitás. A \LaTeX -ben létezik egy `\typein[\prncs]{szöveg}` parancs, mellyel a terminálról olvashatunk be. Egyetlen kötelező argumentumában a terminálra küldött `szöveget` adhatjuk meg. Ha nem adjuk meg az opcionális paraméterét, akkor a begépelte szöveg az `\includeonly` paraméterét határozza meg. Ellenkező esetben az opcionális paraméterként megadott `\prncs` parancsot deklaráljuk. Hasonlóképpen a `\typeout{szöveg}` utasítással írhatunk a terminálra.

...

```
\typein[\elado]{Kérem az eladó nevét!}
\typein[\vevo]{Kérem a vevő nevét!}
{\center Adásvételi szerződés \[1cm]}
Mely köttetett \today\ egyrésztől {\it \elado}, mint Eladó --
továbbiakban Eladó -- másrésztől {\it \vevo}, mint Vevő --
továbbiakban Vevő -- között ...
```

Itt ér véget a mai gyakorlat.

A hallgatók házi feladata a következő:

Ez már a nagy házi feladat! Egy tetszőleges saját, vagy internetről letöltött szöveget kell formázni, amely az alábbi követelményeket teljesíti: van benne folyó szöveg, van benne legalább egy táblázat, van benne legalább egy kép/ábra, van benne legalább egy képlet és tartalmaz irodalomjegyzéket, illetve azokra hivatkozást. A házi feladatban a bruttó (utasításokkal együtt számolt) karakterszám 25 000–30 000 legyen.

A házi feladatokat a <http://mcl.hu/~kgergely/hf> címen található feltöltő scripttel kell beadni. A sikeres feltöltésről a megadott e-mail címre értesítést küld a script. A feladat leadásának határideje a 12. oktatási hét péntek, reggel 8:00.

Az oktató házi feladata a következő:

A névjegykártya forrását sokszorosítani és szétosztani.

Tizenegyedik laborgyakorlat
Dobozok, minioldalak

LR dobozok. Ha egy dobozban balról jobbra egyetlen sort írhatunk, akkor LR (Left to Right) doboznak hívjuk. Már korábban is volt szerencsénk a legegyszerűbb dobozhoz, amelyet az `\mbox{szöveg}` utasítással hívhatunk elő. Az `\fbox{szöveg}` hatása szintén doboz, ám ekkor egy keret is megjelenik a szöveg körül.

A keretezett dobozban az `\fboxrule` tárolja a keret vonalvastagságát, az `\fboxsep` pedig a keret távolságát a szövegtől. Példa:

```
\mbox{sima doboz keret nélkül}
\fbox{sima keretezett doboz}
{\setlength{\fboxrule}{3pt} \fbox{vastag keretes doboz}}
{\setlength{\fboxsep}{0pt} \fbox{térközmentes keretes doboz}}
```


Egyel bonyolultabb lehetőséget kínál az `\makebox[szél][poz]{szöveg}` és a `\framebox[szél][poz]{szöveg}` utasítás, ahol `szél` a doboz szélessége. A szélesség megadható tetszőleges hosszúságmértékben, vagy használható a `\width`, `\height`, `\depth`, vagy `\totalheight` mértékek, melyek a `szöveg` szélességét, magasságát, mélységét és teljes magasságát (magasság + mélység) adják vissza.

A `poz` paraméterrel a `szöveg` pozícióját adhatjuk meg a dobozban. Értéke lehet `l` (left), `r` (right), vagy `s` (stretch–shrunk). Az alábbi példában egy keretezett dobozt láthatunk, melynek szélessége a beírt szöveg szélességének duplája, és a szöveg a doboz közepére kerül.

A keretezetlen `\makebox[2\width][s]{\hfill{ }Helló világ!\hfill}` verzió persze csak akkor látszik, ha van körülötte szöveg.

```
\framebox[2\width][s]{\hfill{ }Helló világ!\hfill}
```

A `\raisebox{emel}[magasság][mélység]{szöveg}` az `emel` paraméter értékével (lehet negatív is!) emeli meg a szöveget. Az opcionális paraméterekkel állíthatjuk, hogy mennyi legyen a virtuális magassága, illetve mélysége a doboznak. Akkor lehet erre szükség, ha nem szeretnénk a sorok távolságát változtatni.

A `\shabox{szöveg}` parancs a `shadow` csomag betöltésével hívható elő. Hatására egy árnyékolt keretes doboz jelenik meg. Stílusparamétereit az `\sboxrule` vonalvastagság, az `\sboxsep` térköz és az `\sdim` árnyékvastagság átállításával módosíthatjuk.

Ide kerül az `\mbox{első}` sor. `\\`

Ide kerül a `\raisebox{2ex}{második}` sor. `\\`

Ide kerül a `\raisebox{2ex}[0pt]{harmadik}` sor. `\\`

Jópofa, hogy éppen `\shabox{így}` írhatunk árnyékolt dobozt.

A `fancybox` csomag további kereteket tartalmaz. A `\shadowbox{szöveg}` hasonló az előbb megismert árnyékolt dobozhoz, de itt az árnyék alja kerül egy vonalba az alapvonallal. Az `\ovalbox{szöveg}` és `\Ovalbox{szöveg}` lekerekített sarkú dobozok, különböző vastagsággal. A `\cornersize{r}` paranccsal változtathatunk a lekerekítés sugarán. A `\doublebox{szöveg}` dupla keretet tesz a `szöveg` köré, ahol a belső keret vastagsága $3/4\text{\fboxrule}$, a külsőé $3/2\text{\fboxrule}$. A `fancybox` csomagnak további parancsai is vannak, lásd a dokumentációját.

```
\usepackage{shadow}
\usepackage{fancybox}
...
\shabox{Ugyanaz}, vagy \shadowbox{nem ugyanaz?}
\doublebox{Hej, de csinos.}
\ovalbox{Ó, de jópofa.} \cornersize{1}
\Ovalbox{Ez meg egészen kerekded.}
```

Bekezdésdobozok. A `\parbox[poz][mag][szpoz]{szélesség}{szöveg}` utasítással több bekezdésnyi szöveget is egy külön dobozba helyezhetünk. A `minipage` környezet is ugyanezekkel az argumentumokkal hívható meg, és hatása is azonos, azonban a `minipage` környezetben lehetnek felsorolások, táblázatok, lábjegyzetek, míg az előbbiben nem. A `poz` paraméter határozza meg a bekezdésdoboz igazítását az alapvonalhoz. Lehetséges értékei: első sorhoz `t` (top), vagy utolsó sorhoz `b` (bottom). Ha elmarad, akkor középre igazít. A `mag` a szövegdoboz magassága, a `szpoz` a szöveg pozíciója a dobozban. Lehetséges értékei: lap tetejére `t` (top), lap aljára `b` (bottom), középre `c` (center) és húzd szét `s` (stretch–shrunk).

```
\hrulefill\parbox{25mm}{25\,mm széles doboz sok szöveggel.}\hrulefill%
\parbox[t]{25mm}{ez is 25\,mm széles doboz, máshol.}\hrulefill%
\parbox[b]{25mm}{ez is 25\,mm széles doboz, ismét máshol.}\hrulefill%
```

Dobozok egymásba is ágyazhatók gond nélkül. Akár nagyobb dobozt rakhatunk egy kisebbbe (néha lehet rá szükség), vagy egyszerűen bármit bekeretezhetünk az `\fbox{}` parancs segítségével. Az alábbi példa egy névjegykártya készítéséhez adhat segítséget.

```
\usepackage{pifont}  
...  
\fbox{\parbox[t][50mm]{90mm}{\vfill \raggedleft  
\hrulefill \shabox{\scriptsize Kovács Alajos} \\  
\tiny nyugdíjas főtörzsőrmester \\[6pt]  
\ding{38} +36-1-234-5678 \\  
\ding{41} alajos@kovacs.hu \vfill}}
```

Vonaldobozok. A `\rule[emelés]{szél}{mag}` paranccsal `szél` szélességű és `mag` magasságú kitöltött téglalapot rajzolhatunk. Az opcionális paraméterrel megemelhetjük, vagy süllyeszthetjük a téglalapot. A téglalap lehet nulla vastagságú is, amit gyámfának nevezünk. A gyámfázás a nyomdászatban gyakran használt (lásd az első példát). Léteznek beépített gyámfák, melyek a sorvastagság méretét tárolják, ezek a `\strut` szöveg és `\mathstrut` matematikai módban.

```
\fbox{\rule{0pt}{1cm}ez a doboz legalább 1\,cm magas}
\fbox{ez} \fbox{egy} \fbox{egyenetlen} \fbox{dobozsor}
\fbox{\strut ez} \fbox{\strut már} \fbox{\strut egyenletes}
\fbox{\strut dobozsor} ...  $(\sqrt{g} + \sqrt{h})$  csúnya, de
 $\left(\sqrt{\mathstrut g} + \sqrt{\mathstrut h}\right)$  szép.
```

Dobozok tárolása. Dobozok szövegrészletek tárolására is alkalmazhatók. A `\newsavebox{\prncs}` utasítással hozhatunk létre egy új, `\prncs` nevű dobozt, melyet háromféleképpen is feltölthetünk. Az `\sbox{\prncs}{szöveg}` utasítást már korábban is használtuk (versidézetek szedésénél), a `\savebox{\prncs}[szél][poz]{szöveg}` annyival több, hogy az opcionális paraméterekkel a doboz szélessége és a dobozban lévő szöveg elhelyezése is befolyásolható. Az `lrbox` környezet szintén az argumentumban megadott `\prncs` doboz feltöltésére alkalmazható (új környezetek definiálásakor lehet hasznos). A feltöltött dobozok tartalmának kiiratása a `\usebox{\prncs}` utasítással történik.

A mellékelt papíron egy példa látható, melyben lementett szövegdobozokkal egy A4-es lap két oldalára tíz kétoldalas névjegykártyát nyomtatunk. Vegyük észre a rugalmas méretek és a dobozok alkalmazását!

Táblázatok formázása

Matematikai módban a `\begin{array}[poz]{oszl}` környezettel, szöveges módban a `\begin{tabular}[poz]{oszl}` környezettel hozhatunk létre tömböket. Az `oszl` paramétert már korábban megismertük, a `poz` paraméter a táblázat illesztésének módját adja meg. Ha `t` (top), akkor a táblázat első sora kerül az alapvonalra, ha `b` (bottom), akkor az alsó sor. Ha nincs megadva ez az opcionális paraméter, akkor a táblázat közepét igazítjuk az alapvonalhoz.

```
\hrulefill \begin{tabular}[t]{c} 1 \\ 2 \\ 3 \\ 4 \end{tabular}%  
\hrulefill \begin{tabular}{c} 1 \\ 2 \\ 3 \\ 4 \end{tabular}%  
\hrulefill \begin{tabular}[b]{c} 1 \\ 2 \\ 3 \\ 4 \end{tabular}%  
\hrulefill \\
```

Automatikus sortörés táblázatban. Ha megadjuk egy táblázat oszlopdefiníciójában, hogy az oszlop adott szélességű legyen, akkor gyakran csúnya a kimenet, mivel a sorkizárt szedés miatt széles szóközök kerülnek a szavak közé. Megoldást jelenthet az alábbi kód, melyhez az `array` csomagot be kell tölteni.

```
\usepackage{array}
...
\newcommand*{\tabsovege}[1]{\let\tmp=#1\let\=#1\tmp}
\begin{tabular}{c|>{\tabsovege{\raggedright}\mbox{}}p{25mm}}
\hline 1 & Az első sorban is látszik milyen hasznos kis
segítség ez. \\ \hline 2 & Hát még a második sorban, ahol
szintén egy hosszú szöveget láthatunk. \\ \hline \end{tabular}
```

A `tabularx` csomaggal illeszthetünk be adott szélességű táblázatot. Használatához a `\begin{tabularx}{szél}[poz]{osz1}` környezetet kell meghívni. Az első argumentuma a táblázat szélességét határozza meg. Hasznos lehet a `\linewidth` méret alkalmazása, amely az aktuális vonalszélességet adja vissza. A következő paraméter opcionális; a korábban megismert `poz` pozícionáló paraméterrel ekvivalens. Az utolsó oszlopban az oszlopdeklarációk szerepelnek, azonban itt megjelenik egy speciális oszlop: a rugalmas oszlopot az `X` definíció jelöli.

```
\begin{tabularx}{\linewidth}{c|X} Akkora, amekkora kell  
& ez meg a maradékot kapja meg mindenképpen akármit is  
írok, legyen az akár nagyon hosszú szöveg \\ Rövidebb  
szöveg & második sor. \end{tabularx}
```

Ha több x típusú oszlop is van, akkor azok azonos szélességűek lesznek. Ha ez nem tetszik, akkor az oszlopdefiníciókban módosíthatjuk az x típusú oszlopok szélességeinek arányát. Az alábbi példa remélhetőleg megvilágítja a használatát:

```
\begin{tabularx}{\linewidth}
  {c|>{\hspace=0.5\hspace}X|c|>{\hspace=1.5\hspace}X}
1 & megérett a meggy & 2 & csipkebokor vessző \\
3 & te leszel a párom & ... \end{tabularx}
```

Az oszlopdefinícióban szereplő két szám összege meg kell, hogy egyezzen az x oszlopok számával. Jelen esetben az oszlopok szélességének aránya: $0.5 / 1.5 = 1 / 3$, azaz a negyedik oszlop szélessége háromszorosa a másodikénak.

További paraméterek az `array` csomaggal. A `tabular` környezet kötelező argumentumában az oszlopdefiníciók kiegészítődnek az alábbiakkal:

l balra zárt oszlop

r jobbra zárt oszlop

c középre zárt oszlop

p{szél} sorkizárt oszlop `szél` szélességgel, az első sor az alapvonalon

m{szél} sorkizárt oszlop `szél` szélességgel, de a sorok közepe kerül az alapvonalra

b{szél} sorkizárt oszlop `szél` szélességgel, de a sorok alja kerül az alapvonalra

>**{dekl}** az adott oszlop elemei előtt minden sorban a `dekl` deklarációt végrehajtja

<**{dekl}** az adott oszlop elemei után minden sorban a `dekl` deklarációt végrehajtja

| függőleges vonal

@**{szöveg}** az oszlopokat elválasztó vonal felülírása térköz lenyeléssel

!**{szöveg}** az oszlopokat elválasztó vonal felülírása térköz lenyelése nélkül

***{n}{oszl}** Több (n darab) azonos oszlopdefinícióval rendelkező oszlop(ok) ismétlése

Más programok kimenetének gyors táblázatba foglalása. A `dcolumn` csomag betöltése után használható az oszlopdefiníciókban a `D{isep}{osep}{jegy}` paraméter, amely megmondja a \LaTeX -nek, hogy mivel jelöltük a tizedespontot (`isep`), mivel szeretnénk helyettesíteni (`osep`) és hány digitet akarunk kiírni (`jegy`).

```
\usepackage{dcolumn}
...
\begin{tabular}{|>{\$}r<{\$}!{\$=\$}D{.}{,}{3}|}
\hline % most jön a Matlab kimenete
1^2 & 1 \\
\sqrt{2} & 1.142 \\
\sqrt{100} & 10 \\
\hline
\end{tabular}
```

Hosszú táblázatok. Ha táblázatunk egy oldalnál hosszabb, akkor a `longtable` csomagot kell betöltenünk. A `longtable` környezettel illeszthetünk be egy oldalnál hosszabb táblázatokat. A csomagnak – és a környezetnek – sok opciója van, tessék utánanézni <http://www.cmis.csiro.au/Graham.Williams/TeX/docs/longtable.pdf>.

Színes táblázatok. A `colortbl` csomaggal a táblázat oszlopai, sorai, cellái kiszínezhetőek. Dokumentáció szintén a weben: <http://www.cmis.csiro.au/Graham.Williams/TeX/docs/colortbl.pdf>

Vonalak cifrázása. A `\hhline` csomaggal lehetséges a táblázatok dupla vonalait „szépíteni”. Dokumentáció szintén a weben: <http://www.ctan.org/tex-archive/macros/latex/required/tools/hhline.dtx>

Stílusparaméterek. Az alábbi hosszméreteket lehet átírni a már ismert `\setlength` parancs segítségével. Az `\arraycolsep` és a `\tabcolsep` az oszlopok közötti távolság fele `array`, illetve `tabular` környezet használata esetén. Az `\arrayrulewidth` paranccsal a táblázatban lévő vonalak vastagsága szabályozható. A `\doublerulesep` kettőzött vonalak esetében a vonalak közötti távolság.

A sorok közötti távolságot az `\arraystretch` paraméter határozza meg. Dimenzió nélküli szám, megváltoztatása a `\renewcommand` paranccsal lehetséges. Alapértéke 1.

Tizenkettedik laborgyakorlat

Úszó objektumok formázása

A `figure` és `table` környezettel hozhatunk létre úszó objektumot. Ha csillagos verzióikat használjuk, akkor két hasábos szedés esetén a táblázat, vagy az ábra mindkét hasábot elfoglalhatja. Hasznos nagy táblázatok, ábrák beillesztésekor.

A `\caption[rövid]{cím}` utasításnak adhatunk egy `rövid` opcionális paramétert is, melynek jelentése: a táblázatok, vagy ábrák listájában a `rövid` címet írd ki. A táblázatok és ábrák jegyzéke a `\listoftables` és a `\listoffigures` utasításokkal iratható ki.

A `wrapfigure` csomag segítségével elérhetjük, hogy minden ábra a beillesztés helyén jelenjen meg.

Ábra, vagy táblázat körbefolytatása szöveggel. A `floatflt` csomagot kell betölteni, majd a `\begin{floatingfigure}{szél}` környezetbe kell az ábrát illeszteni, pl. a korábbról megismert `\includegraphics{}` utasítással. A `floatingfigure` környezet paraméterében határozhatjuk meg, hogy vízszintesen mekkora helyet foglaljunk az ábrának. Az ábra elé és mögé írt szöveg körbefolyja az ábrát. Hasonlóan használható a `floatingtable` környezet.

```
\usepackage{floatflt}
```

```
...
```

```
\begin{floatingfigure}{.5\linewidth} \center
```

```
\resizebox{.45\linewidth}{!}{\includegraphics{maki.eps}}
```

```
\caption{Majom Úr} \label{fig:maki} \end{floatingfigure}
```

A majmok nagyon sok tulajdonságukban emlékeztetnek az emberre...

A `subfigure` csomag betöltése után az ábrákon belül alábrákat is létrehozhatunk a `\subfigure[részcím]{részábra}` utasítással. A részcím csak a részábra alatt jelenik meg. Ha van `\caption`, akkor az egész ábrát kitölti.

```
\usepackage{subfigure}
...
\begin{figure} \center \mbox{%
\subfigure[Az állat]{\includegraphics{maki.eps}} \qqquad %
\subfigure[A ketrec]{\includegraphics{bme.eps}} }
\caption{Beiratnám jó eszű maki egyedemet,
hogy végezze el e Műszaki Egyetemed (kecskerím)}
\end{figure}
```

Az `\includegraphics` stílusparaméterei. Ha a `graphics` csomag van betöltve, akkor eléggé szegényesen paraméterezhető: két opcionális paraméterében megmondhatjuk, hogy milyen bal alsó és jobb felső koordináták közötti képrészletet akarunk beilleszteni. Ha csillaggal használjuk, akkor csak a képrészlet kerül be a dokumentumba, csillag nélkül a teljes kép (de csak a kivágottnak foglal le helyet)

Ha a `graphicx` csomagot használjuk a `graphics` helyett, akkor sokkal több lehetőségünk van. A dokumentációért lásd a 9–12. oldalakat:

<http://www.ctan.org/tex-archive/macros/latex/required/graphics/grfguide.ps>

```
Itt jön egy makifej \resizebox{2cm}{!}{\rotatebox{-90}{
\includegraphics*[0,200][400,600]{maki.eps}}}
de most berakja a teljes képet is, csak csacska módon
\includegraphics[0,200][400,600]{maki.eps}
szöveg most például már nem is látszik.
```

Csomagok és osztályok

Magunk is írhatunk osztályokat és csomagokat. Csomag írására akkor lehet szükség, ha van néhány utasításunk, melyeket gyakran használunk, függetlenül a dokumentumosztálytól. Saját dokumentumosztályra akkor lehet szükségünk, ha egy új osztályt akarunk definiálni, melyben többen is dolgozni fognak. Az osztály definiálás hosszadalmas és alapos munka, ráadásul sok olyan dolgot is ismerni kellene hozzá, amiről nem esett szó (lásd a példa fejezetcím formázó utasítását). Ezért nem foglalkozunk új osztályok definiálásával, részletekért lásd <http://www.tex.ac.uk/tex-archive/macros/latex/doc/clsguide.ps> és <ftp://cam.ctan.org/tex-archive/macros/latex/base/classes.dtx>.

Csomagok esetében az első sorba a `\ProvidesFile` utasítást kell írni, melynek első paramétere a csomag neve, majd opcionálisan megadhatjuk a csomag készítésének dátumát, verziószámot és szöveges ismertetőt. A csomag használatakor hivatkozhatunk e paraméterekre:

```
\usepackage[csomag opciók]{csomag}[dátum]
```


A jenedy.sty file tartalma:

```
\ProvidesFile{jenedy.sty}[2002/12/03 v1.0 Probacsomag]
\setcounter{\secnumdepth}{5} % öt mélységig számozza a fejezeteket
\setcounter{\tocdepth}{4} % a fejezetcímek négy mélységig kerülnek
% be a tartalomjegyzékbe

\makeatletter
\renewcommand\section{\@startsection {section}{1}{\z@}%
    {-3.5ex \@plus -1ex \@minus -.2ex}%
    {2.3ex \@plus.2ex}{\normalfont\tiny\bfseries}}
\makeatother
\newcommand*\JG{\textsf{Jenedy Gábor}}
\newenvironment*{idezet}
    {\begin{list}{}{\setlength{\rightmargin}{\leftmargin}}%
    \item[,,, \ignorespaces] {\unskip'' \end{list}}
    ...
```

További formai elemek

Lábjegyzetek. A korábban megismert `\footnote{szöveg}` paranccsal lehetséges. Opcionális argumentumával `\footnote[szám]{szöveg}` megadhatjuk a használni kívánt számlálóértéket. Ekkor a beépített `footnote` számláló értéke nem változik. A `\footnote{}` utasítás előtt nem szabad szóközt hagyni, lehet az szó végén, vagy írásjel után. A `footnote` számláló értékét a már ismert `\thefootnote` utasítással lehet kiírni. Ha nem tetszik a lábjegyzet formája, az utóbbit kell felülírni.

```
\renewcommand*{\thefootnote}{\roman{footnote}}
Mindenk\footnote{az első lábjegyzet}
másképp\footnote[20]{hát ez meg hogy lett ilyen nagy?!}
Pedig még csak \thefootnote\ az értéke!!!}
csinálja\footnote{kettes számú lábjegyzet}%
\dots\footnote[1]{Vigyázat, becsapós!}
```

A `\footnotemark[szám]` paranccsal egy lábjegyzetjelet illeszthetünk be a folyó szövegbe. Ha nem használjuk az opcionális argumentumot, akkor a sorban következő jelet teszi ki és lépteti a `footnote` számlálót eggyel. Ha használjuk az opcionális paramétert, akkor `számnak` megfelelő lábjegyzetjelet teszi ki és nem bántja a `footnote` számlálót

A `\footnotetext[szám]{szöveg}` a lábjegyzetbe írja `szám` jellel a szöveget, anélkül, hogy a szövegekörnyezetbe jelet tenne. Ha nem adjuk meg az opcionális argumentumot, akkor az aktuális számlálóértéket írja ki, és a számláló értékét nem változtatja.

```
Mindenki\footnotemark\footnotetext{Közös lábjegyzet.}
másképp\footnotemark[\value{footnote}]
csinálja\footnotemark[\value{footnote}]\dots
```

A `\footnote{}` parancs nem használható sem dobozokban (`minipage` környezetben a környezet aljára kerülnek a lábjegyzetek, nem a lap aljára), sem táblázatokban. Ha ilyen környezetekben szeretnénk `\footnote` utasítást normál módon használni, akkor gányolni kell a `\footnotemark` és a `\footnotetext` utasításokkal. Közben persze ügyesen trükközni kell a `footnote` számláló értékével.

Ez a megoldás `\footnote{Megoldás?!} is \fbox{jó%`
`\footnotemark[101]}, \footnotetext[101]{Csak nem szép.}`
bár idétlenül mutat `\footnote{Nocsak.}` az ugrás a számlálóban.
A jó megoldás sem sokkal bonyolultabb, lássuk hát `\fbox{%`
`\strut{}keretezett\footnotemark\ formában\footnotemark}!`
`\addtocounter{footnote}{-1}\footnotetext{Keretezetlenül?}`
`\stepcounter{footnote}\footnotetext{Vagy formátlanul?}`

Lábjegyzetek formai paraméterei. Állítható a lábjegyzetek közötti távolság (`\footnotesep`), a lábjegyzetek és a szöveg közötti vonal vastagsága (`\footnoterule`), melynek 0pt vastagnak kell lennie (lásd a lenti példát). A `\skip\footins` méret változtatásával a szöveg és a lábjegyzet első sora közötti távolság állítható be. További stílusmódosítási lehetőségekért lásd a `footnote` csomagot.

```
\setlength{\footnotesep}{1.5cm} % biztosan túl sok.  
\setlength{\footnoterule}{\rule{\linewidth}{.3pt}\vspace{-.3pt}}  
% a vonalvastagságot minuszba is kihagyjuk, hogy össz 0pt legyen  
\setlength{\skip\footins}{1cm}  
Mindenki\footnote{az első lábjegyzet}  
csinálja\footnote{kettes számú lábjegyzet}%  
\dots\footnote{hármasszámú lábjegyzet!}
```

Széljegyzetek. A `\marginpar{szöveg}` utasítással írhatjuk a szöveget a margóra. Alapértelmezésként mindig a kötés átellenes oldalára kerül a szöveg. Ha ez nem tetszik, akkor `\reversemarginpar` utasítással a másik oldalra kényszeríthetjük a margófeliratot. Vissza a `\normalmarginpar` utasítással léphetünk.

A széljegyzetek formai megjelenését a `\marginparwidth`, a `\marginparsep` és a `\marginparpush` paraméterekkel állíthatjuk. Az első a széljegyzetek szélességét szabályozza, a második a széljegyzet és a szedéstükör távolságát, az utolsó két egymást követő széljegyzet közötti minimális függőleges távolságot állítja be.

A `\marginpar[bal szöveg]{jobb szöveg}` utasításnak létezik egy opcionális paramétere is, mellyel a páros oldalakra eltérő szöveget írhatunk (bal szöveg), mint a páratlan oldalakra (jobb szöveg).

Élőfej, élőláb. A `\pagestyle{sty}` utasítással a dokumentum egészét, a `\thispagestyle{sty}` utasítással pedig az adott lap stílusát határozhatjuk meg. A `sty` stílus lehet `empty`, (nincs fejléc és lábléc), `plain`, (nincs fejléc, a láblécben középen az oldalszám), `headings`, (a lábléc üres, a fejlécben az oldalszám kívül, belül a rész cím és fejezetcímek).

A `fancyhdr` csomag betöltésével és a `fancy` stílus megadásával a fejléc és lábléc tartalma tetszőlegesen befolyásolható. A `\fancyhead[hely]{mit}` és `\fancyfoot[hely]{mit}` parancsokkal írhatunk a fejlécbe és láblécbe. A `hely` argumentum lehet `L` (left), `R` (right), `C` (center), `E` (even – páros oldalon), `O` (odd – páratlan oldalon). Egy utasításon belül több hely is megadható, ezeket vesszővel kell elválasztani. Az alábbi kód a lapok külső oldalára fejlécbe írja az oldalszámot.

```
\usepackage{fancyhdr}
\pagestyle{fancy} \fancyhead[LE,RO]{\thepage} \fancyfoot{}
```


Jegyzékek. A tartalomjegyzék beillesztése a `\tableofcontents` paranccsal lehetséges, az ábrák jegyzéke a `\listoffigures`, míg a táblázatoké a `\listoftables` utasítással iratható ki.

A jegyzékek automatikusan generálódnak fordítás során, de lehetséges beléjük írni is. Ha szeretnénk formázó utasítást írni az egyik jegyzékbe, akkor az `\addtocontents{hova}{mit}` utasítást kell használnunk, ahol a `hova` paraméter a jegyzék típusa (`toc`, `lof`, vagy `lot`). Ha egy elemet akarunk beilleszteni, akkor az `\addcontentsline{hova}{típus}{mit}` utasítást írjuk, ahol a `típus` mondja meg a beillesztett mező típusát (tartalomjegyzék esetében `part`, `chapter`, `section` stb. lehet, a másik két esetben csak `figure`, illetve `table`). A megjelenő oldalszám az utasítás kiadásának helyét mutatja.

További olvasmányok
Konvertálás más formátumokba

A legjobb magyar könyv a témában: Wettl Ferenc, Mayer Gyula, Sudár Csaba, *L^AT_EX kezdőknek és haladóknak*, Panem, 1998, Budapest

A legjobb könyv a témában: Leslie Lamport, *L^AT_EX; A Document Preparation System* Addison-Wesley, 1994.

Aki a T_EX rejtelseit is szeretné felfedezni: Donald E. Knuth, *The T_EXbook*, Addison-Wesley, 1984.

Aki a T_EX-et magyarul szeretné felfedezni: Bujdosó Gyöngyi, Fazekas Attila, *T_EX kezdőlépések*, Tertia, 1997, Budapest

Ha valakinek nincs pénze könyvekre, akkor az Interneten is rengeteg jó forrást fedezhet fel. A „T_EX frequently asked questions” weboldal: <http://www.tex.ac.uk/tex-archive/macros/latex/doc/>

L_AT_EX navigator (és encyclopedia): <http://tex.loria.fr/>

A T_EX-hel és L_AT_EX-hel kapcsolatos anyagok gyűjtőhelye (Comprehensive T_EX Archive Network – CTAN): <ftp://ftp.dante.de>, vagy <ftp://helka.iif.hu/mirrors/ctan/>

A L_AT_EX hivatalos honlapja: <http://www.tex.ac.uk/CTAN/latex/>

A magyar L_AT_EX honlap címe: <http://www.math.bme.hu/latex/>

A `dvi1j`, `dvi1j4` stb. konverterekkel HP LaserJet alatt nyomtatható formátumot állíthatunk elő a DVI kimenetünkből.

A `dvips` alkalmazással PostScript (PS) kimenetet állíthatunk elő. A UNIX világban a PS a kiindulási alap, PostScriptből bármibe (akár mozgó GIF-be is!) `gs`-sel (GhostScript) alakíthatjuk át file-jainkat. A `gs` dokumentációját a `man gs` paranccsal hívhatjuk elő.

Lehetőségünk van közvetlenül PDF formátumba is konvertálni. Erre a `dvipdf`, alkalmazás szolgál, mely DVI-ből készít PDF-et. A `pdflatex` alkalmazással a \LaTeX forrásunkat közvetlenül PDF-be fordíthatjuk, de vigyázat, nem a \LaTeX3 csoport fejleszti, ezért a portabilitás nem garantált és lehetnek benne hibák. Ettől függetlenül sok kellemes szolgáltatása van a `pdflatex`-nek, például az `\includegraphics` parancsnak közvetlenül GIF és JPG file-okat is átadhatunk.

Egyéb konverterek is léteznek, melyek nem részei a $\text{\LaTeX} 2_{\epsilon}$ disztribúciónak. A teljesség igénye nélkül ilyen például az `rtf2latex`, amely RTF file-okat tud \LaTeX forrásba átírni, a `latex2rtf`, amely ugyanezt csinálja csak fordítva, vagy a `latex2html`, amely HTML kimenetet állít elő a \LaTeX forrás alapján. Mivel az alkalmazásokat felsorolni is lehetetlen, a webes keresőket ajánlom további információért és bővebb dokumentációért.

<http://www.google.com>

VÉGE

Ennyi fért a félév anyagába, talán hasznos is volt. Remélem sikerült megismerni a $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ világot és alkalmatok nyílik később használni az itt tanultakat.

A tárgy anyagával kapcsolatban várom észrevételeiteket (mit csináltál volna másképp, mi hiányzott, mi volt sok stb.) a jeneyg@hit.bme.hu e-mail címre. Ha a későbbiekben lenne kérdésed a $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -hel kapcsolatban, akkor is kereshetsz ezen a címen.

Tartalomjegyzék

Nulladik laborgyakorlat

Adminisztratív és egyéb kérdések 1

Első laborgyakorlat

A \LaTeX használatának alapjai 15

Második laborgyakorlat

A szöveg begépelése 26

Harmadik laborgyakorlat

Környezetek, a dokumentum struktúrálása	37
Környezetek	41
Felsorolások, számozott, számozatlan listák	44
Idézetek, versek környezetei	48
Tárgymutató (Index)	51

Negyedik laborgyakorlat

Irodalomjegyzék 54

Hivatkozás egyéb objektumokra 59

Új utasítások és környezetek létrehozása 63

Ötödik laborgyakorlat

Táblázatok 70

Képek, ábrák beillesztése 81

Hatodik laborgyakorlat

A matematikai szedés 86

Hetedik laborgyakorlat

Matematikai szedés II 97

Nyolcadik laborgyakorlat

Tételek a szövegben 108

A szöveg formázása 112

Kilencedik laborgyakorlat

Vízszintes és függőleges térközök 122

Számlálók 129

Tizedik laborgyakorlat

Listák formázása 135

Dokumentumok felszabdalása, egyéb file-ok csatolása 143

Tizenegyedik laborgyakorlat

Dobozok, minioldalak 148

Táblázatok formázása 157

Tizenkettedik laborgyakorlat

Úszó objektumok formázása 166

Csomagok és osztályok 171

További formai elemek 174

További olvasmányok, konvertálás más formátumokba 182

Vége 187