

L^AT_EX 101

Kód szerkezete, szintaktika

Vadon Viktória

2022/23/I. félév

Általános tanácsok

- fordítsunk gyakran
 - minél kevesebb változott, annál könnyebb hibát keresni benne!
- hagyjuk, hogy segítsen a szerkesztő
 - ha elkezdünk parancsot gépelni, sokszor felajánlja, hogy milyen parancsra lehet kiegészíteni – a T_EXStudio többet ismer, mint a T_EXMaker
 - a T_EXStudio a zárójeleket is megpróbálja kiegészíteni (de néha nincs igaza)

Kód/dokumentum szerkezete

HTML

```

<!DOCTYPE ...>
<HTML>
<HEAD>
fej: stílus, szkriptek, etc.
</HEAD>
<BODY>
törzs: tényleges tartalom
</BODY>
</HTML>

```

L^AT_EX

```

\documentclass[...]{...}
preambulum (preamble):
csomagok betöltése,
stílus, saját makrók
\begin{document}
dokumentumtörzs:
tényleges tartalom
\end{document}

```

Minimális dokumentum

érvényes \LaTeX dokumentum (hogy egyáltalán leforduljon):

- a *legelső sorban* a `\documentclass{dokumentumosztály}` parancsot egy érvényes dokumentumosztállyal
- a dokumentumtörzs elejét és végét jelző `\begin{document}` és `\end{document}` parancsokat
- köztük legalább egy *megjelenthető* karaktert (üres dokumentum sem fog fordulni)

Documentclass

- pl. `\documentclass{article}`
- `documentclass` = dokumentumosztály
- tekintsünk rá úgy, mint a HTML-ben a doctype deklarációra
- \LaTeX fájl *kötelező* első sora (ez különbözteti meg egy alacsonyabb szintű plain \TeX fájljától)
- megadja a dokumentum típusát
pl. `article` = cikk, `report` = riport, `book` = könyv, `beamer` = diasor
- ezzel együtt importál oldalbeállításokat, egyéb formázást;
némelyik osztály (pl. `beamer`) extra makrókat is (erről később)

Mi jön a documentclass után? |

...de még a `\begin{document}` előtt?

csomagok betöltése és beállítása

- `package = csomag`
- gondoljunk rá úgy, mint egy programkönyvtárra
- adott célra szánt makrógyűjtemény; a \LaTeX funkcionalitását bővíti. a legtöbb csomagot felhasználók teszik közzé
 - szinte minden célra létezik csomag
 - kompatibilitás nem garantált
- csomagok (és dokumentumosztályok) böngészése:
CTAN (Comprehensive \TeX Archive Network)
- betöltés: `\usepackage{csomagnév}`
- csomagokhoz tartozhatnak opciók/beállítások:
`\usepackage[opciók]{csomagnév}`

Mi jön a documentclass után? II

...de még a `\begin{document}` előtt?

paraméterek és egyéb beállítások, formázás testreszabása

- pl. oldalszámozás, tartalomjegyzék mélysége
- ilyen jellegű parancsok a dokumentumtörzsben is előfordulhatnak
- később látunk majd példákat

saját makrók/parancsok definíciói

- pl. érdemes gyakran használt jelöléseket vagy kifejezéseket parancsba venni, akkor egy helyen módosítható, vagy éppen kevesebbet kell gépelni
- később tanuljuk

Parancsok I

vezérlő karakterek

% \ { } ~ \$ ^ _ & #

- ~: törhetetlen szóköz
- %: komment kezdete, a sor többi része (a sortörést is beleértve!) „nem létezik”
- \ (backslash, visszaperjel, repjel): parancsok első karaktere
- a többi hatásáról később
- egyelőre a legfontosabb: bizonyos karaktereket begépelve nem megjelennek, hanem parancsként működnek. furcsa hiba esetén ellenőrizzük, nem-e gépeltünk be vezérlő karaktert.

Parancsok II

\ + speciális karakter

- egyes vezérlő karakterek kiíratása: $\% = \%$, $\{ = \{$, $\} = \}$ $\$ = \$$, $_ = _$, $\& = \&$, $\# = \#$
- ún. repülő ékezetek, pl. $\^ > \^a = \hat{a}$, $\~a = \tilde{a}$, $\"a = \ddot{a}$, $\'a = \acute{a}$, $\'a = \grave{a}$, stb.
- egyéb, pl. $\backslash =$ sortörés, $\ =$ szóköz = (adott szélességű) szóköz

Parancsok III

\backslash + *betűsorozat*

- pl. $\backslash\text{textbackslash}$ = \backslash , $\backslash\text{LaTeX}$ = \LaTeX
- *csak* az angol ábécé kis-és nagybetűi!
- érzékeny a kis- és nagybetűkre
- vigyázzunk rá, hogy egy parancs ne folyjon össze az őt követő szóval! a \LaTeX úgy nem ismeri fel, meddig tart a parancs
 - de: a parancs „megeszi” az összes őt követő whitespace (szóköz, tab, enter, stb.) karaktert, ld. $\backslash\text{LaTeX nyelv} = \text{\LaTeXnyelv}$
megoldás: \backslash + szóköz, pl. $\backslash\text{LaTeX}\ \text{nyelv} = \text{\LaTeX nyelv}$

Parancs argumentumai I

- argumentum nélküli parancsok
 - láttuk pl. `\\` = sortörés, `\textbackslash` = `\`
 - léteznek pl. felülírásig érvényes formázási parancsok: `\Large` nagyobb betűméret, `\itshape` dőlt betűk, `\centering` középre zárás, stb.
- parancsok 1 argumentummal: jól meghatározott hatáskör
 - pl. `\textit{dőlt betűk}` már nem = *dőlt betűk* már nem
 - argumentumot kapcsos zárójel `{}` határolja
 - zárójelek híján csak a következő karakter, pl. `\textit abc` = *abc*

Parancs argumentumai II

- lehet *több* argumentum, pl.
`\textcolor{blue}{kék szöveg} = kék szöveg`
 - az argumentumok *sorrendje kötött*
példánkban szín neve először, formázandó szöveg később
 - *külön-külön* kerülnek kapcsos zárójelekbe
 - *sokszor* nem tűri meg a zárójelek között a szóközőket! ha furcsa hibát találunk, ellenőrizzük.
- egyes parancsoknak lehet opcionális argumentuma
 - `< \item`, felsorolás elemét jelzi
 - `< \item[*]`: szimbólum megváltoztatható
 - opcionális argumentumot `[]` szögletes zárójel jelzi

Parancs argumentumai III

- lehet opcionális és kötelező argumentum is
 - pl. `\section{Szakasz neve}` = új szakasz (dokumentumegység) kezdete, 1 kötelező argumentum
 - `\section[rövid név]{Szakasz hosszú neve}` > a rövid nevet a tartalomjegyzékben és fejlécben használja
 - opcionális argumentum először
- ha az argumentumaival túl hosszú a parancs...
 - *általában* nem szereti a szóközöket és sortöréseket argumentumok között
 - de kommenttel be lehet csapni!
emlékeztető: % után a sor többi része „nem létezik”, beleértve a sortörést is

```
\parancs[opcionális argumentum]%  
{kötelező argumentum 1}%  
{kötelező argumentum 2}
```

Környezetek I

- környezet (environment): speciális vagy elkülönített szakasz a dokumentumban

- például

```
\begin{center}  
középre zárt szöveg  
\end{center}
```

középre zárt szöveg

- ez miért jobb, mintha egyetlen parancs lenne?
hogy ne egy parancs argumentumaként kelljen akár több paragrafust kezelni! (könnyű szem előtt téveszteni a zárójeleket)
- kezdetét és végét `\begin{környezetnév}` és `\end{környezetnév}` jelzi

Környezetek II

- milyen környezetek léteznek? például:
 - fentebb látott igazítás
 - (előre definiált) speciális formázások, mint pl. forráskód, tételek és definíciók, felsorolás és számozás
 - úsztatott objektumok, pl. ábrák, táblázatok
 - egyéb célból elkülönített szakasz, pl. kézzel formázott címoldal
- mire jó még?
„felülírásig érvényes” formázási parancsok csak a környezet végéig érvényesek!
 - erre a célra elég lehet egy **csoport**: { nyitja és } zárja; gondolhatunk rá úgy, mint egy névtelen, formázatlan környezet.

Környezetek argumentumai I

- hasonlóképp mint parancsoknál, [] opcionális, {} kötelező argumentum
- a `\begin{környezetnév}` után kerül *minden* argumentum, az opcionális is
(avagy, a *környezetnevet* tartalmazó zárójel *nem* számít argumentumnak)
- például theorem = tétel, 1 opcionális argumentum

Theorem (Tétel neve)

Tétel szövege.

```
\begin{theorem}[Tétel neve]  
Tétel szövege.  
\end{theorem}
```

Környezetek argumentumai II

- például: blokk (csak beamer diasorban), 1 kötelező argumentum

cím

törzs

```
\begin{block}{cím}
törzs
\end{block}
```

LaTeX források

Magyarul

- röviden: Gyimesi Eszer könyve
- hosszabban: LaTeX 78 percben
- hosszan: Tómacs Tibor könyve

Angolul

- LaTeX wikibooks
- parancssorból `texdoc latex2e` (quick reference guide),
`texdoc source2e` (commentated source code)
- csomagok dokumentációi
- kérdések: Google > StackOverflow
- online reference manuals
- stb.