

L^AT_EX: több forrásfájl, szakdolgozat sablon, KOMA-script bundle

Vadon Viktória

2022/23/1. félév

- 1 Több forrásfájl
 - Áttekintő: miért, és hogyan?
 - Fordítás több forrásfájl esetén
 - Fájlok formátuma
 - Fájlok betöltése: `\input` vs `\include`

Áttekintő: miért, és hogyan? I

- átláthatóság: hosszú dokumentum esetén hasznos lehet több fájlban menteni a forrást
- kód-újrahasznosítás: pl. saját makrógyűjtemény betöltése
- komponensei:
 - kell egy kitüntetett fő/mester/gyökér dokumentum
 - a mester dokumentumban `\input{fájlnév}` vagy `\include{fájlnév}` paranccsal betöltjük a többi fájlt (mintha bemásolnánk a tartalmukat)
 - a *mester* dokumentumot fordítjuk(!)

Fordítás több forrásfájl esetén I

- fordítás több forrásfájl esetén:
- a *mester dokumentumot* fordítjuk(!)
 - Overleaf-ben magától felismeri, a betöltött file-t recompile-olva automatikusan a fődokumentumot fordítja
 - a szerkesztőben beállítható a mester dokumentum – akkor bármi is van nyitva, mindig a mester dokumentumot fordítja
 - program újraindításával elfelejti, vagy kapcsoljuk ki kézzel, ha máson dolgozunk
 - T_EX makerben: beállítások > jelenlegi dokumentum beállítása fődokumentumként (ctrl+0)
 - T_EX studioban: beállítások > gyökérdokumentum >
 - automatikus észlelés opció (mint Overleaf)
 - vagy fődokumentum rögzítése kézzel (mint T_EX maker)

Fájlok formátuma I

- emlékeztető: \LaTeX dokumentum első sora
`\documentclass{...}`, és egyszer szerepelhet benne a
`\begin{document}`, `\end{document}`
- a *mester dokumentum*:
 - ez **A** dokumentum, ezért szabályos \LaTeX fájl kell legyen, azaz:
 - *kötelezően* tartalmazza a `\documentclass{}`-t, és a
`\begin{document}` és `\end{document}` parancsokat
- a hivatkozott fájlok
 - olyan, mintha bemásolnánk a tartalmukat a mester dokumentumba
 - ezért **tilos** tartalmazniuk `\documentclass{}`-t és a
`\begin{document}` és `\end{document}` parancsokat(!)

Fájlok betöltése: `\input` vs `\include` |

- `\input{relatív/elérés/fájlnev}`,
`\include{relatív/elérés/fájlnev}`
 - ha `.tex` a kiterjesztés, akkor nem kell kiírni
 - írási jogosultság miatt javasolt a fődokumentum mappáját vagy almappáit használni
 - pl. ha `fodok.tex` a szakdoga mappában van, `1fej.tex` pedig a `szakdoga/forrasok` mappában, akkor a relatív hivatkozás `forrasok/1fej`
- `\input{fájlnev}`
 - *nyers* másolás és beillesztés
 - használható preambulumban is, pl. makrógyűjtemény betöltésére
 - több `\input` parancs is egymásba ágyazható

Fájlok betöltése: \input vs \include II

- `\include{fájlnév}`:
 - kifinomultabb, magasabb szintű parancs
 - generál egy `fájlnév.aux` fájlt
 - ebben megjegyzi a fájlban elhelyezett `\label`-öket, section és társai címsorokat, és a számlálók állását a fájl végén
 - `\includeonly{fájl1,fájl2}`:
 - preambulumban használható
 - csak a listában szereplő fájlokat importálja `\include`-dal, a többit „elrejt”
 - egy helyen kontrollálható, mit fordítunk le ténylegesen: gyorsítja a fordítást
 - de, a többi „rejtett” `\include`-os fájl `.aux` fájlját is használja!
 - ezért működnek a kereszthivatkozások, kimaradnak a „rejtett” fájlok oldalszámái, stb., a „rejtett” fájlokban lévő section és társai címsorok is megjelennek a tartalomjegyzékben(!)

Fájlok betöltése: `\input` vs `\include` III

- `\include` egyéb tulajdonságai
 - a fájl előtt és után is oldaltörést tesz
 - csak a dokumentumtörzsben (`\begin{document}` `\end{document}` között) használható
 - `\include` parancsok (belük betöltött fájlok) nem ágyazhatók egymásba
- `.aux` fájl generálása `\include`-olt fájlhoz:
 - egyszer le kell úgy fordítani, hogy nem zárjuk ki az `\includeonly`-val!
 - pl. kommentáljuk ki az `\includeonly`-t (vagy vegyük be a listába a fájlt)
 - utána az `.aux` fájl megmarad, akkor kell újragenerálni, ha módosítjuk magát a forrásfájlt
 - a forrásfájl mappájában generálódik, ezért a \LaTeX -nek írási jogosultság kell rá – ezért javasolt a fő dokumentum almappáit használni

- 2 Szakdolgozat sablon
 - Elérhetőség
 - Csomagok, stílusfájlok
 - Magyar hivatkozások

Szakedolgozat sablon I

- letölthető:
 - honlapomról
 - Alkalmazott Matematika tanszék régi honlapjáról (hamarosan frissül)
 - Matematika Intézet honlapjáról
- fő/mester/gyökér dokumentum: `dolgozat.tex`
 - ld. több fájllos fordítás, 1. szakasz
- a fejezetek tartalma
 - egyrészt demó
 - másrészt magyarázat, hogy mi a tartalmi elvárás!
- borító: nagy része formanyomtatvány, nem sokat kell bele írni

Csomagok, stílusfájlok I

- `\usepackage{csomagnév}` parancs: kb. egy `\input{csomagnév.sty}` parancs
- `styles/dolgozat.sty`: gyakorlatilag a preambulum nagy része: csomagok betöltése, beállítása
 - nyugodtan módosítsuk!
- `styles/cpp.sty`, `styles/java.sty`, `styles/python.sty`, `styles/rust.sty`: listings csomag testreszabása különböző programnyelvekhez
 - színes stílusok definiálása hozzájuk, illetve ezeket a stílusokat használó rövidített parancsok/környezetek
 - pl. python-hoz:
 - `\begin{python}[]` környezet: `\begin{lstlisting}[]`
 - `\pythoninline!!` parancs: `\lstinline!!`
 - `\pythonexternal[] {fájlnév}` parancs: `\lstinputlisting[] {fájlnév}`

Csomagok, stílusfájlok II: magyar hivatkozások

- `styles/refs.sty`: magyarított hivatkozások
 - `\mref`, parancs: `a(z) + \ref + . + hivatkozott objektum típusa magyarul*`
 - ha saját típust definiálunk, mint új tételkörnyezet vagy float, a többi mintájára meg kell(het) adni a magyar nevét
 - `\Mref`: ugyanez, csak a névelő nagy betűvel – mondat elejére
 - `\mpref`, `\Mpref`: oldalhivatkozás
 - alias: `\mpageref`, `\Mpageref`
 - tipp: használjuk a sima `\ref` parancsot a label automatikus kiegészítéséhez, majd írjuk át a magyar verzióra
 - egyenletekhez használjuk az `\eqref` parancsot
 - ha valami bonyolultabb kell, ragozás, többes szám stb., használjuk a `babel` csomag `\aref`, `\Aref` parancsait, és írjuk kézzel a körítést

3 KOMA-script

- KOMA-script bundle
- Dokumentumszerkezet, címsorok
 - Szakaszolás
 - Címsorok formázása
 - Oldalstílus, fejléc, lábléc
- Tartalomjegyzék és listák testreszabása
 - Teljes lista testreszabása
 - Bejegyzéstípusok testreszabása
- Oldalbeállítás
 - Betűméret, oldalbeállítás
 - Margók

KOMA-script bundle I

- MikTeX: koma-script dokumentáció
- a L^AT_EX dokumentumszerkesztési, tipográfiai képességeit javítja
- dokumentumszerkesztési csomagok
 - scrextend, scrhack: KOMA-script funkcionalitás egyéb osztályokban
 - tocbasic: tartalomjegyzék és egyéb jegyzékek testreszabása
 - sclayer-scrpage: fejléc, lábléc testreszabása
 - typearea: oldalbeállítás
- KOMA-script osztályok
 - alap osztályok megfelelői: scrartcl, scrreprt, scrbook, sclttr2
 - beépítve tartalmazzák a fenti csomagokat
- beállítások alkalmazása
 - csomag vagy documentclass opció
 - később: `\KOMAOPTIONS{kulcs1=érték1,kulcs2=érték2}`

Dokumentumszerkezet, címsorok I

- eredeti \LaTeX `\cleardoublepage` parancs: következő páratlan oldalon folytat
 - KOMA-script: `\cleardoubleoddpage`,
`\cleardoubleevenpage`
 - opció: `cleardoublepage=oldalstílus` – beillesztett üres oldal milyen oldalstílust használjon
 - opció: `open=any`, `=left`, `=right`: új part és chapter nyitása új oldalon bárhová esik, csak bal oldalon, csak jobb oldalon
- `\flushbottom`: függőleges nyújtás, hogy minden oldal utolsó sora ugyanott végződjön
 - kikapcsolás: `\raggedbottom`

Szakaszolás I

- szakasz cím parancsok
 - eredeti: `\section[rövid]{Címsor}`, rövid verzió tartalomjegyzékben és fejlécben
 - eredeti: számozatlan szakasz címek `\section*`, stb. – nem kerülnek be a tartalomjegyzékbe
- KOMA-script-tel jobban személyre szabhatók
- számozatlan egységek:
 - `\addpart`, `\addchap`, `\addsec`: part, chapter, section szintű cím sor, ami számozatlan, de bekerül a tartalomjegyzékbe, és felülírja a fejléct(!)
 - `\addpart*`, `\addchap*`, `\addsec*` verzió: számozatlan, és törli a fejléceket(!)

Szakaszolás II

- címsor rövid verziója:
 - `heading=optiontohead`: rövid verzió csak fejlécben
 - `heading=optiontotoc`: rövid verzió csak tartalomjegyzékben
 - `heading=optiontoheadandtoc`: rövid fejlécbe és tartalomjegyzékbe is
 - sőt, még részletesebben:

```
\section[head={fejlécbe},%  
tocentry={tartalomjegyzékbe},%  
reference={hivatkozásba},nonumber=false]{Címsor}
```

- ha szöveggént szeretnénk benne `=t` vagy `,t` használni, használjunk `{ }`-t

Címsorok formázása I

- igazítás: `\raggedsection`, `\raggedchapter`, `\raggedpart` parancsok
 - KOMA-script-ben alapértelmezett: `\raggedright` – balra zárt, *nem* sorkizárt
 - pl. középre igazítás
`\renewcommand{\raggedsection}{\centering}`
 - alap osztályokban alapértelmezett sorkizárás
`\renewcommand{\raggedsection}{}`
- méretezés `headings=big`, `=normal`, `=small`
 - előre definiált betűméretezési- és térközsémák
 - betűtípushoz és többi címsorhoz relatívan méretez

Címsorok formázása II

- `chapterprefix`, `chapterprefix=true`: „fejezet” címszó kiírása a fejezetsorszámmal, fejezetcím új sorban (alap osztályokban alapért.)
 - `chapterprefix=false`: csak fejezet sorszám + fejezetcím egy sorban (KOMA-script alapért.)
 - hasonlóképp `appendixprefix` kulcs
 - két sor távolságának beállítása
`\RedeclareSectionCommand[innerskip=8pt]{chapter}`
 - babel fordít, de `chapterprefix=true` esetén angolosan „fejezet X” sorrendben generál, megoldás:
`\renewcommand*{\chapterformat}%
{\thechapter.~\chapapp}`
 - ez tulajdonképpen csak a „sorszám” formátuma, a fejezet nevét majd ő külön beilleszti, nekünk nem kell!
 - `\chapapp`: fejezet ill. appendix szó beillesztése
 - hasonlóképp `\partformat`-ra
`\renewcommand*{\partformat}{\thepart.~\partname}`

Címsorok formázása III

- betűtípus módosítása
 - beállítás 0-ról indulva: `\setkomafont{típus}{parancsok}`, pl. `\setkomafont{part}{\huge\scshape}`
 - kiegészítés, korábbi beállítások megtartásával: `\addtokomafont{típus}{parancsok}`, pl. `\addtokomafont{part}{\scshape}`
 - disposition típus: minden címsorra vonatkozó betűtípus beállításokhoz
 - pl. `\addtokomafont{disposition}{\sffamily}` – minden címsor sans serif betűkkel
 - chapterprefix, partnumber külön formázhatók

Oldalstílus, fejléc, lábléc I

- *nem* kompatibilis a fancyhdr-rel!
- megszokott alap L^AT_EX oldalstílusok (vagy megfelelőik) léteznek: `empty`, `plain`, `headings`, `myheadings`
 - általában a `headings` stílus megfelelő
 - `myheadings` itt is `\markright`, `\markboth` parancsokkal állítható be
 - az alap osztályokkal ellentétben nem tesz caps lock-ot!
- alap osztályokban: `part`, `chapter` címsorba beépítve `\thispagestyle{plain}`
- KOMAscript: `\titlepagestyle`, `\partpagestyle`, `\chapterpagestyle`, `\indexpagestyle` beállíthatók
 - például `\renewcommand*{\chapterpagestyle}{empty}`
 - alapértelmezés itt is `plain` mindre

Oldalstílus, fejléc, lábléc II

- a `chapterprefix` opcióval a fejlécben is megjelenik „fejezet X. fejezetcím”
- `\chaptermarkformat` újradefiniálható
 - ez is csak a „sorszám” formátuma, a címsor nem kell bele
 - ha csak kivennénk a „fejezet” címszót:

```
%\renewcommand*\chaptermarkformat%  
{\thechapter.~}
```
 - magyar szórenddel:

```
%\renewcommand*\chaptermarkformat%  
{\thechapter.~\chapapp:~}
```

Tartalomjegyzék és listák testreszabása I

Teljes lista testreszabása

- `tocbasic` csomag, vagy KOMA-script osztályokban beépítve
- teljes listára vonatkozó beállítások:
`\setuptoc{kit}{opciók}`
- pl. `\setuptoc{toc}{noindent}`
- opció kikapcsolása: `\unsettoc{kit}{opciók}`
- `kit`: külső lista fájl kiterjesztése
 - pl. tartalomjegyzékhez `.toc`, ábrák listájához `.lof`
 - azaz: ez alapján azonosítja be, hogy melyik listára vonatkozik a beállítás
- megengedett, előre definiált opciók:
 - teljes lista: ld. KOMA-script dokumentáció (15.2 szakasz, 384. oldal)
 - `noindent`: behúzás nélkül

Tartalomjegyzék és listák testreszabása II

Teljes lista testreszabása

- `onecolumn`: egy hasámban, kéthasábos dokumentum esetén is
- `leveldown`: alacsonyabb szintű címsor, pl. `section` helyett `subsection`, stb.
- `totoc`: bevétel a tartalomjegyzékbe
- `numbered`: számozott címsor használata (alapért. számozatlan)
- *már létező* listák kiírása, átnevezése a szokásos módon
 - pl. tartalomjegyzék kiírása `\tableofcontents`, átnevezése `\renewcommand{\contentsname}{Tartalom}`
 - alternatív KOMA-script parancsok (`tocbasic`-kel újonnan létrehozott listákra is)
 - kiírás `\listoftoc[Címsor]{kit}`
 - átnevezés `\renewcommand{\listofkitname}{Címsor}`

Bejegyzéstípusok testreszabása I

- adott szintű/típusú bejegyzések formázása: *beépített stílusok* segítségével

```
\DeclareTOCStyleEntries[opciók]%  
{stílus}{típus1,típus2}
```

- például `\DeclareTOCStyleEntries{dottedtocline}`
`{section,figure}`
- típus: pl. `section`, `figure` – bármilyen lehetséges lista bejegyzés
- létező stílusok:
 - `largetocline`, `dottedtocline`, `undottedtocline`
 - `gobble` – kihagyás
 - a mélységet alapvetően a `tocdepth` számláló kontrollálja itt is; de ezen felül is elrejtethők szintek vagy típusok
 - `tocline` – a legrugalmasabb, ezzel használható a legtöbb kézi opció
 - stb., ld. KOMAscript dokumentáció, 387-390. oldalak

Bejegyzéstípusok testreszabása II

- [opciók] segítségével módosíthatók a stílusok – de nem minden attribútum érvényes minden stílussal!
 - teljes lista: KOMA-script dokumentáció, 15. táblázat, 391. oldaltól
 - `indent=hossz`: behúzás
 - `entryformat=\parancs`, pl. `\entryformat=\itshape`, bejegyzés szövege dőlt betűvel
 - `entrynumberformat=\parancs`: bejegyzés sorszáma – megkapja az `entryformat` parancsot *is*
 - `pagenumberformat=\parancs`: oldalszám formázása
 - `linefill=kód`, pl. `linefill=\hfill` – kitöltés a szöveg és oldalszám között
 - kitöltés egy szimbólum ismétlésével:
`linefill={\TOCLineLeaderFill[szimb]}`

Betűméret, oldalbeállítás I

- oldalbeállítás KOMA-script osztályokban, vagy typearea csomaggal
- nem kompatibilis a geometry-vel(!)
- betűméret: `fontsize=méret`
 - bármilyen méret megengedett, nem csak a szokásos 10pt-12pt!
 - sőt, tetszőleges L^AT_EX mértékegységben megadható, nem csak pt-ben
- papírméret, orientáció
 - pl. `paper=A4,portrait`: A4-es, álló
 - `paper=landscape`: fekvő
 - `paper=letter`: levélpapír, `paper=A4`, stb.
 - `paper=29cm:21cm`, 29cm magas, 21cm széles

Betűméret, oldalbeállítás II

- `twoside`
 - `twoside`, `twoside=true`: kétoldalas
 - külső margó kétszerese a belsőnek(!)
 - `twoside=semi`: margók szimmetrikusak, fejléc és lábléc azonos minden oldalon, de a kötés margó helye váltakozik
 - alapért., vagy `twoside=false`: szokásos egyoldalas margók
- `twocolumn`: szokásos módon kéthasábos dokumentum

Margók I

- margók
- alsó margó kétszerese a felsőnek
- általában: hagyjuk az osztályt/csomagot kiszámolni a margókat
- betűméret alapján számol, hogy átlátható mennyiségű szöveg kerüljön egy oldalra, és egész számú sor férjen ki(!)
- DIV=szám, szám négynél nagyobb egész
 - berácsoz, DIV egységre osztva vízszintesen és függőlegesen is, margókra 1+2 egység, többi a szövegtörzs
- BCOR=hossz: kötésmargó, tetszőleges \LaTeX hosszmértékkel

Margók II

- `headinclude`, `footinclude`
 - ha `true`, fejléc ill. lábléc beszámítása a szövegtörzs méreteibe
 - ha `false` (*általában* ez az alapért.), a fejléc és lábléc a margó része
- fejléc magassága `headheight=hossz`, vagy alapért. `headlines=1.25`, sormagasság 1.25-szöröse
- láblécre hasonlóképp `footheight`, `footlines`
- `headsepline`, `footsepline`: `true/false` kapcsolók, fejléc és lábléc elválasztó vonalzója