

# L<sup>A</sup>T<sub>E</sub>X mint dokumentumszerkesztő

Vadon Viktória

2022/23/I. félév

- 1 Beállítások
  - Documentclass
  - Nyelvek kezelése

# Documentclass

## Emlékeztető

LaTeX dokumentum kötelező első sora

```
\documentclass[opciók]{dokumentumosztály}
```

- `article` = (újság)cikk
  - de egy általánosan használt, alap dokumentumstílus
- `report` = riport/jelentés
- `book` = könyv
- `beamer` = diavetítés
- egyébek, pl. `letter` = levél, `memoir` = a könyvnek egy flexibilisebb változata
- csomagok letöltésével még több! pl. KOMA-script, sadott tudományos folyóiratok cikkstílusai, egyetemek szakdolgozatstílusai, stb.

# Documentclass opciók I

- betűméret
  - pl. 10pt (alapértelmezett), 11pt, 12pt
- papírméret
  - pl. a4paper, a5paper, letterpaper
- draft = vázlat, final = végleges
  - vázlat módban képek helyett csak egy keret jelenik meg, és jelzi a margóra túlfolyást
- landscape = fekvő tájolás (alapértelmezés: álló)
- twocolumn = két hasábos szedés (alapértelmezés: egy hasáb)
- twoside = kétoldalas, oneside = egyoldalas
  - margók tükröződők-e, illetve különböző-e a páros és páratlan fejléc és lábléc
- titlepage, notitlepage: van vagy nincs címoldal
  - ha nincs, az első oldal tetején jelenik meg a cím

# Documentclass opciók II

- csomag opciók is bekerülhetnek globális documentclass opcióként(!)
  - pl. ha egy opciót több csomagnak át akarunk adni
  - ha a dokumentumosztály alapértelmezésben betölti a csomagot, sokszor csak így működik, ugyanis egy csomagot nem szabad többször különböző opciókkal betölteni

Dokumentumosztálytól függően:

- különböző opciók használhatók/érvényesek
- különbözhet az alapértelmezés!
  - article: oneseide, notitlepage
  - book: twoside, titlepage

# Magyar nyelv

- `\usepackage[magyar]{babel}`
  - elválasztási szabályok
    - nem mindentudó, kézzel megadható neki elválasztás, pl. `\hyphenation{Ma-gyar-or-szág}`
    - bárhol szerepelhet a parancs; akár több helyen, a problémás szavak után; vagy közösen, egy vesszővel elválasztott listában
  - központozási szabályok – magyar idézőjel, stb.
  - dátum formátuma
  - automatikusan generált szövegek (mint tartalomjegyzék címsora) fordítása
- karakterkódolás magyar ékezetes betűkhöz
  - `\usepackage{t1enc}` – a babel warning-ot (figyelmeztetést) dob, ha ez nincs betöltve
  - forrásban utf8-as karakterek:  
`\usepackage[utf8]{inputenc}`

# Babel több nyelvvel

- többnyelvű dokumentum esetén pl.  
`\usepackage[english,magyar]{babel}`
- az *utolsónak* betöltött az alapértelmezett
- egyes nyelvek sajnos nem kompatibilisek
- váltás: `\selectlanguage{nyelv}`, adott ponttól az új nyelv szabályai érvényesek
- (rövid) idegen nyelvű szövegrészlet beszúrása
  - `\foreignlanguage{nyelv}{idegen nyelvi szöveg}`
  - `\begin{otherlanguage}{nyelv}`, `\end{otherlanguage}` közé
  - érdemes extra csoportba zárni őket, mert mellékhatásaik lehetnek

## 2 Formázás

- Helykitöltő szöveg
- Betűstílusok
- Méretezés
- Központosítás



# Helykitöltő szöveg – lorem ipsum

- 1 `\usepackage{blindtext}`
  - angol helykitöltő szöveg
  - `\blindtext`: 1 bekezdésnyi
  - `\Blindtext[3]`: 3 bekezdésnyi (tetszőleges szám);  
`\Blindtext` alapértelmezésben 5 bekezdésnyi
- 2 `\usepackage{hulipsum}`
  - „lórum ipse”, magyar zagyva szöveg
  - egy előre megadott zagyva szövegből ragad ki adott paragrafusokat
  - `\hulipsum`: alapértelmezésben első 7 bekezdés
  - `\hulipsum[3]`: 3. bekezdés (max. 150)
  - `\hulipsum[3-5]`: 3-5. bekezdések
- 3 `\usepackage{lipsum}`
  - latin zagyva szöveg „lorem ipsum”
  - használata analóg a magyarral

# Betűstílusok I

- mint *dőlt betűk*, **vastag betűk**, stb.
- pl. `\textit{dőlt betűk}` *dőlt betűk*
- kétféle formájuk létezik
  - 1 argumentumos parancs: hatásköre csak az argumentum, pl. `\textit{dőlt betűk}`
  - deklaráció, argumentum nélküli parancs: felülírásig vagy aktuális csoport/környezet végéig érvényes, pl. `\itshape`
- egymásba ágyazhatók és halmozhatók\*
  - `\textit{dőlt és \textbf{vastag}}`  
*dőlt és **vastag***
  - \* ahol értelme van; kategóriánként (ld. lentebb) csak 1 érvényesülhet
- lentebb táblázatok kategóriánként, a parancs mindkét formájával

# Betűstílusok kategóriánként I

## Betűtípusok (family)

hatás	deklaráció	parancs
roman	<code>\rmfamily</code>	<code>\textrm{szöveg}</code>
sans serif	<code>\sffamily</code>	<code>\textsf{szöveg}</code>
typewriter	<code>\ttfamily</code>	<code>\texttt{szöveg}</code>

## Betűvastagság (series)

hatás	deklaráció	parancs
normál (közepes)	<code>\mdseries</code>	<code>\textmd{szöveg}</code>
<b>vastag</b>	<code>\bfseries</code>	<code>\textbf{szöveg}</code>

## Betűstílusok kategóriánként II

## Betűformák (shape)\*

hatás	deklaráció	parancs
<i>italic</i>	<code>\itshape</code>	<code>\textit{szöveg}</code>
<i>slanted</i>	<code>\slshape</code>	<code>\textsl{szöveg}</code>
SMALL CAPS	<code>\scshape</code>	<code>\textsc{szöveg}</code>
upright	<code>\upshape</code>	<code>\textup{szöveg}</code>

- \* Roman betűtípussal, mert a beamerben alapértelmezett sans serif-ben nincs meg minden shape

*Minden* formázási kategória visszaállítása

hatás	deklaráció	parancs
alapértelmezett	<code>\normalfont</code>	<code>\textnormal{szöveg}</code>

# Egyéb betűformázás

- kiemelés `\emph{kiemelt}` *kiemelt*, hatása *általában* dőlt betűk
  - egymásba ágyazható
  - belső parancs ismét egyenes, harmadik szinten ismét dőlt, stb.<sup>1</sup>  
`\emph{kiemelt 1. szint \emph{kiemelt 2. szint  
\emph{kiemelt 3. szint}}}`  
*kiemelt 1. szint kiemelt 2. szint kiemelt 3. szint*
- kis- és nagybetűk
  - `\MakeUppercase{Nagybetűsre!}` NAGYBETŰSRE!
  - `\MakeLowercase{KISBETŰSRE!}` kisbetűsre!
  - vigyázzunk velük, mert parancsokon kívül *mindenre* hatnak, pl. környezetek nevére is, így furcsa hibákat okozhatnak.

---

<sup>1</sup>csalok, mert beamerben pont nem működik

# Betűméretek I

- relatív méretezés, az alapértelmezett betűmérethez képest
- általában deklarációként használjuk őket, azaz felülírásig érvényes parancsként, pl. `\normalsize: normál` (alapértelmezett) méret
  - de létezik környezet verziójuk is, pl. `\begin{normalsize}` és `\end{normalsize}` közti szövegre hat
  - extra nemkívánatos térköz jelenhet meg a környezettel, ezért ritkábban használjuk
- lentebb: parancsok és demonstrációjuk

## Betűméretek II

## méretezési parancsok

parancs	hatás
<code>\tiny</code>	tiny
<code>\scriptsize</code>	scriptsize
<code>\footnotesize</code>	footnotesize
<code>\small</code>	small
<code>\normalsize</code>	<b>normalsize</b>
<code>\large</code>	large
<code>\Large</code>	Large
<code>\LARGE</code>	LARGE
<code>\huge</code>	huge
<code>\Huge</code>	Huge

# Központosítás

- kötőjel - = -; gondolatjel -- = -; angol em dash --- = —
- `\dots` = ... (... = ...)
- idézőjel: *nem* a szokásos dupla idézőjel! "szó" = "szó", csúnya!
  - nyitó: „visszafelé” idézőjel ‘ (duplázva ha kell)
  - záró: szimpla idézőjel/aposztróf ’ (duplázva ha kell)
  - ‘ ‘szó’ ’ = „szó”
- helyes magyar térközökhöz: `\frenchspacing` a preambulumban valahol
  - mert angolban hosszabb a szóköz a mondat végi írásjelek után
  - idegen nyelvnél visszakapcsolás a `\nonfrenchspacing` paranccsal
- ligatúrák: betűtípus függvényében összeolvad ff, fi, fl;  
elkerülés: `{ }` beillesztése a betűk közé



- 3 Dekoráció, mulatságból
  - Színek
  - Keretezés
  - Tükrözés, forgatás, nyújtás
  - Dekorált szöveg tördelése

# Színek I

- szükséges `\usepackage{xcolor}` csomag betöltése (preambulumban)
- szöveg színezése: `\textcolor{red}{piros szöveg}` piros szöveg
- alternatív: deklaráció, pl. `\color{red}`; felülírásig vagy csoport végéig piros szöveg
- néhány alapszín angol neve szerint  
black, blue, brown, cyan, darkgray, gray, green, lightgray, lime, magenta, olive, orange, pink, purple, red, teal, violet, white, yellow
- csomag opciókkal, pl. `\usepackage[dvips]{xcolor}`, több szín elérhető
- dokumentáció-ban lista, hogy milyen opcióval milyen színek érhetőek el

# Színek II

- oldal háttérszíne: `\pagecolor{szín}`
- minden ezt követő oldalra érvényes. kikapcsolás:  
`\nopagecolor`
- színes doboz (mintha szövegkiemelő lenne):  
`\colorbox{yellow}{szöveg sárga háttérrel}`  
szöveg sárga háttérrel
- színes háttér+keret:  
`\fcolorbox{blue}{yellow}{szöveg kék kerettel %  
és sárga háttérrel}`  
szöveg kék kerettel és sárga háttérrel  
`\fcolorbox{blue}{white}{szöveg kék kerettel}`  
szöveg kék kerettel (fehér háttér)
- baj: nem tesz bele sortöréseket, megoldás: tördelés 4 szakaszban

# Keretezés

- keretezett szöveg: `\framebox{szöveg}`
- *nem* tördeli sorokra a szöveget! ld. tördelés 4 szakaszban
- alapértelmezésben a szöveg szélességéhez igazodik a doboz
- első opcionális argumentum, szélesség, pl.  
`\framebox[5cm]{szöveg}`
- alapértelmezésben középre igazít
- második opcionális argumentum, szöveg igazítása, pl.  
`\framebox[5cm][c]{szöveg}`  
c = közép, l = balra, r = jobbra, s = szöveg nyújtása\*, hogy kitöltse a dobozt
- \* Ehhez ragasztó kell, ld. helykitöltés, 2 szakasz

# Tükrözés, forgatás I

- `\usepackage{graphicx}` csomag betöltése szükséges (preambulum)
- forgatás: `\rotatebox{90}{szöveg}` – forgatás 90°-kal, óramutatóval ellentétesen
- nagyítás és nyújtás `scalebox{2.0}[0.5]{szöveg}` – vízszintesen 2-szeres, függőlegesen 0.5-szörös
- függőleges argumentum opcionális, ha kihagyjuk, egyenlőnek veszi a vízszintessel és nyújtás nélkül nagyít
- negatív szám is lehet, akkor az adott irányban tükröz! (mindkét irányban tükrözve 180°-os elforgatás)
- tükrözés: `\reflectbox{szöveg}` – függőleges tengely körül; tulajdonképpen `\scalebox{-1}[1]{szöveg}`

# Dekorált szöveg tördelése I

- *egyetlen bekezdés* tördelése `\parbox`-szal, pl.  
`\parbox{5cm}{szöveg}` – 5cm szélesre
- ha több bekezdés kell: `minipage` környezet, pl.  
`\begin{minipage}{5cm}`  
első bekezdés  
  
második bekezdés  
`\end{minipage}`
- hivatkozási alapnak hasznos lehet: `\textwidth` = szövegtörzs szélessége, vagy `\linewidth` = az aktuális sor hossza
  - törtrésze is, pl. `\parbox{0.8\linewidth}{szöveg}` – sorszélesség 80%-át kitevő doboz

# Dekorált szöveg tördelése II

- ezeket csomagoljuk be a `\framebox`, stb. parancsokba:

```
\framebox{\parbox{5cm}{szöveg}}
```

automatikusan méretezi a paragrafushoz a keretet

- ha nagyobb helyet akarunk mellette hagyni:

```
\framebox[7cm]{\parbox{5cm}{szöveg}}
```

- függőlegesen plusz hely: parboxon *belül* térköz, ld. helykitöltés 2 szakasz
- hivatkozási alapnak hasznos lehet `\baselineskip =` „sortávolság” (két sor alapvonala közti távolság), vagy `\textheight =` szövegtörzs magassága

- 4 Szöveg tagolása
  - Bekezdések
  - Sorköz
  - Igazítás



# Bekezdés, sortörés, oldaltörés I

## Emlékeztető

whitespace a kódban *többnyire* nem jelenik meg a dokumentumban

- két enter avagy üres sor a kódban = új bekezdés
- szinte minden más whitespace kombináció = egy szóköz
- sortörés: `\\`

- alternatív: bekezdés lezárása `\par` paranccsal
- alternatív: új sor `\newline` paranccsal

## Oldaltörés

- oldaltörés: `\clearpage` parancs
- kétoldalas dokumentumnál (twoside documentclass opció)  
`\cleardoublepage`: következő *páratlan* oldalon (jobb oldalon) folytatja

# Bekezdések I

Alapértelmezés: bekezdések első sora behúzással, kivéve bizonyos környezetek után

- `\noindent`: adott bekezdésnél *nincs* behúzás
- `\indent`: adott bekezdésnél *van behúzás*

Alapértelmezés: nincs térköz a bekezdések között

- helyileg kézzel tehetünk térközt a bekezdések közé
- `\smallskip`, `\medskip`, `\bigskip`: kicsi, közepes és nagy függőleges térköz (alapért 3pt, 6pt, 12pt)
- jó tudni: *nem* kényszerítenek új paragrafust, ezért ha nem paragrafusok közt használjuk, csak a sorközt növeli
- `\smallbreak`, `\medbreak`, `\bigbreak`: hasonló hatás, de új paragrafust is kényszerít, és jelzi a  $\text{\LaTeX}$ -nek, hogy itt érdemes(ebb) oldalt törni

# Bekezdések II

## Általános érvényű felülírás

- `\parindent` = behúzás, `\parskip` = bekezdésköz
- áttekinthetőség és oldalkép szempontjából az egyik legyen 0, a másik pedig nem 0
- átállítás: `\setlength{\parskip}{5pt}`, vagy tőrésel `\setlength{\parskip}{5pt plus 2pt}`
- a behúzás lehet negatív is! függő behúzás jellegű hatás, de belelóg a margóba

# Sorköz

- alapértelmezés: szimpla sorköz
- átállítás: `\linespread{szorzó}` parancs, preambulumba téve egész dokumentumra hat
  - másfeles sorköz `\linespread{1.3}`
  - dupla sorköz `\linespread{1.6}`
  - szimpla sorköz visszaállítása `\linespread{1}`
- dokumentumtörzsben is használható, de akkor ki kell adni utána a `\selectfont` parancsot is, hogy hatályba lépjen(!) adott ponttól felülírásig vagy csoport végéig érvényes

# Igazítás

- alapértelmezés: sorkizárt
- egyéb igazításokra deklaráció (visszavonásig vagy csoport végéig érvényes), vagy környezet (`\begin{környezetnév}`, `\end{környezetnév}`)
- érdemes csoportba zárni a deklarációt
- jó tudni: a környezet extra térközt illeszt be – ez alapján döntjük el, melyik verziót használjuk az adott helyen

## Igazítási parancsok

hatás	deklaráció	környezetnév
balra zárt	<code>\raggedright</code>	<code>flushleft</code>
jobbra zárt	<code>\raggedleft</code>	<code>flushright</code>
középre zárt	<code>\centering</code>	<code>center</code>

- 5 Cím formázása
  - Cím
  - Helykitöltés

# Cím I

Két mód van cím létrehozására – egyszerre csak az egyiket használjuk!

## ① kézzel formázott cím

- kézi címoldal – titlepage környezet
- egy oldalt lefoglal
- azon belül kézi betűstílusok, méretezés, igazítás, helykitöltés, stb.
- akkor használjuk, ha adott követelményeknek kell megfelelni, pl. szakdolgozathoz

## ② automatikusan formázott cím

- 2 sablon: címsorok vs címoldal
- címsor: oldal tetején cím, utána folytatódik a szöveg
- címoldal: egy egész oldalt kitölt a cím
- alapértelmezés: article-ben címsor, book-ban címoldal
- módosítás: documentclass opcióval, titlepage/notitlepage
- tipikusan ezt használjuk – egyszerűbb, gyorsabb

## *Automatikusan formázott cím*

### ① adatok megadása

- `\title{Cím}` – kötelező
- `\author{Szerző neve}`
- `\date{Dátum};`  
alapértelmezés (ha nem adjuk meg) `\date{\today}`, „ma” = dokumentum fordításának napja
- üresen is hagyhatók
- dokumentumosztálytól függően extra mezők  
pl. beamerben `\subtitle{Alcím}`, `\institute{Intézmény}`
- megadásuk nem elég, attól még nem jelenik meg(!)
- akárhol megadhatók, még a kiíratás előtt

### ② kiíratás: `\maketitle`

- a `documentclass` beállításától függően generálja a címsort vagy címoldalt
- dokumentumtörzsbe helyezzük el (`\begin{document}` után)



# Ragasztók gyorstalpaló

- $\LaTeX$  fejében minden dobozokból épül fel: szavak, bekezdések is dobozok
- köztük ún. *ragasztó*: térköz ami nyújtható és/vagy zsugorítható
- pl. többrészes ábrák, címodalak szerkesztésénél jó, ha kézzel tudjuk kontrollálni a térközöket
- jó tudni: ha újra akarjuk definiálni a sorközt, paragrafusközt, stb., akkor plusz-mínusz tűréssel tegyük!

# Helykitöltés parancsai I

- általában
  - h = horizontal = vízszintes
  - v = vertical = függőleges
- `\hspace{1cm}`: 1cm kihagyása vízszintesen
- `\hfil`, `\hfill`: vízszintes irányban *végtelenül* nyúló ragasztó – kitölti a rendelkezésére álló teret
  - ha több van belőle, egyenletesen oszlik szét köztük a szabad hely
  - `\hfill` erősebb mind `\hfil`; ha mindkettő szerepel a sorban, a `\hfil` 0 hosszt vesz fel
  - miért van belőle kettő? a belső parancsok `\hfil`-t, azaz a gyengébbet használják; így felül lehet írni
- `\hss`: hasonló mint a `\hfil`, de végtelenül zsugorítható is – negatív hosszt is felvehet

## Helykitöltés parancsai II

- mindnek megvan a függőleges párja, h helyett v-vel
  - `\vspace{1cm}`
  - `vfil`, `\vfill`
  - de `\vss` nincs(!)
- extra: `\\[0.5cm]`: sortörés parancs, opcionális argumentummal – extra 0.5cm függőleges térközt hagy a sortörés után
- jó tudni: `\null` parancs, nem csinál semmit
  - ott használjuk, ahol mi nem akarunk csinálni semmit, de a  $\LaTeX$  megköveteli tartalom jelenlétét
  - pl. oldal tetején térköz, a `\null` létrehoz egy viszonyítási pontot (anélkül a  $\LaTeX$  nem is hajlandó térközt rakni!)

# Hosszmértékek

A leggyakrabban használtak:

- cm, mm
- pt: pont, amiben a betűméretet is mérjük
- $\text{\LaTeX}$  specifikus em,ex: relatív mérték, a betűkészlet határozza meg
  - em = 'm' szélessége, vízszintes hosszakhoz
  - ex = 'x' szélessége, függőleges hosszakhoz
- nyújtható és zsugorítható hossz megadása pl.  
1cm plus 0.2cm minus 0.2cm
  - végtelen nyújthatósághoz pl. 1cm plus 1fill