



L^AT_EX L^AP_OK

<https://tibortomacs.github.io/latex-tutorial-hu>

Verzió: 2025-01-30

© Tómacs Tibor

* O B C D E F H
Q M L A N a b c d e f g h i
P j k l m n o p q r s t u v w x y z
1 2 3 4 5 6 7 8 9 0
? ! % ' () * + , - . : ;
< > [\] ^ _ ` { | } ~

L^AT_EX L^AP_OK

TÓMÁCS TIBOR



Tómacs Tibor: L^AT_EX
című könyvének rövidített változata
kezdő felhasználók részére

Ha észrevétele, megjegyzése van, kérem írjon a következő címre:
tomacs.tibor@gmail.com



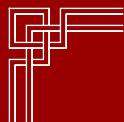
A pdf generálásához használt T_EX-rendszer:
pdfTeX, Version 3.141592653-2.6-1.40.26 (TeX Live 2024) kpathsea version 6.4.0

A borítógrafikát a [Freepik](#) tervezte.

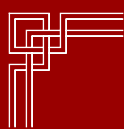


Tartalomjegyzék

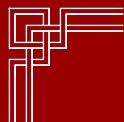
①	L^AT_EX-kurzus videókkal	7
②	Mi a L^AT_EX és hogyan kell telepíteni?	9
	T _E X-rendszerek	9
	Telepítés Windowsra	10
	Telepítés Linuxra	10
	A L ^A T _E X használata online	11
③	Az első L^AT_EX dokumentum elkészítése	12
	Parancs	12
	Dokumentumosztály, argumentum	12
	Környezet	13
	Opció	13
	Csomag, nyelv	13
	Fordítók, belső kódolás	14
	Bemeneti kódolás	15
	Hibakezelés	15
	A forrás és a pdf szinkronizálása	15
	Munkafájlok törlése	16
	Sablonok	16
④	A L^AT_EX alapfogalmai	17
	Parancs	17
	Kötelező argumentum	17
	Opcionális argumentum	17
	Környezet	18
	Blokk	18
	Deklarációs parancs	18
	Komment	19
	A forrásfájl szerkezete	19
	Fontosabb standard dokumentumosztályok	19
	Csomag	20
⑤	Alapvető formai elemek	22
	Szóközök	22
	Foglalt karakterek	23
	Ékezetes betűk	23
	Különleges karakterek	24



Kötőjel	24
Nagykötőjel	24
Gondolatjel	24
Hármaspont	25
Idézőjel	25
Betűváltozatok	25
Alak	26
Testesség	26
Család	26
Kiemelés	26
Betűméretek	26
A szavak elválasztása	27
Sortörések	27
Oldaltörések	28
Függőleges térközök	28
Bekezdések	28
Bekezdések balra zárása	28
Bekezdések jobbra zárása	29
Bekezdések középre zárása	29
Lábjegyzetek	29
Színek	30
Internetcímek	30
Számírás	31
Mértékegységek	31
Teljes példa	32
6 Oldalak kinézete	34
Oldalak szerkezete és méretei	34
Többhasábos szedés	35
Teljes példa	35
7 Listák	37
Számozatlan listák	37
Számozatlan listák extra térközök nélkül	37
Leíró listák	38
Számozott listák	39
Számozott listák extra térközök nélkül	39
Teljes példa	40
8 Kereszthivatkozások	42
9 Képek	44
Teljes példa	45
10 Táblázatok	46
Teljes példa	47



11	Objektumok úsztatása	49
	Képek és táblázatok úsztatása	49
	Úsztatott objektumok címkézése	50
	Teljes példa	50
12	Verbatim, programkódok	52
	Verbatim	52
	Programkódok	53
	Teljes példa	55
13	Képletek	57
	Matematikai mód	57
	Műveleti jelek	58
	Relációjelek	58
	Matematikai zárójelek	59
	Közönséges matematikai jelek	62
	Indexek, gyökjelek, törtek, binomiális együtthatók	62
	Kalligrafikus és dupla szárú betűk	62
	Matematikai ékezetek	62
	Három pont	62
	Görög betűk	63
	Változó hosszúságú vízszintes jelek	63
	Esetek szétválasztása	63
	Mátrixok	64
	Nagy operátorok	64
	„Nolimits” függvények	64
	„Limits” függvények	65
	Új függvények definiálása	65
	Differenciálás, differencia operátor	66
	Matematikai jelek több szerepben	67
	Kiemelt képletek sorszámozása	67
	Több képlet egymás alatt	68
	Több képlet egymás alatt illesztéssel	68
	Teljes példa	69
14	Strukturált művek	72
	Főcím, címlap, kivonat	72
	A főszöveg szintjei	72
	Fej- és láblécek	73
	Alapbeállítások	73
	Fej- és láblécek testreszabása	74
	Tartalomjegyzék	76
	Tételszerű bekezdések	77
	Bibliográfia	79
	Teljes példa	80



15 Elektronikus publikáció	87
A hyperref csomag néhány hasznos parancsa	87
A hyperref csomag néhány opciója	87
Teljes példa	88
16 Szakdolgozat készítése	90
A lehetséges opciók	90
Címoldal	90
Teljes példa	91
Irodalomjegyzék	93







L^AT_EX-kurzus videókkal























Ez a könyv a L^AT_EX magas szintű dokumentumleíró nyelv világába vezeti be az Olvasót. Az itt ismertetett tananyag alapjait az egri Eszterházy Károly Katolikus Egyetem Matematikai és Informatikai Intézetének „Számítógépes szöveg- és kiadványszerkesztés” című előadásai és gyakorlatai képezik.

A könyv kifejezetten a kezdők számára készült. Amennyiben ettől többet is szeretne tudni a rendszer használatáról, akkor olvassa el **Tómacs Tibor: L^AT_EX** című könyvét.



























Az alábbiakban ismertetjük az Eszterházy Károly Katolikus Egyetem L^AT_EX-kurzusának anyagát, amely önálló tanulásra is felhasználható. Minden leckéhez ikonokkal jelölt linkek tartoznak:

-  A leckéhez tartozó pdf. A gyakorlatok esetében ez az előállítandó dokumentum kimenete.
-  A leckéhez tartozó dokumentum megoldásának L^AT_EX-forrása zip fájlba csomagolva.
-  Amennyiben az **Overleaf** oldalon van regisztrációja, akkor ezzel a linkkel közvetlenül is betöltheti oda a lecke megoldásának L^AT_EX-forrását.
-  A lecke itt tekinthető meg videón.

A kurzus anyaga

- Bevezető 
- Telepítés 
- Tananyag 
- Az első dokumentum elkészítése   
- 1. gyakorlat – bekezdések, központosítás, betűméretek, betűtípusok, igazítások, listák, térközök    
- 2. gyakorlat – számozott listák testreszabása, táblázatok, úsztatás, kereszthivatkozások, lábjegyzetek    
- 3. gyakorlat – URL címek, képek, úsztatás, kereszthivatkozások    
- 4. gyakorlat – saját úsztatott környezetek készítése, dobozok, többhasábos szedés, színek kezelése    



5. *gyakorlat* – matematikai képletek    
6. *gyakorlat* – verbatim, programkódok szintaxis kiemelése (highlight)    
7. *gyakorlat* – strukturált dokumentum készítése `article` dokumentumosztályban, tételszerű környezetek    
8. *gyakorlat* – strukturált dokumentum készítése `report` dokumentumosztályban, oldal geometriája, tartalomjegyzék, fej- és lábléc, irodalomjegyzék, hivatkozás irodalomra    
9. *gyakorlat* – szakdolgozat `thesis`-ekf dokumentumosztályban    
Az Eszterházy Károly Katolikus Egyetem szakdolgozatsablonja  
10. *gyakorlat* – prezentáció készítése `beamer` dokumentumosztályban   
- Beadandó feladat az Eszterházy Károly Katolikus Egyetem hallgatóinak 



Mi a L^AT_EX és hogyan kell telepíteni?

DONALD ERVIN KNUTH 1977-ben egy olyan számítógépes programot fejlesztett ki, amely a nyomdászat minden tudását képes modellezni. A programot T_EX-nek keresztelte el, melynek kiejtése „tekh” vagy „tek”, mint a „technika” szóban.

A T_EX segítségével minden megoldható, ami egy kiadvány elkészítése során felmerülhet, de csak fáradságos úton, elemi parancsok használatával tudunk vele dolgozni. Ezért szükség volt olyan makrócsomag létrehozására, amely magasabb szintű programozási nyelven, jóval könnyebben kezelhető. Egy ilyen makrócsomag a L^AT_EX (kiejtése „latekh” vagy „latek”), amely egy magas szintű általános dokumentumleíró nyelvnek tekinthető. Ennek első publikus verzióját LESLIE B. LAMPORT készítette. A L^AT_EX folyamatosan és rendkívül dinamikus fejlődő rendszer.

A kiadvány L^AT_EX-forrása egy szöveges állomány, amely együtt tartalmazza a kiadvány szövegét és a formázáshoz szükséges L^AT_EX parancsokat. 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 éltetni, 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 kereszthivatkozások automatikusan számozódnak. A mai programok közül a L^AT_EX tudja a bekezdéseket a legoptimálisabban tördelni. Minden operációs rendszeren hozzáférhető, továbbá egy rendszeren megírt mű egy másik rendszeren is ugyanazt az eredményt adja, nincs áttördelési effektus.

Évente sok ezer könyv, cikk, oktatási segédanyag, szakdolgozat, doktori disszertáció stb. jelenik meg L^AT_EX-ben. Egyes tudományokban, mint a matematika, fizika, informatika, stb., a használata szabvánnyá vált, a legtöbb tudományos folyóirat csak ebben fogad el kéziratot. Magyarországon például a Typotex Kiadó minden kiadványa L^AT_EX rendszerben készül.

T_EX-rendszerek

A dokumentum L^AT_EX-forrásának pdf formátumba konvertálásához egy úgynevezett T_EX-rendszerre lesz szükségünk. Ilyen rendszer a TeX Live (<https://www.tug.org/texlive>) illetve a MiKTeX (<https://miktex.org>).

A szerkesztett dokumentum forrása egy szöveges állomány, amit bármely editoron létrehozhat. Azonban sokkal könnyebb a szerkesztés, ha olyan editoron dolgozik, amely a L^AT_EX-re lett optimalizálva. Számos ilyen létezik, de véleményem szerint a TeXstudio (<https://www.texstudio.org>) a legjobb.



Telepítés Windowsra

A TeX Live telepítéséhez töltsse le a telepítésvezérlőjét (<https://mirror.ctan.org/systems/texlive/tlnet/install-tl-windows.exe>), futtassa, majd kövesse az utasításokat. Ha serverhiba miatt nem sikerül a letöltés, akkor próbálja ezt a linket: <https://ctan.math.washington.edu/tex-archive/systems/texlive/tlnet/install-tl-windows.exe>. A telepítés kb. egy-két óráig tart. A TeXstudio telepítéséhez töltsse le a telepítőjét (<https://www.texstudio.org>), indítsa el a telepítő fájlt, majd kövesse az utasításokat.

Windowson egy másik lehetőség a *TeXfireplace* (<https://tibortomacs.github.io/texfireplace/>) kompakt keretrendszer, amely egyszerre telepíti a MiKTeX rendszert és a TeXstudiót, továbbá számos olyan kiegészítőt, amely megkönnyíti a rendszer használatát. Ennek telepítése kb. 5-10 perc.

Telepítés Linuxra

A TeX Live telepítéséhez töltsse le a telepítésvezérlőjét (<http://mirror.ctan.org/systems/texlive/tlnet/install-tl-unx.tar.gz>). Ha serverhiba miatt nem sikerül a letöltés, akkor próbálja ezt a linket: <https://ctan.math.washington.edu/tex-archive/systems/texlive/tlnet/install-tl-unx.tar.gz>. A kicsomagolt mappában nyissa meg a terminált. Parancssorba írja be a következőt:

```
sudo ./install-tl
```

Amikor választani kell a TeX Live telepítési lehetőségei közül, akkor írja be, hogy

```
I
```

Várjon a telepítés végéig. A TeX Live kényelmes használatához meg kell adni a pdf_latex program elérési útvonalát. Tegyük fel például, hogy ez az

```
/usr/local/texlive/2023/bin/x86_64-linux
```

mappa. Ekkor írja parancssorba a következőt:

```
sudo gedit /etc/environment
```

Az environment fájlban ki kell egészíteni a

```
PATH="ÚTVONALAK"
```

sort erre:

```
PATH="/usr/local/texlive/2023/bin/x86_64-linux:ÚTVONALAK"
```

Természetesen az `ÚTVONALAK` helyére azt kell írni, ami az environment fájlban eredetileg volt. Mentse az environment fájlt, majd lépjen ki a gedit programból. Ubuntu esetén a gyöker PATH más, mint a normál PATH, ezért még ezt is be kell állítani. Parancssorba írja be, hogy



```
sudo gedit ~/.bashrc
```

majd a `.bashrc` fájlban írja be a következő sort az első sorba:

```
alias sudo='sudo env PATH=$PATH'
```

Mentse a `.bashrc` fájlt és indítsa újra a számítógépet!

A TeXstudio telepítéséhez töltsse le a telepítőjét (<https://www.texstudio.org>), indítsa el a telepítő fájlt, majd kövesse az utasításokat.

A L^AT_EX használata online

Az *Overleaf* (<https://www.overleaf.com>) weboldal regisztrálás után internetes böngészőben ad szerkesztési lehetőséget, továbbá a végeredményt jelentő pdf fájlt egy szerveren található TeX Live rendszer generálja. A rendszer előnyei:

- Saját gépre nem kell telepíteni TeX-rendszert illetve L^AT_EX-editort,
- a dokumentumokat felhőben tárolja,
- a kezelése könnyű és gyorsan átlátható,
- jól dokumentált,
- az alapszolgáltatások ingyenesek,
- az ingyenes verzióban egy másik felhasználóval megosztható a dokumentumunk, prémium tagság esetén többel is,
- rengeteg sablont tartalmaz, bár sokat lelkes amatőrök készítettek, így ezek nem biztos, hogy megfelelnek minden tipográfiai követelménynek.

Vannak hátrányai is az Overleaf használatának:

- Nagyobb terjedelmű dokumentumot még prémium verzióban sem lehet pdf-be konvertálni a fordítási idő korlátozása miatt,
- nem a legfrissebb TeX Live rendszerrel dolgozik,
- a szerkesztő funkciói szegényesek a TeXstudióhoz képest.



Az első L^AT_EX dokumentum elkészítése

Először hozzon létre egy mappát, amibe majd a forrásfájlt elmenti. Fontos, hogy minden dokumentum külön mappában legyen, mert egy dokumentumhoz több fájl is fog tartozni.

Ezután TeXstudióban nyisson meg egy új dokumentumot a **Fájl** **Új** menüvel. Írja be a következőket:

```
\documentclass{article}
\begin{document}
Helló, világ!
\end{document}
```

Mentse el a forrásfájlt a **Fájl** **Mentés** menüvel a korábban létrehozott mappában. Ennek kiterjesztése tex lesz. Az így elkészült forrásfájlt konvertálja pdf fájlba az **Eszközök** **Fordítás és megjelenítés** menüvel vagy az **F5** billentyűvel. A végeredmény megjelenik a TeXstudióban. Az elkészült pdf fájl abban a mappában lesz, ahol a forrásfájl is található, de emellett további úgynevezett *munkafájlok* is generálódtak.

A következőkben megismerkedünk az előző kód részleteivel. Ennek kapcsán megemlítünk néhány fontos fogalmat, ami a L^AT_EX használatánál nélkülözhetetlen.

Parancs

A L^AT_EX alapvető eleme az úgynevezett *parancs*. Minden parancs `\` fordított per-jellel (backslash) kezdődik, amit a parancs neve követ. A parancs neve néhány kivételtől eltekintve általában csak az angol ábécé betűiből állhat, továbbá lehet kis- és nagybetű is. Ilyen parancs például az előző kód első sorában található `\documentclass`. A parancs nevében nem mindegy, hogy kis- vagy nagybetű áll, azaz például `\Documentclass` nem írható `\documentclass` helyett.

Dokumentumosztály, argumentum

A `\documentclass` parancssal kell megadni, hogy milyen osztályba soroljuk a szerkesztett kiadványt. A dokumentumosztály nevét a `\documentclass` parancs argumentumában kell megadni, amit a parancs után kell írni kapcsos zárójelek, azaz `{` és `}` közé. Jelen esetben az `article` dokumentumosztályt használjuk, amit tehát `\documentclass{article}` módon adunk meg.



Környezet

Az előző kód második sorában a `\begin` parancs áll. Ezzel kezdünk el valamit. Most például a dokumentum szövegét szeretnénk elkezdni, amit a `\begin` parancs argumentumába írt `document` szóval tehetünk meg, azaz a `\begin{document}` paranccsal. A dokumentum végét az `\end{document}` jelzi. Minden `\begin` parancsot le kell zárni egy `\end` paranccsal. Egy ilyen párost *környezetnek* nevezünk. A környezet neve a `\begin` és `\end` parancsok argumentuma, azaz például a `\begin{document}` `\end{document}` páros a `document` környezet, melybe a dokumentum szövege kerül minden olyan további paranccsal együtt, amely a szöveg formázását határozza meg.

Opció

A `\documentclass{article}` parancs hatására a dokumentum fő szövege 10 pt betűmérettel fog megjelenni. Ez azért van így, mert az `article` dokumentumosztályban ez az alapbeállítás. De ez a dokumentumosztály elérhetővé teszi még a 11 pt és 12 pt méreteket is. Ha például a 12 pt betűméretre szeretnénk áttérni, akkor ezt az úgynevezett opcionális argumentumban kell megadni szögletes zárójelek, azaz `[és]` között az alábbi módon:

```
\documentclass[12pt]{article}
\begin{document}
Helló, világ!
\end{document}
```

Az `article` dokumentumosztálynak más opciói is vannak. Például az `a4paper` opció az A4 papírmérethez igazítja a margókat. Ez nem alapopció, ezért ha aktiválni akarja, akkor be kell írni az opciók közé. Ha több opciót is megad, akkor azokat vesszővel kell elválasztani. Például

```
\documentclass[12pt,a4paper]{article}
\begin{document}
Helló, világ!
\end{document}
```

Csomag, nyelv

A betöltött dokumentumosztály képességeit úgynevezett *csomagokkal* lehet kiegészíteni. Egy csomagot a `\usepackage` paranccsal lehet betölteni a `\documentclass` után, de még a `\begin{document}` előtt. A csomag nevét a `\usepackage` argumentumában kell megadni. Például a `babel` csomag segítségével lehet megadni a dokumentum nyelvét. A következő kód 2. sorában betöltjük a `babel` csomagot.



```
\documentclass[12pt,a4paper]{article}
\usepackage{babel}
\begin{document}
Helló, világ!
\end{document}
```

A nyelvet a babel csomag opciójaként kell megadni. Például magyar nyelvre így lehet beállítani:

```
\documentclass[12pt,a4paper]{article}
\usepackage[magyar]{babel}
\begin{document}
Helló, világ!
\end{document}
```

Ugyanez angol esetén:

```
\documentclass[12pt,a4paper]{article}
\usepackage[english]{babel}
\begin{document}
Hello, World!
\end{document}
```

A magyar nyelvi beállítások esetén van még egy kiegészítési lehetőség, amit a babel betöltése előtt kell megadni:

```
\documentclass[12pt,a4paper]{article}
\PassOptionsToPackage{defaults=hu-min}{magyar.ldf}
\usepackage[magyar]{babel}
\begin{document}
Helló, világ!
\end{document}
```

Ezzel a magyar nyelv „finomabb” tipográfiai beállításai is aktiválódnak.

Fordítók, belső kódolás

A L^AT_EX forrásfájl pdf fájlba konvertálásához három úgynevezett *fordító* áll a rendelkezésünkre:

- pdflatex
- xelatex
- lualatex

A legtöbb L^AT_EX editor, így a TeXstudio esetén is az alapértelmezett fordító a pdflatex. Ez a DONALD ERVIN KNUTH által 1977-ben létrehozott eredeti fordító továbbfejlesztett verziója. Mi is ezt fogjuk használni.

A pdflatex fordító használata esetén alapbeállítás szerint a pdf fájlban található ékezetes betűk két karakterként jelennek meg, külön az alapbetű és külön a rátett ékezet. Ez elsőre nem tűnik gondnak, de valójában több problémát is okoz.



Például hosszabb szöveg esetén a sorvégi szavak elválasztásánál megakadályozza azon szótagok elválasztását, melyekben ékezetes betű szerepel. Így egy bekezdés tördelése nem lesz ideális. Ahhoz, hogy ez ne így legyen, az úgynevezett *belső kódolást* kell átállítani. Ehhez a fontenc csomagot kell betölteni T1 opcióval:

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

Bemeneti kódolás

Ahhoz, hogy a belső kódolás felismerje, hogy milyen betűt írtunk a forrásfájlunkba, ahhoz az úgynevezett *bemeneti kódolás* típusát kell tudnia a pdf_latex fordítónak. A pdf_latex alapesetben azt feltételezi, hogy UTF-8 a bemeneti kódolás. Mivel a TeXstudio – mint a legtöbb L^AT_EX editor – alapesetben UTF-8 kódolásra van állítva, ezért a kimenet helyes lesz, vagyis ezzel külön nincs teendőnk.

Hibakezelés

Ezen a ponton érdemes kipróbálni a hibakezelést. Például a `\begin{document}` parancsot írja át rosszra, mondjuk így: `\Begin{document}`. Ezután fordítsa le a forráskódot (azaz konvertálja pdf-be). Ekkor egy hibaüzenetet kap, mert a `\Begin` parancs nincs definiálva:

Undefined control sequence. \Begin

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

A forrás és a pdf szinkronizálása

Nagyobb terjedelmű dokumentum esetén sok munkát meg lehet spórolni, ha a forrásfájl adott pozíciójából a pdf fájl megfelelő pozíciójába tud ugrani és viszont. Ezt a célt szolgálja a *SyncTeX* program. A pdf-be konvertálás után TeXstudióban tartsa nyomva a **Ctrl** billentyűt, majd az egér bal gombjával kattintson a forrásfájlban a megfelelő szövegrészre. Ekkor a TeXstudio átugrik a pdf fájl megfelelő részére. Ez visszafelé is működik.



Munkafájlok törlése

Mielőtt bezárná a TeXstudiót, még egy feladatot el kell végezni. A munka elején megnyitott mappában a tex és pdf fájlokon kívül néhány munkafájl is létrejött. Többek között egy log kiterjesztésű naplófájl is, ami az esetlegesen rosszul begépett forráskódból származó hibákat is rögzíti. A munka végeztével ezeket érdemes törölni, amit TeXstudióból könnyen megtehet: **Eszközök** » **Segédfájlok törlése**.

A TeXstudio rengeteg kényelmi szolgáltatást biztosít, melyek segítségével sokkal gyorsabban állíthatja elő a L^AT_EX-forrást. Ezeket ebben a jegyzetben nem tárgyaljuk, hiszen a TeXstudio újabb verzióinak kiadásával megváltozhatnak. Ezért ezeket a funkciókat célszerű önállóan felfedezni és megtanulni a használatukat.

Sablonok

Összegzésként, érdemes a következő sablonokat használni egy dokumentum elkészítésénél kiindulási alapként:

L^AT_EX-sablon magyar nyelvű dokumentumokhoz

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

L^AT_EX-sablon angol nyelvű dokumentumokhoz

```
\documentclass[12pt,a4paper]{article}
\usepackage[T1]{fontenc}
\usepackage[english]{babel}
\begin{document}
Hello, World!
\end{document}
```




A L^AT_EX alapfogalmai

Az első dokumentum elkészítésekor pár fogalomra már kitértünk, most nézzük egy kicsit részletesebben is.

Parancs

A L^AT_EX-ben a dokumentum formázása *parancsokkal* történik. A parancs `\` fordított perjellel (backslash) kezdődik, majd ezt követi a parancs neve, melyben ékezetes betű, szám és szóköz nem szerepelhet, továbbá kis- és nagybetű között különbséget tesz. Például a

```
\Large
```

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

Kötelező argumentum

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

```
\textit{szöveg}
```

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

```
\setcounter{page}{1}
```

az oldalszámot 1-re állítja.

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 `[` és `]` jelek között. Például egy listaelem bevezethető az

```
\item
```

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

```
\item[-]
```



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

```
\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.

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}
```

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.

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 szöközt elnyelik. Például

```
AAA\itshape BBB
```

eredménye

```
AAABBB
```

Komment

Ha a forrásállományba úgynevezett *kommentet* akar elhelyezni, vagyis amit a 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,
```

A forrásfájl szerkezete

A L^AT_EX 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 fordító figyelmen kívül hagyja.

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 T_EX-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 később).

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.

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 kellene. 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 majd külön jelezzük. Például az `\euro` parancs az `eurosym` csomag betöltésével használható, amit így fogunk jelölni:

```
\euro % \usepackage{eurosym}
```



Alapvető formai elemek

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,  
né%  
gy, 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
```

```
LATEXkézikönyv
```

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`

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 határolójele
<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>

Ékezetes betűk

Ékezetes betű közvetlenül a billentyűzetről is bevihető. Ha olyan ékezetes betűre van szüksége, amely nincs a billentyűzeten, akkor használhatja a következő parancsokat:

<code>ó</code>	<code>\' {o}</code>	<code>ò</code>	<code>\` {o}</code>	<code>ō</code>	<code>\= {o}</code>	<code>ř</code>	<code>\v {o}</code>	<code>q̇</code>	<code>\k {o}</code>
<code>ő</code>	<code>\H {o}</code>	<code>ô</code>	<code>\^ {o}</code>	<code>ó</code>	<code>\. {o}</code>	<code>ř</code>	<code>\r {o}</code>	<code>q̇</code>	<code>\d {o}</code>
<code>ö</code>	<code>\" {o}</code>	<code>õ</code>	<code>\~ {o}</code>	<code>ő</code>	<code>\u {o}</code>	<code>q̇</code>	<code>\c {o}</code>	<code>q̇</code>	<code>\b {o}</code>

Az o betű helyére természetesen bármilyen más betű írható.



Különleges karakterek

Felsorolunk néhány gyakran használt karakter:

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

Kötőjel

A kötőjel forrásállományban - módon 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 -- módon 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 is nagykötőjel, így forrásállományban -- módon 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. Amerikai 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. Brit angol szövegben ezt kell tenni:

```
`text text'
```

```
‘text text’
```

Betűváltozatok

A betűváltozatokat alakjuk, testességük és családjuk szerint osztályozzuk.



Alak

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

Testesség

Normál (medium)	<code>\textmd{SZÖVEG}</code> , <code>{\mdseries SZÖVEG}</code>
Félkövér (boldface)	<code>\textbf{SZÖVEG}</code> , <code>{\bfseries SZÖVEG}</code>

Család

Antikva (roman)	<code>\textrm{SZÖVEG}</code> , <code>{\rmfamily SZÖVEG}</code>
Groteszk (sans serif)	<code>\textsf{SZÖVEG}</code> , <code>{\sffamily SZÖVEG}</code>
Írógép (typewriter)	<code>\texttt{SZÖVEG}</code> , <code>{\ttfamily SZÖVEG}</code>

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 betűválozatot használ, de ideiglenesen vissza akar arra térni (álló, normál, antikva), akkor használja a

```
\textnormal{SZÖVEG}, {\normalfont SZÖVEG}
```

parancsokat. Az első több bekezdésre nem használható.

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 betűválozatot, és aszerint emelnek ki. Álló alak esetén dőlt, nem álló alak esetén álló alakra vált.

Betűméretek

Az alapbetűmérethez relatív betűméretek:

szöveg	<code>{\tiny szöveg}</code>	szöveg	<code>{\footnotesize szöveg}</code>
szöveg	<code>{\scriptsize szöveg}</code>	szöveg	<code>{\small szöveg}</code>



szöveg `{\normalsize szöveg}`
szöveg `{\large szöveg}`
szöveg `{\Large szöveg}`

szöveg `{\LARGE szöveg}`
szöveg `{\huge szöveg}`
szöveg `{\Huge szöveg}`

A 12 pt alapbetűméret esetén – mint ez a dokumentum is – a `\huge` és `\Huge` között nincs különbség, de 10 pt és 11 pt esetén igen.

A szavak elválasztása

A L^AT_EX alpból 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 L^AT_EX csak a kötőjelnél tudja elválasztani. Ha ezt felül akarja bírálni, akkor a kötőjel elé gépeljen fordított aposztrófjelet:

```
egyszer` -kétszer % \usepackage[magyar]{babel}
```

Ekkor a kötőjelnél és minden szótagnál el tud választani. Amennyiben nem aktív a magyar nyelv, akkor az előző megoldás helyett használja a `hyphenat` csomag `\hyph{}` parancsát. Például

```
electromagnetic\hyp{}endioscopy % \usepackage{hyphenat}
```

Néha szükség lehet egy adott szó elválasztásának a tiltására is. Ekkor tegye azt az `\mbox` parancs argumentumába. Például

```
\mbox{karóra}
```

Sortörések

A L^AT_EX automatikusan végzi a sortöréseket, de adott esetben ki is kényszerítheti. Sortörést soha ne alkalmazzon új bekezdés nyitáshoz!

```
\\
```

Új sort kezd sorkizárás nélkül.

```
\\[HOSSZ]
```



Ugyanaz mint a `\\` de a következő sor távolsága `HOSSZ` mérettel megnő. Például

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

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).

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.

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!

Bekezdések balra zárása

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



```
\begin{flushleft}  
SZÖVEG  
\end{flushleft}
```

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}
```

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.

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}
```

Színek

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

 black	 gray	 olive	 teal
 blue	 green	 orange	 violet
 brown	 lightgray	 pink	 white
 cyan	 lime	 purple	 yellow
 darkgray	 magenta	 red	

Színezéséhez a következő parancsot használhatja:

```
{\color{SZÍNNÉV}SZÖVEG} % \usepackage{xcolor}
```

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} % \usepackage{xcolor}
```

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.

Internetcímek

Internetcímek az url csomag következő parancsával adhatók meg:

```
\url{URL CÍM}
```

Például

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

http://www.tug.org



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} % \usepackage{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} % \usepackage{siunitx}
```

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

```
\num{1234567}
```

```
1.234.567
```

Mértékegységek

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

```
123\,cm 1200\,km 50\,\% 1000\,Ft 500\,\$ 10\,\AA
```

```
123 cm 1200 km 50 % 1000 Ft 500 $ 10 Å
```

vagy



```
20\,\textcelsius
1\,\textperthousand
5\,\textpertenthousand
```

```
20°C 1‰ 5‱
```

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} % \usepackage{siunitx}
```

paranccsal oldhatja meg. Például

```
\ang{1;;} vagy \ang{2;3;} vagy \ang{4;5;6}
```

```
1° vagy 2°3' vagy 4°5'6''
```

Ugyan ez az siunitx csomag nélkül így oldható meg:

```
$1^\circ$ vagy $2^\circ3'$ vagy $4^\circ5'6''$
```

Teljes példa

Betűtípus és -méret, térköz, törés, bekezdés, lábjegyzet, szín

```
\documentclass[a4paper,12pt]{article}
\usepackage[T1]{fontenc}
\PassOptionsToPackage{defaults=hu-min}{magyar.ldb}
\usepackage[magyar]{babel}
\usepackage{xcolor}
\footnotestyle{rule=fourth}
```

```
\begin{document}
```

```
\begin{flushright}
```

```
\footnotesize\em
```

```
„A jó matematikus nem halmozza az élvezeteket,\\
hanem élvezi a halmazokat.”
```

```
\end{flushright}
```

A halmazelmélet fiatal tudományág, létrejött a XIX. század második felére tehető, ami nem véletlen, hiszen a halmazok vizsgálatához nagyfokú absztrakció szükséges. Ekkorra érték el a matematikai kutatások olyan szintet, hogy az ilyen absztrakció szükségessé és lehetővé vált.

A matematikusok figyelme a halmazok elemeiről a halmazokra irányult. Olyan problémák vezettek ide, melyeket bizonyos halmazokra anélkül sikerült megoldani, hogy azokat az egyes halmazelemekre vonatkoztatták volna (pl. biztosítási matematika, kinetikus gázelmélet).



A legdöntőbb momentum az volt, amikor a végtelen sorok vizsgálata közben felismerték, hogy a véges halmazok tulajdonságaival nem rendelkeznek törvényszerűen a `\emph{végtelen halmazok}` is.

A ma `\emph{naiv halmazelméletnek}` nevezett rendszer megalkotója `\textsc{Georg Cantor}` (1845--1918) volt, akitől a halmaz fogalmának az alábbi körülírása származik: „A halmaz meghatározott, különböző, képzeletünkben vagy gondolatainkban fölfogott dolgok összessége. A kérdéses dolgok a halmaz elemei.”

A továbbiakban az alapvető halmazelméleti fogalmakat -- részhalmaz, halmazok egyenlősége, műveletek, számosság stb. --- ismertnek tételezzük fel, hiszen az analízis tárgyalásakor ezeket az olvasó megismerte.

A végtelen halmazok elméletének kezdettől fogva voltak bírálói, de addig szilárd elmélet volt, míg logikai ellentmondásokat nem fedeztek fel benne. Egyike ezeknek a Russell-féle antinómia`\footnote{Több alakja is ismert, az itt olvasható verzió talán a legnépszerűbb.}`:

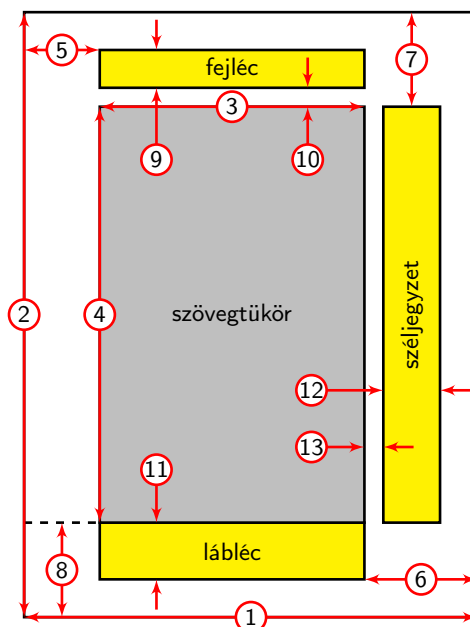
`\emph{\color{blue}`A falu borbélyja az a férfi a faluban, aki azokat és csak azokat a férfiakat borotválja meg a faluban, akik nem maguk borotválkoznak. Kérdés, hogy borotválkozik-e a borbély?`}`

`\end{document}`

Oldalak kinézete

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 a landscape opciót is használja:

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

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

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

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

Többhasábos szedés

Többhasábos szedés esetén használja a multicol csomag multicol environmentjét:

```
\begin{multicols}{HASÁBSZÁM} % \usepackage{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}
```

Teljes példa

Oldal méretei, többhasábos szedés

```
\documentclass[12pt]{article}
\usepackage[a4paper, top=25mm, bottom=25mm,
             outer=25mm, inner=30mm]{geometry}
\usepackage[T1]{fontenc}
\PassOptionsToPackage{defaults=hu-min}{magyar.ldf}
\usepackage[magyar]{babel}
```



```
\usepackage{multicol}
```

```
\begin{document}
```

Ez a dokumentum A4 lapméretű, továbbá minden margó 25\,mm, kivéve a belsőt, amely 30\,mm. A következőkben egy kéthasábos szedést láthatunk.

```
\begin{multicols}{2}
```

A hosszúság, terület, térfogat, ívhossz, felszín, egyszerű alakzatokra már az ókori görögök által definiáltak és számolhatóak voltak.

A sokszögek területének és a poliéderek térfogatának fogalmát először `\textsc{Peano}` és `\textsc{Jordan}` terjesztették ki a sík illetve a tér részhalmazainak egy nagyobb rendszerére a XIX.-század végén. Eszerint egy síkbeli korlátos halmaz külső mértéke legyen az őt lefedő véges sok sokszögből álló alakzatok területének pontos alsó korlátja, belső mértéke pedig a benne fekvő véges sok sokszögből álló alakzatok területének pontos felső korlátja. Ha ezek egyenlőek, akkor a halmazt mérhetőnek, ezen közös értéket pedig a halmaz mértékének nevezzük. Térfogat esetén hasonló az eljárás.

Ez a mértékfogalom egyszerű, de az integrálás céljára nem megfelelő. Az általánosítás területén a fő lépést `\textsc{Lebesgue}` tette meg a XX.-század elején. Az általa alkotott mérték és integrál előnye a nagyobb általánosság, az integrál és a határátmenet felcserélhetősége.

```
\end{multicols}
```

```
\end{document}
```



Listák

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.
```

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} % \usepackage{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.
```

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

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

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.
```

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} % \usepackage{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.
```

Teljes példa

Listák

```
\documentclass[a4paper,12pt]{article}  
\usepackage[T1]{fontenc}  
\PassOptionsToPackage{defaults=hu-min}{magyar.ldf}  
\usepackage[magyar]{babel}  
  
\begin{document}  
  
\begin{enumerate}  
  \item Hány olyan tompaszögű háromszög van, melyben a szögek mérőszáma  
    fokokban három különböző egész szám?  
  \begin{enumerate}  
    \item Adjon részletes indoklást!  
    \item Szerkessze meg az egyik megoldást!  
  \end{enumerate}  
  \item Mi a feltétele annak, hogy egy derékszögű háromszög  
    súlyvonalaiából, mint oldalakból szerkesztett háromszög derékszögű  
    legyen?  
  \begin{enumerate}  
    \item Adjon részletes indoklást!  
    \item Oldja meg úgy is a feladatot, ha az egyik oldalról tudjuk,  
    hogy 3\,cm hosszú!
```




```
\end{enumerate}
\item Két ikerprímszám összege osztható 12-vel, ha a prímszámok 3-nál
nagyobbak.
\begin{enumerate}
\item Írja le az ikerprímszám definícióját!
\item Bizonyítsa be az állítást!
\end{enumerate}
\end{enumerate}

\end{document}
```



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 L^AT_EX-re bizzuk a számozott elemeknél és a kereszthivatkozásoknál a megfelelő számok beírását.

Ha egy számozott elemről kiderül, hogy hivatkozni kell rá, akkor először ezt az elemet címkézzük 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

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

esetén, 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.
```

Helyesebb lenne a mondat, ha a sorszám elé határozott névelőt rakna: „az 1.”, „a 2.”, stb. A magyarban a névelő függ a sorszámtól. Ezt a problémát megoldja a babel csomag magyar opciója. Ilyenkor használja az

```
\aref{CÍMKE} % \usepackage[magyar]{babel}
\Aref{CÍMKE} % \usepackage[magyar]{babel}
```



vagy rómitól különböző számozás esetén az ezekkel egyenértékű

```
\az{\ref{címké}} % \usepackage[magyar]{babel}
\Az{\ref{címké}} % \usepackage[magyar]{babel}
```

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

```
Lásd \aref{lista-proba}.~listaelemet.
\Aref{lista-proba}.~listaelemben olvasható.
```

Lásd a 2. listaelemet. A 2. listaelemben olvasható.

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

```
\pageref{címké}
```

paranccsal írathatja ki. Ennek névelős verziói:

```
\apageref{címké} % \usepackage[magyar]{babel}
\Apageref{címké} % \usepackage[magyar]{babel}
```

melyek rómitól különböző oldalszámozás esetén ezekkel egyenértékűek:

```
\az{\pageref{címké}} % \usepackage[magyar]{babel}
\Az{\pageref{címké}} % \usepackage[magyar]{babel}
```

Például

```
\Aref{lista-proba}.~listaelemet
\apageref{lista-proba}.~oldalon találjuk.
```

A 2. listaelemet a 42. oldalon találjuk.

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

Képek

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

```
\includegraphics[OPCIÓK]{KÉPFÁJL} % \usepackage{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 a könyvtárban kell lennie. Azonban praktikusabb, ha a képfájlokat az aktuális mappa egy almappjába teszi. Ennek az almappának célszerű olyan nevet adni, amelyben nincs ékezetes betű és szóköz. Például, ha a `grafikonok` nevű almappba 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.

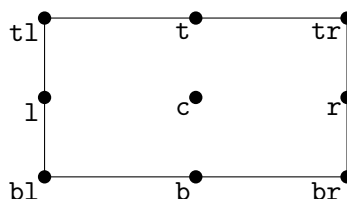
Az `\includegraphics` parancs fontosabb opciói:

`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}
```

Teljes példa

Képek

```
\documentclass[a4paper,12pt]{article}
\usepackage[T1]{fontenc}
\PassOptionsToPackage{defaults=hu-min}{magyar.ldf}
\usepackage[magyar]{babel}
\usepackage{graphicx,url}
```

```
\begin{document}
```

A következő kép innen tölthető le:

```
\begin{center}
\url{https://tibortomacs.github.io/latex-tutorial-hu/lion.pdf}
\end{center}
```

Ez az oroszlán ma már a `\LaTeX` szimbólumává vált. A szélessége legyen `5\,cm`!

```
\begin{center}
\includegraphics[width=5cm]{lion}
\end{center}
```

Gyakorlásképpen forgassuk el:

```
\begin{center}
\includegraphics[width=5cm,origin=c,angle=90]{lion}
\includegraphics[width=5cm,origin=c,angle=-90]{lion}
\end{center}
```

```
\end{document}
```

Táblázatok

A táblázatok elkészítése az egyik legbonyolultabb feladat a L^AT_EX-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|}{Jövedelem (Ft)}&\multicolumn{2}{c|}{év}\\\cline{2-3}  
\multicolumn{1}{l|}{Adó (Ft)}&\multicolumn{2}{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} % \usepackage{multirow}  
\multirow{CELLASZÁM}{SZÉLESSÉG}{SZÖVEG} % \usepackage{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

Teljes példa

Táblázat

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



```
\begin{center}
\begin{tabular}{|l|r|r|r|r|}
\multicolumn{1}{l}{} &
\multicolumn{4}{c}{év}\\
\cline{2-5}
\multicolumn{1}{l}{} &
\multicolumn{1}{c}{\emph{2008}} &
\multicolumn{1}{c}{\emph{2009}} &
\multicolumn{1}{c}{\emph{2010}} &
\multicolumn{1}{c}{\emph{2011}}\\
\hline
\emph{jövedelem (Ft)} & 994\,000 & 1\,231\,500 & 1\,525\,410 &
2\,321\,600\\
\emph{járulék (Ft)} & 165\,000 & 194\,950 & 215\,750 &
235\,850\\
\hline
\end{tabular}
\end{center}

\end{document}
```


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 az objektum nem feltétlenül a forrásban megadott helyre kerül, hanem egy általunk engedélyezett pozícióba: az aktuális oldal tetejére, aljára vagy külön oldalra.

Képek és táblázatok úsztatása

Képek úsztatására a `figure`, míg táblázatok úsztatására a `table` környezet használható. Ezen környezetek opciói:

- `h` Maradjon helyben, ha lehetséges.
- `t` Az aktuális oldal tetejére kerüljön.
- `b` Az aktuális oldal aljára kerüljön.
- `p` Külön oldalra kerüljön.
- `!` Figyelmen kívül hagy minden korlátozást, amely a területre helyezhető úsztatások számával vagy a terület által elfoglalható maximális mérettel kapcsolatos.

Opciónak ezen jelek bármilyen kombinációja használható. A feldolgozás sorrendje minden esetben `!htbp`. Tehát mindegy, hogy például `tb` vagy `bt` van opcióként megadva, először mindenképpen az oldal tetején próbálja elhelyezni az objektumot és csak azután alulra. 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.

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}\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

Az 1. ábrán látható ...

Teljes példa

Képek és táblázatok úsztatása

```
\documentclass[a4paper,12pt]{article}
\usepackage[T1]{fontenc}
\PassOptionsToPackage{defaults=hu-min}{magyar.ldf}
\usepackage[magyar]{babel}
\footnotestyle{rule=fourth}
\usepackage{graphicx,url}
```



```
\begin{document}

\Az{\ref{fig-lion}}.~ábra\footnote{Forrás:
\url{https://tibortomacs.github.io/latex-tutorial-hu/lion.pdf}}
szélessége legyen 5\,cm!

\begin{figure}[ht!]
\centering
\includegraphics[width=5cm]{lion}
\caption{A \LaTeX\ szimbóluma}
\label{fig-lion}
\end{figure}

\begin{table}[ht!]
\centering
\begin{tabular}{|l|r|r|r|r|}
\multicolumn{1}{l}{} & \multicolumn{4}{c}{év}\\
\cline{2-5}
\multicolumn{1}{l}{} & \multicolumn{1}{c}{\emph{2008}} & & & \\
\multicolumn{1}{c}{\emph{2009}} & & \multicolumn{1}{c}{\emph{2010}} & & \\
\multicolumn{1}{c}{\emph{2011}} & & & & \\
\hline
\emph{jövedelem (Ft)} & 994\,000 & 1\,231\,500 & 1\,525\,410 & 2\,321\,600 \\
\emph{járulék (Ft)} & 165\,000 & 194\,950 & 215\,750 & 235\,850 \\
\hline
\end{tabular}
\caption{A jövedelem és járulékok kimutatása}
\label{tablazat-jovedelem}
\end{table}

\Az{\ref{tablazat-jovedelem}}.~táblázat
\az{\pageref{tablazat-jovedelem}}.~oldalon olvasható.

\end{document}
```



Verbatim, programkódok

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 kell 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.

Programkódok

Különböző programnyelvek kódjainak megjelenítésére alkalmas a listings csomag.

```
\begin{lstlisting}[OPCIÓK] KÓD \end{lstlisting} % \usepackage{listings}  
\lstinputlisting[OPCIÓK]{KÓDOT TARTALMAZÓ FÁJL} % \usepackage{listings}
```

Az opciók a következő parancsban is megadhatók:

```
\lstset{OPCIÓK} % \usepackage{listings}
```

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

basicstyle=STÍLUS Kód fontjai (például `basicstyle=\small\ttfamily`).

breaklines Hosszú sorok törése.

backgroundcolor=SZÍN Háttérszín (például `backgroundcolor=\color{red}`).

showspaces Szóköz `_` módon jelölve.

showtabs Tabulátort jelöli.

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

frame=ÉRTÉK 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=SZÍN Keret vonalának színe (például `rulecolor=\color{red}`).

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

keywords={LISTA} A kiemelendő kulcsszavak listája, mely a **LISTA**-ban van felsorolva, vesszővel elválasztva. Például `keywords={begin,end}`.

morekeywords={LISTA} A kulcsszavak listáját ezzel lehet bővíteni.

keywordstyle=STÍLUS A kulcsszavak stílusa. Például `keywordstyle=\bfseries`.

title={KÓDCÍM} Kód címe sorszám nélkül. Ez nem kerül be a kódok jegyzékébe.

caption={KÓDCÍM} 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 töltsse be a `caption` csomagot is.



`label={CÍMKÉ}` Kereszthivatkozás címkéje. Ezt a `\label` parancs helyett kell használni.

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}
```



Teljes példa

Programkód

```
\documentclass[a4paper,12pt]{article}
\usepackage[T1]{fontenc}
\PassOptionsToPackage{defaults=hu-min}{magyar.ldf}
\usepackage[magyar]{babel}
\usepackage{listings,xcolor,caption}

\lstset{
  language=Delphi,
  basicstyle=\footnotesize\ttfamily,
  numbers=left,
  backgroundcolor=\color{lightgray},
  frame=tlbr,
  keywordstyle=\color{blue},
}

\renewcommand{\lstlistingname}{programkód}

\begin{document}

\lstinputlisting[caption=Egy Delphi kód,label=kod-delphi]{prog.pas}

\end{document}
```

Az előbbi kódban a prog.pas fájl tartalma lehet például a következő:

prog.pas

```
unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
  Forms, Dialogs, StdCtrls;

type
  { TForm1 }

  TForm1 = class(TForm)
    Button1: TButton;
    Label1: TLabel;
    procedure Button1Click(Sender: TObject);

  private
    { Private declarations }
```



```
public
  { Public declarations }

end;

var
  Form1: TForm1;

implementation

{$R *.dfm}

procedure TForm1.Button1Click(Sender: TObject);
begin
  Label1.Caption := 'Helló, világ!';
end;

end.
```




Képletek

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. *szövegeközi matematikai mód*. Ennek megadása a következő módon lehetséges:

```
$KÉPLET$
```

Például

```
Bármít is teszünk, $2+2=4$.
```

```
Bármít 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

```
\[  
KÉPLET  
\]
```

vagy

```
\begin{equation*}  
KÉPLET  
\end{equation*}
```

módon lehetséges. Például

```
Bármít is teszünk,  
\[  
2+2=4.  
\]
```

```
Bármít is teszünk,
```

$$2 + 2 = 4.$$



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

```
\text{SZÖVEG}
```

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 bízva 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:

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

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

Műveleti jelek

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

Relációjelek

$=$	$=$	\equiv	<code>\equiv</code>	\approx	<code>\approx</code>	$>$	$>$
$:=$	<code>:=</code>	\sim	<code>\sim</code>	\cong	<code>\cong</code>	\leq	<code>\leq</code>
\doteq	<code>\doteq</code>	\simeq	<code>\simeq</code>	$<$	<code><</code>	\geq	<code>\geq</code>



\ll	<code>\ll</code>	\ni	<code>\ni</code>	\subseteq	<code>\subseteq</code>	$ $	<code>\mid</code>
\gg	<code>\gg</code>	\subset	<code>\subset</code>	\supseteq	<code>\supseteq</code>	\parallel	<code>\parallel</code>
\in	<code>\in</code>	\supset	<code>\supset</code>	$:$	<code>:</code> (arányjel)	\perp	<code>\perp</code>

A nyilak is a relációjelek közé tartoznak:

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

Relációjeleket negálni (áthúzni) a `\not` paranccsal 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:

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

 $a \nmid b, a \nparallel b$

Matematikai zárójelek

Bal oldali (nyitó) zárójelek

$($	<code>(</code>	\langle	<code>\langle</code>	\ulcorner	<code>\ulcorner</code>	\llcorner	<code>\llcorner</code>
$[$	<code>[</code> vagy <code>\lbrack</code>	\lceil	<code>\lceil</code>	\llcorner	<code>\llcorner</code>	\llcorner	<code>\llcorner</code>
$\{$	<code>\{</code> vagy <code>\lbrace</code>	\lfloor	<code>\lfloor</code>	\lvert	<code>\lvert</code>		

Jobb oldali (csukó) zárójelek

$)$	<code>)</code>	\rangle	<code>\rangle</code>	\urcorner	<code>\urcorner</code>	\llcorner	<code>\llcorner</code>
$]$	<code>]</code> vagy <code>\rbrack</code>	\rceil	<code>\rceil</code>	\llcorner	<code>\llcorner</code>	\llcorner	<code>\llcorner</code>
$\}$	<code>\}</code> vagy <code>\rbrace</code>	\rfloor	<code>\rfloor</code>	\lvert	<code>\lvert</code>		

Például



```
\lvert-1\rvert
```

 $| - 1 |$

Közönséges matematikai jeleket ne használjon zárójelként. Például

```
$|-1|$
```

 $| - 1 |$

rossz eredményt ad, mert a program ezt úgy értelmezi, hogy a `|` jelből kivonjuk az 1-et, így a `-` jel körül térközöket hagy.

Amennyiben egy jelet nyitó zárójelként akar értelmezni, akkor tegye elé a

```
\mathopen
```

parancsot. Például a `\mathopen|` ekvivalens az `\lvert` paranccsal.

Amennyiben egy jelet csukó zárójelként akar értelmezni, akkor tegye elé a

```
\mathclose
```

parancsot. Például a `\mathclose|` ekvivalens az `\rvert` paranccsal.

A `\mathopen` és `\mathclose` parancsok használatára tekintsük a következő esetet:

```
$]-1, 1[\setminus\{0\}
```

 $] - 1, 1[\setminus \{0\}$

Az eredmény rossz, hiszen a `]` jel csukó zárójelként van értelmezve, így az utána található `-` jelet kivonásként értelmezi, másrészt a `[` nyitó zárójelként van értelmezve, így az utána található `\` jelet nem tekinti relációnak. A megoldás az, hogy ideiglenesen a `]` jelet nyitó, a `[` jelet pedig csukó zárójelként értelmezzük:

```
$\mathopen]-1, 1\mathclose[\setminus\{0\}
```

 $\mathopen] - 1, 1 \mathclose[\setminus \{0\}$

Természetesen, ha egy jelet nyitó zárójelként van értelmezve és annak is használja, illetve, ha egy jelet csukó zárójelként van értelmezve és annak is használja, akkor az előbbi megoldásra nincs szükség. Például a következő kód helyes eredményt ad:

```
$[-1, 1]\setminus\{0\}
```

 $[-1, 1] \setminus \{0\}$

Az előbbi zárójel mérete nem igazodik a képlethez. Például a következő kód hibás eredményt ad:

```
\[f(\frac{1}{2})=0\]
```

$$f\left(\frac{1}{2}\right) = 0$$

Ilyen esetben a nyitó zárójel elé tegye a



```
\left
```

míg a csukó zárójel elé

```
\right
```

parancsot. Például

```
\[f\left(\frac{1}{2}\right)=0\]
```

$$f\left(\frac{1}{2}\right)=0$$

A `\left` és `\right` parancsok használatával már azt is megadjuk, hogy melyik a nyitó és melyik a csukó zárójel, így ezekkel nem szabad együtt használni a `\mathopen` és `\mathclose` parancsokat. Tehát például a következő kód helyes eredményt ad:

```
\[\left|-\frac{1}{2}\right|\]
```

$$\left|-\frac{1}{2}\right|$$

Néhány esetben nem ad megfelelő eredményt a zárójelek automatikus méretezése. Például

```
\$\left\{\{a,b\},\{c,d\}\right\}\$
```

```
\{\{a,b\},\{c,d\}\}
```

A külső zárójeleknek egy picit nagyobbaknak kellene lenniük, de ezt a közbezárt képlet nem generálja. Ilyenkor rögzített méreteket is használhat. A `\left` helyett

```
\bigl \Bigl \biggl \Biggl
```

illetve `\right` helyett

```
\bigr \Bigr \biggr \Biggr
```

Ezek hatása:

```
\$\Biggl(\biggl(\Bigl(\bigl(\cdot\bigr)\Bigr)\biggr)\Biggr)\$
```

$$\left(\left(\left(\left(\left(\cdot\right)\right)\right)\right)\right)$$

Például az előző képlet a következő módon oldható meg helyesen:

```
\$\bigr\{\{a,b\},\{c,d\}\bigr\}\$
```

```
\{\{a,b\},\{c,d\}\}
```



Közönséges matematikai jelek

$\%$ <code>\%</code>	\Im <code>\Im</code>	\blacksquare <code>\blacksquare</code>	1° <code>1^\circ\circ</code>
\perp <code>\bot</code>	∇ <code>\nabla</code>	\angle <code>\angle</code>	$1'$ <code>1' (1^\prime)</code>
\top <code>\top</code>	∂ <code>\partial</code>	\sphericalangle <code>\sphericalangle</code>	$1''$ <code>1''</code>
\neg <code>\neg</code>	\eth <code>\eth</code>	\lrcorner <code>\lrcorner</code>	$ $ <code> </code> vagy <code>\vert</code>
\forall <code>\forall</code>	\emptyset <code>\emptyset</code>	\flat <code>\flat</code>	$ $ <code>\ </code> vagy <code>\Vert</code>
\exists <code>\exists</code>	∞ <code>\infty</code>	\sharp <code>\sharp</code>	
\nexists <code>\nexists</code>	\triangle <code>\triangle</code>	\natural <code>\natural</code>	
\Re <code>\Re</code>	\square <code>\square</code>	$\#$ <code>\#</code>	

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>	

Kalligrafikus és dupla szárú betűk

$\mathcal{A B C D E F G H I J K L M N}$	<code>\mathcal{A B C D E F G H I J K L M N}</code>
$\mathbf{A B C D E F G H I J K L M N}$	<code>\mathbf{A B C D E F G H I J K L M N}</code>

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>	\dota{a} <code>\dota{a}</code>
\bar{a} <code>\bar{a}</code>	\grave{a} <code>\grave{a}</code>	\dot{a} <code>\dot{a}</code>	\dotta{a} <code>\dotta{a}</code>

Három pont

$1, \dots, n$	<code>1, \ldots, n</code>	$1 \cdots n$	<code>1 \cdots n</code>	\vdots	<code>\vdots</code>
$1 + \cdots + n$	<code>1 + \ldots + n</code>	$\int \cdots \int$	<code>\int \ldots \int</code>	\ddots	<code>\ddots</code>

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

`\dots`

paranccsal. Például



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

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

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>		

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>		

Esetek szétválasztása

Erre a cases környezet használható. Például

```
\[
f(x)=
\begin{cases}
0, & \text{ha } x \in \mathbb{Q}, \\
1, & \text{különben.}
\end{cases}
\]
```

$$f(x) = \begin{cases} 0, & \text{ha } x \in \mathbb{Q}, \\ 1, & \text{különben.} \end{cases}$$



Mátrixok

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

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

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

Nagy operátorok

\sum	<code>\sum</code>	\int	<code>\int</code>	\cap	<code>\bigcap</code>	\oint	<code>\oint</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 \$
\[
\sum
\]
```

$$\sum \quad \quad \quad \sum$$

„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>\varliminf</code>
<code>\cosh</code>	<code>\exp</code>	<code>\sec</code>	<code>\varlimsup</code>

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



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

 $\log_2 x$
 $\log_2 x$

illetve

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

 \int_a^b
 \int_a^b

„Limits” függvények

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

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

```
\sum_{i=1}^{\infty}a_i:=\lim_{n\to\infty}\sum_{i=1}^na_i$
\[
\sum_{i=1}^{\infty}a_i:=\lim_{n\to\infty}\sum_{i=1}^na_i
\]
```

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

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

Új függvények definiálása

Előfordulhat, hogy olyan függvényre van szükség, amely alaphól 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

```
$\Min_{k\in\mathbb{N}}$  
\[  
\Min_{k\in\mathbb{N}}  
\]
```

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

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

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

```
\DeclareMathOperator{PARANCS}{JEL}
```

Például

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

után

```
$\tg^2x$  
\[  
\tg^2x  
\]
```

$$\text{tg}^2 x$$

$$\text{tg}^2 x$$

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

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



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

Matematikai jelek több szerepben

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

```
$f\colon A\to B$ (helyes)\n
$f:A\to B$ (helytelen)
```

```
f: A → B (helyes)\n
f : A → B (helytelen)
```

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

Ha a magyar .ldf 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

```
$2,5\cdot 2=5$\n
$a,b,c$
```

```
2,5 · 2 = 5\n
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, ...
```

Kiemelt képletek sorszámozása

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

```
\begin{equation}\label{CÍMKE}\n
KÉPLET\n
\end{equation}\n
\eqref{CÍMKE}
```

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

```
\Az{\eqref{CÍMKE}} % \usepackage[magyar]{babel}\n
\az{\eqref{CÍMKE}} % \usepackage[magyar]{babel}
```

Például



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

$$x^2 + 2x - 3 = 0 \quad (1)$$

Az (1) miatt ...

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

$$x + y \quad (1)$$

$$x^2 + xy + y^2 \quad (2)$$

illetve

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

$$\begin{array}{r} x + y \\ x^2 + xy + y^2 \end{array} \quad (1)$$

Több képlet egymás alatt illesztéssel

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



```
\begin{align}
1. \text{ SOR } \text{JOBBRA\&BALRA} \& \text{ JOBBRA\&BALRA} \& \text{ JOBBRA} \dots \text{ \label{CÍMKE 1}}\\
2. \text{ SOR } \text{JOBBRA\&BALRA} \& \text{ JOBBRA\&BALRA} \& \text{ JOBBRA} \dots \text{ \label{CÍMKE 2}}\\
\dots \\
N. \text{ SOR } \text{JOBBRA\&BALRA} \& \text{ JOBBRA\&BALRA} \& \text{ JOBBRA} \dots \text{ \label{CÍMKE N}}
\end{align}
```

Például

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

$$\begin{array}{lll} x = y + z & y = bd & z = bc \\ b = 10 & 2c = 56 & d = 44 \end{array} \quad (1)$$

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*}
```

$$\begin{aligned} x &= y + z \\ &= bd + bc \\ &= 1000 \end{aligned}$$

Teljes példa

Képletek

```
\documentclass[a4paper,12pt]{article}
\usepackage[T1]{fontenc}
\PassOptionsToPackage{defaults=hu-min}{magyar.ldf}
\usepackage[magyar]{babel}
\usepackage{mathtools,amssymb}

\DeclareMathOperator{\diff}{d\!}
\DeclareMathOperator{\tg}{tg}

\begin{document}
```



Legyen X egy halmaz. Az $\mathcal{A} \subset \mathcal{P}(X)$ halmazrendszer `\emph{\sigma-algebrának}` nevezzük, ha

```
\begin{enumerate}
\item  $X \in \mathcal{A}$ ,
\item  $\overline{A} = X \setminus A \in \mathcal{A} \quad \text{for all } A \in \mathcal{A}$ ,
\item  $\bigcup_{i=1}^{\infty} A_i \in \mathcal{A}$ , ha  $A_i \in \mathcal{A} \setminus (i \in \mathbb{N})$ .
\end{enumerate}
```

Ekkor az (X, \mathcal{A}) rendezett párt `\emph{mérhető térnek}`, az \mathcal{A} elemeit `\emph{mérhető halmazoknak}` nevezzük.

A $\mu: \mathcal{A} \rightarrow [0, \infty]$ függvényt `\emph{mértéknek}` nevezzük az (X, \mathcal{A}) mérhető téren, ha $\mu(\emptyset) = 0$ és

```
\[
\mu\left(\bigcup_{i=1}^{\infty} A_i\right) =
\sum_{i=1}^{\infty} \mu(A_i)
\]
```

minden $A_i \in \mathcal{A} \setminus (i \in \mathbb{N})$ diszjunkt rendszerre. Ekkor (X, \mathcal{A}, μ) -t `\emph{mértéktérnek}`, $\mu(A)$ -t az A mértékének nevezzük.

Legyen X egy halmaz, $\mathcal{H} \subset \mathcal{P}(X)$, $\nu: \mathcal{H} \rightarrow [0, \infty]$ és μ a ν -höz tartozó külső mérték. $B \subset X$ pontosan akkor μ -mérhető, ha

```
\begin{equation}\label{egyenlet-feltetel}
\nu(A) \geq \mu(A \cap B) + \mu(A \setminus B) \quad \text{for all } A \in \mathcal{H}.
\end{equation}
```

`\Az{\eqref{egyenlet-feltetel}}` szükségessége triviálisan teljesül. Az elégséges voltát később látjuk be.

A véges értékészletű függvényeket `\emph{egyszerű függvényeknek}` nevezzük. Ha X egy halmaz és $A \subset X$, akkor az

```
\[
I_A: X \rightarrow \mathbb{R}, \quad I_A(x) :=
\begin{cases}
1, & \text{ha } x \in A, \\
0, & \text{különben}
\end{cases}
\]
```

függvényt az A `\emph{karakterisztikus függvényének}` nevezzük.

A majorált konvergenciátétel a következőt állítja: Legyen

```
 $(X, \mathcal{A}, \mu)$  mértéktér és  $g, f, f_n: X \rightarrow \mathbb{R}_+$ 
 $(n = 1, 2, 3, \dots)$  mérhető függvények. Ha  $g$  integrálható,  $\int f_n \rightarrow \int g$ 
és  $\lim_{n \rightarrow \infty} f_n = f$ , akkor
```



```
\lim_{n\to\infty}\int\limits f_n-f\diff\mu=0
\quad\text{és}\quad
\lim_{n\to\infty}\int f_n\diff\mu=\int f\diff\mu.
\]
```

A tangens függvényt a következő módon definiáljuk:

```
\[
\tg\colon\mathbb{R}\setminus
\left\{\frac{\pi}{2}+k\pi:k\in\mathbb{Z}\right\}\to\mathbb{R},
\quad
\tg(x):=\frac{\sin(x)}{\cos(x)}.
\]

\end{document}
```



Strukturált művek

Hosszabb, strukturált dokumentumokat a következő módon lehet például tagolni:

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

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.

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.

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 fejlécbe 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.

Fej- és láblécek

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.

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. léníával, a fejléctől és lábléctől. Ezeknek a vonalaknak a vastagságát a következő parancsokkal állíthatja be:

```
\renewcommand{\headrulewidth}{VASTAGSÁG}
```

Fejléc alatti lénia vastagsága, aminek az alapértéke 0.4pt.

```
\renewcommand{\footrulewidth}{VASTAGSÁG}
```

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{SZÖVEG}
```



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

```
\markboth{INFÓ1}{INFÓ2}
```

Ennek hatására a `\leftmark` eredménye `INFÓ1`, míg a `\rightmark` eredménye `INFÓ2` lesz.

```
\markright{INFÓ}
```

Ennek hatására a `\rightmark` eredménye `INFÓ` lesz.

A következő kódot, ha a dokumentumtestbe írja, akkor onnan kétoldalas szédségnél 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 felülbírálható a `\markboth` parancssal. Például

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

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

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}
```

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` paranccsal definiált környezetekkel hozhat létre.

```
\newtheorem{TÉTELNÉV}{TÉTELCÍM}
\newtheorem{TÉTELNÉV}{TÉTELCÍM}[SZÁMLÁLÓÓS]
\newtheorem{TÉTELNÉV}[EGYÜTTNÉV]{TÉTELCÍM}
```

tételnev Létrejön egy **TÉTELNÉV** környezet és egy **TÉTELNÉV** számláló, mely minden újabb ilyen környezet megnyitásakor növekszik eggyel.

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

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

együtnév Egy másik tételszerű környezet neve. A **TÉTELNÉV** és **EGYÜTTNÉV** környezetek számlálói együtt fognak növekedni.

A létrehozott tételszerű környezetet az alábbi módon használhatja:

```
\begin{TÉTELNÉV}[EGYEDI CÍM]
A BEKEZDÉS SZÖVEGE
\end{TÉTELNÉV}
```

Az **EGYEDI CÍM** megadása esetén, az a **TÉTELCÍM** után jelenik meg zárójelben. Hivatkozni a tételszerű bekezdésekre az általános leírásnak megfelelően lehet. Például

```
\newtheorem{tetel}{tétel}
...
\begin{tetel}
A tétel szövege.
\end{tetel}
\begin{tetel}[Cauchy]\label{cauchy}
A következő tétel szövege.
\end{tetel}
\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} % \usepackage{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*{TÉTELNÉV}{TÉTELCÍM} % \usepackage{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}
...

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

□

Bibliográfia

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

```
\begin{thebibliography}{PÉLDACÍMKÉ}
\bibitem{KULCS} ELEMLEÍRÁS
...
\end{thebibliography}
```

PÉLDACÍMKÉ A bibliográfiai elemek címkéi közül a legszélesebb.

KULCS A bibliográfiai elemre `\cite[SZÖVEG]{KULCS}` 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 **SZÖVEG**-ben például megadhatja, hogy melyik oldalra hivatkozik.

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

```
\acite[SZÖVEG]{KULCS} % \usepackage[magyar]{babel}
\Acite[SZÖVEG]{KULCS} % \usepackage[magyar]{babel}
```

vagy az ezzel egyenértékű



```
\az{\cite[SZÖVEG]{KULCS}} % \usepackage[magyar]{babel}
\Az{\cite[SZÖVEG]{KULCS}} % \usepackage[magyar]{babel}
```

parancsokkal. Például

```
Lásd \cite{PlainTeX} és \cite[134.~oldal]{LaTeX}\dots
Lásd \cite{PlainTeX,LaTeX}\dots
Lásd \acite{PlainTeX,LaTeX} könyvekben\dots
\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]... Lásd az [1, 2] könyvekben...

Hivatkozások

- [1] Bujdosó Gyöngyi, Fazekas Attila: \TeX kezdőlépések, Budapest, 1997, Tertia Kiadó.
- [2] Wettl Ferenc, Mayer Gyula, Szabó Péter: \LaTeX kézikönyv, Budapest, 2004, Panem Könyvkiadó.

Teljes példa

Strukturált mű report dokumentumosztályban

```
\documentclass[12pt,twoside]{report}
\usepackage[T1]{fontenc}
\usepackage{fancyhdr} % babel előtt kell betölteni
\usepackage[a4paper, top=40mm, bottom=40mm, inner=40mm, outer=30mm,
headsep=8mm, headheight=15pt]{geometry}
\PassOptionsToPackage{defaults=hu-min}{magyar.ldf}
\usepackage[magyar]{babel}
\usepackage{mathtools,amssymb,amsthm}

\newtheorem{tetel}{Tétel}[chapter]
\newtheorem{kovetkezmény}[tetel]{Következmény}
\theoremstyle{definition}
\newtheorem{definicio}[tetel]{Definíció}

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




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

```
\begin{document}
```

```
\title{Egy példa strukturált dokumentumra}  
\author{Szerző neve}  
\date{dátum}
```

```
\maketitle
```

```
\tableofcontents
```

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

Minden természet- és társadalomtudomány foglalkozik olyan jelenségekkel, melyekben egy bizonyos esemény szükségszerűen bekövetkezik, ha az általunk ismert és figyelembe vett körülmények fennállnak. Ezeket `\emph{meghatározott eseményeknek}` nevezzük.

Bizonyos jelenségeknél az összes számításba jöhető körülmény figyelembe vétele lehetetlen, de legalábbis igen nehéz.

Ennek oka lehet például, hogy a jelenség háttérében meghúzódó körülmények rendszere a tudomány mai állása szerint még nem teljesen feltárt, vagy nem tudjuk mérni őket, vagy számuk túl nagy és kapcsolatuk nagyon bonyolult.

Ilyenkor előfordulhat, hogy a figyelembe vett körülmények összessége nem határozza meg egy esemény bekövetkezésének elegendő okát.

Az ilyen eseményeket `\emph{véletlen eseményeknek}` nevezzük.

Például, amikor egy dobókockával játszunk, akkor nem tudjuk figyelembe venni az összes körülményt -- hogy milyen helyzetből indult, mekkora impulzust kapott, a légellenállást, az asztallal való ütközést, a súrlódást stb.--, csak azt a tényt, hogy feldobtuk.

Ez viszont nem határozza meg a dobás eredményét egyértelműen, így számunkra például a hatos dobása véletlen eseményt jelent.

Ha egy véletlen kimenetelű jelenség sokszor ismétlődhet, akkor `\emph{véletlen tömegjelenségről}` beszélünk.

Az ilyen típusú jelenségekről a véletlenszerűségük ellenére is áttekintést nyerhetünk.

Vegyük példaként a radioaktív bomlást.

Bár minden egyes atommag bomlása véletlennek tekinthető, mégis például egy urántömbben elhelyezkedő sok-sok milliárd atommag esetében már előre meg tudjuk mondani, hogy egy meghatározott időn belül hány százalékuk fog elbomlani.

Ez a bomlás úgynevezett exponenciális törvénye, melyet a valószínűségszámítás segítségével írhatunk le.



Ezt a törvényt a mérések éppúgy alátámasztják, mint bármilyen meghatározott természeti törvényt.

`\emph{A valószínűségszámítás tárgya a véletlen tömegjelenségek vizsgálata, feladata pedig ezen jelenségek törvényszerűségeinek a feltárása.}`

Végezzünk el egy véletlen kimenetelű kísérletet sokszor egymás után. Figyeljük egy lehetséges esemény bekövetkezését.

Ha a kísérlet `n` végrehajtása után `k`-szor fordult elő a figyelt esemény, akkor a `$\frac{k}{n}$` számot az esemény `\emph{relatív gyakoriságának}` nevezzük.

`\emph{A tapasztalat azt mutatja, hogy sok kísérlet esetén a relatív gyakoriság, egy eseménytől függő érték körül ingadozik}.`

A továbbiakban ezt az értéket a vizsgált esemény `\emph{valószínűségének}` fogjuk nevezni.

Ezen tapasztalat alapján axiómákat lefektetve lehetőség nyílik egy matematikai elmélet kidolgozására.

Természetesen egy axiómarendszer akkor jó, ha az elmélet visszaadja a tapasztalatot.

Látni fogjuk a nagy számok törvényeivel foglalkozó fejezetben, hogy ez az elvárás teljesül.

`\chapter{Valószínűségszámítás}`

`\section{Események matematikai modellezése}`

Mindenekelőtt szükségünk lesz olyan eszközökre, amelyek alkalmasak a véletlen események közötti kapcsolatok leírására.

Ezt halmazok segítségével oldjuk meg.

Az események halmazokkal való azonosítása a matematikában a következő példa alapján kézenfekvőnek tűnik:

Amikor egy dobókockával játszunk, az egyes, kettős, hármas, négyes, ötös vagy a hatos oldal lehet felül. A nekik megfelelő halmazok legyenek a következők: `$\{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}$`.

Ezeket a továbbiakban `\emph{elemi eseményeknek}` fogjuk nevezni.

De más események is elképzelhetők.

Például az, hogy páros számot dobok.

Ennek feleltessük meg a következő halmazt: `$\{2, 4, 6\}$`.

Ezt az eseményt `\emph{összetett eseménynek}` fogjuk hívni, mert felbontható nem triviális módon több esemény uniójára: `$\{2, 4, 6\} = \{2\} \cup \{4\} \cup \{6\}$`.

Az is esemény, hogy egytől hatig valamilyen egész szám fog kijönni.

Ezt `\emph{biztos eseménynek}` nevezzük, melyet `Ω`-val jelölünk, és a halmaz megfelelője: `$\Omega = \{1, 2, 3, 4, 5, 6\}$`.



Azt az eseményt, amely a kockajáték szabályai szerint nem fordulhat elő, `\emph{lehetetlen eseménynek}` fogjuk nevezni, és a halmaz megfelelője legyen az üres halmaz.

Végül azt is eseménynek kell tekinteni, ha egy esemény nem következik be.

Például nem egyest dobok. Az ehhez tartozó halmaz:

```
\overline{\{1\}}=\{2, 3, 4, 5, 6\}
```

Vegyük észre, hogy akármelyik eseményt is tekintjük, az a biztos esemény egy részhalmaza.

Milyen fontos tulajdonságai vannak az eseményeknek?

Mindenekelőtt kihangsúlyozzuk, hogy értelemszerűen minden ami teljesül a halmazokra, az teljesül az eseményekre is.

Az események rendszerét az `\Omega` részhalmazainak egy rendszerével reprezentáljuk (jelöljük ezt `\mathcal{F}`-fel).

Az előző példában `\mathcal{F}` az `\Omega` összes részhalmazainak a halmaza, azaz `\Omega \emph{hatványhalmaza}`.

Ennek azonban nem feltétlenül kell teljesülni, mint azt látni fogjuk például a geometriai valószínűségi mező tárgyalásánál.

Általános esetben `\mathcal{F}` az `\Omega` hatványhalmazának egy részhalmaza.

Ennek a tulajdonságait kell megvizsgálni.

Hármat emelünk ki:

```
\begin{enumerate}
```

```
\item Az első, hogy az \Omega esemény, azaz eleme \mathcal{F}-nek.
```

```
\item Azt is láttuk, hogy egy esemény ellentettje is esemény.
```

```
\item Végül nyilvánvaló tulajdonság még, hogy két esemény uniója is esemény.
```

Például páros számot vagy hármast dobok: `\{2, 4, 6\} \cup \{3\} = \{2, 3, 4, 6\}`.

Az általánosabb esetek leírására szolgál az úgynevezett

`\emph{Kolmogorov-féle elmélet}`, mely felteszi, hogy nemcsak véges sok, hanem megszámlálhatóan végtelen sok esemény uniója is esemény.

```
\end{enumerate}
```

A továbbiakban ezen tulajdonságokat választjuk az események axiómarendszereként.

```
\section{A valószínűség matematikai modellezése}
```

Egy másik alapfogalomra, a valószínűségre is szükségünk van.

Tapasztalatunk alapján, ez nagyszámú kísérletek után, körülbelül a relatív gyakorisággal egyezik meg.



Így a valószínűség jól jellemezhető a relatív gyakoriság tulajdonságaival.

A valószínűség egy függvény.

Minden eseményhez hozzárendeli azt a számot, amely körül a relatív gyakoriság ingadozik.

Három tulajdonságát emeljük ki:

```
\begin{enumerate}
```

```
\item A relatív gyakoriság, s így a valószínűség értéke sem lehet negatív.
```

```
\item Ha a biztos esemény relatív gyakoriságát vizsgáljuk, akkor minden kísérlet esetén a bekövetkezések száma és a kísérletek száma megegyezik. Így a hányadosuk minden esetben 1.
```

Ebből az következik, hogy a biztos esemény valószínűsége 1.

```
\item A harmadik tulajdonságot ismét dobókockával szemléltetjük.
```

Tekintsük azokat az eseményeket, amikor az egyes oldal, illetve amikor a kettős vagy hármas oldal van felül.

Az ezeknek megfelelő halmazok az $\{1\}$ és a $\{2, 3\}$.

Ezen két egymást kizáró esemény relatív gyakoriságait megvizsgálva, azt fogjuk tapasztalni, hogy $\frac{1}{6}$ illetve $\frac{1}{3}$ körül ingadozik nagyszámú kísérlet esetén.

Ez természetes, hiszen az oldalak között fizikai jellemzőit tekintve nincs különbség, csupán másképpen jelöljük őket.

Így minden oldalra egyforma eséllyel eshet.

Ha most az $\{1, 2, 3\}$ esemény relatív gyakoriságát vizsgáljuk, akkor az $\frac{1}{2}$ körül ingadozik.

Vagyis ahogy az előre sejthető volt, az előző két érték összeadódik.

Ezt az eredményünket általánosítva azt mondhatjuk, hogy ha az A és B események diszjunktak, akkor az események uniójának a valószínűsége megegyezik az események valószínűségeinek összegével.

Itt is kiterjesztjük az eredményt végtelen esetre.

Eszerint megszámlálhatóan végtelen sok, páronként diszjunkt esemény uniójának valószínűsége megegyezik az események valószínűségeinek összegével.

Bár ez nem következik a szemléletből, mégis elfogadásával a jelenségek egy igen széles köre leírható lesz.

```
\end{enumerate}
```

A valószínűségszámítás Kolmogorov-féle elméletében ezeket a tulajdonságokat választjuk a valószínűség axiómarendszerének.

(Lásd még `\cite[27.~oldal]{FAZEKAS}`, illetve `\cite{RENYI}`.)

Ajánlott feladatgyűjtemények: `\cite{DENKINGER,SOLT}`.)

```
\chapter{Egy példafejezet tételszerű környezetekre}
```

```
\begin{definicio}
```



Definíció szövege.

```
\end{definicio}
```

```
\begin{tetel}
```

Tétel szövege.

```
\end{tetel}
```

```
\begin{tetel}[Tétel címe]
```

Tétel szövege.

```
\end{tetel}
```

```
\begin{proof}
```

Bizonyítás szövege.

```
\end{proof}
```

```
\begin{ko vetke z meny}
```

Következmény szövege.

```
\end{ko vetke z meny}
```

```
\begin{thebibliography}{10}
```

```
\bibitem{DAROCZY} Daróczy Zoltán: \emph{Mérték és integrál}, Budapest,  
1984, Tankönyvkiadó.
```

```
\bibitem{DENKINGER} Denkinger Géza: \emph{Valószínűség-számítási  
gyakorlatok}, Budapest, 1986, Tankönyvkiadó.
```

```
\bibitem{FAZEKAS} Fazekas István: \emph{Valószínűség-számítás}, Debrecen,  
2000, Debreceni Egyetem Kossuth Egyetemi Kiadója.
```

```
\bibitem{HALMOS} P.~R.~Halmos: \emph{Mértékelmélet}, Budapest, 1984,  
Gondolat.
```

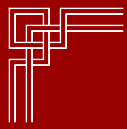
```
\bibitem{KOLMOGOROV} A.~N.~Kolmogorov, Sz. V. Fomin: \emph{A  
függvényelmélet és a funkcionálanalízis elemei}, Budapest, 1981, Műszaki  
Könyvkiadó.
```

```
\bibitem{MOGYORODI} Mogyoródi József, Somogyi Árpád:  
\emph{Valószínűség-számítás I.}, Budapest, 1982, Tankönyvkiadó.
```

```
\bibitem{RENYI} Rényi Alfréd: \emph{Valószínűség-számítás}, Budapest,  
1966, Tankönyvkiadó.
```

```
\bibitem{SAIN} Sain Márton: \emph{Nincs királyi út!}, Budapest, 1975,  
Gondolat.
```

```
\bibitem{SOLT} Solt György: \emph{Valószínűség-számítás}, Budapest, 1993,  
Műszaki Könyvkiadó.
```



```
\bibitem{SZOKEFALVI} Szőkefalvi-Nagy Béla: \emph{Valós függvények és  
függvénsorok}, Budapest, 1954, Tankönyvkiadó.  
\end{thebibliography}  
  
\end{document}
```

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[címké]{SZÖVEG}
```

Ennek helyén a `SZÖVEG` felirat jelenik meg linkként. Erre kattintva a `\label{címké}`-vel létrehozott címkére ugrik.

A `hyperref` csomag néhány opciója

`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).

Teljes példa

Struktúrált elektronikus publikáció report osztályban

```
\documentclass[12pt]{report}
\usepackage[T1]{fontenc}
\usepackage[colorlinks, linktocpage, allcolors=blue, pdfstartview=FitH,
bookmarksnumbered]{hyperref} % babel és geometry előtt kell betölteni
\usepackage[a4paper, top=40mm, bottom=40mm, inner=40mm,
outer=30mm]{geometry}
\PassOptionsToPackage{defaults=hu-min}{magyar.ldf}
\usepackage[magyar]{babel}

\begin{document}

\title{Cím}
\author{Szerző}
\date{dátum}

\maketitle

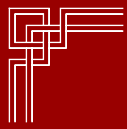
\tableofcontents

\chapter{Fejezet címe}\label{fejezet-pl}

Fejezet szövege.

\section{Szakasz címe}

\chapter{Másik fejezet címe}
```

```
\Az{\ref{fejezet-pl}}.~fejezetben\dots (lásd \cite{DENKINGER}).  
  
\begin{thebibliography}{2}  
\bibitem{DAROCZY} Daróczy Zoltán: \emph{Mérték és integrál}, Budapest,  
1984, Tankönyvkiadó.  
\bibitem{DENKINGER} Denkinger Géza: \emph{Valószínűségszámítási  
gyakorlatok}, Budapest, 1986, Tankönyvkiadó.  
\end{thebibliography}  
\end{document}
```



Szakdolgozat készítése

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

A lehetséges opciók

`twoside` Ha a szakdolgozatot kétoldalasán szeretné kinyomtatni, akkor ezt az opciót alkalmazza! Ne használja egyoldalas nyomtatáshoz illetve elektronikus verzióhoz!

`colorlinks` A linkek színes karakterekkel jelennek meg. Ezt csak a szakdolgozat elektronikus verziójához használja, a nyomtatott verzióhoz nem kell!

`tocnopagenum` Ennek hatására a tartalomjegyzéknek nem lesz oldalszámozása. Ha közvetlenül a címoldal követően van elhelyezve a tartalomjegyzék, akkor az első számozott oldal csak ezután következik.

Címoldal

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 Katolikus 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 Katolikus 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\\egyetemi 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!

Teljes példa

Szakdolgozat thesis-ekf dokumentumosztálllyal

```
\documentclass{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{2025}

\maketitle

\tableofcontents

\chapter{Fejezet címe}

\section{Szakasz címe}
```



```
\begin{thebibliography}{1}  
\bibitem{cimke} \textsc{Szerző}: Cím, Kiadó, Hely, évszám.  
\end{thebibliography}  
  
\end{document}
```

A szakdolgozat készítéséhez a legegyszerűbb, ha a [thesis-ekf-templates.zip](#) sablont használja.



Irodalomjegyzék

- [1] LaTeX2e unofficial reference manual.
URL: <http://tug.org/texinfohtml/latex2e.html>
- [2] Wikibooks.org: LaTeX.
<http://upload.wikimedia.org/wikipedia/commons/2/2d/LaTeX.pdf>
- [3] BUJDOSÓ GYÖNGYI, FAZEKAS ATTILA: *T_EX kezdőlépések*, Budapest, 1997, Tertia Kiadó.
- [4] JOHANNES BRAAMS, DAVID CARLISLE, ALAN JEFFREY, LESLIE LAMPORT, FRANK MITTELBACH, CHRIS ROWLEY, RAINER SCHÖPF: *The L^AT_EX 2_ε Sources*.
<http://mirrors.ctan.org/macros/latex/base/source2e.pdf>
- [5] DONALD ERVIN KNUTH: *The T_EXbook*, Reading/Ma. etc., 1984, Addison-Wesley.
<http://tug.ctan.org/tex-archive/systems/knuth/dist/tex/texbook.tex>
- [6] L^AT_EX3 PROJECT TEAM: L^AT_EX News.
<https://www.latex-project.org/news/latex2e-news/ltnews.pdf>
- [7] TOBIAS OETIKER, HUBERT PARTL, IRENE HYNA, ELISABETH SCHLEGL: *Egy nem túl rövid bevezető a L^AT_EX 2_ε használatába*.
<https://math.bme.hu/latex/dl/latex78.pdf>
- [8] SZABÓ PÉTER: *Magyar nyelvű szöveg szedése MagyarL^AT_EX-hel*.
<http://www.math.bme.hu/latex/magyarldf-doc.pdf>
- [9] TÓMÁCS TIBOR: L^AT_EX
<https://tibortomacs.github.io/latex-tutorial-hu/latex.pdf>
- [10] TIBOR TÓMÁCS: *Thesis class for the Eszterházy Károly Catholic University*.
<http://mirrors.ctan.org/macros/latex/contrib/thesis-ekf/thesis-ekf.pdf>
- [11] WETTL FERENC, MAYER GYULA, SZABÓ PÉTER: *L^AT_EX kézikönyv*, Budapest, 2004, Panem Könyvkiadó.
http://math.bme.hu/latex/lakk_free.pdf