



Tómacs Tibor

Rövid útmutató a

L^AT_EX

használatához

Tómacs Tibor: \LaTeX
című könyvének rövidített változata
kezdő felhasználók részére,
önálló tanuláshoz

EGER, 2020. NOVEMBER 21. 11:46

Tartalomjegyzék

Előszó	5
1. Bevezetés	6
1.1. A \LaTeX jellemzői	6
1.2. Telepítés Windowsra	6
2. Az első lépések	8
2.1. Alapfogalmak	8
2.1.1. Parancs	8
2.1.2. Kötelező argumentum	8
2.1.3. Opcionális argumentum	8
2.1.4. Környezet	9
2.1.5. Blokk	9
2.1.6. Deklarációs parancs	9
2.1.7. Komment	9
2.2. A forrásfájl szerkezete	10
2.2.1. Fontosabb standard dokumentumosztályok	10
2.2.2. Csomag	10
2.3. Az első dokumentum elkészítése	11
3. Alapvető formai elemek	14
3.1. Karakterek	14
3.1.1. Foglalt karakterek	14
3.1.2. Ékezetes betűk	14
3.1.3. Különleges karakterek	15
3.1.4. Szóközök	15
3.1.5. Írásjelek	16
3.2. Betűváltozatok	17
3.2.1. Osztályozás	17
3.2.2. Kiemelés	18
3.3. Betűméretek	18
3.4. Törések	18
3.4.1. A szavak elválasztása	18
3.4.2. Sortörések	18
3.4.3. Oldaltörések	19
3.5. Függőleges térközök	19
3.6. Bekezdések	19
3.6.1. Bekezdések balra zárása	19
3.6.2. Bekezdések jobbra zárása	20

3.6.3.	Bekezdések középre zárása	20
3.7.	Lábjegyzetek	20
3.8.	Színek	20
3.9.	Internetcímek	21
3.10.	Számírás	21
3.11.	Mértékegységek	22
3.12.	Kereszthivatkozások	22
4.	Oldalak kinézete	25
4.1.	Oldalak szerkezete és méretei	25
4.2.	Többhasábos szedés	26
5.	Listák	27
5.1.	Számozatlan listák	27
5.1.1.	Számozatlan listák extra térközök nélkül	27
5.2.	Leíró listák	28
5.3.	Számozott listák	29
5.3.1.	Számozott listák extra térközök nélkül	29
6.	Képek	31
7.	Táblázatok	33
8.	Objektumok úsztatása	35
8.1.	Képek és táblázatok úsztatása	35
8.2.	Úsztatott objektumok címkézése	36
9.	Verbatim, programkódok	37
9.1.	Verbatim	37
9.2.	Programkódok	38
10.	Képletek	40
10.1.	Matematikai mód	40
10.2.	Műveleti jelek	41
10.3.	Relációjelek	41
10.4.	Közönséges matematikai jelek	42
10.5.	Indexek, gyökjelek, törtek, binomiális együtthatók	42
10.6.	Matematikai ékezetek	42
10.7.	Görög betűk	42
10.8.	Kalligrafikus és dupla szárú betűk	43
10.9.	Három pont	43
10.10.	Matematikai zárójelek	43
10.11.	Esetek szétválasztása	44
10.12.	Matematikai jelek több szerepben	44
10.13.	Változó hosszúságú vízszintes jelek	45
10.14.	Mátrixok	45
10.15.	Operátorok, függvények	46
10.15.1.	Nagy operátorok	46
10.15.2.	„Nolimits” függvények	46

10.15.3. „Limits” függvények	46
10.15.4. Új függvények definiálása	47
10.15.5. Differenciálás, differencia operátor	47
10.16. Kiemelt képletek sorszámozása	48
10.17. Több képlet egymás alatt	48
10.18. Több képlet egymás alatt illesztéssel	49
11. Strukturált művek	50
11.1. Főcím, címlap, kivonat	50
11.2. A főszöveg szintjei	50
11.3. Fej- és láblécek	51
11.3.1. Alapbeállítások	51
11.3.2. Fej- és láblécek testreszabása	52
11.4. Tartalomjegyzék	54
11.5. Tételszerű bekezdések	54
11.6. Bibliográfia	56
12. Elektronikus publikáció	58
13. Szakdolgozat készítése	60
14. Gyakorlatok	62

Előszó

Ebben a jegyzetben az Eszterházy Károly Egyetem „*Számítógépes szöveg- és kiadványszerkesztés*” című gyakorlatának tananyaga található, amely a \LaTeX magas szintű dokumentumleíró nyelv világába vezeti be az Olvasót.

Ezzel a rendszerrel 1990-ben ismerkedtem meg. Azóta számos tananyagot, könyvet és cikket szerkesztettem vele. Több folyóirat technikai szerkesztőjeként rengeteg szerzők által elkövetett hibával találkozom, melyek számomra az oktatásban fontosak, hiszen így jobban látom, hogy mire kell a \LaTeX tanításában nagyobb hangsúlyt fektetni. Ezt igyekeztem kamatoztatni ebben a leírásban is.

Akik az itt a leírtaktól többet szeretnének tudni, azoknak a következő könyvet ajánlom:

Tómacs Tibor: \LaTeX

A gyakorlati jegy megszerzéséhez a következő feladatot kell megoldani: [klikk ide!](#)

Reményeim szerint a kurzus elvégzése után az Olvasó természetesnek veszi majd, hogy szakdolgozatának vagy bármely más jellegű publikációjának, dokumentumának elkészítéséhez \LaTeX -rendszert használ. Ha észrevétele, megjegyzése van, kérem írjon a tomacs.tibor@uni-eszterhazy.hu címre.

A \LaTeX -rendszert használók többmilliós táborának jelmondatával kívánok az Olvasónak sok türelmet és kitartó munkát a tanuláshoz!



Dr. Tómacs Tibor
egyetemi docens

1. fejezet

Bevezetés

DONALD ERVIN KNUTH 1977-ben egy olyan számítógépes programot fejlesztett ki, amely a nyomdászat minden tudását képes modellezni. A programot a görög *τέχνη* szó (jelentése: művészet, mesterség; kiejtése: *techné*) első három betűjéből T_EX-nek keresztelte el. Így a kiejtése nem „teksz”, hanem „tekh”, mint a *technika* szóban.

A T_EX nevezhető a nyomdászat assemblerének is, mellyel minden tipográfiai feladat megoldható. Ezzel azonban csak fáradságos úton, elemi parancsok használatával tudunk dolgozni. Ezért szükség volt olyan makrócsomag létrehozására, amely magasabb szintű programozási nyelven, jóval könnyebben kezelhető. Egy ilyen makrócsomag a L^AT_EX, amely egy magas szintű általános dokumentumleíró nyelvnek tekinthető.

1.1. A L^AT_EX jellemzői

A dokumentum L^AT_EX-forrása egy szöveges állomány, mely együtt tartalmazza a kiadvány szövegét és a L^AT_EX parancsait. Így a szerkesztés során nem azt látja, amit a végén lefordítva pdf fájlban kap. Ez a kezdő felhasználónak hátrány, de a gyakorlat megszerzése után már előnyként fogja élvezni, mert ezáltal vizuális szerkesztésre nincs szükség, csak a tartalomra kell figyelni.

A L^AT_EX ingyenes és nyílt forráskódú program, melynek segítségével professzionális tipográfia érhető el, beleértve a matematikai képleteket is. Az irodalomjegyzékek, tartalomjegyzékek, szójegyzékek, lábjegyzetek és keresztshivatkozások automatikusan számozódnak, így állandó utólagos javítgatásokra nincs szükség.

A mai programok közül a L^AT_EX tudja a bekezdéseket a leghatékonyabban tördelni. Minden operációs rendszeren hozzáférhető, továbbá egy rendszeren megírt mű egy másik rendszeren is ugyanazt az eredményt adja, nincs áttördelési effektus. Nagy terjedelmű dokumentum forrása és az eredményt jelentő pdf fájl is csekély méretű, így internetes publikálásra ideális.

1.2. Telepítés Windowsra

A L^AT_EX minden géptípuson és operációs rendszeren hozzáférhető. Az egyik ilyen disztribúció, a Linuxon és Windowson is egyaránt működő *TeX Live*. Ennek telepítéséhez töltsse le a telepítésvezérlőjét ([klikk ide](#)), futtassa, majd kövesse az utasításokat. A telepítés akkor fejeződik be sikeresen, ha megjelenik a „*Welcome to TeX Live!*” felirat.

A szerkesztett dokumentum forrása egy szöveges állomány, amit bármely editoron létrehozhat. Azonban sokkal könnyebb a szerkesztés, ha olyan editoron dolgozik, amely a \LaTeX -re lett optimalizálva. Számos ilyen létezik, de talán a *TeXstudio* a legjobb. Ennek telepítéséhez töltsse le a telepítőjét ([klikk ide](#)), indítsa el a telepítő fájlt, majd kövesse az utasításokat.



Videó: \TeX -rendszer telepítése Windowsra

2. fejezet

Az első lépések

Az első dokumentum létrehozása előtt (lásd a 2.3. szakaszban) célszerű néhány olyan fogalmat megismerni, melyek segítségével ez érthetőbb lesz.

2.1. Alapfogalmak

2.1.1. Parancs

A \LaTeX -ben a dokumentum formázása parancsokkal történik. A parancs \backslash (backslash) jellel kezdődik, majd ezt követi a parancs neve, melyben ékezetes betű, szám és szóköz nem szerepelhet, továbbá kis- és nagybetű között különbséget tesz. Például a

```
 $\backslash$ Large
```

parancs az utána írt szöveget a normál betűméretnél nagyobbbal szedi ki.

2.1.2. Kötelező argumentum

Vannak olyan parancsok, amelyek csak bizonyos *paraméterek* megadásával működnek. Ezeket a paramétereket a parancs *argumentumába* kell beírni $\{ \}$ jelek közé. Például a

```
 $\backslash$ textit{szöveg}
```

a „szöveg” szót dőlten szedi ki. Egy parancsnak több paramétere is lehet. Például

```
 $\backslash$ setcounter{page}{1}
```

az oldalszámot 1-re állítja.

2.1.3. Opcionális argumentum

Egy parancsnak lehet *opciója* is, amit nem kötelező megadni. Ha nem adja meg, akkor az *alapopció* lép érvénybe. Az opciókat a parancs *opcionális argumentumában* kell megadni $[]$ jelek között. Például egy listaelem bevezethető az

```
 $\backslash$ item
```

parancssal, ami az alapértelmezett jelet teszi ki a listaelem elé, de írhat

```
 $\backslash$ item[-]
```

parancsot is, amely egy kötőjelet tesz a listaelem elé. Előfordulhat, hogy egy parancsnak opciója és paramétere is van. Például az

```
 $\backslash$ includegraphics[width=3cm]{abra.jpg}
```


parancs betölti az `abra.jpg` képet 3 cm szélességben. Valamikor több opció is megadható. Ekkor az opciókat vesszővel kell elválasztani. Például

```
\includegraphics[width=3cm,angle=90]{abra.jpg}
```

parancs betölti az `abra.jpg` képet 3 cm szélességben 90 fokkal elforgatva.

2.1.4. Környezet

A `\begin`, `\end` parancspárt *környezetnek* nevezzük, a kettő közötti rész pedig a *környezet belseje*. Ezen parancsok argumentumában kell a környezet nevét megadni. Például `itemize` környezet alatt a `\begin{itemize}`, `\end{itemize}` parancspárt értjük, ami számozatlan listát készít:

```
\begin{itemize}
  \item Listaelem
  \item Listaelem
\end{itemize}
```

2.1.5. Blokk

Vannak olyan parancsok, melyek az utánuk lévő részre valamilyen hatást fejtenek ki. Például az `\itshape` parancs a soron következő szöveget dőlten szedi ki. Ha azt akarja, hogy csak egy adott részre terjedjen ki a hatása, akkor *blokkba* kell zárni. Blokk kapcsos zárójelekkel adható meg. Például

Ez egy `{\itshape nem túl izgalmas}` példa.

esetben csak a „nem túl izgalmas” lesz kisedve dőlten. Blokkot határoz meg egy környezet is. Például

```
\begin{itemize}
  \itshape
  \item Listaelem
\end{itemize}
```

esetén az `\itshape` csak az `itemize` környezetben belül hat.

2.1.6. Deklarációs parancs

Ha egy parancs önmagában nem jelenít meg semmit, nincs se kötelező se opcionális argumentuma, ugyanakkor az utána található részre hatással van, akkor azt *deklarációs parancsnak* nevezzük. Ilyen például az előbb említett `\itshape` parancs is. A deklarációs parancsok az utánuk következő szöveget elnyelik. Például

```
AAA\itshape BBB
```

eredménye

AAABBB

2.1.7. Komment

Ha a forrásállományba ún. *kommentet* akar elhelyezni, vagyis amit a \LaTeX -fordító figyelmen kívül hagy, akkor azon szöveg elejére írjon `%` jelet. A komment vége sortörés. Például:

```
% Ez a szöveg nem jelenik meg fordítás után!  
Ez megjelenik, % de ez megint nem!
```

Ez megjelenik,

2.2. A forrásfájl szerkezete

A \LaTeX forrásfájl szerkezete a következő séma szerint épül fel:

```
\documentclass[opciók]{dokumentumosztály}  
<preambulum>  
\begin{document}  
<dokumentumtest>  
\end{document}
```

Elsőként egy *dokumentumosztályt* kell betölteni a `\documentclass` paranccsal, ami a dokumentum alapstílusát határozza meg. Például az `article` dokumentumosztályt `12pt` opcióval így kell betölteni:

```
\documentclass[12pt]{article}
```

Az ezt követő részt a `\begin{document}` parancsig *preambulumnak* nevezzük. Ide kerülhetnek azok a parancsok, melyek az egész dokumentumra hatással vannak, de megjelenítendő szöveget nem tartalmazhat. A `\begin{document}` és `\end{document}` közötti részt *dokumentumtestnek* nevezzük, mely minden megjelenítendő szöveget és parancsot tartalmaz. Az `\end{document}` után írt szöveget a \LaTeX figyelmen kívül hagyja.

2.2.1. Fontosabb standard dokumentumosztályok

article Előadások, meghívók, kisebb jelentések, programdokumentációk, publikációk stb. készítéséhez. Főbb opciói:

`10pt`, `11pt`, `12pt` A dokumentum alap betűmérete. Alapopció: `10pt`.

`a4paper`, `b5paper`, `letterpaper` Lapméret. Alapopció: `letterpaper`. Bármelyik méretet is választja, a fizikai lapméret minden esetben A4 lesz, amennyiben az alapbeállításokkal telepítette a TeX-rendszert. Ezek az opciók csak a kiválasztott lapméretnek megfelelő margókat állítják be. Ha fizikailag is be akarja állítani a lapméretet, akkor a `geometry` csomagot kell használnia (lásd a 4.1. szakaszt).

`oneside`, `twoside` Egy- illetve kétoldalas szedés. Alapopció: `oneside`.

report Beszámolók, értekezések, diplomamunkák készítéséhez használható. Az opciói ugyanazok, mint az `article` esetében. Alapértékek: `10pt`, `letterpaper`, `oneside`. A részek és fejezetek ebben az osztályban mindig új oldalon kezdődnek.

book Könyvek írásához. Opciói megegyeznek a `report` dokumentumosztályéval. Alapértékek: `10pt`, `letterpaper`, `twoside`.

2.2.2. Csomag

A dokumentumosztály képességeit, stílusát *csomagokkal* bővítheti. Ezeket a *preambulumban* kell betölteni a

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

paranccsal. Például

```
\usepackage[a5paper]{geometry}
```

az oldalt A5 méretre állítja. Ha nincs opció vagy alapopciókat használ, akkor a szögletes zárójelek nem kellenek. Például

```
\usepackage{listings}
```

esetén programkódokat tud megjeleníteni. Ha több opciót is betölt, akkor azokat vesszővel kell elválasztani. Például

```
\usepackage[paperwidth=105mm,paperheight=75mm]{geometry}
```

esetén az oldal szélessége 105 mm és az oldal magassága 75 mm lesz. Ha alapopciókkal több csomagot is betölt, akkor az a következő módon is megtehető:

```
\usepackage{<csomag1>,<csomag2>,<csomag3>}
```

Például

```
\usepackage{listings,fancyhdr}
```


betölti a listings és a fancyhdr csomagokat, amit így is meg lehetett volna tenni:

```
\usepackage{listings}
\usepackage{fancyhdr}
```




Ha egy parancs csak akkor használható, ha előtte betölt egy csomagot, akkor azt ebben a leírásban majd az \in jellel jelezzük. Például az `\euro` parancs az `eurosym` csomag betöltésével használható, amit így fogunk jelölni:

```
\euro  $\in$  eurosym
```


2.3. Az első dokumentum elkészítése

Nyissa meg a TeXstudiót és abban egy új dokumentumot: . Írja be a következőket:

```
\documentclass{article}
\begin{document}
Magyar nyelvű szöveg.
\end{document}
```

Ezt mentse el: . A megjelenő ablakban a mentés előtt hozzon létre egy új mappát az előzetesen kiválasztott helyen az  gombbal. Lépjen be a létrehozott mappába, majd a fájl nevének megadása után . Fontos, hogy ezt minden dokumentum esetén tegye meg, azaz minden dokumentum külön mappában legyen, ugyanis egy dokumentumhoz több fájl is fog tartozni.

Ügyeljen arra, hogy a szerkesztett szöveg UTF-8 kódolású legyen. A TeXstudio alapbeállítás esetén ilyen kódolást használ. Az éppen aktuális kódolás a TeXstudio jobb alsó sarkában olvasható le.

Fordítsa le az így elkészített forrásfájlt . Fordítás után megjelenik a pdf, melyen a „Jóízű félárú sütőtök” mondat látható 10 pt betűmérettel és a lap alján oldalszámozás van. A lap mérete A4 lesz, de a margók az angoloknál szabványos levélpapír mérethez lesznek igazítva.

Az `article` dokumentumosztálynak az alap betűméretre három opciója van (10pt, 11pt, 12pt), a lapméretre pedig többek között van egy `a4paper` opciója. Az előbb azért jelent meg a dokumentum 10 pt betűmérettel, mert a 10pt alapopció. Így ha át akar tér-

ni A4 lapméretnek megfelelő margókra és 12 pt betűméretre, akkor az 1. sort egészítse ki az alábbi módon:

```
\documentclass[a4paper,12pt]{article}
\begin{document}
Magyar nyelvű szöveg.
\end{document}
```

Ékezetes betű nem csak billentyűzetről vihető be, hanem parancsként is. Például, ha az „a” betűre egy vesszőt akar tenni ékezetként (azaz „á” betűt akar), akkor használhatja a `\'a` parancsot. Ezt a megoldást repülő ékezetnek nevezzük. Javítsa ki a 3. sort az alábbi módon:

```
\documentclass[a4paper,12pt]{article}
\begin{document}
Magyar nyelv\H{u} sz\{o}veg.
\end{document}
```

A \LaTeX úgy működik, hogy az ékezetes betűket fordítás előtt repülő ékezetekre konvertálja. Azonban a repülő ékezetek a sorok végén elválasztási hibákat eredményezhetnek bizonyos esetekben, ugyanis ezek két karakternek számítanak: alapbetű és a rátett ékezet. Ezt a problémát oldja meg az ún. belső kódolás megfelelő megválasztása. Ehhez a `fontenc` csomagot kell használni `T1` opcióval. Javítsa vissza a forráskódot az előző állapotba, majd töltsse be a `fontenc` csomagot a 2. sorban:

```
\documentclass[a4paper,12pt]{article}
\usepackage[T1]{fontenc}
\begin{document}
Magyar nyelvű szöveg.
\end{document}
```

Ez még mindig nem elég a helyes elválasztás beállításához, hiszen a \LaTeX nem tudja, hogy milyen nyelven írja a dokumentumot. Jelen esetben magyarul, amit a `babel` csomag `magyar` opciójával kell a forrásban közölni (lásd a 3. sorban):

```
\documentclass[a4paper,12pt]{article}
\usepackage[T1]{fontenc}
\usepackage[magyar]{babel}
\begin{document}
Magyar nyelvű szöveg.
\end{document}
```

A `babel` csomag `magyar` opciója betölti a `magyar.ldf` fájlt, amely a magyar tipográfia megvalósításáért felelős. A `magyar.ldf` első verziójához képest a ma használatos jóval nagyobb tudású, ami úgy van beállítva kompatibilitási okok miatt, hogy alapesetben a régivel legyen egyenértékű. Ahhoz, hogy az új elemek is érvényesülhessenek, a `babel` betöltése előtt át kell állítani a `magyar.ldf` alapbeállításait:

```
\documentclass[a4paper,12pt]{article}
\usepackage[T1]{fontenc}
\PassOptionsToPackage{defaults=hu-min}{magyar.ldf}
\usepackage[magyar]{babel}
\begin{document}
Magyar nyelvű szöveg.
\end{document}
```

Próbaképpen fordítsa le a forráskódot. Ezzel a kóddal tetszőleges magyar nyelvű szöveg kiszedhető.

Ezután próbálja ki a hibakezelést. Például a `\begin{document}` parancsot szándékosan javítsa át rosszra, mondjuk így: `\Begin{document}`. Ezután fordítsa le a forráskódot. Ekkor a \LaTeX -fordító egy hibaüzenetet küld, miszerint a `\Begin` parancs nincs definiálva:

Undefined control sequence. \Begin

Ezt a hibaüzenetet a TeXstudio is kiírja, és a hibás sorra ugrik. Ezután a hibás kódot javítsa vissza jóra. Ismét lefordítva, már nem kap hibaüzenetet.

A munka elején megnyitott mappában a tex és pdf fájlokon kívül néhány munkafájl is létrejött. Többek között egy log kiterjesztésű naplófájl is, ami a rosszul begépelte forráskódból származó hibákat is rögzíti. A munka végeztével ezeket érdemes törölni, amit TeXstudióból könnyen megtehet: [Eszközök](#) > [Segédfájlok törlése](#).



Videó: Az első \LaTeX -dokumentum készítése

A TeXstudio rengeteg kényelmi szolgáltatást biztosít, melyek segítségével sokkal gyorsabban állíthatja elő a \LaTeX -forrást. Ezeket itt nem tárgyaljuk, hiszen az újabb verziók kiadásával megváltozhatnak. Inkább célszerű önállóan felfedezni és gyakorolni a használatukat.

3. fejezet

Alapvető formai elemek

3.1. Karakterek

3.1.1. Foglalt karakterek

Vannak olyan billentyűzetről beírható karakterek, melyek közvetlenül nem jeleníthetők meg, mert a forrásállományban speciális jelentésük van:

<code>\</code>	parancsok ezzel kezdődnek
<code>%</code>	kommentek ezzel kezdődnek
<code>{ }</code>	blokkok, illetve parancsok argumentumainak határai
<code>\$</code>	matematikai mód határolója
<code>&</code>	táblázatoknál kell
<code>#</code>	változót tartalmazó parancs definiálásához kell
<code>_</code>	alsó index
<code>^</code>	felső index
<code>~</code>	törhetetlen szóköz

Ha ezeket meg akarja a pdf-ben jeleníteni, akkor a következő parancsokat használja:

<code>\</code>	<code>\textbackslash</code>	<code>&</code>	<code>\&</code>
<code>%</code>	<code>\%</code>	<code>#</code>	<code>\#</code>
<code>{</code>	<code>\{</code>	<code>_</code>	<code>_</code>
<code>}</code>	<code>\}</code>	<code>^</code>	<code>\textasciicircum</code>
<code>\$</code>	<code>\\$</code>	<code>~</code>	<code>\textasciitilde</code>

3.1.2. Ékezetes betűk

Korábban már volt róla szó, hogy a forrásállomány kódolásának beállítása után ékezetes betű közvetlenül a billentyűzetről is bevihető. De mi van akkor, ha olyan ékezetes betűre van szükségünk, amely nincs a billentyűzeten? Ilyenkor használhatja a repülő ékezeteket.

ó	<code>\' {o}</code>	ò	<code>\` {o}</code>	ō	<code>\= {o}</code>	ö	<code>\v {o}</code>	ø	<code>\k {o}</code>	ōo	<code>\t {oo}</code>
õ	<code>\H {o}</code>	ô	<code>\^ {o}</code>	ó	<code>\. {o}</code>	õ	<code>\r {o}</code>	ø	<code>\d {o}</code>		
ö	<code>\" {o}</code>	õ	<code>\~ {o}</code>	ő	<code>\u {o}</code>	ø	<code>\c {o}</code>	ø	<code>\b {o}</code>		

Az o betű helyére természetesen bármilyen más betű írható.

3.1.3. Különleges karakterek

€	<code>\euro</code> ∈ <code>eurosym</code>	*	<code>\textasteriskcentered</code>
£	<code>\pounds</code>	•	<code>\textbullet</code>
¢	<code>\textcent</code> ∈ <code>textcomp</code>	◦	<code>\textopenbullet</code> ∈ <code>textcomp</code>
®	<code>\textregistered</code> ∈ <code>textcomp</code>	‰	<code>\textperthousand</code> ∈ <code>textcomp</code>
©	<code>\textcopyright</code> ∈ <code>textcomp</code>	§	<code>\S</code>
†	<code>\dag</code>	№	<code>\textnumero</code> ∈ <code>textcomp</code>
‡	<code>\ddag</code>	※	<code>\textreferencemark</code> ∈ <code>textcomp</code>

3.1.4. Szóközök

Forrásállományban egy szóközt a Space billentyű lenyomásával tehet. Több szóköz egymás után a forrásállományban csak egy szóközt jelent a végeredményben, viszont a sor elején található szóköz a végeredményben nem jelenik meg. Szintén szóköznek számít a végeredményben, ha a forrásállományban sortörés van. Ez csak akkor nem igaz, ha a sor végén egy % jel van úgy, hogy közvetlenül előtte nincs szóköz. Például

```
Egy, kettő,      három,  
négy,  
öt, %  
hat.
```

```
Egy, kettő, három, négy, öt, hat.
```

Ha egy parancsnak nincs argumentuma, akkor általában az utána található szóközt nem jeleníti meg. Például

```
\LaTeX kézikönyv
```

```
LATEX kézikönyv
```

Ha ez nem kívánatos eredményt ad, mint itt is, akkor `_` paranccsal kikényszerítheti a szóközt (a `_` jel a szóközt jelenti):

```
\LaTeX\_ kézikönyv
```

```
LATEX kézikönyv
```

Van olyan eset is, amikor egy szóköz után nem szabad sort törni. Például ha azt írja, hogy IV. Béla, akkor a pont után nem lehet sortörés. Ennek érdekében a pont után ún. törhetetlen szóközt kell rakni. Forrásban `~` a törhetetlen szóköz jele:

```
IV.~Béla
```

Ezt érdemes megtenni minden olyan pont után, amikor az nem a mondat végét jelenti. Így az ilyen pontok nem kerülhetnek a sor végére. Vigyázat, ha már valahová tett törhetetlen szóközt, akkor utána ne tegyen még egy szóközt, mert az két szóközt eredményez, és a törhetetlenség is megszűnik:

```
IV.~ Béla (Így helytelen!)
```

```
IV. Béla (Így helytelen!)
```

A törhetetlen szóköznek van egy olyan változata is, ami a normál szóköz méretének

a fele. Ezt mértékszám és mértékegység között, illetve számok ezres csoportosításánál szoktuk használni. Forrásban \, a törhetetlen feles szóköz jele:

```
5\,cm, 14\,216\,123
```

```
5 cm, 14 216 123
```

3.1.5. Írásjelek

Kötőjel ♦ A kötőjel forrásállományban - jellel adható meg. Például

```
levegő-mintavétel; elő- vagy utótag; betűtípus és -méret;  
egy-két ember; 5-6 éves lehet; tudod-e;
```

```
levegő-mintavétel; elő- vagy utótag; betűtípus és -méret; egy-két ember; 5-6 éves  
lehet; tudod-e;
```

Nagykötőjel ♦ A nagykötőjel forrásállományban -- jellel adható meg. Például

```
lásd 15--21.-oldalakon; kelet--nyugati; az orosz TU--154 repülő;  
brazil--magyar meccs;
```

```
lásd 15-21. oldalakon; kelet-nyugati; az orosz TU-154 repülő; brazil-magyar meccs;
```

Gondolatjel ♦ A gondolatjel forrásállományban -- jellel adható meg. Gondolatjel előtt és után is szóköz áll, kivéve, ha írásjel követi. Például

```
Ilyen korán -- mondta Éva -- nem fogok felkelni!  
Sokszor vitatkoztak -- legtöbbször semmiségekért --, de szerették  
egymást.
```

```
Ilyen korán – mondta Éva – nem fogok felkelni! Sokszor vitatkoztak – legtöbbször  
semmiségekért –, de szerették egymást.
```

Hármaspont ♦ A hármaspont forrásállományban a

```
\dots
```

paranccsal adható meg. Ehelyett soha ne használjon három darab pontot egymás után írva. Például

```
\dots várom a párom \dots\ üres a polc\dots
```

```
... várom a párom ... üres a polc...
```

Idézőjel ♦ Idézőjelként soha ne használja a forrásban a " **Shift** + **2** jelet! Ez tipográfiai hiba. Az idézőjel nyelvenként változó. Magyar szöveg esetén a következőt kell tenni:

```
,,Idézett szöveg.,"
```


„Idézett szöveg.”

Tehát a nyitó idézőjel a forrásban két vessző, míg a záró idézőjel a forrásban **Shift** + **1** (aposztrófjel) kétszer egymás után. Angol szövegben ezt kell tenni:

```
``text text''
```

“text text”

Tehát a nyitó idézőjel a forrásban a fordított aposztrófjel **AltGr** + **7** kétszer egymás után, míg a záró idézőjel a forrásban **Shift** + **1** (aposztrófjel) kétszer egymás után.

3.2. Betűváltók

3.2.1. Osztályozás

A betűváltókat alakjuk, testességük és családjuk szerint osztályozzuk.

Alak (shape)

Álló (upright): `\textup{<szöveg>}`, `{\upshape <szöveg>}`
Döntött (slanted): `\textsl{<szöveg>}`, `{\slshape <szöveg>}`
Dőlt (italics): `\textit{<szöveg>}`, `{\itshape <szöveg>}`
Kiskapitális (small caps): `\textsc{<szöveg>}`, `{\scshape <szöveg>}`

Testesség (series)

Normál (medium): `\textmd{<szöveg>}`, `{\mdseries <szöveg>}`
Félkövér (boldface): `\textbf{<szöveg>}`, `{\bfseries <szöveg>}`

Család (family)

Antikva (roman): `\textrm{<szöveg>}`, `{\rmfamily <szöveg>}`
Groteszk (sans serif): `\textsf{<szöveg>}`, `{\sffamily <szöveg>}`
Írógép (typewriter): `\texttt{<szöveg>}`, `{\ttfamily <szöveg>}`

Az alak, testesség és család keverhetőek. Például

```
\textit{\textbf{\textsf{szöveg}}}
```

szöveg

A `\text`-el kezdődő parancsokat több bekezdésre nem lehet alkalmazni. Amikor nem alap betűváltót használ, de ideiglenesen vissza akar arra térni (álló, normál, antikva), akkor használja a

```
\textnormal{<szöveg>}, {\normalfont <szöveg>}
```

parancsokat. Az első több bekezdésre nem használható.

3.2.2. Kiemelés

Amikor egy szót, vagy gondolatot ki akar emelni, használja az

```
\emph{<szöveg>}, {\em <szöveg>}
```

parancsokat. (Az első megoldás több bekezdésre nem használható.) Ezek figyelik az aktuális betűváltozatot, és aszerint emelnek ki. Álló alak esetén dőlt, nem álló alak esetén álló alakra vált. Kiemelésre ne használjon félkövér típust, mert az a címekre van fenntartva.

3.3. Betűméretek

szöveg	<code>{\tiny szöveg}</code>	SZÖVEG	<code>{\Large szöveg}</code>
szöveg	<code>{\scriptsize szöveg}</code>	SZÖVEG	<code>{\LARGE szöveg}</code>
szöveg	<code>{\footnotesize szöveg}</code>	SZÖVEG	<code>{\huge szöveg}</code>
szöveg	<code>{\small szöveg}</code>	SZÖVEG	<code>{\Huge szöveg}</code>
szöveg	<code>{\normalsize szöveg}</code>		
SZÖVEG	<code>{\large szöveg}</code>		

3.4. Törések

3.4.1. A szavak elválasztása

A \LaTeX alaphól sorkizártan szedi a szöveget, így a sorvégi szavak elválasztása hosszabb szövegek esetén elkerülhetetlen. Amikor beállította a nyelvet, akkor a szavak nagy részét helyesen el tudja választani a program, de teljesen nem lehet automatizálni. Például a „karóra” szó esetében kétféle elválasztás is lehetséges, aszerint, hogy mit jelent: kar-ó-ra vagy ka-ró-ra. Ha azt tapasztalja, hogy egy adott helyen egy adott szót rosszul választ el, akkor alkalmazhatja az `\-` ún. *puha elválasztójelet*. Például

Már nem volt a szarkánál a kar`\-ó`-ra, mikor felrepült a ka`\-ró`-ra.

Ebben az esetben az adott szót csak a `\-` módon megjelölt helyeken lehet elválasztani. Ha egy szóban kötőjel van, akkor azt a \LaTeX csak a kötőjelnél tudja elválasztani. Ha ezt felül akarja bírálni, akkor a kötőjel elé gépeljen fordított aposztrófjelet:

egyszer```-kétszer

Ekkor a kötőjelnél és minden szótagnál el tud választani. Néha szükség lehet egy adott szó elválasztásának a tiltására is. Ekkor tegye azt az `\mbox` parancs argumentumába. Például

```
\mbox{karóra}
```

3.4.2. Sortörések

A \LaTeX automatikusan végzi a sortöréseket, de adott esetben ki is kényszerítheti. Sortörést soha ne alkalmazzon új bekezdés nyitásához!

```
\\
```

Új sort kezd sorkizárás nélkül.

```
\\[<hossz>]
```

Ugyanaz mint a `\\` de a következő sor távolsága *<hossz>* mérettel megnő. Például

```
\\[2mm]
```

3.4.3. Oldaltörések

A \LaTeX maga végzi az oldaltöréseket, de adott esetben ki is kényszerítheti.

```
\newpage
```

Új oldalt (illetve többhasábos szedésnél új hasábot) kezd. Az utolsó sort vízszintesen, azután pedig az oldalt (vagy hasábot) függőlegesen feltölti térközzel.

```
\clearpage
```

A `\newpage` parancstól annyiban különbözik, hogy többhasábos szedésnél is új oldalt kezd, másrészt az új oldal kezdése előtt megjeleníti az ún. úszó objektumokat (lásd később).

3.5. Függőleges térközök

Extra függőleges térköz a következő parancsokkal érhető el:

```
\smallskip
```

```
\medskip
```

```
\bigskip
```

A `\medskip` a `\smallskip` kétszeresével, míg a `\bigskip` a `\medskip` kétszeresével egyezik meg. Ezek a parancsok akkor működnek, ha a forrásban előttük egy üres sor van. Ezeket a parancsokat például kiemelten fontos bekezdések elé lehet tenni. Üres sort soha ne alkalmazzon függőleges térközként, mert a lap alján és tetején kifejtve tipográfiai hibát eredményez.



Videó: Betűtípusok és -méretek, térközök, törések

3.6. Bekezdések

Új bekezdés esetén a forrásállományban hagyni kell egy üres sort, vagy ki kell adni a

```
\par
```

parancsot. Gyakori hiba, hogy új bekezdés helyett sortörést alkalmaznak. Ez tipográfiai hiba, kerülje!

3.6.1. Bekezdések balra zárása

Ilyenkor a bekezdést kezdő sor is a bal margónál kezdődik és nincs a jobb oldalon kiegészítés, így szóelválasztások sincsenek. Megvalósítása:

```
\begin{flushleft}
```

```
<szöveg>
```

```
\end{flushleft}
```

3.6.2. Bekezdések jobbra zárása

Képzeld el egy balra zárt szöveget, de most minden sort toljon el úgy, hogy a sorvégek a jobb margóhoz kerüljenek. Ez a jobbra zárás. Megvalósítása:

```
\begin{flushright}
<szöveg>
\end{flushright}
```

3.6.3. Bekezdések középre zárása

Képzeld el egy balra zárt szöveget, de most minden sort toljon el középre. Ez a középre zárás. Megvalósítása:

```
\begin{center}
<szöveg>
\end{center}
```

Például

```
\begin{center}
Ez egy hosszabb szöveg, ami középre van zárva,
így szóelválasztások sincsenek benne.
De a sortörések pontjait mi is meg tudjuk adni:\\
Ez külön sorba kerül.\\ Ez is külön sorba kerül.\\
\end{center}
```

Ez egy hosszabb szöveg, ami középre van zárva, így szóelválasztások sincsenek benne. De a sortörések pontjait mi is meg tudjuk adni:
Ez külön sorba kerül.
Ez is külön sorba kerül.

3.7. Lábjegyzetek

Ahová lábjegyzetet szeretne írni, ott adja ki a

```
\footnote{<lábjegyzet szövege>}
```

parancsot. A `\footnote` előtt nem lehet szóköz. Ha a jegyzet egy adott szóra vonatkozik, akkor a parancsot közvetlenül a szó után írja, ha egy mondatra vagy mondatrészre, akkor az azt lezáró írásjel után.

A `magyar.ldf` fájl `defaults=hu-min` opciója a lábjegyzetek fölé nem tesz vízszintes vonalat. Ha mégis szeretne tenni, akkor írja be a következőt a preambulumba:

```
\footnotestyle{rule=fourth}
```

3.8. Színek

Színek kezelésére az `xcolor` csomag használható. A definiált színnevek a következők:

 black	 cyan	 green	 magenta
 blue	 darkgray	 lightgray	 olive
 brown	 gray	 lime	 orange



Színezéséhez a következő parancsot használhatja:

```
{\color{<színnév>}<szöveg>}
```

Például

```
{\color{red}Piros szöveg.} Fekete szöveg.
```

Piros szöveg. Fekete szöveg.

Ha olyan színre van szükség, amely nincs definiálva az `xcolor` csomagban, akkor használja a szín ún. RGB kódját:

```
{\color[RGB]{<red>,<green>,<blue>}<szöveg>}
```

ahol `<red>`, `<green>` és `<blue>` mindegyike 0 és 255 közötti egész szám lehet. Az első a vörös, a második a zöld, a harmadik a kék mennyiségét jelenti. Például

```
{\color[RGB]{122,0,67}Színes szöveg.} Fekete szöveg.
```

Színes szöveg. Fekete szöveg.

3.9. Internetcímek

Internetcímek az `url` csomag következő parancsával adhatók meg:

```
\url{<URL cím>}
```

Például

```
\url{http://www.tug.org}
```

http://www.tug.org

3.10. Számírás

Az 5 vagy annál több jegyű egész számokat ezres csoportosítással kell leírni. A csoportosítás jobbról balra történik. Nem kell csoportosítani a 4 jegyű egész számokat, kivéve abban az esetben, ha egy táblázat olyan oszlopában található, amelyben szerepel 4-nél több jegyű egész szám is. Így lehet elérni, hogy a megfelelő számjegyek mindig egymás alatt legyenek.

A csoportosító jel a magyarban a feles törhetetlen szóköz (`\,`) vagy a pont, az angolban pedig a vessző. Tehát például

Helyesen `1\,234\,567` vagy `1.234.567`, de `9999`.

Helyesen `1 234 567` vagy `1.234.567`, de `9999`.

Az ezres csoportosítás automatizálható a

```
\num{<szám>} \in siunitx
```

paranccsal. Például

```
\num{1234567}
```

```
1 234 567
```

A `\num` parancs csak a 4-nél több jegyű egész számokra alkalmazza az ezres csoportosítást. Ha az előzőekben leírt kivétel esetén szükséges a 4 jegyű számok ezres csoportosítása is, akkor alkalmazza az `siunitx` csomag `group-minimum-digits=4` opcióját. Ha ezt az opciót csak egy blokkon belül lokálisan szeretné bekapcsolni, akkor használja a következő parancsot a blokk elején:

```
\sisetup{group-minimum-digits=4}
```

Ha ezres csoportosítójelnek nem az alapértelmezett feles törhetetlen szóközt, hanem például a pontot szeretné, akkor használja az `siunitx` csomag `group-separator={.}` opcióját. Ekkor

```
\num{1234567}
```

```
1.234.567
```

3.11. Mértékegységek

Mértékszám és mértékegység közé mindig feles törhetetlen szóközt kell tenni. Például

```
123\,cm 1200\,km 50\,\% 1000\,Ft 500\,\$
```

```
123 cm 1200 km 50 % 1000 Ft 500 $
```

vagy

```
100\,\textcelsius ∈ textcomp  
1\,\textperthousand ∈ textcomp  
1\,\textpertenthousand ∈ textcomp
```

```
100°C 1‰ 1‱
```

Ez alól a szabály alól egy kivétel van, amikor szöveget írunk fokban, percben és másodpercben. Ekkor nincs a mértékszám után térköz. Ennek írását legkönnyebben az

```
\ang{<fok>;<perc>;<másodperc>} ∈ siunitx
```

paranccsal oldhatja meg. Például

```
\ang{1;;} vagy \ang{1;2;} vagy \ang{1;2;3}
```

```
1° vagy 1°2' vagy 1°2'3''
```

3.12. Kereszthivatkozások

Egy dokumentumban sok olyan elem lehet, amit számozunk és hivatkozunk is rá. Ezek az ún. kereszthivatkozások. Természetesen ezeket nem érdemes a forrásban konkrétan beírni, hiszen egy ilyen szám a szerkesztés során még változhat, így állandóan javítani kellene, ami egy idő után sok hibát eredményezne. Erre az a megoldás, hogy a \LaTeX -re bízunk a számozott elemeknél és a kereszthivatkozásoknál a megfelelő számok beírását.

Ha egy számozott elemről kiderül, hogy hivatkozni kell rá, akkor először ezt az elemet címkézzé meg a

```
\label{<címke>}
```

paranccsal. A `<címke>` tetszőleges lehet, de azért érdemes kerülni az ékezetes betűket, szóközt és aktív karaktereket (magyarban ilyen a fordított aposztróf jel, kettőspont, kérdőjel, felkiáltójel és a pontosvessző).

Például, a későbbiekben látni fogjuk, hogy egy számozott listát a `enumerate` környezettel hozhat létre, melyben minden listaelemet `\item` paranccsal indítunk:

```
\begin{enumerate}
\item Ez egy listaelem.
\item Ez egy másik listaelem.
\end{enumerate}
```

Ha a 2. listaelemre akar hivatkozni, akkor a kódban a 3. sort így módosítsa:

```
\item\label{lista-proba} Ez egy másik listaelem.
```

1. Ez egy listaelem.
2. Ez egy másik listaelem.

Címkével ellátott elemre a

```
\ref{<címke>}
```

paranccsal tud hivatkozni. Az előző példát folytatva:

```
Lásd \ref{lista-proba}.~listaelemet.
```

Lásd 2. listaelemet.

Sokkal szebb lenne a mondat, ha a sorszám elé határozott névelőt rakna: „az 1.”, „a 2.”, stb. Amint látja a magyarban a névelő függ a sorszámtól. Ezt a problémát is megoldja a `magyar.ldf`. Ilyenkor használja az

```
\az{\ref{<címke>}}
\Az{\ref{<címke>}}
```

parancsokat, attól függően, hogy a sorszám előtti névelőt kis vagy nagy kezdőbetűvel szeretné:

```
Lásd \az{\ref{lista-proba}}.~listaelemet.\\
\Az{\ref{lista-proba}}.~listaelemben olvasható.
```

Lásd a 2. listaelemet.

A 2. listaelemben olvasható.

Amikor megcímkéz egy elemet, akkor nem csak az adott sorszámot tudja a \LaTeX , hanem azt is, hogy az adott elem melyik oldalon található. Adott címkéhez tartozó oldalszámot a

```
\pageref{<címke>}
```

paranccsal írathatja ki. Ennek névelős verziói:

```
\az{\pageref{<címke>}}
\Az{\pageref{<címke>}}
```

Például

```
\Az{\ref{lista-proba}}.~listaelemet  
\az{\pageref{lista-proba}}.~oldalon találjuk.
```

A 2. listaelemet a 23. oldalon találjuk.

Létezik még ezeken kívül is hivatkozási forma (egyenlet, irodalomjegyzék), de ezeket majd az adott fejezetekben tárgyaljuk.



Videó: Bekezdések, lábjegyzetek, színek,
kereszthivatkozások

Feladat

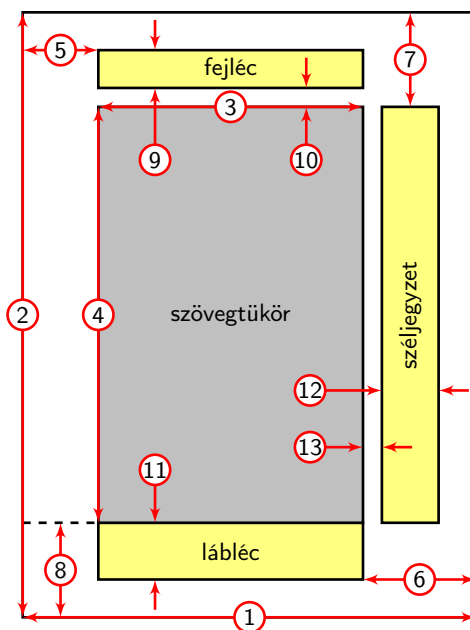
Szerkessze meg következő dokumentum \LaTeX -forrását: [Betűtípusok és -méretek, térközök, törések, bekezdések, lábjegyzetek, színek, kereszthivatkozások](#). A feladat megoldása letölthető innen: [Megoldás](#).

4. fejezet

Oldalak kinézete

4.1. Oldalak szerkezete és méretei

Egy oldal szerkezete a következő ábrán látható. Főbb részei: szövegtükör, margók, fejléc, lábléc, széljegyzet.



Az ábrán számokkal jelölt méreteket a `geometry` csomag opcióival állíthatja be, melyek a következők:

1. `paperwidth=<méret>` Oldal szélessége.
2. `paperheight=<méret>` Oldal magassága.
3. `textwidth=<méret>` Szövegtükör szélessége.
4. `textheight=<méret>` Szövegtükör magassága.
5. `inner=<méret>` Belső margó szélessége. A belső margó a lapok kötése felőli margó. Egyoldalas dokumentum esetén ez a bal margót, míg kétoldalas dokumentum esetén páratlan oldalon a bal, illetve páros oldalon a jobb margót jelenti.
6. `outer=<méret>` Külső margó (belső margóval ellentétes oldalon) szélessége.
7. `top=<méret>` Felső margó magassága.
8. `bottom=<méret>` Alsó margó magassága.
9. `headheight=<méret>` Fejléc magassága.

10. `headsep=<méret>` Fejléc és szövegtükör távolsága.
11. `footskip=<méret>` Lábléc magassága.
12. `marginparwidth=<méret>` Széljegyzet területének szélessége.
13. `marginparsep=<méret>` Széljegyzet és szövegtükör távolsága.

Ha szabványos méretet akar (A0–A6, B0–B6), akkor az `a0paper`, `...`, `a6paper`, `b0paper`, `...`, `b6paper` opciók valamelyikét kell betölteni. Például

```
\usepackage[b5paper]{geometry}
```

Ha ugyanezt a méretet szeretné, de 90 fokkal elforgatva, akkor használja a `landscape` opciót is:

```
\usepackage[b5paper,landscape]{geometry}
```

Ha egyedi méreteket akar, akkor például a következőt kell tenni:

```
\usepackage[paperwidth=105mm,paperheight=75mm]{geometry}
```

Fontos, hogy ezek fizikailag is beállítják a lap méretét, nem úgy, mint a standard dokumentumosztályok lapméretre vonatkozó opciói, melyek csak a margókra vannak hatással.

4.2. Többhasábos szedés

Többhasábos szedés esetén használja a `multicol` csomag `multicols` környezetét:

```
\begin{multicols}{<hasábszám>} ∈ multicol
<szöveg>
\end{multicols}
```

A `<hasábszám>` maximum 9 lehet. A hasábok közötti távolság 10 pt. Ennek átállítása például 1 cm-re:

```
\setlength{\columnsep}{1cm}
```

A hasábok közötti vonalvastagság 0 pt. Ennek átállítása például 1 pt-ra:

```
\setlength{\columnseprule}{1pt}
```

Feladat

Szerkessze meg következő dokumentum L^AT_EX-forrását: [Oldal méretei, többhasábos szedés](#). A feladat megoldása letölthető innen: [Megoldás](#).

5. fejezet

Listák

5.1. Számozatlan listák

Számozatlan listákra az `itemize` környezet használható. Minden listaelemet `\item` parancs vezet be.

```
\begin{itemize}
  \item <listaelem>
  \item <listaelem>
\end{itemize}
```

E környezetek négy szint mélységig ágyazhatók egymásba. Például:

Lista előtti szöveg.

```
\begin{itemize}
  \item Listaelem az első szinten.
  \begin{itemize}
    \item Listaelem a második szinten.
    \item Újabb listaelem a második szinten.
  \end{itemize}
  \item Egy másik listaelem az első szinten.
\end{itemize}
```

Lista utáni szöveg.

Lista előtti szöveg.

- Listaelem az első szinten.
 - Listaelem a második szinten.
 - Újabb listaelem a második szinten.
- Egy másik listaelem az első szinten.

Lista utáni szöveg.

5.1.1. Számozatlan listák extra térközök nélkül

Az `itemize` környezet minden listaelem között hagy egy extra függőleges térközt. Ha ezt nem akarja, akkor használja a `paralist` csomag `compactitem` környezetét. Ezt pontosan úgy kell használni, mint az előzőekben ismertetett `itemize` környezetet.

```

\begin{compactitem} ∈ paralist
  \item <listaelem>
  \item <listaelem>
\end{compactitem}

```

Például

```

Lista előtti szöveg.
\begin{compactitem}
  \item Listaelem az első szinten.
    \begin{compactitem}
      \item Listaelem a második szinten.
      \item Újabb listaelem a második szinten.
    \end{compactitem}
  \item Egy másik listaelem az első szinten.
\end{compactitem}
Lista utáni szöveg.

```

Lista előtti szöveg.

- Listaelem az első szinten.
 - Listaelem a második szinten.
 - Újabb listaelem a második szinten.
- Egy másik listaelem az első szinten.

Lista utáni szöveg.

5.2. Leíró listák

A leíró listákra, azaz a szótárszerű felsorolásokra a `description` környezet való. Minden listaelemet `\item[<címke>]` parancs előz meg. E környezetek hat szint mélységig ágyazhatók egymásba.

```

\begin{description}
  \item[<címke>] <listaelem>
  \item[<címke>] <listaelem>
\end{description}

```

Például

```

\begin{description}
  \item[Címke] szöveg szöveg szöveg szöveg szöveg szöveg szöveg
    szöveg szöveg szöveg szöveg szöveg szöveg
  \item[Másik címke] szöveg szöveg szöveg szöveg szöveg szöveg szöveg
    szöveg szöveg szöveg szöveg szöveg szöveg
\end{description}

```

Címke. szöveg szöveg szöveg szöveg szöveg szöveg szöveg szöveg szöveg szöveg szöveg szöveg szöveg szöveg szöveg szöveg szöveg szöveg

Másik címke. szöveg szöveg szöveg szöveg szöveg szöveg szöveg szöveg szöveg szöveg szöveg szöveg szöveg szöveg szöveg szöveg szöveg szöveg

5.3. Számozott listák

Számozott listákra az `enumerate` környezet való. Minden listaelemet `\item` parancs előz meg.

```
\begin{enumerate}
  \item <listaelem>
  \item <listaelem>
\end{enumerate}
```

E környezetek négy szint mélységig ágyazhatók egymásba. Például:

Lista előtti szöveg.

```
\begin{enumerate}
  \item Listaelem az első szinten.
  \begin{enumerate}
    \item Listaelem a második szinten.
    \item Újabb listaelem a második szinten.
  \end{enumerate}
  \item Egy másik listaelem az első szinten.
\end{enumerate}
```

Lista utáni szöveg.

Lista előtti szöveg.

1. Listaelem az első szinten.
 - a) Listaelem a második szinten.
 - b) Újabb listaelem a második szinten.
2. Egy másik listaelem az első szinten.

Lista utáni szöveg.

5.3.1. Számozott listák extra térközök nélkül

Ha nem akarja, hogy a listaelemek között legyen extra függőleges térköz, akkor az `enumerate` környezet helyett használja a `paralist` csomag `compactenum` környezetét. Használata pontosan megegyezik az `enumerate` környezettel.

```
\begin{compactenum} ∈ paralist
  \item <listaelem>
  \item <listaelem>
\end{compactenum}
```

Például

Lista előtti szöveg.

```
\begin{compactenum}
  \item Listaelem az első szinten.
  \begin{compactenum}
    \item Listaelem a második szinten.
    \item Újabb listaelem a második szinten.
  \end{compactenum}
  \item Egy másik listaelem az első szinten.
```

`\end{compactenum}`

Lista utáni szöveg.

Lista előtti szöveg.

1. Listaelem az első szinten.
 - a) Listaelem a második szinten.
 - b) Újabb listaelem a második szinten.
2. Egy másik listaelem az első szinten.

Lista utáni szöveg.



Videó: Listák

Feladat

Szerkessze meg következő dokumentum \LaTeX -forrását: [Listák](#). A feladat megoldása letölthető innen: [Megoldás](#).

6. fejezet

Képek

Képek beillesztése esetén használja a `graphicx` csomag következő parancsát:

```
\includegraphics[opciók]{képfájl} ∈ graphicx
```

A *képfájl* megadásakor a kiterjesztést nem kell megadni. Azaz például, ha az `abra.jpg` képet kell beilleszteni, akkor

```
\includegraphics{abra}
```

Fontos, hogy ebben az esetben a képfájlnak az aktuális könyvtárban kell lennie. Azonban praktikusabb, ha a képfájlokat az aktuális mappa egy almappájába teszi. Ennek az almappának célszerű olyan nevet adni, amelyben nincs ékezetes betű és szóköz. Például, ha a `grafikonok` nevű almappába helyezi az `abra.jpg` képet, akkor a következő módon jeleníthetjük meg:

```
\includegraphics{grafikonok/abra}
```

Gyakori hiba, hogy a teljes elérési utat megadják. Például

```
\includegraphics{C:/minta/grafikonok/abra.jpg} % ÍGY SOHA!
```

Ez rossz megoldás, hiszen ekkor a forrás csak ezen az útvonalon fog lefordulni, azaz nem lesz hordozható.

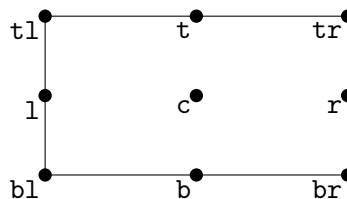
Az `\includegraphics` parancs fontosabb opciói:

`width=<szélesség>` A kép szélessége (például `width=5cm`).

`height=<magasság>` A kép magassága (például `height=5cm`). A `width` és a `height` együttes megadásával a képet torzíthatjuk is.

`angle=<fok>` Kép forgatásának szöge fokban. A pozitív érték az óra járásával ellentétes irány.

`origin=<origó>` Forgatás középpontja. Az *origó* értékei a következők lehetnek: `t1`, `t`, `tr`, `l`, `c`, `r`, `bl`, `b`, `br` (alapérték: `bl`). Ezek magyarázata a következő ábrán látható:



A következő példában a kép szélességét 3 cm-re állítjuk és elforgatjuk 90 fokkal az óra járásával megegyező irányban a középpontja körül:

```
\includegraphics[width=3cm,angle=-90,origin=c]{abra}
```

Feladat

Szerkessze meg következő dokumentum \LaTeX -forrását: [Képek](#). A feladat megoldása letölthető innen: [Megoldás](#).

7. fejezet

Táblázatok

A táblázatok elkészítése az egyik legbonyolultabb feladat a \LaTeX -ben. Nem tárgyaljuk általánosan az ide vonatkozó parancsokat, csak példákon keresztül tekintjük át a lehetőségeket a teljesség igénye nélkül. Az első példa:

```
\begin{tabular}{lrrr}  
Budapest & 7:00 & 9:30 & 13:15\\  
Dömsöd & 7:58 & 10:40 & 14:38\\  
\end{tabular}
```

Budapest	7:00	9:30	13:15
Dömsöd	7:58	10:40	14:38

Tehát táblázat a `tabular` környezettel készíthető. Ennek paraméterében kell megadni, hogy hány oszlop van, és a tartalmuk hogyan legyen igazítva. Az előző példában az `lrrr` azt jelenti, hogy 4 oszlop van, az első balra (`l` mint `left`), a többi 3 pedig jobbra (`r` mint `right`) legyen igazítva. Ha egy oszlopot középre akar igazítani, akkor azt a `c` (mint `center`) betűvel jelezze. A `&` az ún. tabulátor jel, ami két oszlop elválasztását jelzi. A `\\` sortörést jelöl. A táblázatba vonalakat is húzhat:

```
\begin{tabular}{|l|rrr|}  
\hline  
Budapest & 7:00 & 9:30 & 13:15\\ \cline{2-4}  
Dömsöd & 7:58 & 10:40 & 14:38\\ \hline  
\end{tabular}
```

Budapest	7:00	9:30	13:15
Dömsöd	7:58	10:40	14:38

Ahol függőleges vonalat akar húzni, oda a `tabular` környezet paraméterében rakjon `|` (`AltGr` + `W`) jelet. Ahová vízszintes vonalat akar húzni, oda a `tabular` környezetben tegyen `\hline` parancsot. Ha egy vízszintes vonalat nem akar teljesen meghúzni, csak mondjuk a 2. oszloptól a 4. oszlopig, akkor `\hline` helyett használjon `\cline{2-4}` parancsot.

Cellákat vízszintesen is összevonhat a

```
\multicolumn{<cellaszám>}{<cellaforma>}{<szöveg>}
```

paranccsal. A `<cellaszám>` az összevont cellák számát jelenti. A `<cellaforma>` az adott összevont cellára vonatkozó formázás, amely pontosan úgy történik, mint a `tabular`

környezet paraméterében. Ez a parancs akkor is célravezető, ha nem összevonni akar, csak az adott cellának a formázását akarja megváltoztatni az általánosan megadotthoz képest. Ilyenkor a `<cellaszám>` értelemszerűen 1. Például

```
\begin{tabular}{|l|rr|}
\cline{2-3}
\multicolumn{1}{l|}{&\multicolumn{2}{c|}{év}}\ \ \ \cline{2-3}
\multicolumn{1}{l|}{&\multicolumn{1}{c}{2002}&
\multicolumn{1}{c}{2003}}\ \ \ \hline
Jövedelem (Ft) & 994\,000 & 1\,231\,500\ \
Adó (Ft) & 165\,000 & 194\,950\ \ \ \hline
\end{tabular}
```

	év	
	2002	2003
Jövedelem (Ft)	994 000	1 231 500
Adó (Ft)	165 000	194 950

Cellák függőleges összevonását a következő paranccsal teheti meg:

```
\multirow{<cellaszám>}{*{<szöveg>}} ∈ multirow
\multirow{<cellaszám>}{<szélesség>}{<szöveg>} ∈ multirow
```

Például

```
\begin{tabular}{|l|c|}
\hline
\multirow{2}{*{Egysoros szöveg}} & 1\ \
& 2\ \ \hline
\multirow{3}{3cm}{3\,cm széles szöveg törve} & 3\ \ \cline{2-2}
& 4\ \ \cline{2-2}
& 5\ \ \hline
\end{tabular}
```

Egysoros szöveg	1
	2
3 cm széles szöveg törve	3
	4
	5

Feladat

Szerkessze meg következő dokumentum \LaTeX -forrását: [Táblázat](#). A feladat megoldása letölthető innen: [Megoldás](#).

8. fejezet

Objektumok úsztatása

A táblázatok és képek beillesztésénél előfordulhat, hogy azok az adott oldalon már nem férnek el, és a következő oldalra való áthelyezésével az oldal alja telítetlen marad. Ennek megoldására született az úgynevezett „úsztatás”. Ez azt jelenti, hogy a problémás objektumot áthelyezi egy általunk megadott helyre (az aktuális oldal aljára, tetejére, vagy külön oldalra), az oldalt pedig telíti a soron következő szöveggel.

8.1. Képek és táblázatok úsztatása

Képek úsztatására a `figure`, míg táblázatok úsztatására a `table` környezetet használhatók. Ezen környezetek opciói:

- `h` Maradjon helyben, ha lehetséges.
- `t` Az aktuális oldal tetejére kerüljön.
- `b` Az aktuális oldal aljára kerüljön.
- `p` Külön oldalra kerüljön.
- ! Ekkor megszűnnek bizonyos korlátozások, így az objektum nagyobb eséllyel kerül arra helyre, ahová szeretnénk.

Opciónak ezen betűk bármilyen kombinációja használható. A betűk sorrendje mindegy, ugyanis az objektum a legelső olyan helyre kerül, amelyet az opció megenged. Ez alól csak a `h` kivétel, aminek mindennel szemben elsőbbsége van. Nézzünk néhány példát.

```
\begin{figure}
\centering
\includegraphics{fig}
\end{figure}
```

Mivel itt nem adtunk meg opciót, így az alapérték érvényesül, mely `tbp`. Ez azt jelenti, hogy ebben az esetben a képet először megpróbálja a lap tetejére, ha oda nem kerülhet, akkor a lap aljára, ha oda sem, akkor külön oldalra tenni.

```
\begin{figure}[th]
\centering
\includegraphics{fig}
\end{figure}
```

Ebben az esetben a képet először megpróbálja helybenhagyni, de ha oda nem kerülhet, akkor a lap tetejére teszi.

```
\begin{figure}[!ht]
```

```
\centering
\includegraphics{fig}
\end{figure}
```

A képet bizonyos korlátozások feloldása mellett, de nem mindenáron próbálja helyben tartani. Ha oda nem kerülhet, akkor a lap tetejére teszi.

8.2. Úsztatott objektumok címkézése

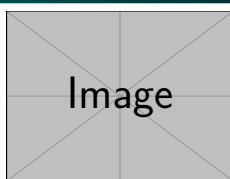
Sokszor előfordul, hogy a képekre, táblázatokra hivatkozni szeretnénk. Ilyenkor célszerű nekik automatikus sorszámot és címet adni. Ezen feladatok elvégzésére szolgál a

```
\caption{< cím >}
```

parancs. Például

```
\begin{figure}[!ht]
\centering
\includegraphics[width=3cm]{example-image}
\caption{Egy példa ábra úsztatására}\label{fig-pelda}
\end{figure}
\Aref{fig-pelda}~ábrán látható \dots
```

Ennek hatására a képet megjeleníti középen és felcímkézi. A címkébe aszerint kerül „ábra” vagy „táblázat” felirat, hogy `figure` vagy `table` környezetbe raktuk a `\caption` parancsot. A sorszám automatikus.



1. ábra. Egy példa ábra úsztatására

Az 1. ábrán látható ...



Videó: Képek és táblázatok

Feladat

Szerkessze meg következő dokumentum \LaTeX -forrását: [Képek és táblázatok úsztatása](#). A feladat megoldása letölthető innen: [Megoldás](#).

9. fejezet

Verbatim, programkódok

9.1. Verbatim

A verbatim olyan része a forrásállománynak, amely úgy jelenik meg a dokumentumban, mint a forrásállományban. Ha a verbatim szöveg nem hosszabb egy input sornál, akkor használja a

```
\verb|verbatim szöveg|  
\verb*|verbatim szöveg|
```

parancsokat. A | határolójel lehet bármely más, szóköztől, *-tól és betűtől különböző jel, ami nem szerepel a verbatim szövegben. Például

```
\verb|\LaTeX\ könyv|\  
\verb+\LaTeX\ kód+
```

```
\LaTeX\ könyv  
\LaTeX\ kód
```

A `\verb` helyett `\verb*` parancsot írva, az eredményben a szóközők helyén `␣` jelenik meg. Például

```
\verb*|\LaTeX\ könyv|\  
\verb**\LaTeX\ kód+
```

```
\LaTeX\␣könyv  
\LaTeX\␣kód
```

A `\verb` illetve `\verb*` parancsok nem tehetők más parancsok argumentumába.

Ha egy input sornál többet akar beírni verbatimként, akkor használja a `verbatim` vagy `verbatim*` környezeteket. Például

```
\begin{verbatim}  
\LaTeX\ könyv  
\LaTeX\ kód  
\end{verbatim}  
\begin{verbatim*}  
\LaTeX\ könyv  
\LaTeX\ kód  
\end{verbatim*}
```

```

\LaTeX\ könyv
\LaTeX\ kód

\LaTeX\_\könyv
\LaTeX\_\kód

```

Ezek a környezetek nem tehetők parancsok argumentumába.

9.2. Programkódok

Különböző programnyelvek kódjainak megjelenítésére alkalmas a `listings` csomag.

```

\begin{lstlisting}[\langle opciók \rangle] \langle kód \rangle \end{lstlisting} ∈ listings
\lstinputlisting[\langle opciók \rangle]{\langle kódot tartalmazó fájl \rangle} ∈ listings

```

Az opciók a következő parancsban is megadhatók:

```

\lstset{\langle opciók \rangle} ∈ listings

```

Tekintsük át az előbbi parancsok néhány hasznos opcióját. Az értékekben szereplő színekre vonatkozó kódok az `xcolor` csomag betöltésével működnek.

`basicstyle=\langle stílus \rangle` Kód fontjai (például `basicstyle=\small\ttfamily`).

`breaklines` Hosszú sorok törése.

`backgroundcolor=\langle szín \rangle` Háttérszín (például `backgroundcolor=\color{red}`).

`showspaces` Szóköz `_` módon jelölve.

`showtabs` Tabulátort jelöli.

`numbers=\langle típus \rangle` Kód sorainak számozása. Ha a `\langle típus \rangle` `none` (alapértelmezés), akkor nincs számozás, ha `left`, akkor bal oldalon van számozás, ha `right`, akkor jobb oldalon van számozás.

`frame=\langle érték \rangle` Keretvonalak rajzolása. Az érték a `trbl` bármilyen részhalmaza lehet (`t` fent, `b` lent, `r` jobbra, `l` balra). Például, ha fent és bal oldalon akarunk vonalat húzni, akkor `frame=tl`.

`rulecolor=\langle szín \rangle` Keret vonalának színe (például `rulecolor=\color{red}`).

`language=\langle programnyelv \rangle` Programnyelv kulcsszavainak, megjegyzéseinek a kiemelését tölti be. Az előre definiált nyelvek listája megtalálható a csomag leírásában. Például `language=Delphi`.

`keywords=\{ \langle lista \rangle \}` A kiemelő kulcsszavak listája, mely a `\langle lista \rangle`-ban van felsorolva, vesszővel elválasztva. Például `keywords={begin,end}`.

`morekeywords=\{ \langle lista \rangle \}` A kulcsszavak listáját ezzel lehet bővíteni.

`keywordstyle=\langle stílus \rangle` A kulcsszavak stílusa. Például `keywordstyle=\bfseries`.

`title=\{ \langle kódcím \rangle \}` Kód címe sorszám nélkül. Ez nem kerül be a kódok jegyzékébe.

`caption=\{ \langle kódcím \rangle \}` Kód címe sorszámmal, címkével. Ha címkének például a „kód” szót szeretné, akkor használja ezt a parancsot: `\def\lstlistingname{kód}`. Ha magyar nyelvű dokumentumot ír, akkor még töltsé be a `caption` csomagot is.

`label=\{ \langle címke \rangle \}` Kereszthivatkozás címkéje. Ezt a `\label` parancs helyett kell használni. A parancs azért nem használható, mert azt a \LaTeX már a programkód részének tekintené.

Például

```

\def\lstlistingname{kód}
\lstset{language=Delphi,basicstyle=\footnotesize\ttfamily,

```

```

keywordstyle=\color{blue},numbers=left,frame=trbl}

\begin{lstlisting}[caption={Trim függvény},label={kod-trim}]
function Trim(s:string):string;
var i:integer;
begin
result:='';
for i:=1 to length(s) do if s[i]<>' '
then result:=result+s[i];
end;
\end{lstlisting}
\Aref{kod-trim}.~kódban \dots

```

1. kód. Trim függvény

```

1 function Trim(s:string):string;
2 var i:integer;
3 begin
4 result:='';
5 for i:=1 to length(s) do if s[i]<>' '
6 then result:=result+s[i];
7 end;

```

Az 1. kódban ...

Az előző eredmény a következő módon is elérhető. Az alábbi programkódot mentse el egy trim.pas nevű szövegfájlba a tex kiterjesztésű forrásállomány mellé:

```

function Trim(s:string):string;
var i:integer;
begin
result:='';
for i:=1 to length(s) do if s[i]<>' '
then result:=result+s[i];
end;

```

Ezután az lstlisting környezet helyett használja a \lstinputlisting parancsot a dokumentumban:

```
\lstinputlisting[caption={Trim függvény},label={kod-trim}]{trim.pas}
```

Feladat

Szerkessze meg következő dokumentum L^AT_EX-forrását: [Programkód](#). A feladat megoldása letölthető innen: [Megoldás](#).

10. fejezet

Képletek

10.1. Matematikai mód

Ha matematikai képletet akar szerkeszteni, akkor használja az `amsmath` és `amssymb` csomagokat. Az `amsmath` helyett használható annak kiterjesztése, a `mathtools` csomag is.

Ha egy képlet kb. akkora mint egy szó, akkor azt a szövegbe illesztjük, mint például a $\sin^2 \alpha + \cos^2 \alpha = 1$ esetén. Ez az ún. *szövegközi matematikai mód*. Ennek megadása a következő módon lehetséges:

```
$\langle képlet \rangle$
```

Például

Bármit is teszünk, `$2+2=4$`.

Bármit is teszünk, $2 + 2 = 4$.

Ha a képlet nagyobb, bonyolultabb, vagy fontossága miatt ki kell emelni, akkor külön sorba kell szedni, mint például az

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} \quad \forall x \in \mathbb{R}$$

esetén. Ez az ún. *kiemelt matematikai mód*. Ennek megadása a következő két sor bármelyikével lehetséges:

```
\[\langle képlet \rangle\]
```

```
\begin{equation*}\langle képlet \rangle\end{equation*}
```

Például

Bármit is teszünk, `\[2+2=4.\]`

Bármit is teszünk,

$$2 + 2 = 4.$$

Bizonyos esetekben előfordulhat, hogy egy képletben magyarázó vagy összekötő szöveget kell beiktatni. Ilyenkor ideiglenesen ki kell lépni a matematikai módból a

```
\text{\langle szöveg \rangle}
```

paranccsal. Például

```
\[1+1=2 \text{ és } 2+2=4\]
```


$$1 + 1 = 2 \text{ és } 2 + 2 = 4$$

A képletben rosszul jelent meg a szöveg, pedig a forrásban volt szóköz a szöveg előtt és után. Ennek az a magyarázata, hogy a matematikai módban begépett szóközöket a L^AT_EX felülbírálja, a nagyon speciális tipográfia miatt nem bízza a szerzőre. Ilyenkor az a megoldás, hogy a szóközöket szöveg módban adja ki:

```
\[1+1=2 \text{ és } 2+2=4\]
```

$$1 + 1 = 2 \text{ és } 2 + 2 = 4$$

De ez még mindig nem tökéletes. Ugyanis a képletekben maguktól megjelenő térközök miatt nem különül el jól a szöveg. Ilyenkor lehet használni a `\quad` parancsot:

```
\[1+1=2\quad\text{és}\quad2+2=4\]
```

$$1 + 1 = 2 \quad \text{és} \quad 2 + 2 = 4$$

10.2. Műveleti jelek

<code>+</code>	<code>\mp</code>	<code>\setminus</code>	<code>\vee</code>
<code>-</code>	<code>\cdot</code>	<code>\cap</code>	<code>\star</code>
<code>/</code>	<code>\times</code>	<code>\cup</code>	<code>*</code>
<code>\pm</code>	<code>\div</code>	<code>\wedge</code>	<code>\circ</code>

10.3. Relációjelek

<code>=</code>	<code>\approx</code>	<code>\ll</code>	<code>\subseteq</code>
<code>:=</code>	<code>\cong</code>	<code>\gg</code>	<code>\supseteq</code>
<code>\equiv</code>	<code><</code>	<code>\in</code>	<code>:</code> (arányjel)
<code>\sim</code>	<code>></code>	<code>\ni</code>	<code> </code>
<code>\simeq</code>	<code>\leq</code>	<code>\subset</code>	<code>\parallel</code>
	<code>\geq</code>	<code>\supset</code>	<code>\perp</code>

A nyilak is a relációjelek közé tartoznak:

<code>\leftarrow</code>	<code>\leftharpoonup</code>	<code>\Uparrow</code>
<code>\rightarrow</code> vagy <code>\to</code>	<code>\leftharpoondown</code>	<code>\Downarrow</code>
<code>\leftrightarrow</code>	<code>\rightharpoonup</code>	<code>\Updownarrow</code>
<code>\Leftarrow</code>	<code>\rightharpoondown</code>	<code>\nearrow</code>
<code>\Rightarrow</code>	<code>\uparrow</code>	<code>\searrow</code>
<code>\Leftrightarrow</code>	<code>\downarrow</code>	<code>\swarrow</code>
<code>\mapsto</code>	<code>\updownarrow</code>	<code>\nwarrow</code>

Relációjeleket negálni (áthúzni) a `\not` parancssal lehet. Például

```
$a\not=b$
```

$$a \neq b$$

Néhány esetben ez nem ad megfelelő eredményt:

`$a\not\mid b$, $a\not\parallel b$`

`a \nmid b, a \nparallel b`

Ezek helyett külön tervezésű negált reláció jelet kell használni:

`$a\nmid b$, $a\nparallel b$`

`a \nmid b, a \nparallel b`

10.4. Közöséges matematikai jelek

<code>% \%</code>	<code>\exists \exists</code>	<code>\partial \partial</code>	<code>\angle \angle</code>
<code>\perp \bot</code>	<code>\nexists \nexists</code>	<code>\emptyset \emptyset</code>	<code>\measuredangle \measuredangle</code>
<code>\top \top</code>	<code>\Re \Re</code>	<code>\infty \infty</code>	<code>1^\circ 1^\circ</code>
<code>\neg \neg</code>	<code>\Im \Im</code>	<code>\triangle \triangle</code>	<code>1' 1'</code>
<code>\forall \forall</code>	<code>\nabla \nabla</code>	<code>\square \square</code>	<code>1'' 1''</code>

10.5. Indexek, gyökjelek, törtek, binomiális együtthatók

x_a	<code>x_{a}</code>	$\sqrt{2}$	<code>\sqrt{2}</code>	$\binom{a}{b}$	<code>\binom{a}{b}</code>
x^b	<code>x^{b}</code>	$\sqrt[3]{5}$	<code>\sqrt[3]{5}</code>		
x_a^b	<code>x_{a}^{b}</code>	$\frac{a}{b}$	<code>\frac{a}{b}</code>		

10.6. Matematikai ékezetek

\hat{a}	<code>\hat{a}</code>	\vec{a}	<code>\vec{a}</code>	\breve{a}	<code>\breve{a}</code>	\ddot{a}	<code>\ddot{a}</code>
\tilde{a}	<code>\tilde{a}</code>	\acute{a}	<code>\acute{a}</code>	\check{a}	<code>\check{a}</code>	\dddot{a}	<code>\dddot{a}</code>
\bar{a}	<code>\bar{a}</code>	\grave{a}	<code>\grave{a}</code>	\dot{a}	<code>\dot{a}</code>	\ddddot{a}	<code>\ddddot{a}</code>

10.7. Görög betűk

α	<code>\alpha</code>	μ	<code>\mu</code>	ω	<code>\omega</code>	Θ	<code>\Theta</code>
β	<code>\beta</code>	ν	<code>\nu</code>	F	<code>\digamma</code>	Λ	<code>\Lambda</code>
γ	<code>\gamma</code>	ξ	<code>\xi</code>	ε	<code>\varepsilon</code>	Ξ	<code>\Xi</code>
δ	<code>\delta</code>	π	<code>\pi</code>	ϑ	<code>\vartheta</code>	Π	<code>\Pi</code>
ϵ	<code>\epsilon</code>	ρ	<code>\rho</code>	\varkappa	<code>\varkappa</code>	Σ	<code>\Sigma</code>
ζ	<code>\zeta</code>	σ	<code>\sigma</code>	ϖ	<code>\varpi</code>	Υ	<code>\Upsilon</code>
η	<code>\eta</code>	τ	<code>\tau</code>	ϱ	<code>\varrho</code>	Φ	<code>\Phi</code>
θ	<code>\theta</code>	υ	<code>\upsilon</code>	ς	<code>\varsigma</code>	Ψ	<code>\Psi</code>
ι	<code>\iota</code>	ϕ	<code>\phi</code>	φ	<code>\varphi</code>	Ω	<code>\Omega</code>
κ	<code>\kappa</code>	χ	<code>\chi</code>	Γ	<code>\Gamma</code>		
λ	<code>\lambda</code>	ψ	<code>\psi</code>	Δ	<code>\Delta</code>		

10.8. Kalligrafikus és dupla szárú betűk

$ABCDEFGHIJKLMN$ `\mathcal{ABCDEFGHIJKLMN}`
 $ABCDEFGHIJKLMN$ `\mathbb{ABCDEFGHIJKLMN}`

10.9. Három pont

$1, \dots, n$ `1, \dots, n` $1 \cdots n$ `1 \cdots n` $:$ `\vdots`
 $1 + \cdots + n$ `1 + \dots + n` $\int \cdots \int$ `\int \dots i \int` \ddots `\ddots`

Az `\dots` az alpvonalra, míg a `\cdots` függőlegesen középre teszi a három pontot. Több esetben ez automatizálható a

`\dots`

paranccsal. Például

```
$1, \dots, n$\n$1 + \dots + n$
```

```
1, \dots, n\n1 + \dots + n
```

10.10. Matematikai zárójelek

(([`\lceil`]] `\rfloor`
[[[`\lfloor`]] `\}` `\}` `\|` `\|`
{ `\{` | | } `\rangle`
< `\langle`)) `\rceil`

A matematikai zárójeleket övező térközök nagyon speciálisan viselkednek, továbbá az sem mindegy, hogy nyitó vagy csukó zárójelről van szó. Ezt közölni kell a forráskódban. A nyitó zárójel elé

`\left`

míg a csukó zárójel elé

`\right`

parancsot kell írni. Példaként nézzük meg a következő két sor eredménye közötti különbséget:

```
$a(b+c)d$\n$a\left(b+c\right)d$
```

```
a(b + c)d\na (b + c) d
```

Van olyan eset, amikor a `\left` és `\right` parancsok elhagyása teljesen rossz eredményt ad. Például

```
$|-7|$$\n$\left|-7\right|$
```

```
| - 7|  
|-7|
```

Az első eset azért rossz, mert a program azt hiszi, hogy a `|` jelből kivonjuk a 7-et. Így a `-` jel körül térközöket hagy. Felmerül a kérdés, hogy a \LaTeX miért nem tudja, hogy például a `]` jel egy csukó zárójel? Hiszen ekkor nem kellene elé rakni a `\right` parancsot. Ez azért van, mert a matematikában egyáltalán nem biztos, hogy `]` valóban csukó zárójel. Például gondoljon a `]0, 1]` félig nyílt, félig zárt intervallumra. Az sem biztos, hogy például a `{` jellel zárójelet akar kifejezni. Gondoljon az esetek szétválasztására:

$$f(x) = \begin{cases} 1, & \text{ha } x < 1, \\ 0, & \text{ha } x \geq 1. \end{cases}$$

Vagyis a zárójelek nem automatizálhatók a képletekben. A felhasználónak kell megmondani, hogy mi számít nyitó és mi csukó zárójelnek.

A `\left` és `\right` parancsok nem csak a térközöket, hanem a zárójelek nagyságát is beállítják. Például

```
\left(1+\left(x+y\right)^2\right)^3
```

```
(1 + (x + y)^2)^3
```

10.11. Esetek szétválasztása

Erre a `cases` környezet használható. Például

```
\[  
f(x)=  
\begin{cases}  
0 & \text{ha } x\$ \text{ racionális,}\\ 1 & \text{ha } x\$ \text{ irracionális.}\\ \end{cases}  
\]
```

$$f(x) = \begin{cases} 0 & \text{ha } x \text{ racionális,} \\ 1 & \text{ha } x \text{ irracionális.} \end{cases}$$

10.12. Matematikai jelek több szerepben

Vannak olyan matematikai jelek, amelyeknek többféle szerepe is lehet. Például

```
 $f:A \rightarrow B$  (helyes)  
 $f:A \rightarrow B$  (helytelen)
```

```
 $f: A \rightarrow B$  (helyes)  
 $f : A \rightarrow B$  (helytelen)
```

A második megoldás azért rossz, mert ott az szerepel, hogy f aránylik az A -hoz.

Ha a magyar.1df fájlt defaults=hu-min opcióval töltötte be, akkor a vessző matematikai üzemmódban két szám között tizedesvesszőként értelmezett, de egyéb esetben megmarad az eredeti szerepe. Például

```
$2,5\cdot2=5$\$
$a,b,c$
```

2,5 · 2 = 5
a, b, c

Ha két szám között a vesszőt nem tizedesvesszőként használja, akkor a vessző után tegyen egy szóközt:

```
$1, 2, 3,\dots$
```

1, 2, 3, ...

10.13. Változó hosszúságú vízszintes jelek

\widehat{xz}	<code>\widehat{xz}</code>	$A \xrightarrow{f \circ g} B$	<code>A\xrightarrow{f \circ g} B</code>
\overline{xz}	<code>\overline{xz}</code>	$\overbrace{xxxxzzz}^n$	<code>\overbrace{xxxxzzz}^n</code>
\underline{xz}	<code>\underline{xz}</code>	$\underbrace{xxxxzzz}_n$	<code>\underbrace{xxxxzzz}_n</code>
\overrightarrow{xz}	<code>\overrightarrow{xz}</code>		

10.14. Mátrixok

Mátrixokat többek között a következő környezetekkel lehet készíteni: `pmatrix`, `bmatrix`, `vmatrix`. Ezek belsejében a táblázatokhoz hasonló módon kell megadni a mátrix elemeit. Például

```
\[
\begin{pmatrix}
a & b \\
c & d
\end{pmatrix} \quad
\begin{bmatrix}
a & b \\
c & d
\end{bmatrix} \quad
\begin{vmatrix}
a & b \\
c & d
\end{vmatrix}
\]
```

$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad \begin{vmatrix} a & b \\ c & d \end{vmatrix}$

10.15. Operátorok, függvények

10.15.1. Nagy operátorok

Σ	<code>\sum</code>	\int	<code>\int</code>	\cap	<code>\bigcap</code>	\oint	<code>\oiint</code>
\prod	<code>\prod</code>	\iint	<code>\iint</code>	\cup	<code>\bigcup</code>	\iiint	<code>\iiint</code>

A nagy operátorok más méretben jelennek meg szövegszerű illetve kiemelt matematikai módban. Például

```
\sum
\[\sum\]
```

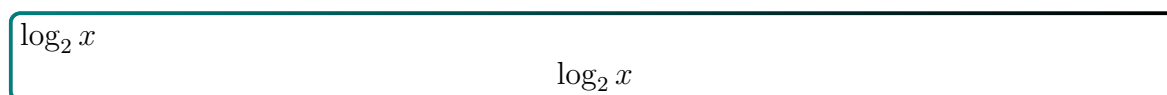


10.15.2. „Nolimits” függvények

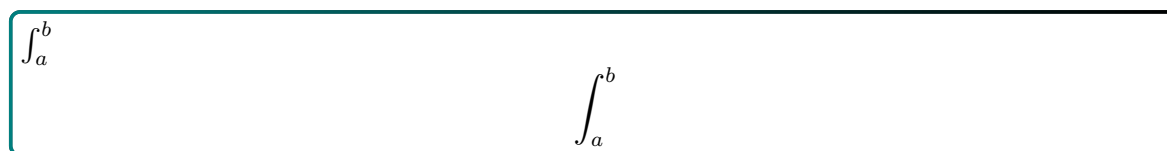
<code>\arccos</code>	<code>\cot</code>	<code>\hom</code>	<code>\sin</code>
<code>\arcsin</code>	<code>\coth</code>	<code>\ker</code>	<code>\sinh</code>
<code>\arctan</code>	<code>\csc</code>	<code>\lg</code>	<code>\tan</code>
<code>\arg</code>	<code>\deg</code>	<code>\ln</code>	<code>\tanh</code>
<code>\cos</code>	<code>\dim</code>	<code>\log</code>	<code>\lim</code>
<code>\cosh</code>	<code>\exp</code>	<code>\sec</code>	<code>\limsup</code>

A „nolimits” függvények és az integráljel indexei mindig mellette jelennek meg. Például

```
\log_2x
\[\log_2x\]
```



```
\int_a^b
\[\int_a^b\]
```



10.15.3. „Limits” függvények

<code>\det</code>	<code>\lim</code>	<code>\limsup</code>	<code>\min</code>
<code>\inf</code>	<code>\liminf</code>	<code>\max</code>	<code>\sup</code>

A nagy operátorok (az integráljel kivételével) és a „limits” függvények indexei szövegszerű matematikai módban mellette jelennek meg, de kiemelt matematikai módban alatta és fölötte. Például

$\sum_{i=1}^{\infty} a_i := \lim_{n \rightarrow \infty} \sum_{i=1}^n a_i$
 $\backslash\sum_{i=1}^{\infty} a_i := \lim_{n \rightarrow \infty} \sum_{i=1}^n a_i \backslash$

$$\sum_{i=1}^{\infty} a_i := \lim_{n \rightarrow \infty} \sum_{i=1}^n a_i$$

$$\sum_{i=1}^{\infty} a_i := \lim_{n \rightarrow \infty} \sum_{i=1}^n a_i$$

10.15.4. Új függvények definiálása

Előfordulhat, hogy olyan függvényre van szükség, amely alpból nem áll rendelkezésre. Például a magyarban a tangens jele tg, amelynek csak az angol verziója (tan) definiált. Ilyenkor mi is gyárthatunk újakat. Új „limits” függvény a következő paranccsal definiálható a preambulumban:

```
\DeclareMathOperator*{<parancs>}{<jel>}
```

Például

```
\DeclareMathOperator*{\Min}{Min}
```

után

```

 $\Min_{k \in \mathbb{N}}$ 
 $\backslash\Min_{k \in \mathbb{N}} \backslash$ 

```

$$\text{Min}_{k \in \mathbb{N}} \qquad \text{Min}_{k \in \mathbb{N}}$$

Új „nolimits” függvény a következő paranccsal definiálható a preambulumban:

```
\DeclareMathOperator{<parancs>}{<jel>}
```

Például

```
\DeclareMathOperator{tg}{tg}
```

után

```

 $tg^2 x$ 
 $\backslash tg^2 x \backslash$ 

```

$$tg^2 x \qquad tg^2 x$$

10.15.5. Differenciálás, differencia operátor

Derivált függvényt a ' (Shift + 1) billentyűvel írhatunk. Például

$f'(x)$, $f''(x)$

$$f'(x), f''(x)$$

Az integrálásnál és deriválásnál szokásos differencia operátor jelet magunknak kell definiálni a preambulumban:

```
\DeclareMathOperator{\diff}{d\!}
```

Ezután például

```
\[\int f(x)\diff x\quad\text{és}\quad\frac{\diff f(x)}{\diff x}\]
```

$$\int f(x) dx \quad \text{és} \quad \frac{df(x)}{dx}$$

10.16. Kiemelt képletek sorszámozása

A kiemelt képletek sorszámozására használja az `equation` környezetet. Hivatkozás esetén `\ref` helyett az `\eqref` parancs használható:

```
\begin{equation}\label{<címke>}  
<képlet>  
\end{equation}  
\eqref{<címke>}
```

Ha automatikus határozott névelőt akar elé tenni, akkor a következőket kell használni:

```
\Az{\eqref{<címke>}}  
\az{\eqref{<címke>}}
```

Például

```
\begin{equation}\label{egyenlet-masodfoku}  
x^2+2x-3=0  
\end{equation}  
\Az{\eqref{egyenlet-masodfoku}} miatt \dots
```

$$x^2 + 2x - 3 = 0 \tag{1}$$

Az (1) miatt ...

10.17. Több képlet egymás alatt

Ha több kiemelt képletet ír egymás alá, akkor nem ad jó végeredményt a `\[...\]`, `equation*` vagy az `equation` környezetek egymás utáni alkalmazása, mert túl nagy lesz közöttük a függőleges térköz. Ilyenkor használja a `gather` környezetet.

```
\begin{gather}  
<1. képlet>\label{<címke 1>}\\  
<2. képlet>\label{<címke 2>}\\  
...  
<n. képlet>\label{<címke n>}  
\end{gather}
```

Ha nem akar képletszámozást, akkor a `gather*` környezetet használja. Ha csak egy sort nem akar számozni, akkor annak végére tegye a `\notag` parancsot. Például

```
\begin{gather}  
x+y \label{egyenlet-pelda-a}\\  
x^2+xy+y^2\label{egyenlet-pelda-b}  
\end{gather}
```


$$x + y \tag{1}$$

$$x^2 + xy + y^2 \tag{2}$$

```
\begin{gather}
x+y \tag{1}
x^2+xy+y^2\label{egyenlet-pelda}
\end{gather}
```

$$x + y \tag{1}$$

$$x^2 + xy + y^2 \tag{1}$$

10.18. Több képlet egymás alatt illesztéssel

Egymás alatti képletekben lehetnek olyan elemek, amelyeket egymáshoz kell illeszteni.

```
\begin{align}
<1. sor jobbra>&<balra> & <jobbra>&<balra> & <jobbra> ... \label{<címke 1>}\\
<2. sor jobbra>&<balra> & <jobbra>&<balra> & <jobbra> ... \label{<címke 2>}\\
...
<n. sor jobbra>&<balra> & <jobbra>&<balra> & <jobbra> ... \label{<címke n>}
\end{align}
```

Például

```
\begin{align}
x&=y+z & y&=bd & z&=bc \label{egyenlet-pelda-a}\\
b&=10 & 2c&=56 & d&=44 \label{egyenlet-pelda-b}
\end{align}
```

$$x = y + z \qquad y = bd \qquad z = bc \tag{1}$$

$$b = 10 \qquad 2c = 56 \qquad d = 44 \tag{2}$$

A `\notag` parancs itt is ugyanúgy használható, mint a `gather` környezetben. Az `align*` környezet pontosan azt csinálja, mint az `align`, de nem tesz ki képletszámokat.

```
\begin{align*}
x&=y+z=\\
&=bd+bc=\\
&=1000
\end{align*}
```

$$x = y + z =$$

$$= bd + bc =$$

$$= 1000$$

Feladat

Szerkessze meg következő dokumentum \LaTeX -forrását: [Képletek](#). A feladat megoldása letölthető innen: [Megoldás](#).

11. fejezet

Strukturált művek

Hosszabb, strukturált dokumentumokat a következő módon lehet például tagolni:

- cím
- kivonat
- tartalomjegyzék
- főszöveg szintjei
 - fejezetek
 - szakaszok
 - alszakaszok
 - al-alszakaszok
 - paragrafusok
 - alparagrafusok
- bibliográfia

11.1. Főcím, címlap, kivonat

A mű címét, szerzőjét és dátumot a következő parancsokkal adhatja meg.

```
\title{<cím>}  
\author{<szerző>}  
\date{<dátum>}  
\maketitle
```

A `\title`, `\author` és `\date` parancsok írhatók preambulumba is, de a `\maketitle` csak a dokumentumtestbe. A `<dátum>` alapértéke a `\today` parancs, ami a fordítás idején az aktuális dátumot jelenti. Ezen parancsok argumentumaiba lábjegyzetek is írhatók a

```
\thanks{<szöveg>}
```

paranccsal. A `\maketitle` a rendelkezésre álló adatokból elkészíti a címet. Ezután nyithat egy `abstract` környezetet (kivéve a `book` osztályt), melybe a mű rövid kivonatát írhatja.

11.2. A főszöveg szintjei

A főszöveg szintjeinek címeit a következő parancsokkal adhatja meg:

```
\chapter{<cím>}
```

Fejezet címe (`article` osztályban nincs).

```
\section{<cím>}
```

Szakasz címe.

```
\subsection{<cím>}
```

Alszakasz címe.

```
\subsubsection{<cím>}
```

Al-alszakasz címe.

```
\paragraph{<cím>}
```

Paragrafus címe.

```
\subparagraph{<cím>}
```

Alparagrafus címe.

Például

```
\chapter{A fejezet címe}
```

1. fejezet

A fejezet címe

Ha egy számozott szint esetén csak egyetlen szintnek nem akar sorszámot, akkor használja az előző parancsok ún. csillagos változatát (például `\section*{<cím>}`). Ilyenkor a cím nem kerül a tartalomjegyzékbe és a fejlécebe sem.

A szintekre pontosan úgy lehet hivatkozni, mint azt az általános esetre leírtuk. Például

```
\subsection{Ez az alszakasz címe}\label{subsec-pelda}
```

...

Lásd `\aref{subsec-pelda}`.~alszakaszban.

1.1. Ez az alszakasz címe

...

Lásd az 1.1. alszakaszban.

11.3. Fej- és láblécek

11.3.1. Alapbeállítások

Egy hosszabb dokumentumban célszerű, ha minden oldalon találunk utalást arra, hogy az a dokumentum mely részén van: hányadik oldalon, melyik szinten és melyik alszinten. Ezek `book` osztályban automatikusan megjelennek. A másik két osztályban (`article`, `report`) ehhez adja ki a

```
\pagestyle{headings}
```

parancsot. Ennek hatása:

- lábléc üres
- oldalszám a fejléc külső margójánál
- szint információi a fejlécben

- egyoldalas szedésnél minden oldalon a belső margónál
 - kétoldalas szedésnél páros oldalon a belső margónál
- alszint információi a fejlécben
- egyoldalas szedésnél nincs
 - kétoldalas szedésnél páratlan oldalon a belső margónál.

További oldalstílusok:

```
\pagestyle{empty}
```

Üres fej- és lábléc.

```
\pagestyle{plain}
```

Üres fejléc, a lábléc közepén oldalszám.

```
\pagestyle{myheadings}\markboth{<infó1>}{<infó2>}
```

- lábléc üres
- oldalszám a fejléc külső margójánál
- *<infó1>* a fejlécben
 - egyoldalas szedésnél nincs
 - kétoldalas szedésnél páros oldalon a belső margónál
- *<infó2>* a fejlécben
 - egyoldalas szedésnél minden oldalon a belső margónál
 - kétoldalas szedésnél páratlan oldalon a belső margónál
- *<infó1>* és *<infó2>* bármikor megváltoztatható a `\markboth` paranccsal. Külön csak az *<infó2>* is megadható a `\markright{<infó2>}` paranccsal.

Ha egy konkrét oldalra vonatkozóan meg akarja az oldalstílust változtatni, akkor az adott szövegrészhez gépelje be:

```
\thispagestyle{<stílus>}
```

ahol a *<stílus>*: `headings`, `myheadings`, `empty` vagy `plain`. A `report` és `book` osztályokban az új részt és az új fejezetet nyitó oldalak `plain` stílusra váltanak, majd a következő oldaltól visszatér az eredeti stílusra.

11.3.2. Fej- és láblécek testreszabása

A testreszabáshoz a `fancyhdr` csomag használható. Ezt a csomagot már a `babel` előtt be kell tölteni. Ennek a csomagnak van egy saját stílusa `fancy` néven, melynek hatása:

- lábléc közepén az oldalszám
- szint információi a fejlécben
 - egyoldalas szedésnél a külső margónál
 - kétoldalas szedésnél a belső margónál
- alszint információi a fejlécben
 - egyoldalas szedésnél a belső margónál
 - kétoldalas szedésnél a külső margónál.

Ezt a stílust átalakíthatja a

```
\fancyhead[<hely>]{<szöveg>}
```

```
\fancyfoot[<hely>]{<szöveg>}
```

parancsokkal. A *<hely>* lehetséges értékei: `LE`, `CE`, `RE`, `LO`, `CO`, `RO`. A betűk jelentései: `L` bal mező, `C` közép mező, `R` jobb mező, `E` páros oldal, `O` páratlan oldal. Tehát például `LE` a bal mezőt jelenti a páros oldalakon.

Minden testreszabás előtt adja ki a

```
\fancyhf{}
```

parancsot, mely a korábban definiált fej- és lábléc beállításokat törli.

Lehetőség van a főszöveget elválasztani egy vonallal, az ún. léniával, a fejléctől és lábléctől. Ezeknek a vonalaknak a vastagságát a következő parancsokkal állíthatja be:

```
\renewcommand{\headrulewidth}{\langle vastagság \rangle}
```

Fejléc alatti lénia vastagsága, aminek az alapértéke 0.4pt.

```
\renewcommand{\footrulewidth}{\langle vastagság \rangle}
```

Lábléc feletti lénia vastagsága, aminek az alapértéke 0pt.

A fej- és láblécek tartalmának beállításaira a következő parancsok használhatók:

```
\leftmark
```

Az `article` osztály esetén a szakaszcímet tárolja, `report` illetve `book` osztály esetén pedig fejezetcímet.

```
\rightmark
```

Az `article` osztály esetén az alszakaszcímet tárolja, `report` illetve `book` osztály esetén pedig szakaszcímet.

```
\thepage
```

Kiírja az aktuális oldalszámot.

```
\nouppercase{\langle szöveg \rangle}
```

Alapesetben a szintinformációk nagybetűkkel jelennek meg a fej- és láblécben. Ha ezt nem akarjuk, akkor használjuk ezt a parancsot.

```
\markboth{\langle infó1 \rangle}{\langle infó2 \rangle}
```

Ennek hatására a `\leftmark` eredménye `\langle infó1 \rangle`, míg a `\rightmark` eredménye `\langle infó2 \rangle` lesz.

```
\markright{\langle infó \rangle}
```

Ennek hatására a `\rightmark` eredménye `\langle infó \rangle` lesz.

A következő kódot, ha a dokumentumtestbe írja, akkor onnan kétoldalas szedésnél a következő beállítások érvényesülnek:

- lénia nincs
- lábléc üres
- oldalszám a fejléc külső margójánál
- szint információi a fejlécben páros oldalon a belső margónál
- alszint információi a fejlécben páratlan oldalon a belső margónál.

```
\pagestyle{fancy}
```

```
\fancyhf{}
```

```
\fancyhead[LE,RO]{\normalfont\normalsize\thepage}
```

```
\fancyhead[LO]{\nouppercase{\sffamily\small\rightmark}}
```

```
\fancyhead[RE]{\nouppercase{\sffamily\small\leftmark}}
```

```
\renewcommand{\headrulewidth}{0pt}
```

Ha számozatlan szintet használ, például

```
\chapter*{Bevezetés}
```

akkor a fejlécben nem jelenik meg ez az információ. Ez felülbíráható a `\markboth` paranccsal. Például

```
\chapter*{Bevezetés}\markboth{Bevezetés}{Bevezetés}
```

Ekkor a `\leftmark` és a `\rightmark` eredménye is „Bevezetés”.

11.4. Tartalomjegyzék

A dokumentumnak arra a pontjára, ahol a tartalomjegyzéket meg akarja jeleníteni, adja ki a következő parancsot:

```
\tableofcontents
```

Amikor egy szintnyitó parancsnak a csillagos verzióját alkalmazza, akkor ez a cím nem lesz sorszámozva, nem kerül az élőfejbe és a tartalomjegyzékbe. Hogy mégis bekerüljön a tartalomjegyzékbe az oldalszámmal együtt, a szintnyitó parancs után gépelje be a következőt:

```
\addcontentsline{toc}{<szint>}{<cím>}
```

Például

```
\section*{Előszó}
\addcontentsline{toc}{section}{Előszó}
```

11.5. Tételszerű bekezdések

Sokszor lehet szükség olyan bekezdések írására, melyeknek típuscímet vagy sorszámot kell adni. Ilyen például a matematikában a tétel, bizonyítás, definíció. Ezek az ún. tételszerű bekezdések, melyeket a `\newtheorem` parancssal definiált környezetekkel hozhat létre.

```
\newtheorem{<tétnév>}{<tételcím>}
\newtheorem{<tétnév>}{<tételcím>}[<számlálóos>]
\newtheorem{<tétnév>}[<együtnév>]{<tételcím>}
```

<tétnév> Létrejön egy *<tétnév>* környezet és egy *<tétnév>* számláló, mely minden újabb ilyen környezet megnyitásakor növekszik eggyel.

<tételcím> Ez lesz a tételszerű bekezdés típuscíme (definíció, megjegyzés stb.). Ezen cím mellett megjelenik a *<tétnév>* számláló aktuális értéke is.

<számlálóos> Egy már korábban definiált számláló, általában valamelyik szint számlálója (chapter, section stb.). Ennek változásakor a *<tétnév>* nevű számláló lenullázódik. A *<számlálóos>* és a *<tétnév>* számláló együtt jelenik meg (például 2.1. tétel).

<együtnév> Egy másik tételszerű környezet neve. A *<tétnév>* és *<együtnév>* környezetek számlálói együtt fognak növekedni.

A létrehozott tételszerű környezetet az alábbi módon használhatja:

```
\begin{<tétnév>}[<egyedi cím>]
<A bekezdés szövege>
\end{<tétnév>}
```

Az *<egyedi cím>* megadása esetén, az a *<tételcím>* után jelenik meg zárójelben. Hivatkozni a tételszerű bekezdésekre az általános leírásnak megfelelően lehet. Például

```
\newtheorem{tétel}{tétel}
...
\begin{tétel}
```

```

A tétel szövege.
\end{tétel}
\begin{tétel}[Cauchy]\label{cauchy}
A következő tétel szövege.
\end{tétel}
\Aref{cauchy}.~tételből következően\dots

```

1. tétel. *A tétel szövege.*

2. tétel (Cauchy). *A következő tétel szövege.*

A 2. tételből következően...

```

\newtheorem{tétel}{tétel}[section]
\newtheorem{defin}[tétel]{definíció}
...
\section{Szakasz címe}
\begin{tétel}
A tétel szövege.
\end{tétel}
\begin{defin}
A definíció szövege.
\end{defin}

```

1. Szakasz címe

1.1. tétel. *A tétel szövege.*

1.2. definíció. *A definíció szövege.*

Alapesetben a tételszerű bekezdésekben a cím félkövéren, a szöveg pedig dőlten jelenik meg. Ezek stílusait magunk is beállíthatjuk a következő paranccsal:

```
\theoremstyle{<stílusnév>} ∈ amsthm
```

A *<stílusnév>* értékei a következők lehetnek:

- plain** A cím félkövér, a szöveg dőlt. Ez az alapérték.
- definition** A cím félkövér, a szöveg álló antikva.
- remark** A cím dőlt, a szöveg álló antikva.

Ha a tételszerű környezetnek nem akar számozást, akkor használja a következőt:

```
\newtheorem*{<tétnév>}{<tételcím>} ∈ amsthm
```

Matematikai tételek, lemmák, következmények bizonyítására van egy előre definiált **proof** környezet az amsthm csomagban. Például

```

\begin{proof}
A bizonyítás szövege.
\end{proof}

```

Bizonyítás. A bizonyítás szövege. □

A □ az ún. Q.E.D. jel. Ha nem akarunk Q.E.D. jelet, akkor írjuk be a következőt:

```
\renewcommand{\qedsymbol}{}

```

Ha a bizonyítás kiemelt matematikai képlettel zárul, akkor a képlet utáni sorba kerül a Q.E.D. jel, ami csúnya:

$$e^{i\pi} + 1 = 0.$$

□

Ilyenkor használja a `\qedhere` parancsot:

```
\begin{proof}
...
\[ \mathrm{e}^{i\pi} + 1 = 0. \qedhere \]
\end{proof}
```

$$e^{i\pi} + 1 = 0.$$

□

A `proof` környezet opcióval is használható. Például

```
\begin{tétel}\label{xy}
A tétel szövege.
\end{tétel}
\begin{proof}[\Aref{xy}.~tétel bizonyítása]
A bizonyítás szövege.
\end{proof}
```

1. tétel. *A tétel szövege.*

Az 1. tétel bizonyítása. A bizonyítás szövege.

□

11.6. Bibliográfia

Bibliográfiát `thebibliography` környezettel lehet készíteni, a bibliográfiai elemeket pedig a `\bibitem` paranccsal adhatja meg.

```
\begin{thebibliography}{\langle példacímke \rangle}
\bibitem{\langle kulcs \rangle} \langle elemleírás \rangle
...
\end{thebibliography}
```

\langle példacímke \rangle A bibliográfiai elemek címkéi közül a legszélesebb.

\langle kulcs \rangle A bibliográfiai elemre `\cite[\langle szöveg \rangle]{\langle kulcs \rangle}` paranccsal lehet hivatkozni a dokumentumban. Ilyenkor az adott ponton az adott elem címkéje [] jelek között jelenik meg. Egyszerre több kulcsot is megadhat, ezeket vesszővel kell elválasztani.

A *\langle szöveg \rangle*-ben például megadhatja, hogy melyik oldalra hivatkozik.

Magyar nyelvű dokumentum esetén a hivatkozások elé automatikus névelőt is rakhat az

```
\az{\cite[\langle szöveg \rangle]{\langle kulcs \rangle}}
\Az{\cite[\langle szöveg \rangle]{\langle kulcs \rangle}}
```

parancsokkal. Például

```
Lásd \cite{PlainTeX} és \cite[134.~oldal]{LaTeX}.
Lásd \cite{PlainTeX,LaTeX}.
\begin{thebibliography}{2}
\bibitem{PlainTeX} Bujdosó Gyöngyi, Fazekas Attila:
```



```
\TeX\ kezdőlépések, Budapest, 1997, Tertia Kiadó.  
\bibitem{LaTeX} Wettl Ferenc, Mayer Gyula, Szabó Péter:  
  \LaTeX\ kézikönyv, Budapest, 2004, Panem Könyvkiadó.  
\end{thebibliography}
```

Lásd [1] és [2, 134. oldal]. Lásd [1, 2].

Hivatkozások

- [1] Bujdosó Gyöngyi, Fazekas Attila: T_EX kezdőlépések, Budapest, 1997, Tertia Kiadó.
- [2] Wettl Ferenc, Mayer Gyula, Szabó Péter: L^AT_EX kézikönyv, Budapest, 2004, Panem Könyvkiadó.

Feladat

Szerkessze meg következő dokumentum L^AT_EX-forrását: [Struktúrált mű article osztályban](#). A feladat megoldása letölthető innen: [Megoldás](#).

12. fejezet

Elektronikus publikáció

Az elkészült dokumentumot átalakíthatja elektronikus publikációvá is. Ehhez töltsse be a `hyperref` csomagot. Ekkor az elkészült pdf fájlban automatikusan készül vázlatfa (bookmarks) és kis vázlatképek (thumbnails), továbbá linkké válnak a hivatkozások, URL címek.

Ha a `babel` és `geometry` csomagokat is használja, akkor azokat a `hyperref` után hívja meg.

A `hyperref` csomag néhány hasznos parancsa:

```
\url{<URL cím>}
```

Internetcímet adhat meg. Ez nem rakható parancsok argumentumaiba.

```
\href{<URL cím>}{<szöveg>}
```

Internetcímet adhat meg. A pdf-ben a `<szöveg>` jelenik meg, melyre kattintva betölti az `<URL cím>`-et. Ez nem rakható parancsok argumentumaiba.

```
\hyperref[<címke>]{<szöveg>}
```

Ennek helyén a `<szöveg>` felirat jelenik meg linkként. Erre kattintva a `\label{<címke>}`-vel létrehozott címkére ugrik.

A `hyperref` csomag néhány opciója:

`unicode` A vázlatfában helyesen jelenjenek meg az ő Ő ú Ű betűk.

`bookmarks=false` Ne készüljön vázlatfa. Alaphelyzetben készül.

`bookmarksopen` Alaphelyzetben a vázlatfában csak a legfelső szint látszik. Ezzel az opcióval minden szint nyitott lesz.

`bookmarksnumbered` A vázlatfában a címek legyenek számozottak.

`linktocpage` A jegyzékekben az oldalszámok legyenek a linkek. Alaphelyzetben a címek a linkek.

`breaklinks` Linkek sorvégi törésének engedélyezése.

`colorlinks` A linkek színes karakterrel legyenek kiemelve. Alaphelyzetben színes kerettel jelennek meg.

`hidelinks` A linkek ne legyenek színnel vagy kerettel kiemelve.

`hyperfootnotes=false` A lábjegyzet jelölője ne legyen link.

`pdfpagemode=FullScreen` A pdf megnyitásakor csak a lap jelenik meg a teljes képernyőn, a lehető legnagyobb nagyításban.

`pdfstartview=<érték>` Ha az `<érték>` `Fit`, akkor a pdf megnyitásakor az ablakban a lehető legnagyobb nagyítást alkalmazza. Ha `FitH`, akkor a pdf megnyitásakor az ablak

teljes szélességére nagyít. Ha `FitV`, akkor a pdf megnyitásakor az ablak teljes magasságára nagyít.

`linkcolor=<szín>` A `\ref` által létrehozott link színe.

`pagecolor=<szín>` A `\pageref` által létrehozott link színe.

`citecolor=<szín>` A `\cite` által létrehozott link színe.

`urlcolor=<szín>` Az `\url` és `\href` által létrehozott link színe.

`allcolors=<szín>` Minden link színe.

`linkbordercolor=<szín>` A `\ref` által létrehozott link keretének színe.

`citebordercolor=<szín>` A `\cite` által létrehozott link keretének színe.

`urlbordercolor=<szín>` Az `\url` és `\href` által létrehozott link keretének színe.

`allbordercolors=<szín>` Minden link keretének színe.

`pdfborder={0 0 <szám>}` A link keretének vastagsága `<szám>` pont (ha ez 0, akkor nincs keret).

Feladat

Szerkessze meg következő dokumentum \LaTeX -forrását: [Struktúrált elektronikus publikáció report osztályban](#). A feladat megoldása letölthető innen: [Megoldás](#).

13. fejezet

Szakdolgozat készítése

A `thesis-ekf` osztály olyan szakdolgozatok megírására alkalmas, amely megfelel az Eszterházy Károly Egyetem szabályzatának. Ebben a dokumentumosztályban a `geometry`, `hyperref` és `graphicx` csomagok automatikusan betöltődnek, így ezeket nem szabad ismét betölteni! A lehetséges opciók:

`twoside` Ha a szakdolgozatot kétoldalasan szeretné kinyomtatni, akkor ezt az opciót alkalmazza! Ne használja egyoldalas nyomtatáshoz illetve elektronikus verzióhoz!
`tocnopagenum` Ennek hatására a tartalomjegyzéknek nem lesz oldalszámozása. Ha közvetlenül a címoldalt követően van elhelyezve a tartalomjegyzék, akkor az első számozott oldal csak ezután következik.

A címoldal a `\maketitle` paranccsal hozható létre. Ehhez előtte az adatokat a következő parancsokkal lehet megadni:

```
\logo{<képbetöltés>}
```

Logó betöltéséhez kell használni. Például

```
\logo{\includegraphics{eszterhazy-logo-hu}}
```

Ha nem adja meg, akkor az Eszterházy Károly Egyetem logója fog automatikusan megjelenni. Ha nem akar logót, akkor írja be a `\logo{}` parancsot.

```
\institute{<intézet neve>}
```

Ezzel adja meg az intézmény nevét. Ha az Eszterházy Károly Egyetem logóját használja, akkor az egyetem nevét nem kell feltüntetni, mert azt a logó már tartalmazza. Ekkor elég csak az intézet neve. Például

```
\institute{Matematikai és Informatikai Intézet}
```

```
\title{<dolgozat címe>}
```

Ezzel adja meg a dolgozat címét.

```
\author{<név>\&\&<szak>}
```

Ezzel adja meg a szerző nevét és szakját. Például

```
\author{Tóth István\&\&matematika BSc}
```

```
\supervisor{<név>\&\&<beosztás>}
```

Ezzel adja meg a témavezető nevét és beosztását. Például

```
\supervisor{Dr. Nagy János\&\&főiskolai docens}
```

```
\city{<város>}
```

Ezzel adja meg a város nevét, ahol az intézmény található. Például

```
\city{Eger}
```

```
\date{<évszám>}
```

Ezzel adja meg a dolgozat leadásának évét. Az évszám után ne tegyen pontot!

Egy példa a használatra:

```
\documentclass[twoside]{thesis-ekf}
\usepackage[T1]{fontenc}
\PassOptionsToPackage{defaults=hu-min}{magyar.ldf}
\usepackage[magyar]{babel}
\begin{document}
\institute{Matematikai és Informatikai Intézet}
\title{A szakdolgozat címe}
\author{Szerző neve\\szak}
\supervisor{Tanár neve\\beosztás}
\city{Eger}
\date{2020}
\maketitle
\tableofcontents
\chapter{Fejezet címe}
\section{Szakasz címe}
\begin{thebibliography}{1}
\bibitem{cimke} \textsc{Szerző}: Cím, Kiadó, Hely, évszám.
\end{thebibliography}
\end{document}
```

A legegyszerűbb, ha a következő sablont használja: [klikk ide](#).



Videó: Szakdolgozat készítése

Feladat

Szerkessze meg következő dokumentum L^AT_EX-forrását: [Szakdolgozat készítése](#). A feladat megoldása letölthető innen: [Megoldás](#).

14. fejezet

Gyakorlatok

Oldja meg a következő gyakorlatokat! Ha egy probléma megoldását nem találja ebben a jegyzetben, akkor nézzen utána a [Tómacs Tibor: L^AT_EX](#) könyvben!

- [1. gyakorlat](#). Bekezdések, központozás, betűméretek, betűtípusok, igazítások, listák, térközök.
- [2. gyakorlat](#). Listák, táblázatok, úsztatás, kereszthivatkozások, lábjegyzetek.
- [3. gyakorlat](#). URL, képek, úsztatás, kereszthivatkozások.
- [4. gyakorlat](#). Saját úsztatott környezet, dobozok, többhasábos szedés, színek.
- [5. gyakorlat](#). Matematikai képletek.
- [6. gyakorlat](#). Verbatim, programkódok.
- [7. gyakorlat](#). Strukturált mű `article` dokumentumosztályban, tételszerű környezetek, matematikai képletek.
- [8. gyakorlat](#). Strukturált mű `report` dokumentumosztályban, margók, tartalomjegyzék, fej- és lábléc, irodalomjegyzék.
- [9. gyakorlat](#). Szakdolgozat készítése `thesis-ekf` dokumentumosztályban.
- [10. gyakorlat](#). Prezentáció készítése `beamer` dokumentumosztályban.