



L^AT_EX TÉVHITEK

TÓMÁCS TIBOR

30 L^AT_EX tévhit, amiről érdemes tudni!

Verzió: 2025-09-28

latex-tutorial-hu

L^AT_EX tévhitek

TÓMÁCS TIBOR

A füzet szabadon letölthető az alábbi linkről:

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

További L^AT_EX témájú kiadványokat talál magyar nyelven a latex-tutorial-hu oldalon.

Ha észrevétele, megjegyzése van, kérem írjon a következő címre:

tomacs.tibor@gmail.com

Verzió: 2025-09-28

A borítógrafikát a [Freepik](https://www.freepik.com) tervezte.

A pdf generálásához használt T_EX-rendszer:

pdfTeX, Version 3.141592653-2.6-1.40.28 (TeX Live 2025) kpathsea version 6.4.1

Előszó

A \TeX -rendszerrel az 1990-es évek elején találkoztam először. Ezen belül a Plain \TeX makrócsomagot használtam. Több cikket, egy könyvet és a doktori disszertációt is ezzel a rendszerrel írtam. Ezeken keresztül megtapasztalhattam a Plain \TeX korlátait, ami egy összetettebb dokumentum megírását kényelmetlenné teszi. A Plain \TeX hátránya az is, hogy tipográfiai ismeretek nélkül használva nagy eséllyel kifogásolható kimenetet eredményez. Emiatt mai szemmel már több formai hibát is látok az akkor írt műveimben.

Ezek érlelték bennem a gondolatot, hogy a strukturált művekre jóval alkalmasabb \LaTeX makrócsomagra térjek át, amely a helyes tipográfiára is sokkal nagyobb hangsúlyt fektet. A váltásra a végső lökést 2004-ben egy konferenciakiadvány elkészítésére való felkérés adta. Ezután 2005-ben megbízást kaptam az *Annales Mathematicae et Informaticae* folyóirat műszaki szerkesztésére.

A \LaTeX elsajátításához rengeteg hazai és külföldi lektorált szakkönyvön rágtam át magam, továbbá a tipográfia mesterségébe is igyekeztem minél jobban elmélyülni. Beleástam magam a felhasználói szinten túl a fejlesztői munkába is. A következő csomagoknak és osztályoknak vagyok a szerzője és fejlesztője: *bookcover*, *fgruler*, *huaz*, *hulipsum*, *numspell*, *thesis-ekf*. Ezek a csomagok a *Comprehensive TeX Archive Network (CTAN)* oldalról érhetők el, de részét képezik a TeX Live és MiKTeX disztribúcióknak is. A \LaTeX Windowson történő használatához kifejlesztettem egy könnyen és gyorsan telepíthető kompakt keretrendszert is *TeXfireplace* néven.

2009-től átvettem az Eszterházy Károly Katolikus Egyetem Matematikai és Informatikai Intézetében a \LaTeX oktatását. Ennek a kurzusnak az elsődleges célja, hogy a hallgatók \LaTeX -ben tudják megírni a TDK illetve szakdolgozataikat.

Elkötelezett híve vagyok a \LaTeX hazai népszerűsítésének és oktatásának. Ebben a témában több könyvet is írtam, melyek a *latex-tutorial-hu* oldalról tölthetők le. Ezekhez számos kiegészítő feladat és videó is készült, melyek az előbbi könyvekből, illetve a *honlapomról* is elérhetőek.

Szerkesztői és oktatói munkám során rengeteg tipikus szerzői hibával, tévhitel és tipográfiai pontatlansággal találkozom. Ennek a füzetnek a létrejöttét az motiválta, hogy okulásként megosszak ezekből néhányat másokkal is. Bízom benne, hogy azok is haszonnal forgatják majd ezt a kis gyűjteményt, akik már magabiztosan használják a publikációik megírásához a \LaTeX nyelvet. Talán ők is felismernek néhány berögzült rossz szokásukat.

Tómacs Tibor
egyetemi docens

Tévhitek jegyzéke

1. Egysoros számozatlan kiemelt képletet dupla dollárjelek között kell megadni	3
2. Egyenletek illesztésére az <code>eqnarray*</code> környezetet kell használni	4
3. Képletekben a kettőspont karaktert lehet használni nem műveleti jelként is	5
4. A halmazkivonás jeleként a <code>\backslash</code> parancsot kell használni	6
5. A <code> </code> karaktert kell használni az abszolút érték jeleként	6
6. Nyitó határolójelet lehet használni záróként is és viszont	7
7. Az operátorneveket a képletekben egyszerűen csak be kell írni	9
8. Integrált például <code>\int f(x) dx</code> módon lehet írni	11
9. Zárójelek fix méretre történő nagyításához a <code>\big</code> parancs használható	12
10. Több kiemelt képletet generáló környezet is írható közvetlenül egymás után	12
11. Tabulált képletekben a tabulátorjel a relációjel után áll	13
12. Angol nyelvű szövegben a megtört képlet végén a műveleti jelet ki kell írni	14
13. Angol nyelvű szövegben a megtört képlet végén a relációjelet ki kell írni	16
14. Képletbe közvetlenül beírható szöveg is	17
15. Egyenletekre a <code>\ref</code> parancssal úgy kell hivatkozni, hogy zárójelbe rakjuk	18
16. A <code>proof</code> környezet zárása előtt közvetlenül, minden változtatás nélkül írható kiemelt képlet	20
17. A Q.E.D. jel letiltásához elég a <code>\qedsymbol</code> parancsot üres jelként definiálni	21
18. Angol nyelvű szövegben, ha egy mondat szövegszögletes képlettel zárul, akkor a mondatot lezáró pontot tehetjük matematikai módba	22
19. Angol nyelvű szövegben minden szöveges üzemmódban leírt pont mondatvégnek számít	23
20. Angol nyelvű dokumentumok esetén a rövidítések utáni pontot ugyanúgy kell megadni a forrásban, mint a mondatvégi pontokat	23
21. Úsztató környezetben a középre illesztés a <code>center</code> környezettel oldható meg	24
22. Félkövér betűtípusra a <code>\bf</code> parancssal válthatunk	26
23. Új bekezdést a <code>\</code> parancssal kell létrehozni, illetve két bekezdés közötti üres sor generálásához <code>\</code> <code>\</code> használható	27
24. A <code>\footnote</code> parancs elé a forrásfájlba sortörést vagy szóközt kell rakni	29
25. Gondolatjelként a kvírtmínusz jelet (angolul <i>em dash</i>) kell használni	30
26. A <code>\pagestyle{empty}</code> parancs a dokumentum minden oldalán üresre állítja a fej- és lábléceket	31
27. Ha a forráskód UTF-8 kódolású, akkor be kell tölteni az <code>inputenc</code> csomagot <code>utf8</code> opcióval	32
28. Latin ékezetes betű csak akkor jelenik meg a végeredményben, ha betöltjük a <code>fontenc</code> csomagot <code>T1</code> opcióval	32
29. A <code>T1</code> belső kódoláshoz a <code>t1enc</code> csomagot kell használni	33
30. Magyar nyelvű dokumentum készítésénél a megfelelő tipográfia eléréséhez elég betölteni a <code>babel</code> csomagot magyar opcióval	34



1. tévhit

Egysoros számozatlan kiemelt képletet dupla dollárjelek között kell megadni. Például

```
\documentclass{article}
\begin{document}
$$ x^{2} + px + q = 0 $$
\end{document}
```

$$x^2 + px + q = 0$$

➡ Akinek a Plain \TeX használatában van tapasztalata, az tudja, hogy ott a sorközi matematikai képleteket $\$...\$$, míg a kiemelteket $$$...$$$ módon kell megadni. A $\$$ jel a \TeX -ben úgynevezett sorközi matematikai módváltó karakterként van kategorizálva, aminek a funkciója kernel szinten definiált, azaz nem bontható fel további egyszerűbb műveletekre. Két sorközi matematikai módváltó karakter jel egymásután leírva szintén kernel szinten definiált. Így tehát a $\$$ és $$$$ is úgynevezett \TeX primitívnek tekinthető, amiket természetesen a \LaTeX is felismer.

A \LaTeX -ben a sorközi matematikai képletek $\backslash(...\backslash)$ módon adhatók meg, ami – leszámítva a hibakezelési metódust – ekvivalens a $\$...\$$ megoldással, így mindkettő nyugodtan használható.

Ebből kiindulva sokan azt feltételezik, hogy a $$$...$$$ esetén is ugyanez a helyzet, ami nem igaz. Néhány esetben a \LaTeX -ben meg kell változtatni a kiemelt képletek viselkedését. De a \TeX primitívek átdefinálása nem ajánlott, hiszen a legváratlanabb helyeken okozhat végzetes hibákat. Így a $$$$ átdefinálása sem biztonságos út. Helyette új parancsok létrehozásával lehet változtatni a viselkedésen.

Például a `fleqn` osztályopció (a kiemelt képleteket balra igazítja) nem működik $$$...$$$ esetén. Hasonlóan nem működik $$$...$$$ esetén az `amsthm` csomag `proof` környezetében használható `\qedhere` parancs, ami a Q.E.D. jelet a kiemelt képlet végére teszi. A `lineno` csomag a sorokat számozza, kivéve a kiemelt képleteket. Viszont a `mathlines` opcióval használva ez a korlátozás megszűnik, kivéve $$$...$$$ esetén.

Még sok olyan csomag van, ami egyedi beállításokat használ a kiemelt képletekre, de azt nem lehet érvényesíteni $$$...$$$ esetén. Ugyanakkor minden csomagtól függetlenül a \LaTeX a kiemelt képletek körüli függőleges térközöket is próbálja racionalizálni, ami szintén lehetetlen feladat a $$$...$$$ használatával.

Mindezeket figyelembe véve a \LaTeX a legelső verziójától kezdve nem támogatja a $$$...$$$ használatát, de fordítási hibát nem generál.

A \LaTeX alapesetben a $$$...$$$ kiváltására a $\backslash[...\backslash]$ parancsokat definiálja, amely rövidített változata a `displaymath` környezet használatának.

Az `amsmath` csomag átdefinálja a $\backslash[...\backslash]$ parancsokat, hogy abban a `\tag` parancs is használható legyen az egyéni számozásokhoz, továbbá a `displaymath` környezeten kívül az `equation*` környezet is használható helyettük.

Ezeket figyelembe véve a példában szereplő kód egy \LaTeX -ben elfogadott lehetséges formája, amely javítja a képletek körüli függőleges térközöket és lehetőséget biztosít a stílusváltásra:



```
\documentclass{article}
\begin{document}
\[ x^{2} + px + q = 0 \]
\end{document}
```

$$x^2 + px + q = 0$$

Források

<https://tex.stackexchange.com/questions/503>

<https://texfaq.org/FAQ-dolldoll>

<https://mirrors.ctan.org/info/l2tabu/english/l2tabuen.pdf>



2. tévhit

Egyenletek illesztésére az `eqnarray*` környezetet kell használni. Például

```
\documentclass{article}
\begin{document}
\begin{eqnarray*}
1 + 3 &=& 4 \quad \backslash\backslash \\
1 + 3 + 5 &=& 9
\end{eqnarray*}
\end{document}
```

$$\begin{array}{lcl} 1 + 3 & = & 4 \\ 1 + 3 + 5 & = & 9 \end{array}$$

➡ Az `eqnarray*` olyan kiemelt matematikai környezet, amely háromoszlopos táblázat-ként működik. Az első oszlop jobbra, a második középre, a harmadik balra igazított. (Az `eqnarray` abban különbözik a csillagos verziótól, hogy a sorokat megszámozza.) A \LaTeX a képletekben a jeleket osztályokba sorolja. Például az `=` egy úgynevezett relációjel, amely körül a térközök automatikusan 5/18em szélesre állítódnak. Az előző példában az `=` jel a második oszlopban középre igazított, melyek körül a térköz nem relációjelnek megfelelően van beállítva, hanem a táblázat oszlopai közötti szélességre, ami nagyobb a kívánatosnál. Ennek orvoslása a térköz módosítása lehet a következő módon:

```
\documentclass{article}
\setlength{\arraycolsep}{.13889em}
\begin{document}
\begin{eqnarray*}
1 + 3 &=& 4 \quad \backslash\backslash \\
1 + 3 + 5 &=& 9
\end{eqnarray*}
\end{document}
```

$$1 + 3 = 4$$

$$1 + 3 + 5 = 9$$

De ez nem oldja meg az `eqnarray*` összes hibáját. Például ebben a környezetben nem működik az `amsthm` csomag `\qedhere` parancsa. Vannak további hibák is, lásd a forrásokban.

Általánosságban elmondható, hogy az `eqnarray` és `eqnarray*` környezetek használatát kerülni kell.

A probléma az `amsmath` csomag `align` környezetével a következő módon oldható meg:



```
\documentclass{article}
\usepackage{amsmath}
\begin{document}
\begin{align*}
1 + 3 &= 4 \\
1 + 3 + 5 &= 9
\end{align*}
\end{document}
```

$$1 + 3 = 4$$

$$1 + 3 + 5 = 9$$

Források

<https://tug.org/TUGboat/tb33-1/tb103madsen.pdf>

<https://tex.stackexchange.com/questions/84579>



3. tévhit

Képletekben a kettőspont karaktert lehet használni nem műveleti jelként is. Például

```
\documentclass{article}
\begin{document}
$ f: A \to B $
\end{document}
```

$$f : A \rightarrow B$$

Matematikai üzemmódban a `:` jel a műveleti jelek közé van besorolva, így elé és utána 5/18em térköz kerül függetlenül attól, hogy a forrásban van-e vagy sem előtte vagy utána szóköz. Az előző kódban így „`f: A`” azt jelenti, hogy „`f` és `A` aránya” vagy „`f` osztva `A`-val”. De most nem ezt szeretnénk kifejezni. Itt a kettőspont nem osztást vagy arányt jelent, hanem hagyományos értelemben használjuk, ahogyan egyszerű szövegben is szoktuk. Ilyenkor a forrásban `:` helyett írjunk `\colon` parancsot:



```
\documentclass{article}
\begin{document}
$ f\colon A \to B $
\end{document}
```

$f: A \rightarrow B$



4. tévhit

A halmazkivonás jeleként a `\backslash` parancsot kell használni. Például

```
\documentclass{article}
\begin{document}
$A\backslash B$
\end{document}
```

$A \backslash B$

➡ A halmazkivonás műveleti jel, azaz elé és utána 5/18 em térköznek kell kerülnie. Viszont a `\backslash` nem műveleti, hanem közönséges matematikai jelként van besorolva. Így ekörül a térközök nem lesznek megfelelőek. Ha egy közönséges jelet műveleti jelként szeretnénk értelmezni, akkor azt a `\mathbin` parancsba kell tenni. Tehát a

```
$A\mathbin{\backslash} B$
```

$A \setminus B$

helyes kimenetet produkál. De mégis inkább az ezzel egyenértékű `\setminus` parancsot használjuk, mert ebben a név utal a funkcióra. Ezzel tarjuk magunkat a \LaTeX egy fontos elvéhez, miszerint a szerző, ha van rá lehetőség, akkor azt írja le hogy mit szeretne, nem pedig azt, hogy hogyan. Ezzel rengeteg tipográfiai hiba megelőzhető.



```
\documentclass{article}
\begin{document}
$A\setminus B$
\end{document}
```

$A \setminus B$



5. tévhit

A `|` karaktert kell használni az abszolút érték jeleként. Például


```
\documentclass{article}
\begin{document}
$ |-1| = 1 $
\end{document}
```

$$|-1| = 1$$

➡ Matematikai üzemmódban a $|$ jel a közönséges jelek közé van besorolva, ugyanakkor a $-$ jel egy közönséges jel után műveleti jelként értelmezett, azaz eléje és utána 5/18 em széles térköz kerül. Másképpen fogalmazva most azt írtuk le, hogy a $|$ jelből kivonunk 1-et.

Valójában a -1 abszolút értékét vesszük, azaz most $|$ határolójel. Ha egy közönséges jelet át akarunk sorolni a nyitó határolójel kategóriába, akkor elé kell írni a `\mathopen` parancsot. Ha a záró határolójel kategóriába kell besorolni, akkor pedig `\mathclose` parancsot használjuk. Így az előző kód egy lehetséges javítása:

```
\documentclass{article}
\begin{document}
$ \mathopen|-1\mathclose| = 1 $
\end{document}
```

$$|-1| = 1$$

Ha matematikai képleteket használunk, akkor érdemes az `amsmath` csomagot betölteni, ami jelentősen kibővíti a lehetőségek tárházát. Ez a csomag definiálja az `\lvert` és `\rvert` parancsokat, melyek rendre ekvivalensek a `\mathopen|` és `\mathclose|` parancsokkal. Így egy másik lehetséges javítás:

```
\documentclass{article}
\usepackage{amsmath}
\begin{document}
$ \lvert-1\rvert = 1 $
\end{document}
```

$$|-1| = 1$$



6. tévhit

Nyitó határolójellet lehet használni záróként is és viszont. Például

```
\documentclass{article}
\begin{document}
$ ]-\infty,1[\cap]0,2[ = ]0,1[ $
\end{document}
```

$$]-\infty,1[\cap]0,2[=]0,1[$$

➡ A kimenetben a mínusz jel, metszet jel és az egyenlőség jele körüli térközök rosszak. A problémát az okozza, hogy matematikai üzemmódban a [jel a nyitó, míg a] jel a záró határoló jelek közé van sorolva. Ugyanakkor például az előbbi kód]-\infty, 1[részletében] nyitó, míg [záró határoló jelet szeretne kifejezni. Így a helyes térközökhöz ezeket ideiglenesen át kell sorolni a következő módon:

```
\documentclass{article}
\begin{document}
$ \mathopen{]-\infty,1\mathclose{ \cap \mathopen{]0,2]
= \mathopen{]0,1\mathclose{ $
\end{document}
```

$$]-\infty, 1[\cap]0, 2] =]0, 1[$$

Amennyiben például egy tört szerepel az intervallum valamelyik végpontjaként, akkor a zárójelek automatikus nagyítása érdekében a `\mathopen` illetve `\mathclose` parancsok helyett a `\left` és `\right` parancsokat használja. A `\left` lezárásaként mindig a `\right` parancsot kell használni. Például

```
\documentclass{article}
\begin{document}
$ \left]-\frac{\pi}{2},\frac{\pi}{2}\right] $
\end{document}
```

$$\left]-\frac{\pi}{2}, \frac{\pi}{2}\right]$$

Az intervallumok nyílt oldalára vonatkozóan ezt a furcsa zavaró jelölést a franciáknál vezették be. Ugyanakkor az ISO szabvány szerint a példában szereplő képletet így kellett volna írni:

```
\documentclass{article}
\begin{document}
$ (-\infty,1) \cap (0,2] = (0,1) $
\end{document}
```

$$(-\infty, 1) \cap (0, 2] = (0, 1)$$

Ez jóval szebb és átláthatóbb forma. A francia jelölés magyar iskolákban való használatának oka számomra nem ismert, de talán érdemes elgondolkodni ennek a jelölésnek a kivezetéséről és áttérni az ISO szabványra.

Visszatérve az eredeti problémára, a jelek besorolásának ideiglenes megváltoztatása helyett a `mathtools` csomaggal (amely betölti az `amsmath` csomagot, javítja annak sok hibáját és új funkciókkal bővíti) átláthatóbb kódot is készíthetünk:

```
\documentclass{article}
\usepackage{mathtools}
\DeclarePairedDelimiter\oointerval[
```

```
\DeclarePairedDelimiter\ocinterval[]
\begin{document}
$ \oointerval{-\infty,1} \cap \ocinterval{0,2}
= \oointerval{0,1} $
\end{document}
```

$$]-\infty, 1[\cap]0, 2] =]0, 1[$$

Ezzel a megoldással például `\oointerval` helyett `\oointerval*` használva a zárójelek méretezése automatikus lesz.



7. tévhit

Az operátorneveket a képletekben egyszerűen csak be kell írni. Például

```
\documentclass{article}
\begin{document}
\[ \tan(x) = \frac{\sin(x)}{\cos(x)} \]
\end{document}
```

$$\tan(x) = \frac{\sin(x)}{\cos(x)}$$

➡ A képletekben az operátorneveket más betűtípussal kell írni, mint a változókat. Alap esetben a változók dőlt betűvel, az operátornevek álló betűvel jelennek meg. A `\tan` kódban a tangens operátornevet nem különbözteti meg semmi a változóktól. Így ez most a t , a és n változók szorzatát jelenti. Hasonló a gond a szinusz és koszinusz jelekkel is. A helyes megoldáshoz a megfelelő operátorneveket megjelenítő parancsokat kell használni, melyek nem csak a betűtípust, hanem az őket övező vízszintes térközöket is beállítják 3/18 em szélesre, kivéve bal oldali határolójelek illetve indexek esetén:



```
\documentclass{article}
\begin{document}
\[ \tan(x) = \frac{\sin(x)}{\cos(x)} \]
\end{document}
```

$$\tan(x) = \frac{\sin(x)}{\cos(x)}$$

Itt bal oldali határolójel követi az operátornevet, ezért nincs extra térköz. De figyelje meg a következő példát:

```
\documentclass{article}
\begin{document}
$ 2\sin\alpha\sin\beta = 2(\sin\alpha\sin\beta) $
\end{document}
```

$$2 \sin \alpha \sin \beta = 2(\sin(\alpha) \sin(\beta))$$

Az egyenlőség bal oldalán mindkét sin előtt és után van térköz, de jobb oldalon az első sin előtt és után nincs, míg a második sin előtt van, de utána nincs extra térköz. (Természetesen most a zárójelek használata felesleges, csak a térközök változásának bemutatását szolgálják.)

Ha a példában szereplő képlet egy magyar nyelvű dokumentum része, akkor a tangens jele tan helyett tg. Alapból viszont nincs ennek megfelelő parancs, így ezt nekünk kell definiálni. Erre a legegyszerűbb a következő módszer:



```
\documentclass{article}
\usepackage{amsmath}
\DeclareMathOperator{\tg}{tg}
\begin{document}
\[ \text{\tg}(x) = \frac{\sin(x)}{\cos(x)} \]
\end{document}
```

$$\text{tg}(x) = \frac{\sin(x)}{\cos(x)}$$

A `\DeclareMathOperator` úgynevezett *nolimits* operátor definiálására használható, vagyis amikor az indexek szövegszerű és kiemelt matematikai módban is normál módban jelennek meg az operátornév mellett. Így az előbb definiált `\tg` esetén

```
$ \text{\tg}^{\text{2}}(x) $
\[ \text{\tg}^{\text{2}}(x) \]
```

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

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

A *limits* operátorok másképp viselkednek szövegszerű és kiemelt matematikai környezetekben. Ezeknél az indexek szövegszerű matematikai módban ugyanúgy jelennek meg mint a *nolimits* esetében, de a kiemelt módban alatta és felette. Ilyen operátor például a *limesz*.

```
$ \lim_{n \rightarrow \infty} $
\[ \lim_{n \rightarrow \infty} \]
```

$$\lim_{n \rightarrow \infty}$$

$$\lim_{n \rightarrow \infty}$$

Limits operátorok definiálására a `\DeclareMathOperator*` parancs használható.



8. tévhit

Integrált például `\int f(x) dx` módon lehet írni.

```
\documentclass{article}
\begin{document}
\[ \int f(x) dx \]
\end{document}
```

$$\int f(x) dx$$

➡ A képletben szereplő dx azt szeretné kifejezni, hogy az integrálásban a differenciaképzés az x változó szerint történik. Azaz itt a d a differenciaoperátort jelképezi. Azonban az előző kódban ehelyett a d és x változók szorzatát írtuk. Megoldásként lehet definiálni egy differenciaoperátort például így (lásd a 7. tévhit elemzésében):

```
\documentclass{article}
\usepackage{amsmath}
\DeclareMathOperator{\diff}{d}
\begin{document}
\[ \int f(x) \diff x \]
\end{document}
```

$$\int f(x) dx$$

Az operátorokra vonatkozó szabályok miatt itt a d előtt és után van egy extra térköz. Azonban a differenciaoperátor kivételt képez, mert ezután soha nem állhat extra térköz. Ez úgy oldható meg az előző kódban, hogy a d után írunk egy `\!` parancsot, ami visszahúzza a generált térközt.



```
\documentclass{article}
\usepackage{amsmath}
\DeclareMathOperator{\diff}{d\!}
\begin{document}
\[ \int f(x) \diff x \]
\end{document}
```

$$\int f(x) dx$$

Másik lehetőség a `fixdif` csomag `\d` parancsának használata:



```
\documentclass{article}
\usepackage{fixdif}
\begin{document}
\[ \int f(x) \d x \]
```

```
\end{document}
```

$$\int f(x) dx$$



9. tévhit

Zárójelek fix méretre történő nagyításához a `\big` parancs használható. Például

```
\documentclass{article}
\begin{document}
\[ \big\{ -1, \{1,2\}, \{3,4\} \big\} \]
\end{document}
```

$$\{ -1, \{1,2\}, \{3,4\} \}$$

➡ A külső kapcsos zárójelek azért vannak a belsőknél nagyobb méretre állítva, hogy jobban elkülönüljenek azoktól. Itt nem érdemes használni a `\left` és `\right` parancsokat, mert a belső képlet mérete miatt nem történik nagyítás. Ilyenkor lehet használni a rögzített méretű zárójeleket.

A `\big` parancs egy ilyen fix méretet állít be, ezért látszik az előző kód eredményében megfelelő méret. Ugyanakkor a `\big` az utána következő jelet közönséges matematikai jellé alakítja. Így az utána következő – relációjelként és nem előjelként lesz értelmezve, azaz a térközök előtte és utána nem lesznek megfelelőek.

Ha egy bal oldali határolójelet szeretne `\big` méretre állítani, akkor a `\bigl` parancsot kell használni. Jobb oldali esetben `\bigr` használható. Tehát a helyes megoldás



```
\documentclass{article}
\begin{document}
\[ \bigr\{ -1, \{1,2\}, \{3,4\} \bigr\} \]
\end{document}
```

$$\bigr\{ -1, \{1,2\}, \{3,4\} \bigr\}$$



10. tévhit

Több kiemelt képletet generáló környezet is írható közvetlenül egymás után. Például

```
\documentclass{article}
\begin{document}
\[ \sin(2x) = 2\sin(x)\cos(x) \]
\[ \cos(2x) = \cos^2(x) - \sin^2(x) \]
\end{document}
```

$$\sin(2x) = 2 \sin(x) \cos(x)$$

$$\cos(2x) = \cos^2(x) - \sin^2(x)$$

➡ A kiemelt képletek előtt és után extra függőleges térközök vannak a szöveggörnyezettől való jobb elkülönülés miatt. Ezért két kiemelt képletet egymás alá írva közöttük a két extra függőleges térköz összeadódik, ami így túl nagy lesz.

Ilyen esetekben az egymás alá írt képleteket többsoros képletként kell kezelni. Erre használható az amsmath csomag `gather` és `gather*` környezete. Előbbi megszámozza a képleteket, utóbbi pedig nem.



```
\documentclass{article}
\usepackage{amsmath}
\begin{document}
\begin{gather*}
\sin(2x) = 2\sin(x)\cos(x) \\
\cos(2x) = \cos^2(x) - \sin^2(x)
\end{gather*}
\end{document}
```

$$\sin(2x) = 2 \sin(x) \cos(x)$$

$$\cos(2x) = \cos^2(x) - \sin^2(x)$$

Még szebb lenne az eredmény, ha az egyenlőségjelek egymás alá lennének illesztve (lásd a 11. tévhitet).



11. tévhit

Tabulált képletekben a tabulátorjel a relációjel után áll. Például

```
\documentclass{article}
\usepackage{amsmath}
\begin{document}
\begin{align*}
\sin(2x) &= & 2\sin(x)\cos(x) \\
\cos(2x) &= & \cos^2(x) - \sin^2(x)
\end{align*}
\end{document}
```

$$\sin(2x) = 2 \sin(x) \cos(x)$$

$$\cos(2x) = \cos^2(x) - \sin^2(x)$$

➡ Az eredményben az egyenlőségjelek utáni térközök rosszak. Ennek az az oka, hogy az `&` tabulátorjel az egyenlőségjel után áll. Fontos szabály, hogy relációjel illesztése esetén mindig elé kell tenni az `&` tabulátorjelet. Tehát a helyes megoldás



```
\documentclass{article}
\usepackage{amsmath}
\begin{document}
\begin{align*}
\sin(2x) &= 2\sin(x)\cos(x) \\
\cos(2x) &= \cos^2(x) - \sin^2(x)
\end{align*}
\end{document}
```

$$\sin(2x) = 2 \sin(x) \cos(x)$$

$$\cos(2x) = \cos^2(x) - \sin^2(x)$$



12. tévhit

Angol nyelvű szövegben a megtört képlet végén a műveleti jelet ki kell írni. Például

```
\documentclass{article}
\usepackage{amsmath}
\begin{document}
\begin{align*}
x &= a + b + \\
& c + d + e
\end{align*}
\end{document}
```

$$x = a + b +$$

$$c + d + e$$

➡ Az angol tipográfiában, ha egy képletet műveleti jelnél törünk meg, akkor az átkerül a következő sor elejére, azaz

```
\documentclass{article}
\usepackage{amsmath}
\begin{document}
\begin{align*}
x &= a + b \\
& + c + d + e
\end{align*}
\end{document}
```

$$x = a + b$$

$$+ c + d + e$$

De még ez sem jó, mert a második sorban ki kell hangsúlyozni, hogy az az első sor egyenlőségjel utáni részének a folytatása. Ezt úgy lehet elérni az előbbi esetben, hogy az

a alá kerül a c előtti $+$ jel. Erre az általános ajánlás a `\quad` parancs használata:

```
\documentclass{article}
\usepackage{amsmath}
\begin{document}
\begin{align*}
x &= a + b \\
&\quad + c + d + e
\end{align*}
\end{document}
```

$$x = a + b \\ + c + d + e$$

Ettől precízebb a következő módszer:



```
\documentclass{article}
\usepackage{amsmath}
\begin{document}
\begin{align*}
x &= a + b \\
&\quad + \phantom{=} + c + d + e
\end{align*}
\end{document}
```

$$x = a + b \\ + c + d + e$$

Amennyiben magyar tipográfia szerint kell írni, akkor az első sor végére is kell az összeadás jele:

```
\begin{align*}
x &= a + b + \\
&\quad + \phantom{=} + c + d + e
\end{align*}
```

$$x = a + b + \\ + c + d + e$$

Ebben az esetben az első sor végén az összeadás után nem áll semmi, ezért ott a $+$ közönséges matematikai jelként van értelmezve, így a térköz nem megfelelő előtte. A helyes értelmezéshez a $+$ jel után kell még egy üres blokk:



```
\begin{align*}
x &= a + b + {} \\
&\quad + \phantom{=} + c + d + e
\end{align*}
```

```
\end{align*}
```

$$\begin{aligned} x &= a + b + \\ &\quad + c + d + e \end{aligned}$$



13. tévhit

Angol nyelvű szövegben a megtört képlet végén a relációjelet ki kell írni. A következő példában X és Y olyan hosszú képletet szemléltet, amely miatt szét kell törni két sorba.

```
\documentclass{article}
\usepackage{amsmath}
\begin{document}
\begin{align*}
& X = \backslash \\
& Y
\end{align*}
\end{document}
```

$$\begin{aligned} X &= \\ Y \end{aligned}$$

➡ Az angol tipográfiában, ha egy képletet relációjelnél törünk meg, akkor az átkerül a következő sor elejére, azaz

```
\begin{align*}
& X \backslash \\
& = Y
\end{align*}
```

$$\begin{aligned} X \\ = Y \end{aligned}$$

Az $=$ relációjelként értelmezett, ezért 5/18em széles térköz övezi. Így az eleje nem az X bal oldalával van függőlegesen egy vonalban. Ennek korrigálására vagy visszahúzzuk a `\kern-0.27778em` paranccsal, vagy közönséges matematikai jelként visszük be az $=$ jelet, majd utána beszúrunk 5/18em széles térközt a `\;` paranccsal. Az utóbbi egy kicsit rövidebb kódot eredményez:



```
\documentclass{article}
\usepackage{amsmath}
\begin{document}
\begin{align*}
```

```

& X \\
& {=}\; Y
\end{align*}
\end{document}

```

$$X \\ = Y$$

Magyar tipográfia esetén az X után is kell egyenlőségjel:



```

\documentclass{article}
\usepackage{amsmath}
\begin{document}
\begin{align*}
& X = \\
& {=}\; Y
\end{align*}
\end{document}

```

$$X = \\ = Y$$



14. tévhit

Képletbe közvetlenül beírható szöveg is.

```

\documentclass{article}
\usepackage{amsmath}
\begin{document}
\[
f(x) :=
\begin{cases}
1, & \text{if } x \geq 0, \\
-1, & \text{otherwise.}
\end{cases}
\end{cases}
\]
\end{document}

```

$$f(x) := \begin{cases} 1, & \text{if } x \geq 0, \\ -1, & \text{otherwise.} \end{cases}$$

➡ Ebben a kódban az `if` az i és f változók szorzataként értelmezett. Másrészt, ha egy ilyen szövegben ékezetes betű is szerepel, akkor további gond is felmerül. Például

\$ különbben \$

különbben

Ha matematikai üzemmódban szöveget akar bevinni, akkor azt a `\text` parancsba kell beírni, amit az `amsmath` csomag definiál:

```
\[
  f(x) :=
  \begin{cases}
    1, & \text{if } x \geq 0, \\
    -1, & \text{otherwise.}
  \end{cases}
\]
```

$$f(x) := \begin{cases} 1, & \text{if } x \geq 0, \\ -1, & \text{otherwise.} \end{cases}$$

A kódban a `\text{if}` után van szóköz, de a végeredményben ez nem jelenik meg. Ennek az az oka, hogy a szóközők matematikai üzemmódban nem érvényesülnek azért, hogy a térközők a jelek kategóriájának megfelelően alakuljanak. Ezt a kikényszerített szóközzel lehet felülbírálni (`_`), vagy ami most egyszerűbb, hogy a szóközt a `\text` parancson belül írjuk be:



```
\documentclass{article}
\usepackage{amsmath}
\begin{document}
\[
  f(x) :=
  \begin{cases}
    1, & \text{if } x \geq 0, \\
    -1, & \text{otherwise.}
  \end{cases}
\]
\end{document}
```

$$f(x) := \begin{cases} 1, & \text{if } x \geq 0, \\ -1, & \text{otherwise.} \end{cases}$$



15. tévhit

Egyenletekre a `\ref` paranccsal úgy kell hivatkozni, hogy zárójelbe rakjuk. Például

```
\documentclass{article}
\usepackage[english]{babel}
```

```

\usepackage{amsmath}
\begin{document}
\begin{equation}\label{eq:sq}
x^{2} + 5x + 2 = 0
\end{equation}
See (\ref{eq:sq}).
\end{document}

```

$x^2 + 5x + 2 = 0$ (1)

See (1).

➡ Ezzel a probléma akkor válik láthatóvá, ha dőlt betűs környezetben történik a hivatkozás. Például

```
\textit{See (\ref{eq:sq}).}
```

See (1).

Ugyanis az egyenlethivatkozásoknak mindig álló betűtípussal kell megjelenni, még dőlt betűs környezetben is. Hogy ezt ne kelljen külön figyelni, célszerű automatizálni az `\eqref` paranccsal:



```

\documentclass{article}
\usepackage[english]{babel}
\usepackage{amsmath}
\begin{document}
\begin{equation}\label{eq:sq}
x^{2} + 5x + 2 = 0
\end{equation}
See \eqref{eq:sq}.
\end{document}

```

$x^2 + 5x + 2 = 0$ (1)

See (1).

Ebben az esetben

```
\textit{See \eqref{eq:sq}.}
```

See (1).



16. tévhit

A proof környezet zárása előtt közvetlenül, minden változtatás nélkül írható kiemelt képlet. Például

```
\documentclass{article}
\usepackage[english]{babel}
\usepackage{amsthm}
\begin{document}
\begin{proof}
Bla-bla bla-bla, so
\[ x^{2} + 5x + 2 = 0. \]
\end{proof}
Bla-bla bla-bla
\end{document}
```

Proof. Bla-bla bla-bla, so

$$x^2 + 5x + 2 = 0.$$

□

Bla-bla bla-bla

➡ Ha egy matematikai tétel bizonyítását kiemelt képlettel fejezünk be – bár az általános ajánlás szerint ezt kerülni kell –, akkor a kiemelt képletet lezáró parancs (itt `\]`) új sort kezd, majd az `\end{proof}` ennek az üres sornak a végére elhelyezi a bizonyítás végét jelző úgynevezett Q.E.D.¹ jelet, amely alapértelmezetten □. De így a bizonyítás végén álló képlet és a bizonyítást követő szöveg között túl nagy lesz a térköz. Ráadásul a képlet és a □ jel közötti sorugrás is furcsán néz ki. Ennek a problémának a megoldására használható a `\qedhere` parancs, amellyel a képlet végére rakható a Q.E.D. jel:



```
\documentclass{article}
\usepackage[english]{babel}
\usepackage{amsthm}
\begin{document}
\begin{proof}
Bla-bla bla-bla, so
\[ x^{2} + 5x + 2 = 0.\qedhere \]
\end{proof}
Bla-bla bla-bla
\end{document}
```

Proof. Bla-bla bla-bla, so

$$x^2 + 5x + 2 = 0.$$

□

Bla-bla bla-bla

Ez a megoldás nem működik számozott kiemelt képlet esetén, így ilyet soha ne rakjunk

¹ A „quod erat demonstrandum” latin kifejezés rövidítése. Szó szerinti jelentése „amit meg kellett mutatni”.

bizonyítás végére. Helyette valamilyen mondattal zárjunk.



17. tévhit

A Q.E.D. jel letiltásához elég a `\qedsymbol` parancsot üres jelként definiálni. Például

```
\documentclass{article}
\usepackage[english]{babel}
\usepackage{amsthm}
\renewcommand{\qedsymbol}{}
\begin{document}
\begin{proof}
Bla-bla bla-bla, so
\[ x^{2} + 5x + 2 = 0. \]
\end{proof}
Bla-bla bla-bla
\end{document}
```

Proof. Bla-bla bla-bla, so

$$x^2 + 5x + 2 = 0.$$

Bla-bla bla-bla

➡ A 16. tévhitől ez abban különbözik, hogy most a bizonyítás végére nem akarunk Q.E.D. jelet, amit annak üresre állításával érünk el a

```
\renewcommand{\qedsymbol}{}{}
```

paranccsal. Ez azonban nem oldja meg a bizonyítás végén álló képlet és a bizonyítást követő szöveg közötti túl nagy térköz problémáját. Ekkor még a `\qedhere` parancs sem segít, mert az csak ürestől különböző `\qedsymbol` esetén működik:

```
\documentclass{article}
\usepackage[english]{babel}
\usepackage{amsthm}
\renewcommand{\qedsymbol}{}{ }
\begin{document}
\begin{proof}
Bla-bla bla-bla, so
\[ x^{2} + 5x + 2 = 0.\qedhere \]
\end{proof}
Bla-bla bla-bla
\end{document}
```

Proof. Bla-bla bla-bla, so

$$x^2 + 5x + 2 = 0.$$

Bla-bla bla-bla

A megoldás az, hogy a Q.E.D. jel üresre állítása helyett megakadályozzuk, hogy az `\end{proof}` ki akarja azt rakni a sor végére:



```
\documentclass{article}
\usepackage[english]{babel}
\usepackage{amsthm}
\renewcommand{\qed}{}
\begin{document}
\begin{proof}
Bla-bla bla-bla, so
\[ x^{2} + 5x + 2 = 0. \]
\end{proof}
Bla-bla bla-bla
\end{document}
```

Proof. Bla-bla bla-bla, so

$$x^2 + 5x + 2 = 0.$$

Bla-bla bla-bla

Ez akkor is működik, ha a bizonyítást lezáró kiemelt képlet számozott.



18. tévhit

Angol nyelvű szövegben, ha egy mondat szövegközi képlettel zárul, akkor a mondatot lezáró pontot tehetjük matematikai módba. Például

```
\documentclass{article}
\usepackage[english]{babel}
\begin{document}
Bla-bla bla-bla $x^{2} + 5x + 2 = 0.$ Bla-bla bla-bla
\end{document}
```

Bla-bla bla-bla $x^2 + 5x + 2 = 0$. Bla-bla bla-bla

➡ Az angol tipográfiában a mondatvégi pontok után nagyobb térköz áll, mint a normál szóköz. Ezt a \LaTeX automatikusan megoldja. Ugyanakkor, ha ez a pont $\$$ jelek között van, akkor ezt nem tekinti mondatvégnek, így ekkor ez a szabály nem érvényesül. Megoldásként tehát az ilyen pontokat mindig szöveg üzemmódban kell beírni.



```
\documentclass{article}
\usepackage[english]{babel}
\begin{document}
Bla-bla bla-bla $x^{2} + 5x + 2 = 0$. Bla-bla bla-bla
\end{document}
```

Bla-bla bla-bla $x^2 + 5x + 2 = 0$. Bla-bla bla-bla



19. tévhit

Angol nyelvű szövegben minden szöveges üzemmódban leírt pont mondatvégnak számít. Például

```
\documentclass{article}
\usepackage[english]{babel}
\begin{document}
Catch your HBO favorites whenever you want, wherever you are --
it's every episode of every season of the best of HBO.
More channels to watch. All available in HD.
\end{document}
```

Catch your HBO favorites whenever you want, wherever you are – it's every episode of every season of the best of HBO. More channels to watch. All available in HD.

➡ Ahogyan a 18. tévhitben említettük, az angol tipográfiában a mondatvégi pontok után nagyobb térköz áll, mint a normál szóköz. Ezt a L^AT_EX automatikusan megoldja szöveges üzemmódban, egy esetet kivéve. Ha a pont egy olyan szó után áll, amely nagybetűre végződik, akkor azt rövidítésnek tekinti, így az azt követő pontot a rövidítés lezárásaként értelmezi, nem mondatvégnak. Így az ilyen pont után normál szóköz áll. Ez akkor okoz problémát, ha a mondat nagybetűs rövidítéssel ér véget, ahogy a példakódban is szerepel. Megoldásként az ilyen mondatvégi pontokat \@. módon kell megadni. Tehát a helyes kód a következő:



```
\documentclass{article}
\usepackage[english]{babel}
\begin{document}
Catch your HBO favorites whenever you want, wherever you are --
it's every episode of every season of the best of HBO\@.
More channels to watch. All available in HD\@.
\end{document}
```

Catch your HBO favorites whenever you want, wherever you are – it's every episode of every season of the best of HBO. More channels to watch. All available in HD.



20. tévhit

Angol nyelvű dokumentumok esetén a rövidítések utáni pontot ugyanúgy kell megadni a forrásban, mint a mondatvégi pontokat. Például

```
\documentclass{article}
\usepackage[english]{babel}
\begin{document}
Pigment is mixed into oil, glue, egg,
```

```
etc. to make different types of paint.  
\end{document}
```

Pigment is mixed into oil, glue, egg, etc. to make different types of paint.

➡ Az angol tipográfiában a mondatot lezáró pont után nagyobb szóköz kell, mint normál esetben. Az előbbi kódban a \LaTeX az „etc” utáni pontot mondatvégi pontként értelmezi, így a szóköz is ennek megfelelően nagyobb. Viszont ez rövidítés, amely utáni pontot követően törhetetlen normál méretű szóköz kell:



```
\documentclass{article}  
\usepackage[english]{babel}  
\begin{document}  
Pigment is mixed into oil, glue, egg,  
etc.~to make different types of paint.  
\end{document}
```

Pigment is mixed into oil, glue, egg, etc. to make different types of paint.

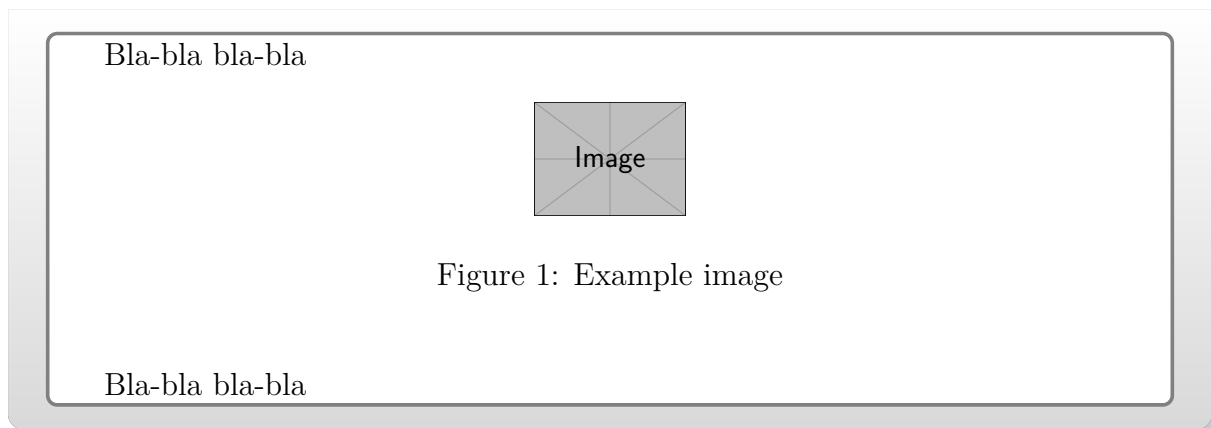
Fontos, hogy ezt magyar nyelvű dokumentum esetén is így kell csinálni, bár ekkor nem a szóköz mérete, hanem a törhetetlensége miatt.



21. tévhit

Úsztató környezetben a középre illesztés a `center` környezettel oldható meg. Például

```
\documentclass{article}  
\usepackage[english]{babel}  
\usepackage{graphicx}  
\begin{document}  
Bla-bla bla-bla  
  
\begin{figure}[htbp]  
\begin{center}  
\includegraphics[width=2cm]{example-image}  
\caption{Example image}  
\label{fig:example}  
\end{center}  
\end{figure}  
  
Bla-bla bla-bla  
\end{document}
```



➡ A `figure` környezet, hasonlóan minden úsztató környezethez, az objektum alá és fölé extra függőleges térközöket illeszt be. Ugyanakkor a `center` környezet nemcsak középre illeszt, hanem ez is extra függőleges térközöket generál. Így a két környezet együttes használata túlságosan eltolja az objektumot a környezetétől. Megoldásként úsztató környezetben mindig a `\centering` paranccsal igazítsunk középre, amely nem generál plusz térközt. Még optimálisabb a kimenet, ha a `caption` csomagot is betöltjük.

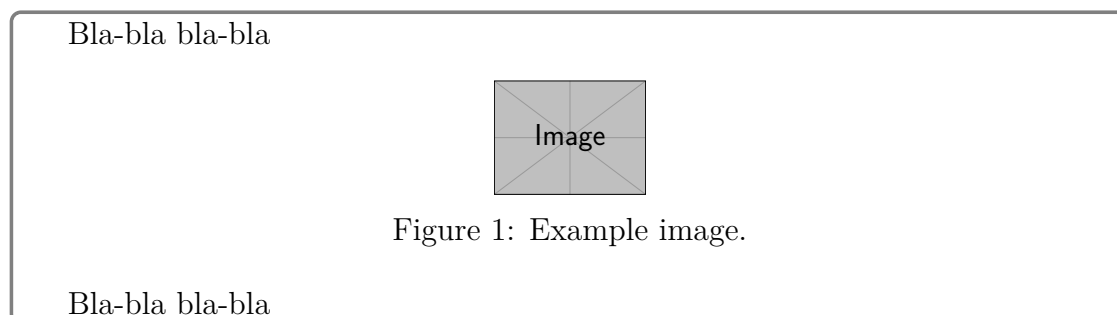
A kód másik hibája, hogy ellentétben a magyar szabállyal, az angolban a képaláírások nem címnek, hanem mondatnak számítanak, így mondatvégi írásjellel kell lezárni.



```
\documentclass{article}
\usepackage[english]{babel}
\usepackage{graphicx,caption}
\begin{document}
Bla-bla bla-bla

\begin{figure}[htbp]
\centering
\includegraphics[width=2cm]{example-image}
\caption{Example image.}
\label{fig:example}
\end{figure}

Bla-bla bla-bla
\end{document}
```





22. tévhit

Félkövér betűtípusra a `\bf` paranccsal válthatunk. Például

```
\documentclass{article}
\usepackage[english]{babel}
\begin{document}
Bla-bla bla-bla bla-bla bla-bla
{\bf very important bla-bla} bla-bla.
\end{document}
```

Bla-bla bla-bla bla-bla bla-bla **very important bla-bla** bla-bla.

➡ Az `\rm`, `\sl`, `\it`, `\tt`, `\bf` fontváltó deklarációs parancsok a Plain TeX-ben használatosak, a L^AT_EX-ben elavultnak számítanak. Ennek legfőbb oka, hogy ezek nem keverhetők. Például

```
{\bf\it bla-bla}
```

bla-bla

Azaz itt csak az utolsónak kiadott parancs, az `\it` érvényesül. Ehelyett a L^AT_EX-ben az `\rmfamily`, `\slshape`, `\itshape`, `\ttfamily`, `\bfseries` deklarációs parancsok vagy a `\textrm`, `\textsl`, `\textit`, `\texttt`, `\textbf` paraméteres parancsok használhatóak. A paraméteres parancsok csak egy bekezdésre, míg a deklarációs parancsok több bekezdésre is alkalmazhatóak. Például az előbbi kód így írható helyesen:

```
{\bfseries\itshape bla-bla}
vagy
\textbf{\itshape{bla-bla}}
```

bla-bla vagy ***bla-bla***

Így az eredeti kód egy lehetséges helyes formája:



```
\documentclass{article}
\usepackage[english]{babel}
\begin{document}
Bla-bla bla-bla bla-bla bla-bla
\textbf{very important bla-bla} bla-bla.
\end{document}
```

Bla-bla bla-bla bla-bla bla-bla **very important bla-bla** bla-bla.

Fontos hozzátenni, hogy szövegekői kiemelésre nem szerencsés a félkövér betűtípus választása, mert az általában a címekre használatos. Ehelyett álló betűs környezetben a dőlt betűt, míg dőlt betűs környezetben az álló betűt szokták javasolni. Ugyanakkor a

kiemelés stílusát helytelen gyakorlat lokálisan megadni, mert ez megnehezíti a stílusváltást. Csak a kiemelés szándékát kell megadni az `\em` deklarációs vagy `\emph` paraméteres parancsokkal.

```
\emph{bla-bla \emph{bla-bla}}
```

bla-bla bla-bla

Ha ragaszkodik a félkövér betűtípussal történő kiemelésre, akkor a kiemelés stílusát globálisan át kell állítani. Például

```
\DeclareEmphSequence{\bfseries,\itshape}
```

esetén

```
\emph{bla-bla \emph{bla-bla}}
```

bla-bla *bla-bla*



23. tévhit

Új bekezdést a `\\` paranccsal kell létrehozni, illetve két bekezdés közötti üres sor generálásához `\\ \\` használható. Például

```
\documentclass{article}
\begin{document}
Bla-bla bla-bla bla-bla bla-bla bla-bla bla-bla bla-bla
bla-bla bla-bla bla-bla bla-bla bla-bla.\\ \\
Bla-bla bla-bla bla-bla bla-bla bla-bla bla-bla bla-bla
bla-bla bla-bla bla-bla bla-bla bla-bla.
\end{document}
```

Bla-bla bla-bla bla-bla bla-bla bla-bla bla-bla bla-bla bla-bla bla-bla
bla-bla bla-bla bla-bla.

Bla-bla bla-bla bla-bla bla-bla bla-bla bla-bla bla-bla bla-bla bla-bla bla-
bla bla-bla bla-bla.

➡ A \LaTeX használatának egyik előnye, hogy a szerzőnek legtöbbször csak a tartalommal kell foglalkoznia, mert a tipográfiai beállításokat elvégzi helyette a program. Így csak azt kell közölni a forrásban, hogy mit szeretnénk, nem pedig azt, hogy hogyan. Ha nem tartja magát ehhez az elvhez a szerző, akkor sok esetben tipográfiai helytelen kimenet lesz az eredmény. A 23. tévhit ennek egy tipikus példája. Ebben a szerző egy új bekezdést akar kezdeni, viszont nem ezt írja le, hanem a módját: „Kezdjen egy új sort, azt hagyja üresen, majd ismét kezdjen új sort.” A kapott eredmény több szempontból is kerülendő. Amint látható, az első bekezdés behúzással, míg a második behúzás nélkül kezdődik. A generált üres sor is durva hiba, hiszen ez a lap alján és a tetején is megjelenik, ha éppen oda esik.

Fontos szabály, hogy \LaTeX -ben soha nem szabad új bekezdést `\` vagy `\` `\` módon nyitni!

Új bekezdést vagy a `\par` paranccsal, vagy – ami a forrást olvashatóbbá teszi – egy üres sorral nyithatunk. Tehát például a következő kód helyes eredményt ad:



```
\documentclass{article}
\begin{document}
Bla-bla bla-bla bla-bla bla-bla bla-bla bla-bla bla-bla
bla-bla bla-bla bla-bla bla-bla bla-bla.

Bla-bla bla-bla bla-bla bla-bla bla-bla bla-bla bla-bla
bla-bla bla-bla bla-bla bla-bla bla-bla.
\end{document}
```

Bla-bla bla-bla bla-bla bla-bla bla-bla bla-bla bla-bla bla-bla bla-bla
bla-bla bla-bla bla-bla.

Bla-bla bla-bla bla-bla bla-bla bla-bla bla-bla bla-bla bla-bla bla-bla
bla-bla bla-bla bla-bla.

Amennyiben egy adott bekezdés elé – a fontosságát hangsúlyozandó – extra függőleges térköz akarunk illeszteni, akkor azt ne egy üres sor generálásával tegyük, mert az szerencsétlen esetben megjelenik a lap alján illetve a tetején is. Ehelyett egy olyan rugalmas térköz tegyük be, amelynek lap alján illetve tetején nincs hatása. Ez elérhető a `\smallskip`, `\medskip` vagy `\bigskip` parancsok valamelyikével. Például



```
\documentclass{article}
\begin{document}
Bla-bla bla-bla bla-bla bla-bla bla-bla bla-bla bla-bla
bla-bla bla-bla bla-bla bla-bla bla-bla.

\bigskip
Bla-bla bla-bla bla-bla bla-bla bla-bla bla-bla bla-bla
bla-bla bla-bla bla-bla bla-bla bla-bla.
\end{document}
```

Bla-bla bla-bla bla-bla bla-bla bla-bla bla-bla bla-bla bla-bla bla-bla
bla-bla bla-bla bla-bla.

Bla-bla bla-bla bla-bla bla-bla bla-bla bla-bla bla-bla bla-bla bla-bla
bla-bla bla-bla bla-bla.

Ha minden bekezdés elé szeretnénk extra függőleges térköz, akkor az előző megoldás kényelmetlen és nem teszi lehetővé a stílusváltást. Ehelyett a két bekezdés közötti extra függőleges térköz beállító `\parskip` hosszúságparancsot állítsuk át pozitív értékre. A következő kódban például minden bekezdés egy `\bigskip` méretű extra függőleges térközzel kezdődik.



```
\documentclass{article}
\setlength{\parskip}{\bigskipamount}
\begin{document}
Bla-bla bla-bla bla-bla bla-bla bla-bla bla-bla bla-bla
bla-bla bla-bla bla-bla bla-bla bla-bla.

Bla-bla bla-bla bla-bla bla-bla bla-bla bla-bla bla-bla
bla-bla bla-bla bla-bla bla-bla bla-bla.
\end{document}
```

Bla-bla bla-bla bla-bla bla-bla bla-bla bla-bla bla-bla bla-bla
bla-bla bla-bla bla-bla.

Bla-bla bla-bla bla-bla bla-bla bla-bla bla-bla bla-bla bla-bla
bla-bla bla-bla bla-bla.

Források

<https://tex.stackexchange.com/questions/82664/when-to-use-par-and-when-nwline-or-blank-lines>



24. tévhit

A `\footnote` parancs elé a forrásfájlba sortörést vagy szóközt kell rakni. Például

```
\documentclass{article}
\begin{document}
Bla-bla bla-bla bla-bla
\footnote{Bla-bla bla-bla bla-bla.}
\end{document}
```

Bla-bla bla-bla bla-bla ¹

¹Bla-bla bla-bla bla-bla.

➡ A forrásfájlban a sortörés a végeredményben szóközként jelenik meg. Így a kimenet szempontjából mindegy, hogy a `\footnote` elé sortörést vagy szóközt teszünk a forrásba, a pdf-ben a lábjegyzet jelölő index előtt egy szóköz lesz látható. A szabály viszont éppen ezt tiltja, azaz a lábjegyzet indexének tapadnia kell arra a szóra, amire vonatkozik. Ha a lábjegyzet egy mondatra vagy tagmondatra vonatkozik, akkor az azt lezáró írásjelre kell tapadnia. Ennek érdekében a `\footnote` elé ne rakjon szóközt. Ha az olvashatóság kedvéért ragaszkodik a `\footnote` új sorba rakásához, akkor az előző sor végére illesszen be egy `%` karaktert, amivel ignorálja a sortörés szóközként történő kifejtését. Például



```
\documentclass{article}
\begin{document}
Bla-bla bla-bla bla-bla%
```

```
\footnote{Bla-bla bla-bla bla-bla.}
bla-bla bla-bla bla-bla.\footnote{Bla-bla bla-bla bla-bla.}
\end{document}
```

Bla-bla bla-bla bla-bla¹ bla-bla bla-bla bla-bla.²

¹Bla-bla bla-bla bla-bla.

²Bla-bla bla-bla bla-bla.



25. tévhit

Gondolatjelként a kvírtmínusz jelet (angolul *em dash*) kell használni. Például

Az öreg fánál állva --- mely már gyerekkoromban is igen
öreg volt --- gyakran elgondolkodtunk a jövőn.

Az öreg fánál állva — mely már gyerekkoromban is igen öreg volt — gyakran
elgondolkodtunk a jövőn.

➡ A magyar nyelvben tilos használni a kvírtmínuszt. Gondolatjelként mindig nagyköte-
jelet (angolul *en dash*) kell írni úgy, hogy előtte és utána is szóköz áll, kivéve, ha vessző
követi. Nagykötejelet -- vagy `\textendash` módon adhatunk meg. A nagykötejelet gon-
dolatjelként való használata az angolban is megengedett ugyanezekkel a szabályokkal.
Például



Az öreg fánál állva -- mely már gyerekkoromban is igen
öreg volt -- gyakran elgondolkodtunk a jövőn.

Az öreg fánál állva – mely már gyerekkoromban is igen öreg volt – gyakran
elgondolkodtunk a jövőn.



He asked \textendash well, more like demanded \textendash to be
given the promotion.

He asked – well, more like demanded – to be given the promotion.

Angolban viszont kvírtmínusz is használható gondolatjelként, de ekkor sem előtte, sem
utána nem állhat szóköz. Arra ügyelni kell, hogy egy dokumentumon belül nem szabad
használni gondolatjelként a kvírtmínuszt és a nagykötejelet is. Kvírtmínusz --- vagy
`\textemdash` módon adható meg. Például



He asked\textemdash well, more like demanded\textemdash to be
given the promotion.

He asked—well, more like demanded—to be given the promotion.

vagy



He asked---well, more like demanded---to be given the promotion.

He asked—well, more like demanded—to be given the promotion.



26. tévhit

A `\pagestyle{empty}` parancs a dokumentum minden oldalán üresre állítja a fej- és lábléceket.

➡ Itt egy példa arra, hogy ez az állítás nem igaz:

```
\documentclass{article}
\usepackage{lipsum}
\pagestyle{empty}
\title{Title}
\author{Author}
\begin{document}
\maketitle
\lipsum
\end{document}
```

Ebben az esetben az első oldal láblécében látható az oldalszám, de a második oldalon már nem. Ennek az a magyarázata, hogy a `\maketitle` aktiválja a `\thispagestyle{plain}` parancsot is, amely az aktuális oldal stílusát `plain`-re állítja függetlenül attól, hogy addig mi volt a beállítás (jelen esetben `empty`). A `plain` stílus pedig alapértelmezésben a láblécbe középre illeszti az oldalszámot.

Nem csak a `\maketitle` aktiválja a `\thispagestyle{plain}` parancsot, hanem a `report` illetve `book` dokumentumosztályok `\chapter` és `\chapter*` parancsai is.

Ha ezt a szándékos viselkedést meg szeretné akadályozni, akkor egy lehetőség a `plain` stílus üresre állítása a

```
\makeatletter\let\ps@plain\empty\makeatother
```

kóddal. Másik lehetőség a

```
\pagenumbering{gobble}
```

parancs használata, amely az oldalszámozást letiltja. Ez csak akkor hatásos, ha a `plain` stílus úgy van beállítva, hogy csak oldalszámozást generál. Harmadik lehetőség, hogy a `\maketitle`, `\chapter` és `\chapter*` parancsok után beírjuk a

```
\thispagestyle{empty}
```

parancsot. Ez utóbbi lokális megoldás, ami nem túl kényelmes és nem teszi lehetővé a stílusváltást. Tehát például a következő esetben már minden oldal fej- és lábléce üres lesz:



```
\documentclass{article}
\usepackage{lipsum}
\pagestyle{empty}
\pagenumbering{gobble}
\title{Title}
\author{Author}
\begin{document}
\maketitle
\lipsum
\end{document}
```



27. tévhit

Ha a forráskód UTF-8 kódolású, akkor be kell tölteni az `inputenc` csomagot `utf8` opcióval, azaz a preambulumban be kell írni a `\usepackage[utf8]{inputenc}` parancsot.

➡ Egy \LaTeX forráskód háromféle fordítóval konvertálható pdf-be: `pdflatex`, `lualatex`, `xelatex`. Az `inputenc` csomag a `pdflatex` fordítóhoz készült, a `lualatex` és `xelatex` fordítókkal nem kompatibilis. Az utóbbi két fordító csak UTF-8 kódolású forráskód esetén működik. A `pdflatex` 2018 előtti verziói alapesetben a forráskód ASCII kódolását feltételezték, UTF-8 esetén be kellett írni a

```
\usepackage[utf8]{inputenc}
```

parancsot a preambulumba. 2018-tól azonban már UTF-8 az alapértelmezett kódolás, azaz innentől nincs szükség az előbbi kódra. Ma leginkább már csak a régi, nem UTF-8 kódolású forrásfájlok lefordításához használjuk az `inputenc` csomagot. Például kelet-európai ISO 8859-2 (Latin-2) kódolás esetén a

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

parancsra lesz szükségünk, továbbá ekkor csak a `pdflatex` használata lehetséges.

Összegezve tehát, *UTF-8 kódolású forrásfájl esetén egyetlen fordítónál sem kell külön parancsot kiadni a bemeneti kódolás típusának megadásához.*



28. tévhit

Latin ékezetes betű csak akkor jelenik meg a végeredményben, ha betöltjük a `fontenc` csomagot `T1` opcióval, azaz a preambulumban ehhez be kell írni a `\usepackage[T1]{fontenc}` parancsot.

➡ Ez az állítás nem igaz. Próbálja ki például a következő kódot:

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

```
Jóízű félárú sütötök.  
\end{document}
```

Jóízű félárú sütötök.

A `pdflatex`, `lualatex`, `xelatex` fordítók mindegyikével megjelennek a pdf-ben az ékezetes betűk. Viszont `pdflatex` használata esetén azt fogjuk tapasztalni, hogy a pdf-ből kimásolva a szöveget és azt beillesztve egy editorba, a következő eredményt kapjuk:

J´o´iz˘u f´el´ar´u s˘ut˘ot˘ok.

Ennek a következő a magyarázata. A `pdflatex` esetén például az ó betű először konvertálódik `\'o` parancs alakba, majd az alapértelmezett OT1 belső kódolás a pdf-ben megjelenítendő fontkészletből kiválasztja a ´ és o karaktereket, végül egymásra helyezi azokat. Így kapjuk meg az ó betűt. Ez a fajta metódus akkor előnyös, ha a pdf-ben olyan fontkészletet használunk, amely nem tartalmaz ékezetes betűket. Viszont így az ékezetes betűk két karakterként lesznek értelmezve, ami a szöveg pdf-ből történő kimásolásánál jól látszik. De ez egy másik, ennél lényegesebb problémát is okoz. Nevezetesen a szóelválasztási algoritmust nem tudja kezelni az ékezetes betűket tartalmazó szótagokra, így a sorvégi elválasztásokat nem lehet optimálisan megoldani.

Ennek a problémának a kezeléséhez olyan fontkészletet kell használni a pdf-ben, amely tartalmaz ékezetes betűket, és át kell térni olyan belső kódoláshoz, amely például az ó betűhöz a neki megfelelő karaktert választja ki a fontkészletből. Ilyen belső kódolás a T1. Ennek megadásához van szükség a

```
\usepackage[T1]{fontenc}
```

kódra. Át lehet térni az alapértelmezettől eltérő fontkészletre is, de arra ügyelni kell, hogy az kompatibilis legyen a T1 belső kódolással.

Amennyiben `lualatex` vagy `xelatex` fordítót használ, akkor ne töltsse be a `fontenc` csomagot, mert a megfelelő belső kódolás kiválasztása automatikusan történik.



29. tévhit

A T1 belső kódoláshoz a `t1enc` csomagot kell használni.

➡ A \LaTeX legelső hivatalos verziójában még csak a `t1enc` csomag használatával volt lehetőség a T1 belső kódolásra való váltásra. Azóta elkészült a `fontenc` csomag, amely több lehetőséget is biztosít. Így ma már a

```
\usepackage[T1]{fontenc}
```

használata javasolt, a `t1enc` elavultnak számít, csak kompatibilitási okokból tartották meg.

Források

<https://davidcarlisle.github.io/uk-tex-faq/FAQ-t1enc.html>



30. tévhit

Magyar nyelvű dokumentum készítésénél a megfelelő tipográfia eléréséhez elég betölteni a `babel` csomagot magyar opcióval, azaz a preambulumban csak be kell írni a `\usepackage[magyar]{babel}` parancsot. Például

```
\documentclass{article}
\usepackage[magyar]{babel}
\begin{document}
Szöveg
\end{document}
```

➡ Magyar nyelvű dokumentumokban nagy eséllyel írunk latin ékezetes betűket, így a 28. tévhitnél leírtak alapján, ha `pdflatex` fordítót használunk, akkor át kell térni a T1 belső kódolásra. Ez a

```
\usepackage[magyar]{babel}
```

sor beírásával nem történik meg automatikusan, így erről külön kell gondoskodni.

De nem csak ez a gond. A `babel` csomag magyar opciója úgy működik, hogy betölti a `magyar.ldf` fájlt, amely a magyar tipográfia megvalósításáért felelős. Egy ilyen `ldf` (language definition) fájl mérete az 1980-as években az akkori kapacitási lehetőségek miatt jelentősen korlátozva voltak. Így a magyar tipográfiai elemek jelentős része nem került kódolásra a `magyar.ldf` első verzióiban. A későbbiekben, amikor már nem voltak méretkorlátozások az `ldf` fájlokra, akkor újraírták a `magyar.ldf` fájlt, beépítve a hiányzó tipográfiai elemek kódolását is. Azonban kompatibilitási okok miatt, a `babel` csomag magyar opciója alapesetben csak azokat a kódrészleteket fogja betölteni, melyek az eredeti `magyar.ldf`-ben is benne voltak. Hogy ez ne így legyen, és a magyar tipográfiának a lehető legjobban megfelelő végeredményt kapjuk, ahhoz ki kell adni a `babel` betöltése előtt a

```
\PassOptionsToPackage{defaults=hu-min}{magyar.ldf}
```

parancsot. Tehát a preambulumban a dokumentumosztály betöltése után a magyar tipográfiát a következő kóddal lehet betölteni:



```
\usepackage[T1]{fontenc} % Csak pdflatex fordító esetén kell.
\PassOptionsToPackage{defaults=hu-min}{magyar.ldf}
\usepackage[magyar]{babel}
```

Megemlítjük, hogy az előbbi beállítás esetén a lábjegyzetek felett nem lesz vonal, ami a magyar tipográfiában elfogadott. Ha mégis ragaszkodik a hagyományosabbnak mondható vonalhoz, akkor a magyar szabályok értelmében a hossza a szövegtükör szélességének negyede, nem pedig az angoloknál elfogadott 2 inch. Ennek eléréséhez a `babel` betöltése után még írja be a

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

parancsot.

A `lualatex` és `xelatex` fordítóknál nem csak a `babel`, hanem a `polyglossia` csomag is használható. Magyar nyelv esetén például ezt kell írni a preambulumba:

```
\usepackage{polyglossia}  
\setmainlanguage{hungarian}
```

Jelenleg ennek a tudása még a `magyar.1df` első verzióinál is csekélyebb. A fejlesztésére remélhetőleg a jövőben lesz olyan vállalkozó, aki jól ismeri a magyar tipográfiai szabályokat.