

L^AT_EX dokumentumszerkesztés II.

Kereszthivatkozások, több forrásfájl

Vadon Viktória

2023/24/I. félév

1 Hivatkozások

- Alapok
- Kattintható hivatkozások: `hyperref`
- Külső linkek hozzáadása, vagy belső linkek eltávolítása
- Linkek formázása
- Linkek a tartalomjegyzékben
- Hivatkozás „automatizálása”

Kereszthivatkozások, alapok I

- keresztivatkozások eredetileg *nyomtatásra* szánva
 - cél: tudjuk, hogy hol és/vagy milyen néven keressük, amikor visszautalunk egy korábbi definícióra, szakaszra, oldalra, stb.
- ezen alapszik a beépített alap funkcionalitás:
 - a hivatkozott objektum **sorszámát** illeszti be, pl. 3. definíció, 1.2. subsection, stb.
 - a „körítést” nekünk kell hozzáírni¹
 - így mi haszna? a sorszámok, oldalszámok automatikusan frissülnek
- ma már (csomagok segítségével) kattintással (is) navigálhatók a keresztivatkozások

¹legalábbis extra csomagok nélkül

Kereszthivatkozások működése

két komponens:

- hivatkozási pontok, horgonyok elhelyezése `\label{mylabel}` paranccsal
- hivatkozás, link elhelyezése `\ref{mylabel}` vagy `\pageref{mylabel}` paranccsal
 - `\ref` az *objektum sorszámát* illeszti be – ahol értelme van
 - ha nincs értelme, a legutóbbi számozott szakaszcímsor sorszámát adja
 - `\pageref` az oldalszámot illeszti be, ahová a label esik – ennek mindig és mindenhol van értelme
- egy label-re több hivatkozás is mutathat
- a szerkesztő megjegyzi az elhelyezett label-öket, `\ref{}` gépelésekor segít a kiegészítésben!

Mire figyeljünk I

- fordítási tanácsok
 - a label-öket külső segédfájlban tárolja, onnan olvassa be!
 - előny: előre is lehet hivatkozni (azaz a label-re elhelyezett hivatkozás hamarabb is lehet, mint maga a label)
 - hátrány: két fordítás kell, hogy a `\ref{}` parancs működjön
 - addig „undefined label” warning, ?? a `\ref` helyén
- tanácsok a `\label{név}`-ökhöz
 - a név tetszőleges, **egyedi** string
 - csak warning, ha nem egyedi; az utolsó előfordulás lesz érvényes
 - konvenció az elnevezéshez: `tipus:nev` formában
 - a típus az objektum típusa, sokszor rövidítve
 - pl. egy Bevezetés nevű section-höz `\label{s:bevez}`

Mire figyeljünk II

- label elhelyezése:
 - számozott objektumon belül, ha van rá lehetőség, pl. ábrában:

```
\begin{figure} % ábra környezet
\caption{Teljes gráf} % ábra felirata
\label{f:teljes_graf}
% kép beillesztése...
\end{figure}
```
 - vagy közvetlenül az objektum után, pl. szakaszcímsor:

```
\section{Bevezetés}
\label{s:bevez}
```
- tanácsok `\ref`, `\pageref`-hez
 - ha nem számozott objektumhoz került a label, hanem pl. hosszú szöveg közepére, számozatlan szakaszba, lehetőleg `\pageref`-et használjunk
 - csomagokkal lesz majd trükközés és alternatíva

Kattintható hivatkozások: hyperref I

- hyperref csomag – dokumentáció (link)
- `\usepackage{hyperref}`
- !! **utoljára** betöltendő csomag!
- !! egyes hatásai nem (vagy másképp) látszanak a \LaTeX szerkesztő beépített .pdf olvasójában – ellenőrizzük Acrobat Reader-ben is!
- sokrétű funkcionalitás, pl.:
- automatikusan .pdf könyvjelzőket készít a szakaszokból
- automatikusan kattintható hiperlinkeket helyez el minden szakaszhoz a tartalomjegyzékben

Kattintható hivatkozások: hyperref II

- automatikusan kattintható linkké alakítja az összes `\ref` és `\pageref` eredményét
 - ha a label-t nem egy számozott objektummal helyeztük el, *nem mindig* jó helyre navigál
 - megoldás: `\phantomsection` parancs
 - közvetlen ezután helyezzük el a label-t
 - „szellem szakaszcímsor”, láthatatlan, de navigációs pontként szolgál
 - például

```
\section{Lórum ipse}
\hulipsum
\phantomsection\label{hulipsum_kozep}
\hulipsum
```


Külső linkek hozzáadása, vagy belső linkek eltávolítása

- külső link elhelyezése:
 - `\href{url}{megjelenítendő szöveg}`
 - `\url{url}`: ekvivalens `\href{url}{url}`-lel
 - hiperlink nélkül, de formázva: `\nolinkurl{url}`
- hiperlinkek eltávolítása a kereszthivatkozásokból:
 - ha más célból használjuk a `hyperref` csomagot
 - `implicit=false` – precízen: nem definiálja újra a belső parancsokat, azaz a `\ref` stb. parancsokból nem csinál hiperlinket, *de* külső url-ek elhelyezhetők
 - `draft` („vázlat”) csomag opció – minden hiperlinket eltávolít, külső url-eket is (nyomtatáshoz pl.)
 - *csak az adott hivatkozás* ne legyen hiperlink: `\ref*` és `\pageref*` parancs

Linkek formázása I

- hyperref csomag beállítása:
 - vagy csomag opcióként
`\usepackage[beállítások]{hyperref}`
 - vagy betöltés után (még preambulumban)
`\hypersetup{beállítások}`
- linkek jelölése
 - alapértelmezett: színes kerettel
 - ez a szerkesztő beépített pdf-nézegetőjében nem biztos, hogy megjelenik, de pl. Adobe-val igen
 - linkek színezése: `colorlinks` opció
 - kiskapitálissal szedve: `frenchlinks` opció
 - ne jelezze sehogy (de attól még kattinthatók): `hidelinks` opció

Linkek formázása II

- színek: link típusától (kereszthivatkozás, irodalomjegyzék, külső url, stb.) függ
 - xcolor csomaggal név szerint elérhetők a színek
 - értelemszerűen, a választott opció alapján a keret vagy a szöveg színét adjuk meg
 - linkcolor, linkbordercolor: kereszthivatkozás vagy kerete, alapért: red
 - urlcolor, urlbordercolor: külső url vagy kerete, alapért: magenta
 - citecolor, citebordercolor: idézet (irodalomjegyzék) vagy kerete, alapért: green
 - allcolors, allbordercolors: minden link vagy keretének színe

- például

```
\usepackage[colorlinks,linkcolor=blue,%  
urlcolor=purple,citecolor=red]{hyperref}
```

Linkek a tartalomjegyzékben

- a hyperref a tartalomjegyzékben a szakaszok nevét és/vagy oldalszámát hiperlinkké tudja alakítani
 - linktoc=section: szakasz neve a link (alapértelmezés)
 - linktoc=page, vagy linktocpage: oldalszám a link
 - linktoc=all: mindkettő
 - linktoc=none: egyik sem

Hivatkozás „automatizálása” I

- emlékeztető: `\ref` és `\pageref` alapesetben a hivatkozott dolog sorszámát illeszti be, a „körítést” nekünk kell írni
- egyes csomagok segítenek a „körítés” generálásában
 - magyar nyelven nehézkes, a szórend miatt, alább csak egy kis ízelítő és ajánló további olvasásra:
 - magyar babel-lel a számhoz igazodó *a/az* névelő, kis- vagy nagybetűvel: `\aref`, `\Aref`, `\apageref`, `\Apageref`
 - `hyperref` csomaggal `\autoref` és `\autopageref` parancsok: a sorszám *elé* írja a dolog típusát
 - pl. `figure 1, page 3, stb.`, angolul egyszerű és hasznos
 - `babel` ugyan lefordítja, **de** a formátum nem változtatható, magyarul `ábra 1, oldal 3` még mindig bugyután hangzik :(`zsákutca`
 - létezik hiperlink nélküli *-ozott verzió is

Hivatkozás „automatizálása” II

- `varioref` csomag – dokumentáció
 - `\vpageref` parancs: hivatkozott pont távolságától függően ezen az oldalon, vagy az előző oldalon, vagy $a(z) x.\backslash$ oldalon típusú mondatokat generál
 - `\vref` parancs = sorszám + `\vpageref`, de típust ez sem tud közéilleszteni, ahogy magyarul kéne :(
 - magyarul `\ref` + típus kézzel + `\vpageref` hasznos lehet
 - tud tartományokat is kezelni (nem részletezzük, ld. csomag dokumentáció)
- `cleveref` csomag – dokumentáció
 - mindent (is) tud: tartományok kezelése, típus kiegészítése, *teljesen* személyre szabható formátum
 - teljesen magyar nyelvre lehet szabni
 - viszont a személyre szabása meglehetősen bonyolult, itt nem merülünk bele – külön szorgalmi diasorban magyarul vagy csomag dokumentációban angolul lehet róla olvasgatni

- 2 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/gyöker 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 I

- `\input{relatív/elérés/fájlnév}`,
`\include{relatív/elérés/fájlnév}`
 - 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{fájlnév}`
 - *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{fáj11,fáj12}`:
 - 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 kereszt-hivatkozások, 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