

## 2.2. feladatsor

### Speciális szoftverek – L<sup>A</sup>T<sub>E</sub>X

Vadon Viktória

2025/26/I. félév

#### Szükséges csomagok

(A csomagok közül némelyik az alap L<sup>A</sup>T<sub>E</sub>X telepítés része, így nem mindet kell kézzel telepíteni)

- hyperref
- varioref, cleveref

#### Korábban már használt, de most is szükséges csomagok

- babel, babel-hungarian, t1enc
- xcolor

## 1. Vetítés, demonstráció

### 1. Feladat (Kereszthivatkozások).

- az előző órai fájl egy másolatában érdemes tovább dolgozni – több oldal, vannak benne `\section` és hasonló címsorok, és listák.
- néhány `\label` – pl. `\section` elejére, végére, számozott `\item`-hez
- `\ref`, `\pageref` rájuk
  - szerkesztő kiegészítés
  - fordítás, `.aux` fájl

### 2. Feladat (`\input` VS `\include`).

- készítsünk egy új *minimális dokumentumot* `article` osztállyal – ő lesz a `main.tex`, többfájlos fordítás fődokumentuma
  - ha offline szerkesztőben dolgozunk, jelöljük is ki fődokumentumnak!
- készítsünk két másik `.tex` fájlt, pl. `forras1.tex` és `forras2.tex`, amiket importálni fogunk – ezekbe nem kell (sőt nem is szabad!) `\documentclass`, sem `\begin{document}`, `\end{document}`!

- c) `forras1`-ben helyezzünk el egy `section` címsort és egy `\label`-t
- d) `forras2`-ben szintén egy `section` címsort és egy `\ref`-et a másikban lévő `\label`-re
- e) `\input`-tal importáljuk őket a `main.tex`-be és fordítsuk le! figyeljük meg a formázást is.
- f) kommentáljuk ki `\input{forras1}`-et és fordítsunk így is. mi történt a hivatkozással?
- g) vegyük ki a kommentelést és cseréljük le az `\input` parancsokat `\include`-ra. mi változott? nézzük meg a formázást és a forrásmappát is!
- h) nézzünk bele a `forras1.aux` fájlba!
- i) a `preambula`-ba helyezzünk el egy `\includeonly{forras2}`-t. mi történik, ha így fordítjuk le a fájlt?

## 2. Kötelező feladatok

### 3. Feladat (Kereszthivatkozások +).

- a) folytassuk a munkát a kereszthivatkozások demo dokumentumban.
- b) helyezzünk el további `label`-öket a következő helyeken:
  - egy mélyebb szintű szakasz címsor, pl. `\subsubsection` után
  - valahová a függelékbe
  - egy másik számozott lista egy eleméhez
  - ha van: egy számozatlan `\item[ ]` elemhez, leíró lista egy eleméhez
- c) az első `section` elején helyezzünk el mindre külön-külön `\ref` és `\pageref` hivatkozásokat, és figyeljük meg, mit ír ki!
- d) magyar `babel`-lel használhatjuk az `\aref`, `\Aref`, `\apageref` és `\Apageref` parancsokat is. próbáljuk ki őket. miben különböznek a sima `\ref`, `\pageref` verzióktól?
- e) töltsük be a `hyperref` csomagot, és nyissuk meg a `.pdf` fájlt `Abode Reader`rel (vagy más elérhető pdf olvasóval). mi változott?
- f) linkek megjelenése: állítsuk át, hogy színes keret helyett a linkek szövege legyen színes. állítsunk be hozzájuk tetszőleges színt is! – most már a `LATEX` szerkesztőnk beépített `.pdf` olvasójában is látjuk a különbséget.
- g) teszteljük le, hová ugranak a hiperlinkek!
  - ha valamelyik nem jó helyre visz, orvosoljuk a problémát, pl. `\phantomsection`-nel
- h) helyezzünk el egy weblap hivatkozást `\href`-fel!
- i) a tartalomjegyzékben legyen a szakaszok neve és az oldalszám is hiperlink! – tipp: `hyperref` csomag beállítás

### 3. Választható feladatok

#### 4. Feladat (Számozatlan szakasz).

- a) folytassuk a munkát az előző dokumentumban.
- b) (a függelék előtt, ha van) hozzunk létre egy számozatlan `\section*`-t, és generáljunk utána legalább 2 oldalnyi zagyva szöveget!
- c) figyeljük meg, mi történik a futó fejléccel és lábléccel a számozatlan szakaszban!
  - orvosoljuk is a problémá(ka)t!
- d) van-e rá *egyszerű* megoldás, hogy egy számozatlan `\section*` bekerüljön a tartalomjegyzékbe? ha van, implementáljuk!
- e) a `\section*` után helyezünk el egy `label`-t!
- f) a fájl elejéről hivatkozunk erre a `label`-re. mi történik? mi lesz a `\ref` által generált felirat, és hová ugrik?
  - van-e *egyszerű* megoldás ezekre a problémákra? ha van, implementáljuk!

#### 5. Feladat (Hivatkozások „automatizálása”).

- a) folytassuk a munkát az előző dokumentumban.
- b) próbáljuk ki a `hyperref` csomag `\autoref` és `\autopageref` parancsait néhány különböző „típusú” `label`-re (pl. egy szakaszcímsor, egy `\item`, ...)
- c) próbáljuk ki a `varioref` csomag `\vref`, `\pageref` parancsait:
  - `autoref`-hez hasonlóan teszteljük több „típuson”
  - ugyanarra a `label`-re helyezünk el `\vpageref` hivatkozásokat a `label`-hez képest előző, azonos és következő oldalon (+ ha elég hosszú a dokumentum, akkor még távolabb is egyet)! mi történt?
- d) teszteljük a `cleveref` csomag `\cref` és `\cpageref` parancsait is – erről a kiegészítő/szorgalmi diasorban olvashatnak, vagy a csomag dokumentációjában.
  - ezt is nézzük meg több különböző `label` „típuson”!
  - tegyünk egy `\cref` hasába több `label`-t!
  - mit csinál `\crefrange`, `\cpagerefrange`?
  - hogyan tudjuk magyar szórendre átállítani a generált szövegeket?