

Memento 1. kurzus: LaTeX bevezető

Segédanyag: Sykora Henrik

2018. szeptember 7.

Tartalomjegyzék

1. Bevezető	1
2. Alapok	1
2.1. Dokumentum szerkezete	1
2.2. Dokumentum elkezdése	2
3. Csomagok, parancsok és környezetek	2
3.1. Csomagok használata	2
3.2. Parancsok	3
3.2.1. Parancsok argumentum nélkül	3
3.2.2. Parancsok argumentummal	3
3.3. Környezetek	3
3.3.1. Egyenletek	3
3.3.2. Vektorok, Mátrixok	4
3.3.3. Felsorolások	5
3.3.4. Ábrák	5
3.4. Táblázatok	6
3.5. Hivatkozások	6
3.6. Saját parancsok létrehozása	6
4. Oldalkép	7
4.1. Margók kézi megadása	7
5. Hasznos linkek	7

1. Bevezető

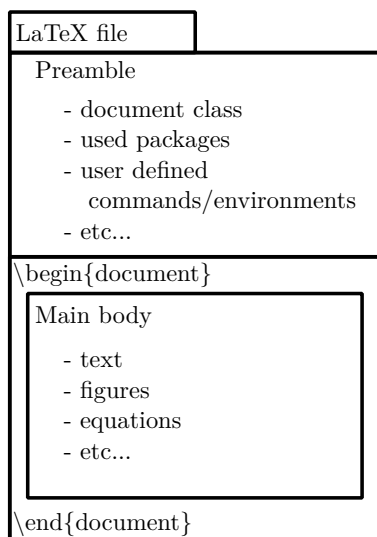
Amit most a képernyőtökön láttok (vagy a kezetekben fogtok) a MeMento LaTeX kurzusának munkalapja. Ehhez tartozik egy szöveges fájl (*LaTeX_Example_text.txt*) is, ami egy nyers szöveget tartalmazza, és ennek a munkalap segítségével fogunk megformázni.

A LaTeX használata során egy egyszerű szöveges file-t (.tex kiterjesztéssel) szerkesztünk, amiből a LaTeX fordítója gyakorlatilag létrehoz egy *PDF* dokumentumot. Ennek az a jelentősége, hogy a LaTeX fileokat bármelyik LaTeX disztribúcióval is fordítjuk le, akkor ugyanolyan formázású dokumentumokat kapunk. Ezzel gyakorlatilag elkerülhető, hogy egy dokumentumunk szétessen ha egy új verzióval nyitjuk meg, mint ahogy ezt tapasztalhattunk néhány régebbi Word dokumentumnál.

2. Alapok

2.1. Dokumentum szerkezete

Egy dokumentum két fő részből áll: *preamble*-ből és a *main body*-ből (lásd 1. ábra). A *preamble* tartalmazza az összes fontos formázási beállítást amelyek a dokumentum oldalképét, stílusát határozza



1. ábra. \LaTeX file szerkezete

meg. Ezek természetesen felülírhatóak manuálisan is, de alapvetően itt adjuk meg, hogy milyen dokumentum típust írunk (pl.: report/book/article stb...), mekkora margókat szeretnénk használni, milyen *package*-eket szeretnénk használni. Itt adhatóak meg egyénileg definiált parancsok (*command*) és környezetek (*environment*) is (részleteket lásd a 3.2 alfejezetben).

A *main body*-ban van az összes olyan tartalom, ami megjelenítésre is kerül. Ilyen a szöveg, a képek, egyenletek, táblázatok stb. Az ide beírt szöveget különböző parancsokkal formázhatjuk amelyek a LaTeXben már előre definiáltak, vagy mi magunk hozunk létre a *preamble*-ben (lásd a 3.6 fejezetben).

2.2. Dokumentum elkezdése

Hogy nekikezdjünk az első LaTeX dokumentumunknak, írjuk be a szöveges file első soraként a következőt:

```
 $\documentclass[11pt, a4paper]\{article\}.$ 
```

Ezzel definiáltuk, hogy a dokumentumunk alapértelmezett betűmérete 11pt, A4-es méretű papírra készül és az "article" osztályba (*class*) tartozik (az osztályokról a ezen az oldalon olvashattok részletesebben).

A következő fontos teendő, hogy a megjelenítendő részt (a *main body*-t) körülvegyük a *document* környezettel az alábbi módon (környezetekről részletesebben a 3.3 fejezetben olvashatsz):

```
 $\begin\{document\}$   

  ...main body...  

 $\end\{document\}$ 
```

Ha most lefordítjátok a dokumentumot, akkor láthatjátok, hogy sem az oldalkép, sem a szöveg nem olyan formában jelenik meg, mint ahogy azt a feladatlapon látjátok. A következő fejezetek foglalkoznak azzal, hogy miként lehet a kívánt módon formázni egy dokumentumot LaTeX-ban.

3. Csomagok, parancsok és környezetek

3.1. Csomagok használata

A LaTeX különböző csomagok segítségével dolgozik amiket ha használni akarunk, akkor azokat a *preamble*-ben hozzá kell adni a $\usepackage\{*packagename*\}$ parancssal. Ha megfigyeljük az előbb fordított dokumentumunkat, akkor láthatjuk, hogy az ékezetes karakterek (á,é,í,ó,ö,ú,ü,ű) nem jelennek meg. Ez azért van, mert a LaTeX által alapértelmezetten használt karakterkódolás nem tartalmazza ezeket, és emiatt be kell hívni az utf8-as kódolást a $\usepackage[utf8]\{inputenc\}$ parancs *preamble*-be írásával. Az esetek nagy részében sok csomagra is szükségünk van, ezeket akár egy parancsban is behívhatjuk: $\usepackage\{*packagename1*, *packagename2*, \dots\}$.

3.2. Parancsok

LaTeX-ben minden (!) parancs, vagy környezet ”\”-el kezdődik. Ha ezt a karaktert akarjuk használni, akkor azt egy külön parancssal kell meghívni (”\textbackslash”).

3.2.1. Parancsok argumentum nélkül

Ebben a részben lesz szó a parancsok használatáról. A parancsok általában lehetnek egyszerű utasítások, mint pl. a `\LaTeX` parancs, ami a \LaTeX szimbólumát jeleníti meg, vagy a `\textbackslash`, amely pedig a ”\” karaktert jelenti. Fontos, hogy egy ilyen parancs van esetén, hogy ha a szót szóköz követi, akkor szükséges egy ”\” a parancs végén is: ”...`\LaTeX`\ szimbólumát...”. Új bekezdést a `\par`-al, új sort pedig a `\\`-el lehet indítani.

3.2.2. Parancsok argumentummal

Sok parancsnak argumentumra van szüksége. Ilyen típusú parancsokkal adhatjuk meg, hogy ha például azt szeretnénk, hogy ha a végleges dokumentumunkban *dőlt* vagy **félkövér** legyen, de parancsokkal tudjuk például megadni a fejezetek címét is. Próbáljuk is ki, ebben a szövegrészben szereplő ”dőlt” és ”félkövér” szavakon a `\textit{...}` és `\textbf{...}` parancsokat.

A parancsok fejezet (*section*), alfejezet (*subsection*) vagy az al-alfejezet (*subsubsection*) is a hozzájuk tartozó parancsokkal adhatóak meg. Keressük meg a nyers szövegben a fejezetek címét, és használjuk rajtuk a fenti angol elnevezéseknek megfelelő parancsokat (`\section{...}`, stb...). Ennek a megadási módnak a nagy előnye, hogy a `\tableofcontents` parancsot a *main body* elejére vagy végére írva olyan tartalomjegyzéket hoz létre a LaTeX, amelyben kattintva a kiválasztott részhez ugrik (`\usepackage{hyperref}` csomag behívása még szükséges). Megfigyelhetjük, hogy a hivatkozások színes téglalappal vannak körbevéve: ezek jelzik, hogy az adott szövegrész kattintható, de nyomtatásban nem jelennek meg.

3.3. Környezetek

Az egyik legfontosabb LaTeX eszköz a környezet. Az egyenletek, táblázatok, felsorolások, képek mind környezetként hívhatóak be egy fájlba, amelyek szépen megformázva kerülnek megjelenítésre. Ezekre nézzünk néhány példát.

3.3.1. Egyenletek

Az egyenletek megjelenítése és bevitele a LaTeX egyik nagy erőssége. Két fő megadási lehetőség van. Az egyik az, ha az egyenleteket két ”\$” közé helyezzük, és akkor a szövegbe illeszti be a formulát, a másik pedig, hogy meghívjuk az egyenlet környezetet, amely középre rendezve, megszámozva jeleníti meg a bevitt formulát. Ezt a következőképpen adhatjuk meg:

```
\begin{equation}
...equation...
\end{equation}
```

Egy egyenlet pedig az alábbi szerint néz ki:

$$\ddot{x} + 2D\alpha\dot{x} + \alpha^2x = 0. \quad (1)$$

Ennek a kódja az alábbi módon néz ki:

```
\begin{equation}
\ddot{x}+2D\alpha \dot{x}+\alpha^2 x = 0.
\end{equation}
```

Itt a két-, és egy pontokat az x fölött a `\ddot{x}` és `\dot{x}` parancsok csinálják. Ezek a parancsok kizárólag matematikai környezetben működnek, ha simán szövegre szeretnénk alkalmazni, akkor a program hibaüzenetet dob. Ilyen parancsok a görög betűk meghívása (`\alpha`), és minden olyan parancs, ami matematikai műveletet ábrázol. Ha szövegben szeretnénk matematikai műveletet használni,

vagy pl egy α -t írni, akkor azt egy lokális matematikai környezetbe kell tenni a $\backslash(\alpha)$ vagy a $\$alpha$ módon. Néhány példát ezekre a matematikai operátorokra az 1. táblázat tartalmaz.

Parancs	Megjelenített karakterek
<code>\infty</code>	∞
<code>x^{n+1}</code>	x^{n+1}
<code>x_{n+1}</code>	x_{n+1}
<code>\frac{a}{b}</code>	$\frac{a}{b}$
<code>\sqrt{a}</code>	\sqrt{a}
<code>b\;</code> <code>\mathbf{a}</code>	b a
<code>\int_a^b x dx</code>	$\int_a^b x dx$
<code>\neq, <, >, \leq, \geq, \equiv, \sim</code>	$\neq, <, >, \leq, \geq, \equiv, \sim$
<code>\cdot, \dots, \vdots, \cdots, \ddots</code>	$\cdot, \dots, \vdots, \cdots, \ddots$

1. táblázat. Néhány "equation" környezetben használható LaTeX parancs

Továbbá ha zárójeleket szeretnénk használni, akkor érdemes ezt a `\left(...\right)` parancsokkal tenni, mert ekkor lehet a zárójeleket halmozni, és ha a zárójelben lévő dolgoknak nagyobb a szokásosnál, akkor a LaTeX hozzáskálazza a zárójeleket is:

$$2\left(\left(\frac{a}{b} + c\right) - d\right) + d\left((a + b)^2 - 1\right) \quad \text{vs.} \quad 2\left(\left(\frac{a}{b} + c\right) - d\right) + d\left((a + b)^2 - 1\right) \quad (2)$$

Több (pl. 3 db) egyenlet esetén az egyenleteket lehet egymás alá, rendezve is megjeleníteni az `align` környezetben sortöréssel (`\`) és a `&` karakterrel:

```
\begin{align}
...lhs1... &= ...rhs1... \\
...lhs2... &= ...rhs2... \\
...lhs3... &= ...rhs3...
\end{align}
```

Az egyenlőségjelek előtt szereplő `&` karakter a rendezés viszonyítási pontját jelöli. Esztétikai szempontok miatt ezt érdemes az egyenlőség jel elé vagy után rakni, mert akkor az egyenlőségjelek egymás fölött lesznek.

3.3.2. Vektorok, Mátrixok

Ez egy rendkívül speciális környezet, mert kizárólag matematikai környezetekben hívható meg, továbbá szükséges hozzá az `amsmath` package is (`preamble`-be: `\usepackage{amsmath}`). Egy mátrixra egy példa:

$$\mathbf{M} = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} \quad (3)$$

Ezt az alábbi kóddal lehet létrehozni:

```
\begin{equation}
\mathbf{M}=
\begin{pmatrix}
a & b & c \\
d & e & f \\
g & h & i
\end{pmatrix}
\end{equation}
```

Megfigyelhetjük, hogy amikor a mátrix környezetet elkezdjük, akkor a `pmatrix` környezetet hívjuk meg. A `matrix` előtt lévő "p" a "parenthesis" rövidítése, és ennek hatására a LaTeX a mátrixot

zárójelbe rakja, ezzel megspórolva nekünk azt, hogy a mátrix elé és után a `\left(\right)` parancsokat írjunk. Léteznek sima `matrix` és `bmatrix` környezetek is, amelyek rendre a zárójel nélküli, és szögletes zárójelekkel ("bracket") körbevett mátrixokat jelentik.

3.3.3. Felsorolások

A LaTeX-ben a felsorolásokat is környezetként kell létrehozni. Többféle felsorolástípus létezik, ezek közül a legegyszerűbbel fogunk foglalkozni, az `itemize` környezettel (a többről itt olvashatsz). Ennek a kódja a következőképpen néz ki:

```
\begin{itemize}
  \item 1. elem
  \item 2. elem
\begin{itemize}
  \item második szint 1. elem
  \item második szint 2. elem
\end{itemize}
\end{itemize}
```

Ennek a kimenete:

- 1. elem
- 2. elem
 - második szint 1. elem
 - második szint 2. elem

Látható, hogy az `itemize` környezetben a `\item` paranccsal indítunk új elemet, továbbá a különböző szinteket az `itemize` környezetek egymásba ágyazásával hozzuk létre.

3.3.4. Ábrák

Ahhoz, hogy a dokumentumokba ábrákat tudjunk behelyezni, a `graphicx` *package*-re van szükség. Ekkor a következő sorok a `.tex` fájl mellett elhelyezett képet beillesztik a szövegbe:

```
\begin{figure}[htb!]
  \centering
  \includegraphics[width=3cm]{teszt.png}
  \caption{Próba felirat}
  \label{fig:teszt}
\end{figure}
```

Ennek az eredménye az alább látható 2. ábra:



2. ábra. Próba felirat

Az ábra beillesztéséhez használt kódrészletek a `figure` környezetben a következőket jelentik:

htb "here/top/bottom": a `figure` körülbelüli helyét adja meg

- ! Felülírja a LaTeX elhelyezési alapbeállítását, az általunk kijelölt helyet erőlteti
- `\centering` Középre igazítja az ábrát
- `\includegraphics` Behelyezi az ábrát a dokumentumba
- `[width=5cm]` Ábra méretet állítja be (`width/height`). További lehetőségek itt találhatóak.
- `teszt.png` Beillesztendő fájl neve
- `\caption` Ábrafelirat
- `\label` Kattintható hivatkozás létrehozásához címke, ez a `\ref` paranccsal hívható meg.

Megfigyelhető, hogy az ábrafelirat a "Figure 1" jelzőt kapjuk. Hogy a magyar címkézést használja a LaTeX, ahhoz a `babel` csomagot kell behívni "magyar" beállítással (`\usepackage[magyar]{babel}`). Ettől kezdve a "x. ábra." címke kerül a képek alá.

3.4. Táblázatok

A táblázatokat a mátrixokhoz hasonlóan kell készíteni, viszont itt van még néhány további formázási lehetőség: szöveg igazításának helye, szegélyek, stb...

```
\begin{table}[h!]
\centering
\begin{tabular}{|l|c|r|}
\hline
aa & aa & aa\\
\hline \hline
aaa & aaa & aaa\\
aaaa & aaaa & aaaa\\
\hline
\end{tabular}
\label{tab:teszt}
\caption{Teszt táblázat}
\end{table}
```

Ennek a kimenete:

aa	aa	aa
aaa	aaa	aaa
aaaa	aaaa	aaaa

2. táblázat. Teszt táblázat

3.5. Hivatkozások

A hivatkozások gyakorlatilag parancsok, amelyeknek a szövegben általunk létrehozott címkéket (`\label`-öket) tudunk argumentumként megadni. Ezek alapján két fő lépése van a hivatkozásnak: létrehozunk egy objektumot amihez hozzáadunk egy címkét (pl.: (`\label{fig:teszt}`)), majd azt behivatkozzuk a `\ref{*label*}` paranccsal. Hivatkozhatóak így képek, mint 1. vagy 2. ábra, vagy akár táblázatok (1. táblázat). Az egyenletekhez érdemes az `\eqref{*label*}` parancsot használni, mert akkor az egyenlet számát a hivatkozás automatikusan zárójelbe teszi, pl.: (1) egyenlet.

3.6. Saját parancsok létrehozása

A LaTeX-ben a *preamble*-ben lehetőség van saját parancsot is létrehozni. Ez akkor praktikus, ha valamit sokszor használunk, és nem akarjuk mindig a hosszú kódot beírni, vagy akkor ha valami olyan jelölést használunk, amit lehet, hogy ki akarunk a későbbiekben cserélni: így csak azt kell megváltoztatni, hogy a parancs mit csinál. Például ebben a dokumentumban a kód formátum sokszor került felhasználásra:

```
\newcommand{\asCode}[1]{\small\fontfamily{qcr}\selectfont #1}}
```

Itt a saját készítésű `asCode` parancs annyit csinál, hogy a szöveget Courier betűtípussal írja a nekünk megfelelő méretben. Az új parancsot a `newcommand` paranccsal hozhatjuk létre, amelynek első argumentumként megadhatjuk, az új parancs nevét (`asCode`), beállításként a várt argumentumok számát (ami most 1 db), majd második argumentumként pedig az elvégzendő utasításokat. Az utasításban a #1 az 1. argumentum beírás helyét jelöli.

Megjegyzés: a kódként történő formázást a `\verb|...|` beépített parancs vagy a `\begin{verbatim} ... \end{verbatim}` környezet segítségével is megtehetjük.

Továbbá lehet argumentum nélküli parancsot is létrehozni: pl.:

```
\newcommand{\prea}{\textit{preamble}}
```

4. Oldalkép

Az oldalkép formázására nagyon sok lehetőség van. Megadható a `\documentclass` argumentumaiként, hogy mekkora legyen az alapértelmezett betűméret, a papírméret, a dokumentumstílus, vagy hogy egy- vagy kétoldalas dokumentumot szeretnénk (pl. a margókat aszerint szedi).

4.1. Margók kézi megadása

A margók kézzel történő megadására is van lehetőség. Ehhez szükséges a `geometry` csomag meghívása, amely a *preamble*-ben aktiválja a `\geometry` parancsot. Pl. Ennek a dokumentumnak a következők a margóbeállításai:

```
\geometry{a4paper,  
  left=20mm, top =20mm,  
  total={170mm,257mm}}
```

Tehát a bal oldali és felső margó 20 mm, és az oldal amire a szöveg kerül az pedig 170 mm széles és 257 mm magas. A további oldalbeállítások itt megtekinthetők.

5. Hasznos linkek

- LaTeX help: <https://www.sharelatex.com/learn>
- LaTeX help 2: <https://en.wikibooks.org/wiki/LaTeX>
- LaTeX template-ek: <https://www.latextemplates.com/>
- A legfontosabb: www.google.com