

L^AT_EX: irodalomjegyzék és szójegyzék

Vadon Viktória

2022/23/I. félév

1 Irodalomjegyzék

- Bevezetés
- BibT_EX
- Fordítás
- Adatbázis: .bib fájl felépítése
- Idézetek/forráshivatkozások – a .tex fájlban
- Irodalomjegyzék (lista) – a .tex fájlban
- Integráció hyperref-fel
- Adatbázis mélyebben

Bevezetés

- több módszer is létezik irodalomjegyzék kezelésére
- a leghatékonyabbat vesszük

Irodalomjegyzék kezelésének komponensei:

- 1 egy irodalomjegyzék a fájlban – kb. számozott lista, elemei az idézetek
 - minden beazonosításhoz és megkereséshez szükséges adattal
 - pl. könyv címmel, szerzővel, kiadóval, kiadás évével és sorszámával, stb.
- 2 idézetek/forráshivatkozások a szövegben, `\cite` paranccsal – kb. keresztshivatkozások a lista elemeire
- 3 opcionális, de kifejezetten hasznos: külső „adatbázis”, `.bib` fájl(ok)
 - feldolgozás BibT_EX motorral

Irodalomjegyzék BibT_EX-kel

Miért használjunk .bib fájl(oka)t és BibT_EX-et?

- a dokumentumban lévő irodalomjegyzéket (listát) automatikusan generálja belőle!
 - automatikusan rendez és egységes formázást alkalmaz – csomagok nélkül néhány előre definiált stílus közül választhatunk (később bővebben)
 - az adatbázis lehet nagyobb, de az adott dokumentumba csak azt a bejegyzést emeli be, amire ténylegesen hivatkozunk
 - ezt a `\cite` parancsok alapján ismeri fel – azaz nem csak keresztshivatkozások
- áttekinthető és újrahasznosítható
 - minden forrás egy helyen, több különböző dokumentumba „importálható”
 - akár több .bib fájl is „importálható” egy dokumentumban, témakörönként rendszerezhetők az idézetek

Fordítás BibT_EX-kel

.bib fájl feldolgozása: BibT_EX motor

- nehézség: több fordítási lépés

Fájl fordítása BibT_EX használatakor:

- 1 pdflatex (szokásos fordítás) – \cite parancsok olvasása, kulcsok kigyűjtése
- 2 bibtex – adott kulcsú bejegyzések kiolvasása a .bib fájl(ok)ból, .bbl segéd fájl generálása
- 3 pdflatex – irodalomjegyzék lista elkészítése
- 4 pdflatex – \cite parancsokból kereszthivatkozás a listára

Fordítás szerkesztőkben

- T_EXMaker
 - gyorsfordítás (pdf_latex+pdf megjelenítése): F1
 - BibT_EX: F11
 - vagy: kiválasztás a legördülő listából és nyíl gomb
 - érdemes (legalább) az egyik billentyűt memorizálni, akkor nem kell mindig váltogatni a legördülő listában!
- T_EXStudio
 - fordítás (alapértelmezésben p_ldf_latex-kel) és megjelenítés: F5, vagy dupla zöld nyíl
 - BibT_EX: F8, vagy Eszközök menü > Bibliográfia
- Overleaf
 - egyetlen recompil-e-lal automatikusan elvégzi mind a négy lépést

Adatbázis: .bib fájl |

a .bib fájl szintaxisa eltér a \LaTeX fájlétól! példa:

- `@book{lampport94,`
 `author = {Leslie Lamport},`
 `title = {\LaTeX}: a document preparation system},`
 `publisher = {Addison Wesley, Massachusetts},`
 `year = {1994}`
 `}`

- `@unpublished{abrharber03,`
 `title = {\TeX} for the Impatient},`
 `author = {Paul W. Abrahams and`
 `Kathryn A. Hargreaves and Karl Berry},`
 `year = {2003},`
 `url = {http://tug.ctan.org/info/impatient/book.pdf}`
 `}`

Adatbázis: .bib fájl II

- minden bejegyzés után üres sor (majd hyperref helyes működéséhez)
- **@book**: bejegyzés típusa, pl. könyv
- a típus után a teljes bejegyzést egy pár kapcsos zárójel határolja { }
- kulcs után és minden mező között elválasztó vessző ,
 - utolsó mező után már nem kell
- **lamport94**: kulcs, egyedi azonosító
 - ez alapján tudunk majd a bejegyzésre hivatkozni (gondoljunk rá úgy, mint egy \label-re)
 - tetszőleges, de érdemes rendszert vinni bele
 - gyakori pl. szerző(k rövidítése) + évszám

Adatbázis: .bib fájl III

- bejegyzés adatai, mezői: `mezőnév = {érték}`, vagy `mezőnév = "érték"` párok
 - a mezőnevek között, bejegyzés típusától függően, van kötelező, és vannak megengedett mezők
 - értéket vagy kapcsos zárójelek, vagy idézőjelek határolják – érdemes konzisztensnek lenni
 - az értékben használhatók \LaTeX parancsok
 - csomagoljunk kapcsos zárójelbe mindent, ahol kis-és nagybetű nem változhat – többek között \LaTeX parancsokat
- szerzők közé `and` kulcsszó
- megjegyzés elhelyezése: nincs külön komment karakter!
 - de bejegyzések közti szöveget vagy érvénytelen mezőtípust nem dolgoz fel – pl. tegyünk kérdőjelet a mezőnév elejére

Idézetek/forráshivatkozások – a .tex fájlban I

- idézés/forráshivatkozás elhelyezése: `\cite{kulcs}` parancs, ahol a kulcs egy .bib fájlbeli bejegyzés kulcsa
 - Lamport könyve `\cite{lamport94}` a `\LaTeX\` nyelv eredeti leírása...
Lamport könyve [2] a `LaTeX` nyelv eredeti leírása...
- opcionális argumentum: tetszőleges kiegészítő szöveg, pl. ha oldalszámmra, adott tételre szeretnénk hivatkozni
Lamport könyvében `\cite[12.\ old]{lamport94}` megtaláljuk...
Lamport könyvében [2, 12. old] megtaláljuk...

Idézetek/forráshivatkozások – a .tex fájlban II

- egy `\cite` parancsban több hivatkozás is lehet:

`\TeX` és `\LaTeX` források

`\cite{abrharber03,lamport94}`...

`TEX` és `LATEX` források `[1,2]`...

- figyeljünk rá: itt a kulcsok felsorolt sorrendjében illeszti be a sorszámukat – ha szükséges, cseréljük meg kézzel a kulcsokat, hogy növekvő legyen a sorrend!
- `\nocite` parancs: „láthatatlan” idézet – az irodalomjegyzékbe (listába) beilleszti a bejegyzést, de nem tesz rá keresztshivatkozást
 - `\nocite{*}`: a `.bib` fájl összes bejegyzését beilleszti

Irodalomjegyzék (lista) – a .tex fájlban I

- irodalomjegyzék stílusa: pl. `\bibliographystyle{plain}`
 - bejegyzések formázása, rendezés, lista számozása
 - kötelező megadni az irodalomjegyzék beszúrása előtt
- *előre definiált stílusok:*
 - `plain`: ábécé-rend, számozott lista
 - `unsorted`: unsorted, azaz a felbukkanási sorrendben listáz és számoz
 - `abbrev`: szerzők keresztneve rövidítve
 - `alpha`: számozás helyett szerzők neve+évszám típusú rövidítés
 - `acm`: szerzők neve kiskapitális betűkkel, vezetéknev elől
 - csomagokkal egyéb opciók, ld. pl. `LATEX Wikibooks`

Irodalomjegyzék (lista) – a .tex fájlban II

- irodalomjegyzék forrása(i) + listázás:
`\bibliography{forras/matek,forras/compsci}`
 - forrásfájlok vesszővel elválasztott listában
 - *relatív* elérési úttal
 - .bib kiterjesztést nem kötelező kiírni
 - dokumentum adott pontján, számozatlan section-ben (vagy ha létezik, chapter-ben) helyezi el a listát – tipikusan a dokumentum végén szokás
 - előtte definiálni kell a stílust!
- babel lefordítja a címsorát – article-ben Hivatkozások, book-ban Irodalomjegyzék
 - kézzel átnevezhető
 - article-ben: `\renewcommand{\refname}{Források}`
 - book-ban: `\renewcommand{\bibname}{Források}`

Integráció hyperref-fel I

- `\usepackage{hyperref}` használatával automatikusan kattintható hiperlinkké alakulnak a `\cite` parancsok
 - linkek formázása: `citebordercolor=` vagy `citecolor=` a `hyperref` beállításai közé
 - emlékeztető, `hyperref` beállítása: betöltéskor `\usepackage[opciók]{hyperref}`, vagy csomag betöltés után `\hypersetup{opciók}`
 - emlékeztető: színekhez `xcolor` csomag

Integráció hyperref-fel II

- extra funkcionalitás: „vissza” linkek elhelyezése az irodalomjegyzék listából az oda mutató `\cite` parancsok helyére
 - hyperref csomag beállításai közé:
 - alapértelmezésben ki van kapcsolva, `backref=none`
 - vissza link a legközelebbi szakaszolási egység címsorhoz: `backref=section` (vagy röviden `backref`)
 - vissza link az adott oldalra: `backref=page` (vagy röviden `pagebackref`)
 - csak beamer documentclass-ban: vissza link diára `backref=slide`
 - ennek a funkciónak a helyes működéséhez szükséges az üres sor a `.bib` fájl bejegyzései között!

Hogyan szerzünk vagy szerkesztünk .bib fájlt? I

- tudományos újságok honlapjáról vagy Google Scholarból általában letölthető .bib/BibTEX formátumban az adott cikk idézete!
 - egy bejegyzés, mezők kitöltve
 - elég bemásolni a saját .bib fájlunkba (sorkihagyásra figyeljünk)
- ha mégis kézzel kell: táblázat a típusokról és hozzájuk tartozó kötelező és megengedett mezőkről \LaTeX Wikibooks-ban (link) (kicsit görgessünk le)
 - nem teljes a táblázat, de elég jó!
 - táblázat fölött referencia linkek

Hogyan szerzünk vagy szerkesztünk .bib fájlt? II

- tudnivalók a szerző(k) nevéről...
 - több szerző: and kulcsszóval elválasztva
 - alapvetően angolosan Keresztnév Vezetéknév sorrendben(!)
 - több vezetéknév esetén kapcsos zárójelezzük
John {von Neumann}, vagy „Vezetéknév, Keresztnév”,
vesszővel elválasztva: von Neumann, John
- néhány gyakori típus: következő diákon

Gyakori típusok és mezők .bib fájlban I

- megjegyzések
 - ha kötelező mező hiányzik, hibát dob
 - ami nem kötelező vagy megengedett mező, az nem fog megjelenni, de hibát nem okoz! extra információ vagy megjegyzés tárolható így, pl. kivonat
 - vigyázzunk, ha opcionális mezőt elgépelünk, akkor csak figyelmen kívül hagyja, nem dob hibát
 - abécé rendben lesznek – magunknak érdekesebb fontossági sorrendben a .bib fájlban, szerző(ke)t és címet előre, stb.
- @article – újságcikk
 - kötelező: author (szerző), journal (újság), title (cím), year (év)
 - megengedett: month (hónap), note (megjegyzés–itt lehet pl. linket megadni), number (folyóirat sorszáma), pages (oldaltartomány), volume (kötet)
- @book – könyv
 - kötelező: publisher (kiadó), title, year

Gyakori típusok és mezők .bib fájlban II

- megengedett: address (postai cím), author, edition (kiadás), editor (szerkesztő), month, note, number, series (sorozat/széria), volume
- @masterthesis, @phthesis – szakdolgozat, disszertáció
 - kötelező: author, school (iskola), title, year
 - megengedett: address, month, note, type (típus)
- @unpublished: még nem publikált (pl. arXiv cikkek)
 - kötelező: author, note, title
 - megengedett: month, year
- @misc: egyéb (pl. weboldal)
 - megengedett: author, howpublished (publikálás módja – pl. webcím), month, note, title, year

2 Irodalomjegyzék testreszabása – biblatex

- A csomag célja
- Szintaxis
- Fordítás biber-rel
- Stílus: számozás, rendezés
- Integráció hyperref-fel
- Egyéb beállítások: rövidítés, dátum, ...
- Irodalomjegyzék kiírása
- Irodalomjegyzék több részletben

biblatex – csomag célja

- biblatex csomag – dokumentáció link
- csomag célja: BibT_EX továbbfejlesztése, irodalomjegyzék (teljes) testreszabása
- jó hír: babel-lel együttműködik
 - babel *után* töltjük be a biblatex-et
 - generált szavak (pl. „oldal”, „és”) automatikus fordítása
 - biblatex és babel használatához (warning formájában) javasolja a csquotes csomagot
- .bib fájl feldolgozása
 - BibT_EX-kel, ahogy eddig
 - **biber** programmal – jelentősen növeli a funkcionalitást, extra mezők, stb.
 - ugyanaz a .bib fájl feldolgozható akármelyikkel, ott nem változik a szintaxis – a BibT_EX csak figyelmen kívül hagyja a számára ismeretlen extra mezőket

biblatex – szintaxis |

```
% preambulom
\usepackage[backend=biber,style=numeric,%
egyéb opciók]{biblatex}
\addbibresource{forras/matek.bib}
\addbibresource{forras/compsci.bib}
% dokumentumtörzs
\printbibliography[opciók]
```

- részben a **style=** opció veszi át `\bibliographystyle` szerepét
 - **numeric** a plain stílusra emlékeztet, a hivatkozások számozását jelenti
 - de a rendezést itt külön tudjuk szabályozni egyéb opciókkal!
 - stílusok, rendezés **CROSSREF**

biblatex – szintaxis II

- `\addbibresource`, `\printbibliography`: források beolvasása és irodalomjegyzék kiírása itt elkülönül!
 - mire figyeljünk: forrásokat külön-külön kell betölteni `\addbibresource`-szal
 - előnye: kiíráskor `\printbibliography` opcióival formázás, címsor megváltoztatása, stb.
- `backend=biber`, vagy `backend=bibtex`:
 - biber-rel nagyobb funkcionalitás, de mindkettő jól működő opció
 - mindkét motor/compiler .bbl fájlt generál, de *különböző szintaxissal!*
 - ezért mindig azt adjuk meg biblatex-nek, amellyikkel ténylegesen fordítunk
- rendszertől/verziótól függően, `backend=` és `style=` kötelező opciók, hibát dob(hat) ha hiányzik valamelyik

Fordítás biber-rel I

Overleaf: backend= beállítás alapján automatikusan választ compiler-t, továbbra is egyetlen recompile-lal elvégzi az összes szükséges lépést

T_EXStudio:

- elég beállítani az alapértelmezett bibliográfia fordítót:
Beállítások menü > T_EXStudio beállításai > fordítás fül >
Alapértelmezett bibliográfia eszköz > legördülő menüből
BibT_EX vagy biber
- ennek megfelelően adjuk meg a biblatex csomag opcióját
- ezután a fordítás megegyezik a 3. szakaszbeli BibT_EX-es fordítással

Fordítás biber-rel II

T_EXmaker:

- Felhasználó menü > Felhasználói parancsok > Felhasználói parancsok szerkesztése
 - biber néven, `biber %` parancs
 - ezzel létrehoztunk egy felhasználói parancsot, a legördülő lista alján megjelenik
- biblatex csomagnak is adjuk meg a `backend=biber` opciót
- a fordítás menete megegyezik a 3. szakaszbeli BibT_EX-es fordítással, csak a második BibT_EX lépést cseréljük ki az új biber parancsra
- warning-ok formájában szól is mindig, hogy mi a következő lépés: „please (re)run latex/biber”

Stílus: számozás, rendezés I

- stílus: `style=`
 - `=numeric`: számozás a forráshivatkozás helyén és az irodalomjegyzék listában is
 - `=alphabetic`: szerző(k) neve+évszám alapú rövidítés a forráshivatkozás helyén és az irodalomjegyzék listában is
 - `=authoryear`: forráshivatkozás helyén szerző(k) teljes neve+évszám, irodalomjegyzék listában nincs jelzés(!)
- rendezés: `sorting=`
 - n,t,y betűk sorrendje adja
 - n = name: szerző, vagy szerkesztő
 - t = title: cím
 - y = year: évszám
 - de nem minden kombináció érvényes(!)
 - *néhány* érvényes: nty, nyt, ynt
 - `=none`: nem rendez, idézés sorrendjében

Integráció hyperref-fel

- `\cite` parancsok hiperlinkké alakítása: `hyperref` (ekvivalens: `hyperref=true`) a **biblatex csomag opciói közé**
 - nem automatikus!
- „vissza” linkekhez `backref` (ekvivalens: `backref=true`) a **biblatex csomag opciói közé**
 - nem kompatibilis a `hyperref` csomag `backref` opciójával(!)

Egyéb beállítások: rövidítés, dátum, ...

- csomag opciók
- `abbreviate`: ha igaz, generált szavak rövidítése, pl. oldal helyett old.
- `giveninits`: ha igaz, keresztnevek rövidítése egy betűre
- `sortcites`: ha igaz, az egy `\cite` parancson belüli hivatkozásokat rendezi
- `date=`: (babel hatással van erre is!)
 - `year`: csak év
 - `long`: pl. 2022. október 2.
 - `comp`: long, tartományok összevonásával
 - `short`: pl. 2022. 10. 2.
 - `terse`: short, tartományok összevonásával

Irodalomjegyzék kiírása

- `\printbibliography[opciók]` parancs
- címsor stílus/sablon: `heading=`
 - `bibliography`: alapértelmezett, számozatlan
 - `bibintoc`: számozatlan, de bekerül a tartalomjegyzékbe(!)
 - `bibnumbered`: számozott
- név/címsor szövegének megváltoztatása:
`title=Irodalomjegyzék`
- `prenote={szöveg}`: szöveg beszúrása a címsor és a tényleges irodalomjegyzék lista közé
- `postnote={szöveg}`: szöveg beszúrása a lista után
- `\printbibheading`: csak a címsor kiírása – hasznos lehet, ha utána több részletben írjuk ki az irodalomjegyzéket
 - itt is működik pl. `title`, `prenote`

Irodalomjegyzék több részletben I

- forrás típusára (mint @article, @book) szűrés:
 - *egyetlen* típus kiválasztása:
`\printbibliography[type=article]`
 - típus(ok) kizárása:
`\printbibliography[nottype=unpublished,%
nottype=misc]`
- könyvnél hasznos lehet pl. fejezetenként külön irodalomjegyzéket
- ún. refsection egységek hozhatók létre
 - nem feltétlen esik egybe a szakaszolással, de hasonlóképp egy lineáris felosztása a dokumentumnak
 - refsection-önként kiíratható az irodalomjegyzék
 - biblatex automatikusan számozza és `refsection:sorszám` `\label`-lel látja el őket

Irodalomjegyzék több részletben II

- refsection-ök automatikus létrehozása/elkezdése adott szintű szakaszolási egységgel, pl. section-nel: `refsection=section` csomag opció
 - `refsection=section+` variáns: section és magasabb szintű egységek (part, és chapter ha létezik) kezdetén új refsection
 - számozatlan egységekre is generál refsection-t – ezért a refsection sorszáma nem feltétlen egyezik meg a section sorszámaival!
- kézzel:
 - `\begin{refsection}`, `\end{refsection}` környezet
 - `\newrefsection` parancs: `\section`-höz hasonlóan, elkezd egy refsection-t, ami a következő refsection kezdetéig tart
 - automatikusan a dokumentum végéig tart az utolsó – `\endrefsection` paranccsal korábban lezárható

Irodalomjegyzék több részletben III

- egy refsection-ön belül a `\printbibliography` csak az adott refsection hivatkozásait listázza
- kiírás sorszám szerint, ami máshol is működik:
`\printbibliography[section=2]`
 - vigyázat, a refsection sorszáma szerint – ha a dokumentum elején van egy tartalomjegyzék számozatlan section-ben/chapter-ben, rögtön elcsúszott!
 - heading stílusok rész-irodalomjegyzékhez, az irodalomjegyzék mintájára: `subbibliography`, `subbibintoc`, `subbibnumbered`

Több részletes irodalomjegyzék – haladó I

- ha az összes szakaszé kell: `\bibbysection` parancs, végigiterál az összes `refsection`-ön (az üreseket kihagyva)
- ehhez valami általános címsor kell, vagy muszáj automatizálni
- példa az automatizálásra:

```
% preambulumb  
\usepackage[... ,refsection=chapter,...]{biblatex}  
\defbibheading{subbib}{\section*{%  
\ref{refsection:\therefsection}.\ fejezet}}  
% dokumentumtörzs  
\printbibheading % fő címsor  
\bibbysection{heading=subbib} % címsor és  
% tartalomjegyzék minden refsection-re  
% - ami itt chapter
```

- `\defbibheading`: saját címsor sablon definiálása

Több részletes irodalomjegyzék – haladó II

- subbib: séma neve – erre hivatkozunk később
- `\section*`: csak formázás
- `\therefsection` a változó, ami a `refsection` sorszámokon fut végig a `\bibbysection` parancsban
- `refsection:\therefsection`: ez pont a `refsection`-nel automatikusan elhelyezett label
- ha a fenti label-t beillesztjük a `\ref`-be, a legközelebbi szakaszolási egység sorszámát adja vissza – jelen esetben egy chapter-ét, amivel automatikusan generáltuk az új `refsection`-t
- `nameref` csomaggal `\ref` lecserélhető `\nameref`-re, ami a chapter *címét* illeszti be!
 - nyilván a köritő szöveget is változtatni kell

Több részletes irodalomjegyzék – haladó III

- további tagolás: resection tovább bontható úgynevezett refsegment-ekre
 - automatikus label:
refsegment: **sectionsorszám**segmentsorszám
 - hasonló parancsok és szintaxis...
 - `\begin{refsegment}`, `\end{refsegment}`
 - `\newrefsegment`, `\endrefsegment`
 - `\printbibliography[segment=1]`
 - `\usepackage[... ,resection=chapter,%
refsegment=section,...]{biblatex}`
 - együtt használva a refsegment kisebb egységet kapjon!
 - `\defbibheading{bysegment}{\section*{Források: ~%
\nameref{refsegment:\therefsection\therefsegment}}}`
 - `\bibbysegment[heading=bysegment]`

- 3 Tárgymutató (index)
 - Mi az a tárgymutató?
 - Áttekintés
 - Fordítás
 - Tárgymutató egyéb funkciói
 - Integráció
 - Személyre szabás – imakeidx

Tárgymutató/névmutató L^AT_EX-kel

- tárgymutató készítésének bekapcsolása
 - preambulumba
 - `\usepackage{makeidx}`
 - `\makeindex`
 - ahol listáztatni szeretnénk (tipikusan valahol a dokumentum végén): `\printindex`
- milyen fogalmakat listázzon?
 - a dokumentumban (`\label`-höz hasonlóan) nekünk kell jelölni a listázandó fogalmat ahol előfordul (az adott oldalra fog hivatkozni)
 - `\index{fogalom}` paranccsal (ld. 4 szakasz)
- feldolgozáshoz `makeindex` motor (compiler)
 - ez is többlépcsős fordítás lesz (ld. 3 szakasz)

Fordítás tárgymutatóval

Overleaf-ben automatikusan kezeli a lépéseket, elég egy recompile!
T_EXmaker, **T_EXStudio**:

- 1 szokásos `pdflatex`: `\index` parancsok kigyűjtése `.idx` segéd fájlba
- 2 `makeindex`: `.idx` fájl tényleges feldolgozása, bejegyzések formázása, rendezése, stb. – eredménye az `.ind` segéd fájl
 - T_EXmaker: MakeIndex a legördülő menüből, vagy F12
 - T_EXStudio: Eszközök > Szójegyzék, vagy F9
- 3 szokásos `pdflatex`: a megformázott tárgymutató beillesztése a dokumentumba, ahol a `\printindex` parancsot találja – új oldal, számozatlan section vagy chapter

\index parancs szintaxisa

- egyéb funkciók: `\index{fogalom}` parancsba beépítve
- **szintaxisa eltér a megszokottól!**
- nem fogad el extra argumentumokat – mert a tartalmát *nem* a \LaTeX , hanem a `makeindex` segédprogram dolgozza fel!
- azaz, minden infót ebben az egy argumentumban kell átadni
- az argumentum tartalma nem (feltétlen) csak a listázandó fogalom, hanem egy sztring, „vezérlő” karakterekkel!
 - speciális karakterek: @, !, |
 - kírásuk idézőjellel escape-elve "@, "!, "|
 - az idézőjelet magát is escape-elni kell(!) ""
- szóközöket tud kezelni
- nem kell, hogy a fogalom egyedi legyen – több oldalszám is tartozhat hozzá

Rendezés VS ékezetek és formázás

- ékezetes betűkre nem dob hibát, de nem tudja őket ábécé rendbe tenni (babel-lel sem!)
- parancsok, formázás esetén szintén baj van az ábécé renddel
- @ speciális karakterrel külön megadható a **formázatlan, ékezet nélküli rendezendő verzió** és a **formázott és/vagy ékezetes megjelenítendő verzió**:

```
\index{rendezendo@\textit{megjelenítendő}}
```

- a *teljes sztring* = `rendezendo@megjelenítendő` alapján azonosít egy bejegyzést
- azaz ha többször szerepel ugyanaz a fogalom, figyeljünk rá, hogy egyezzen meg a *teljes sztring*! különben több különböző bejegyzést készít

Al-bejegyzések I

- a **fogalom** alatt néha szeretnénk felsorolni a **tulajdonságait**
- vagy a fogalom egy **kategória**, és szeretnénk alatta felsorolni az **elemeit** vagy részkategóriáit
- pl. listák kulcsszó alatt szeretnénk megjeleníteni az itemize, enumerate kulcsszavakat
- ! speciális karakterrel:

```
\index{listak!itemize}
```

```
\index{listak!enumerate}
```

Al-bejegyzések II

- a **!** kétszer is használható, pl. ha enumerate kulcsszó alatt szeretnénk utalni a számozásra is:

```
\index{listak!enumerate!szamozas}
```

- nem szükséges a korábbi szinteknek korábban szerepelniük, vagy egyáltalán bárhol ezen a parancson kívül, így is listába veszi őket!
- kell **mindhárom szint**, akkor is, ha a listák>enumerate al-bejegyzés már létezik(!)
- rossz példa:

```
\index{enumerate!szamozas}
```

így létrehoz egy új „enumerate” bejegyzést az első szinten, és második szinten a „szamozas” al-bejegyzést

- miért? neki egyértelmű azonosítás kell, megengedi, hogy egy kulcsszó több helyen is szerepeljen al-bejegyzésként vagy éppen a saját bejegyzéseként

Al-bejegyzések III

- kombinálható a @ jellel, bejegyzés és al-bejegyzés (és al-al-bejegyzés) külön-külön formázhatók:

```
\index{listak@listák!itemize@\texttt{itemize}}
```
- itt is figyeljünk, bejegyzést és al-bejegyzést is *teljes sztringje* alapján, @ „argumentummal” együtt azonosít – ha ugyanarra a bejegyzésre vonatkozik, egyezzen meg a *teljes sztring!*
- **rossz** példa:

```
\index{listak@listák}  
\index{listak!itemize}
```

létrehoz egy „listák” bejegyzést, és egy *másik* „listak” bejegyzést „itemize” al-bejegyzéssel!

Oldalszámok I

- az oldalszám beillesztése automatikus, az `\index` parancs helyéből generálja
- minden, ami az oldalszámra vonatkozik: `\index` parancs argumentumának **végére(!)**, `|` karakter után
- oldalszám formázása: `|` után elhelyezett `\` nélküli(!) paranccsal, pl. vastag betűs oldalszám:

```
\index{listak!enumerate|textbf}
```

- nem halmozható a formázás
- további parancsokat figyelmen kívül hagy, vagy csak verbatim bemásol

Oldalszámok II

- ha egy fogalom többször szerepel, a szereplő oldalszámokat vesszővel elválasztva sorolja fel
- oldaltartomány is hivatkozható, elejére:
`\index{listak| (}`
végére:
vége: `\index{listak|)}`
- tartomány oldalszámainak formázása: nyitó paranccsal együtt
`\index{listak| (textbf}`
- záró parancshoz írt formázást figyelmen kívül hagy; erre is a nyitó parancs formázása érvényes

Lásd és lásd még

- „lásd” ill. „lásd még” utalások célja: átirányítás egy szinonimához, **felváltja** az oldalszámot(!)
- utalás elhelyezése: see (lásd) vagy seealso (lásd még) kulcsszóval, pl.

```
\index{itemize|see{listák, \texttt{itemize}}}
```

- a babel lefordítja majd a kulcsszót
- tegyük kapcsos zárójelbe az utalást!
- az utalás tetszőleges szöveg, a makeindex nem dolgozza fel(!)
- nem ellenőrzi, hogy létező bejegyzésre utalunk-e, és nem hozza létre automatikusan!
- L^AT_EX parancsok, formázás stb. használható benne
- utalás al-bejegyzésre: tetszőleges formátum, a példában vessző, de lehet / vagy > jel is

Integráció

- `showidx` csomag: margón jelzi, hogy hol és milyen `\index` parancsokat definiáltunk – csak vázlat/szerkesztés alatt álló dokumentumban használjuk
- integráció `hyperref` csomaggal
 - `hyperref` betöltésével alapértelmezett módon a tárgymutató oldalhivatkozásai kattintható hiperlinkekké alakulnak
 - ekvivalens `\usepackage[hyperindex=true]{hyperref}` opcióval
 - kikapcsolás: `hyperindex=false` csomag opció

Személyre szabás I

- imakeidx csomaggal (dokumentáció link) lehetséges több oszlopos elrendezés, témánként csoportosított tárgymutató, stb.
- makeidx csomag *helyett* töltjük be, néhány új parancs, plusz `\makeindex` parancs opcionális argumentumot kap
 - `\usepackage{imakeidx}`
 - `\indexsetup{opciók}`
 - `\makeindex[opciók]`
- `\indexsetup` parancs – preambulumba, opciói:
 - `level=\section*`: milyen szakaszolási egységet használjon a címsor
 - `toclevel=section`: tartalomjegyzékben milyen szinten jelenik meg
 - `noclearpage` – nem kezd új oldalon

Személyre szabás II

- `\makeindex` néhány opciója (több: ld. a dokumentációban)
 - opció pl. `columns=2`: 2 oszlopos elrendezés
 - oszlopok távolsága, alapértelmezett `\columnsep=35pt`
 - oszlopok közt választóvonal: `columnseprule`
 - címsor átnevezése: `title=Tárgymutató`
 - `intoc`: automatikusan beveszi a tartalomjegyzékbe
- `\printindex` előtt használható:
`\indexprologue[térköz]{szöveg}`
 - **szöveg**: tárgymutató címsora és az azt követő szólista közé szűrt szöveg, leírás
 - **térköz** – opcionális: függőleges térköz a prologue és a szólista között, alapért. `\bigskip`
- egyéb formázási beállítások: ún. *stílusfájlokkal* (nem megyünk bele)

Több tárgymutató

- tárgymutató több részre szedhető, pl. tematikusan, több `\makeindex` paranccsal és indexek (tárgymutatók) elnevezésével:

```
\makeindex % alapértelmezett index
```

```
\makeindex[name=matek] % második, elnevezett index
```

- egyéb opciók (title, stb.) külön megadhatók rájuk
- `\indexsetup` is használható köztük
- `\index` parancs opcionális argumentuma határozza meg, melyik indexbe kerül:
 - `\index{valami}`: alapértelmezett indexbe
 - `\index[matek]{átmérő}`: matek nevű indexbe
- külön-külön írathatók ki `\printindex`-szel:
 - `\printindex`: alapértelmezett index kiírása
 - `\printindex[matek]`: matek nevű index kiírása

4 tocbibind

- tocbibind – bibliográfia és index a tartalomjegyzékbe

Integráció a tartalomjegyzékbe I

- emlékeztető: irodalomjegyzék és tárgymutató (maga a tartalomjegyzék is) alapértelmezésben számozatlan section (vagy chapter, ha van)
- bevételük a tartalomjegyzékbe:
 - automatikusan*: tocbibind csomaggal
 - * az alapértelmezett csomagok és módszerek, BibTeX és makeidx használatával – biblatex és imakeidx másképp működnek a színtalok mögött, nem kompatibilis
 - kézzel – általános számozatlan szakasz (emlékeztető):

```
\section*{Számozatlan}
```

```
\phantomsection
```

```
\addcontentsline{toc}{section}{Számozatlan}
```

- `\phantomsection`: csak hyperref-fel – hiperlink ugrási pontját garantálja

Integráció a tartalomjegyzékbe II

- kézzel – módosítás generált tartalomra, pl. irodalomjegyzék:

```
\clearpage %% book: \cleardoublepage
\phantomsection
\addcontentsline{toc}{section}{Hivatkozások}
%% book: section helyett chapter
\bibliographystyle{plain}
\bibliography{forrasok.bib}
```
- `\addcontentsline` és `\phantomsection` akkor mutat csak jó helyre, ha a `\clearpage` és a `\bibliography` közé kerülnek. miért?
 - az irodalomjegyzék több oldal is lehet, a végén nem jó helyen lenne
 - a `\bibliography` új oldalt kezd ha nem tesszük meg előtte kézzel, ezért `\clearpage` nélkül pedig túl koránra mutatna az oldalszám és hivatkozás!