

A firstindexletter csomag dokumentációja

Bihari Zsolt info@tengerikajak.net

v1.0 – 2026. június 23.

Kivonat

A `firstindexletter` csomag célja, hogy a `xindy` vagy `makeindex` által generált tárgymutatókban optikailag kiemelve minden betűcsoport legelső elemének kezdőbetűjét. Ezzel a terjedelmes jegyzék kevesebb oldalra tördelhető.

A csomag rugalmas formázási lehetőségeket (méret, szín, fontcsalád) biztosít, és beépített, háromszintű betűpár-egalizálási (kerning) logikával védi ki a nagy méretkülönbségekből adódó tipográfiai (és optikai) ütközéseket.

Tartalomjegyzék

		6.1. A csomag fejléce és függőségei	5
		6.2. A <code>\makeatletter</code> szerepe . . .	5
1. Bevezetés	2	6.3. A csomag bekapcsolása és a <code>lettergroup</code> feldolgozása . .	6
2. Használat és követelmények	2	6.4. A megjelenés alapértelmezett paraméterei	6
2.1. Használati javaslat magyar dokumentumokhoz	2	6.5. A kerningrendszer alapötlete .	7
2.2. Overleaf használata	3	6.6. Csoportkezdés felismerése . .	8
3. Kinézet testreszabása	3	6.7. A <code>\fil@grabfirst</code> makró részletes működése	8
3.1. Betűméret	3	6.8. A <code>theindex</code> környezet átdefiniálása	9
3.2. Betűtípus	3	6.9. Az <code>\item</code> lokális újradefiniálása	10
3.3. Szín	3	6.10. Miért nem a <code>.ind</code> fájl írja át a csomag?	10
4. Tipográfiai egalizálás (Kerning)	4	6.11. Használati javaslat magyar dokumentumokhoz	11
4.1. Kivételista (Magas szárú betűk)	4	6.12. A matematikai és speciális indexbejegyzések kérdése . . .	11
4.2. Első csoport: Legnagyobb behúzás	4	6.13. Mit lehet biztonsággal testreszabni?	11
4.3. Második csoport: Közepes behúzás	4	6.14. Technikai korlátok és sajátosságok	12
4.4. Harmadik csoport: Kis behúzás	4	6.15. Záró megjegyzés	12
5. Működési elv (Technikai háttér)	5		
6. Teljes referencia	5	7. Példafejezet	12

1. Bevezetés

Hagyományos tárgymutatók készítésekor a `makeindex` vagy `texindy` motorok gyakran tesznek egy úgynevezett `lettergroup` fejléct (pl. **A**, **B**... **Z**) a betűcsoportok elé.

Esetünkben azonban nem a \TeX rendszerben megszokott különálló betűkből álló fejléct, hanem az adott csoport legelső bejegyzésének megnövelt, kiemelt kezdőbetűjét valósítjuk meg. Ennek manuális szerkesztése az `.ind` fájlban hosszas és hibalehetőségekkel teli folyamat. A `firstindexletter` csomag ezt a folyamatot teljesen automatizálja.

2. Használat és követelmények

A csomag betöltése önmagában még nem változtatja meg az indexet. A speciális kezdőbetűs indexkép bekapcsolásához az alábbi két sor szükséges:

```
\usepackage{firstindexletter}
\FirstIndexLetterEnable
```

A csomag automatikusan betölti az `imakeidx` és az `xcolor` csomagokat. Fontos, hogy a dokumentumban a tárgymutató generálásakor az `\indexspace` parancsnak (amit a generáló motorok használnak a betűcsoportok elválasztására) benne kell lennie a generált `.ind` fájlban, mert a csomag ez alapján ismeri fel az új csoportok kezdetét.

2.1. Használati javaslat magyar dokumentumokhoz

Magyar dokumentumokban a csomagot célszerű a `texindy` magyar rendezési szabályaival együtt használni. A minimális beállítás:

```
\usepackage{firstindexletter}
\FirstIndexLetterEnable
\makeindex[
  name=indexfile,
  title=Index,
  columns=3,
  noautomatic
]
```

A `\FirstIndexLetterEnable` kapcsolja be a csomag speciális működését. Enélkül az index a generált `.ind` fájl eredeti szerkezete szerint jelenik meg.

A magyar többjegyű kezdőbetűk közül a csomag a következőket mindig önálló iniciáléként kezeli:

Cs, Gy, Ly, Ny, Sz, Ty, Zs

A `Dz` és a `Dzs` kezelése külön szabályozható. Alapértelmezésben a `Dz` és `Dzs` kezdetű bejegyzések a `D` csoporthoz fűződnek.

Ha a `Dz` külön csoportként jelenjen meg:

```
\FirstIndexLetterSeparatedZ
```

Ha a `Dzs` is önálló csoport legyen:

```
\FirstIndexLetterSeparatedDzs
```

A `\FirstIndexLetterSeparatedDzs` automatikusan bekapcsolja a `Dz` külön kezelését is.

2.2. Overleaf használata

Overleaf alatt a dokumentum `pdflatex` fordítóval lefordítható, de a `texindy` futtatása nem minden projektbeállítás mellett történik meg ugyanúgy, mint helyi gépen. Ez különösen akkor fontos, ha a magyar rendezéshez saját `.xdy` modul vagy külön `texindy` parancssori beállítás szükséges.

A jelen dokumentum indexe helyi gépen az alábbi paranccsal készül:

```
pdflatex manual-hu.tex
texindy -L hungarian -M hungarian-utf8 helyseg.idx
pdflatex manual-hu.tex
```

Overleaf használatakor a biztos módszer az, hogy a `helyseg.ind` fájlt helyi gépen generáljuk le, majd ezt a kész fájlt feltöltjük az Overleaf projektbe. Ezután az Overleaf már nem az index rendezését végzi el, hanem a kész `.ind` fájlt olvassa be a `\printindex` parancsnál.

Ez azért előnyös, mert így pontosan ugyanaz az indexszerkezet kerül a dokumentumba Overleafen is, mint helyi fordításkor. Különösen fontos ez a `\lettergroup{...}` sorok miatt, mivel a `firstindexletter` csomag ezekre is támaszkodik a betűcsoportok tipográfiai kezelésénél.

3. Kinézet testreszabása

Az első betű fizikai megjelenését három `\renewcommand` segítségével állíthatjuk be a dokumentum preambulumban.

3.1. Betűméret

A kezdőbetű méretének beállításához az `\IndexInitialSize` parancsot használjuk.

Alapértelmezett értéke `\LARGE`.

```
\renewcommand\IndexInitialSize{\Huge}
```

3.2. Betűtípus

A kezdőbetű családját az `\IndexInitialFamily` makró tárolja. Itt használhatunk beépített rövidítéseket, de hivatkozhatunk külső betűkészletekre is.

Alapértelmezés: `\fontfamily{cmr}\selectfont`.

```
% Pelda vastag talpatlan (sans-serif) betűkészletre:
\renewcommand\IndexInitialFamily{\sffamily\bfseries}
```

3.3. Szín

A szín beállításához az `\IndexInitialColor` parancsot írjuk át, amely az `xcolor` csomag színneveit fogadja. Alapértelmezés: `black`. Jelen leírásban kékre álltunk:

```
\renewcommand\IndexInitialColor{blue}
```

4. Tipográfiai egalizálás (Kerning)

Mivel a kiemelt kezdőbetű lényegesen nagyobb a szót folytató karaktereknél, egyes betűk (különösen az **F**, **P**, **T**, **V**, **W**, **Y**) esetében optikai űr keletkezhet a kezdőbetű és a második betű között. Ennek kiküszöbölésére a csomag háromszintű (nagy, közepes, kis behúzás) automatikus kerning-mechanizmussal rendelkezik.

További tipográfiai probléma, hogy ha a második betű magas szárú, (pl. *h*, *l*, *b*, *t*), a negatív távolság miatt fizikailag is összeakadhat a nagy kezdőbetű kalapjával (pl. *Thira*, *Florence*). A csomag ezért külön megadható távolságot kezel az ilyen betűpárookra.

4.1. Kivételista (Magas szárú betűk)

Ide soroljuk azokat a kisbetűket, melyek felfelé kinyúlnak.

```
\renewcommand\IndexKerningAscenders{hbkdti}
```

Ez az angolszász betűktől eltérő (pl. görög, cirill) esetekben megváltoztatható.

Példa a *h* és az *l* betűkre, amikor eltérő a balra tolás mértéke:



4.2. Első csoport: Legnagyobb behúzás

Jellemzően a **T** és az **Y** betűket soroljuk ide, mivel ezek alatt hatalmas optikai űr tátong.

```
\renewcommand\IndexKerningGroupOneLetters{TY}
\renewcommand\IndexKerningGroupOneValue{-.5ex}
\renewcommand\IndexKerningGroupOneAscenderValue{-0.2ex}
```

4.3. Második csoport: Közepes behúzás

Ide a **V** betű és társai illenek.

```
\renewcommand\IndexKerningGroupTwoLetters{V}
\renewcommand\IndexKerningGroupTwoValue{-.4ex}
\renewcommand\IndexKerningGroupTwoAscenderValue{-0.1ex}
```

4.4. Harmadik csoport: Kis behúzás

Ideális a **P**, **F** és a **W** betűkhöz.

```
\renewcommand\IndexKerningGroupThreeLetters{PFW}
\renewcommand\IndexKerningGroupThreeValue{-0.3ex}
\renewcommand\IndexKerningGroupThreeAscenderValue{0ex}
```

5. Működési elv (Technikai háttér)

A csomag futásidőben újraértelmezi az `\item` makrót a `theindex` környezetben belül. Miután érzékeli a környezet kezdetét, vagy egy `\indexspace` parancsot, az elkövetkező `\item` esetében izolálja az `\expandafter` segítségével a szó legelső (#1), illetve második (#2) karakterét. Az első karaktert beburkolja a formázó parancsokba, majd az `\in@` operátorral egymás után megvizsgálja, melyik kerning csoportba tartozik, és ehhez mérten `\hspace` paranccsal igazítja hozzá a második (#2) karaktert. A stíluslapba beépített `LATEX` makrókkal ez a módszer sokkal kényelmesebb, mintha BASH-ba kilépve `sed` vagy `awk` scriptekkel írnánk át az `.ind` fájl fizikai tartalmát. Ráadásul a megoldás platformfüggetlen.

6. Teljes referencia

Ez a fejezet a `firstindexletter.sty` teljes működését soronkénti logikában, de még hanglehető emberi nyelven ismerteti. A cél nem csupán az, hogy a csomagot használni lehessen, hanem az is, hogy később biztonsággal módosítható, karbantartható és továbbfejleszthető maradjon.

A csomag alapötlete az, hogy a tárgymutató minden betűcsoportjában az *első tényleges bejegyzés* kezdőbetűje optikailag nagyobb legyen, mint a többi szó. A csomag nem a forrás `\index{}` bejegyzéseit alakítja át, és nem is a generált `.ind` fájlt írja át külső programmal, hanem a `LaTeX` indexkörnyezetének futásidejű viselkedését módosítja.

6.1. A csomag fejléce és függőségei

A csomag eleje:

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{firstindexletter}[2026/05/31 First item initial
in each index group]

\RequirePackage{imakeidx}
\RequirePackage{xcolor}
```

A `\NeedsTeXFormat{LaTeX2e}` sor azt jelzi, hogy a csomag `LaTeX2e` környezetet igényel. Ez ma gyakorlatilag szabványos fejléc.

A `\ProvidesPackage` sor két dolgot szolgáltat: a csomag nevét és egy rövid azonosító leírást. Ez egyrészt dokumentációs célú, másrészt a logfájlban is megjelenik, így könnyebb azonosítani, hogy a fordítás során pontosan melyik csomagverzió töltődött be.

A csomag automatikusan betölti az `imakeidx` és az `xcolor` csomagokat. Az `imakeidx` szükséges az index deklarálásához és nyomtatásához, az `xcolor` pedig a nagyított kezdőbetű színezéséhez kell.

6.2. A `\makeatletter` szerepe

A következő sor:

```
\makeatletter
```

lehetővé teszi, hogy a csomag olyan belső nevű makrókat definiáljon és használjon, amelyek `@` karaktert tartalmaznak. A `LaTeX`-ben ez bevett technika a „belső”, csomagszintű

makrók elkülönítésére. A felhasználó számára szánt makrók rendszerint nem tartalmaznak @-t, a belső segédmakrók viszont igen.

6.3. A csomag bekapcsolása és a lettergroup feldolgozása

A következő blokk:

```
\newif\iffil@enabled
\fil@enabledfalse

\newcommand{\FirstIndexLetterEnable}{%
  \global\fil@enabledtrue
}
```

egy logikai kapcsolót vezet be. A `\newif\iffil@enabled` létrehoz egy új feltételes változót, amelynek két állapota lehet: igaz vagy hamis.

Alapértelmezésben a csomag ezt hamisra állítja:

```
\fil@enabledfalse
```

tehát önmagában a csomag még nem módosítja az index viselkedését.

A felhasználó számára szánt parancs:

```
\newcommand{\FirstIndexLetterEnable}{%
  \global\fil@enabledtrue
}
```

globálisan igazra állítja ezt a kapcsolót. Ennek gyakorlati jelentése az, hogy ha a dokumentumban meghívjuk a

```
\FirstIndexLetterEnable
```

parancsot, akkor a csomag később a `theindex` környezetben belül átmenetileg újra-definiálja a `\lettergroup`, a `\indexspace` és a `\item` makrókat. A `\lettergroup` nem egyszerűen üres makróvá válik: a csomag az argumentumából effektív csoportnevet képez, és ez alapján dönti el, hogy valóban új betűcsoport kezdődött-e.

Ez tipikusan akkor hasznos, ha az indexet `noautomatic` módban használjuk, vagy ha nem akarunk külön csoportfejléctet, hanem annak tipográfiai szerepét a legnagyobbított első szókezdő betű veszi át. A `D`, `Dz` és `Dzs` esetében ez a feldolgozás azért különösen fontos, mert a csomag beállításától függően ezeket külön csoportként vagy azonos effektív csoport részeként kezeli.

6.4. A megjelenés alapértelmezett paraméterei

A következő blokk adja meg az alap stílusbeállításokat:

```
\providecommand\IndexInitialSize{\LARGE}
\providecommand\IndexInitialFamily{\fontfamily{cmr}\selectfont}
\providecommand\IndexInitialColor{black}
```

Itt fontos, hogy `\providecommand` szerepel, nem pedig `\newcommand`. Ennek az az értelme, hogy ha a felhasználó korábban már definiálta ezeket a makrókat, akkor a csomag nem írja felül őket. Ha még nem léteznek, akkor létrehozza őket a megadott alapértékekkel.

A három makró feladata:

- `\IndexInitialSize`: a kiemelt kezdőbetű mérete;
- `\IndexInitialFamily`: a kiemelt kezdőbetű betűcsaládja és stílusa;
- `\IndexInitialColor`: a kiemelt kezdőbetű színe.

Az alapértelmezett méret a `\LARGE`, az alapértelmezett betűcsalád a Computer Modern Roman, a szín pedig fekete.

A dokumentumban ezek szabadon átírhatók például így:

```
\renewcommand\IndexInitialSize{\Huge}
\renewcommand\IndexInitialFamily{\sffamily\bfseries}
\renewcommand\IndexInitialColor{blue}
```

6.5. A kerningrendszer alapötlete

A nagyított kezdőbetű és a szó második betűje között gyakran optikailag kellemetlen rés keletkezik. Ez főként akkor látszik, ha az első betű felső vagy átlós szerkezete nagy üres teret hagy maga után, például T, Y, V, W, P, F esetén.

Erre szolgál a következő paraméterezés:

```
\providecommand\IndexKerningAscenders{hlbkdti}

\providecommand\IndexKerningGroupOneLetters{TY}
\providecommand\IndexKerningGroupOneValue{-1.2ex}
\providecommand\IndexKerningGroupOneAscenderValue{-0.2ex}

\providecommand\IndexKerningGroupTwoLetters{VW}
\providecommand\IndexKerningGroupTwoValue{-0.8ex}
\providecommand\IndexKerningGroupTwoAscenderValue{-0.1ex}

\providecommand\IndexKerningGroupThreeLetters{PF}
\providecommand\IndexKerningGroupThreeValue{-0.4ex}
\providecommand\IndexKerningGroupThreeAscenderValue{0ex}
```

Ez a rendszer három negatív térközcsoporthoz kezel.

Kivétellista: magas szárú második betűk.

Az `\IndexKerningAscenders` azoknak a kisbetűknek a listája, amelyek magas szárúak, vagyis felfelé nyúlnak: h, l, b, k, d, t, i. Ha ilyen betű áll a nagyított kezdőbetű után, akkor az agresszív negatív behúzás könnyen összeütközést vagy csúnya túl közel kerülést okozhat. Ezért minden kerningcsoporthoz két érték tartozik: egy általános, és egy enyhébb változat ascenderes második betűhöz.

Első csoport.

Ez a csoport a legerősebb negatív térközt adja. Ide alaptól a T és Y tartozik. Ezeknél a betűknél a nagy kezdőbetű és a következő kisbetű közötti optikai rés különösen nagy.

Ha pontosan a nagy kezdőbetű után írjuk a szó további részét, így néz ki:

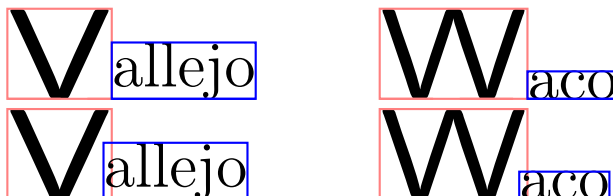


Ha a normál betűmérettel és `-.5ex` (vagyis fél „x” betűszélességgel) balra toljuk jelen csomag kihasználásával, akkor ezt az összképet kapjuk:



Második csoport.

Ez a csoport közepes erősségű negatív térközt ad. Ide alpból a V és W került. Ezeknél szintén gyakori a zavaró rés, de általában nem akkora, mint T vagy Y esetén.



Harmadik csoport.

Itt enyhébb negatív térközt használunk. Ide alpból a P és F tartozik. Ezeknél is lehet optikai hiányérzet, de kisebb mértékű.



A felhasználó ezeket mind átírhatja `\renewcommand`-dal, ahogyan a mintadokumentumban is szerepel.

6.6. Csoportkezdés felismerése

A következő sor:

```
\newif\iffil@indexgroupstart
```

egy újabb logikai kapcsolót hoz létre. Ez fogja jelezni, hogy éppen egy index-betűcsoport első eleme következik-e.

A csomag nem próbálja meg a betűrendet tartalmilag értelmezni. Nem nézi, hogy a következő szó milyen betűvel kezdődik, és nem végez szótári összehasonlítást sem. Ehelyett a generált index szerkezetét figyeli: amikor a `theindex` környezet elindul, vagy amikor `\indexspace` jelenik meg, a csomag úgy tekinti, hogy a következő `\item` lesz a következő betűcsoport első eleme.

Ez egyszerű, gyors és stabil megoldás, de fontos feltétele, hogy a generált `.ind` fájl valóban használja az `\indexspace` parancsot a csoportok elválasztására.

6.7. A `\fil@grabfirst` makró részletes működése

A csomag kulcsfontosságú makrója a `\fil@grabfirst`. Ez veszi át a következő `\item` tartalmából a karaktereket. Az új architektúrában a klasszikus, egymásba ágyazott `\if-else` láncolatok helyett egy modernebb, robusztusabb megoldást használunk, amely elkerüli a TeX token-beolvasási (*expansion*) hibáit.

1. A makró első körben lementi az első két betűt egy belső makróba (`\fil@currentpair`).

2. Ezt követően makró-összehasonlítással (`\ifx`) megvizsgálja a magyar kétjegyű más-salhangzókat. Például létrehoz egy `\fil@checkcs{Cs}` makrót, és összehasonlítja a beolvasottal.
3. Ha egyezést talál, meghívja a `\fil@doublepair` formázást.
4. **Kivétel a Dzs esetén:**
Ha a vizsgált két betű `Dz`, a makró még nem véglegesíti a formázást, hanem az `\expandafter` parancs segítségével meghívja a `\fil@grabthirdforDzs` segédmakrót.
5. Ez a segédmakró beolvassa a következő, harmadik karaktert. Ha az `s`, akkor alkalmazza a `\fil@triplepair{D}{z}{s}` parancsot. Ha valami más következik, akkor egy egyszerű `\fil@doublepairDz` formázást alkalmaz, vagyis az adott szó elejét `Dz`-ként kezeli, nem pedig `Dzs`-ként.

Ezzel a letisztult makró-összehasonlításos módszerrel a csomag kezeli a magyar egyjegyű, kétjegyű és a `Dzs` esetében háromjegyű kezdőbetűket is.

6.8. A theindex környezet átdefiniálása

A csomag egyik legfontosabb része a következő blokk:

```
\let\fil@oldtheindex\theindex
\renewcommand\theindex{...}
```

Itt a csomag előbb elmenti a `theindex` eredeti definícióját `\fil@oldtheindex` néven, majd újradefiniálja a környezetet.

Ez azért jó megoldás, mert nem akarja nulláról újraírni az index teljes tördelését. Ehelyett előbb meghívja az eredeti `theindex`-et, és csak utána végez lokális beavatkozást a környezeten belül.

Az új definíció négy fontos dolgot csinál.

1. A csomag bekapcsolt állapotának ellenőrzése.

A környezet elején a csomag megvizsgálja az `\iffil@enabled` kapcsolót. Ha a felhasználó nem hívta meg a `\FirstIndexLetterEnable` parancsot, akkor a csomag nem avatkozik be az index feldolgozásába.

2. A lettergroup feldolgozása

Bekapcsolt állapotban a csomag a `\lettergroup` parancsot nem pusztán üres makróvá teszi, hanem a csoportnevet átadja a `\fil@handlelettergroup` segédmakrónak. Ez a segédmakró dönti el, hogy a kapott csoportnév valóban új effektív csoportot jelent-e. Így kezelhető az is, hogy a `Dz` és a `Dzs` külön betűként jelenjen-e meg, vagy a `D`, illetve `Dz` effektív csoportjához tartozzon.

3. Az első elem jelölése.

Amikor a `\lettergroup` valódi új effektív csoportot jelez, a csomag globálisan beállítja, hogy a következő `\item` egy új csoport első eleme lesz.

4. Az `\indexspace` újradefiniálása.

A csomag elmenti az eredeti `\indexspace` definícióját, majd újradefiniálja úgy, hogy az eredeti térközt ne azonnal szűrje be, hanem előbb csak egy függő állapotot állítson be a `\fil@pendingindexspace` kapcsolón keresztül.

Ez nagyon fontos: amikor az `indexmotor` egy új betűcsoport előtt `\indexspace`-et szűr be, a csomag még megvárja a következő `\lettergroup` információját. Ha az effektív csoport valóban megváltozott, akkor szükség esetén meghívja a korábban elmentett eredeti `\indexspace` parancsot; ha viszont például a `Dz` vagy `Dzs` az aktuális beállítás szerint nem külön csoport, akkor elkerülhető a felesleges térköz.

6.9. Az `\item` lokális újradefiniálása

A csomag ezután az `\item` makrót is újradefiniálja a `theindex` környezetben belül:

```
\let\fil@olditem\item
\def\item{%
  \fil@olditem
  \iffil@indexgroupstart
    \global\fil@indexgroupstartfalse
    \expandafter\fil@grabfirst
  \fi
}%
```

Ez a működés szíve.

A logika a következő:

1. Először meghívja az eredeti `\item` viselkedést, tehát az `indexlista` normál szerkezete megmarad.
2. Ezután megnézi, hogy a `\iffil@indexgroupstart` kapcsoló igaz-e.
3. Ha nem igaz, akkor semmi különös nem történik: ez nem csoportkezdő elem, tehát nincs nagy kezdőbetű.
4. Ha igaz, akkor ezt az állapotot azonnal hamisra állítja, hogy csak a csoport első eleme kapja meg a speciális kezelést.
5. Végül az `\expandafter\fil@grabfirst` hívással ráengedi a következő szövegre a kezdőbetű-fogyasztó makrót.

A `\expandafter` itt azért szükséges, mert a cél az, hogy a `\item` után következő első két karakter ténylegesen paraméterként bekerüljön a `\fil@grabfirst` makróba. Ez finom TeX-szintű technika, és ezért is nehéz a háromjegyű `Dzs` kezelése: a kétkarakteres elfogyasztás itt még stabil, a háromkarakteres viszont már sokkal kényesebb.

6.10. Miért nem a `.ind` fájlt írja át a csomag?

Felmerülhet, hogy egyszerűbb lenne a generált `.ind` fájlt egy külső szkripttel, például `sed`, `awk` vagy Python segítségével módosítani. A csomag mégsem ezt teszi.

Ennek okai:

- a makrós megoldás platformfüggetlenebb;

- nem igényel külön shell-szintű utófeldolgozást;
- egyetlen `.sty` fájlba zárható;
- jobban együttműködik online rendszerekkel, amennyiben a szükséges indexgenerálás maga elérhető;
- a dokumentum tipográfiai paraméterei közvetlenül \LaTeX -ből vezérelhetők.

A hátránya az, hogy a \TeX tokenfogyasztási modellje miatt bizonyos többkarakteres nyelvi egységek kezelése bonyolultabb megoldást igényel.

6.11. Használati javaslat magyar dokumentumokhoz

Magyar szövegben, ahol kétjegyű kezdőbetűk is előfordulnak, célszerű az indexet így deklarálni:

```
\usepackage{firstindexletter}
\FirstIndexLetterEnable
\makeindex[name=helyseg,title=Index,columns=3,noautomatic]
```

Ennek oka, hogy a külön lettergroup-fejlécek és a megnagyobbított első szókezdő betű együtt rendszerint redundáns tipográfiai hatást adnak. A `noautomatic` opció és a `\FirstIndexLetterEnable` együtt tiszta, visszafogott, könyvszerű indexképet eredményez.

6.12. A matematikai és speciális indexbejegyzések kérdése

A mintadokumentumban látható, hogy bizonyos matematikai alakokkal vegyes indexkulcsok problémásak lehetnek, különösen akkor, ha a rendezési kulcs és a nyomtatandó alak nem tiszta szöveggént van elválasztva. A stabil gyakorlat az, hogy a rendezési kulcs legyen egyszerű, jól rendezhető szöveg, és csak a nyomtatandó részben szerepeljenek matematikai vagy különleges formázások.

Ezért működőképes például az ilyen szerkezet:

```
\index[helyseg]{deviation kszi@ $(D^2\text{\textbf{x}}i)$ }
```

míg a túl agresszíven formázott vagy nem kellően „tiszta” rendezési kulcs könnyen hibához vagy csúnya indexkimenethez vezethet.

6.13. Mit lehet biztonsággal testre szabni?

A csomag jelenlegi állapotában biztonságosan módosítható:

- a kiemelt kezdőbetű mérete;
- a kiemelt kezdőbetű betűcsaládja és stílusa;
- a kiemelt kezdőbetű színe;
- az ascender-lista;
- az 1., 2. és 3. kerningcsoport betűlistája;

- az ezekhez tartozó normál és ascenderes negatív térközértékek;
- a Dz és Dzs külön vagy összevont effektív csoportként való kezelése.

Nem tekinthető biztonságosan módosíthatónak komolyabb TeX-ismeret nélkül:

- a `\fil@grabfirst` paraméterszáma;
- a `\item` újradefiniálási logikája;
- a `\expandafter` használata;
- a két- és háromjegyű magyar kezdőbetűk kezelésének mély szerkezete.

6.14. Technikai korlátok és sajátosságok

1. A működés feltételezi, hogy az indexgenerátor (`makeindex` vagy `xindy`) a `\indexspace` paranccsal jelzi a betűcsoportok határát.
2. A kerning nem igazi, fontszintű OpenType/TrueType kerning, hanem egy kézzel szabályozható, statikus optikai korrekció (negatív `\hspace`).
3. A betűcsoport-kezelés továbbra is a generált index szerkezetére (`\indexspace`) épül, nem pedig nyelvi vagy szemantikai elemzésre.
4. A csomag elsődlegesen latin betűs, hagyományos indexstruktúrákra készült.

6.15. Záró megjegyzés

Jelen csomag tipográfiai célú megoldást ad arra, hogy a tárgymutató betűcsoportjai külön fejléc helyett vizuálisan hangsúlyos első szókezdő betűvel induljanak. A jelenlegi megoldás kis méretű, egyszerűen beépíthető, és optikai korrekciója jól hangolható.

7. Példafejezet

Az alábbi szójegyzék jelen csomag indexjegyzékének érdekében lett elkészítve. A szavak az alábbi módszerrel lettek listázva:

```
Amsterdam\index[helyseg]{Amsterdam}
Austin\index[helyseg]{Austin}
...
```

A `texindy` programot a magyar névsorba rendezés szabályai szerint így futtatjuk:

```
pdflatex dokumentum.tex
texindy -L hungarian -M hungarian-utf8 helyseg.idx
pdflatex dokumentum.tex
```

... ahol előbb legeneráltuk az alapidokumentumot, majd a keletkező `.idx` fájlból az `.ind` fájlt, végül csatoltuk a formázott indexfájlt az alapidokumentumhoz – ami már a pdf-ben is megjelenik.

Amsterdam Austin Atlanta Athens Accra Adelaide Albany Anchorage Boston Baltimore
 Brisbane Belfast Berlin Birmingham Buffalo Boise Chicago Cleveland Cairo Calgary
 Colombo Cincinnati Columbus Dallas Denver Detroit Dublin Dakar Darwin
 Dayton Dover Edmonton Eugene El Paso Erie Evansville Everett Entebbe Europe Frank-
 furt Florence Fargo Fresno Flint Fremont Fukuoka Freetown Geneva Glasgow Galveston
 Gary Genoa Georgetown Glendale Guangzhou Houston Honolulu Halifax Havana Hart-
 ford Helsinki Hobart Hanoi Indianapolis Islamabad Irvine Ithaca Incheon Ibadan Iquique
 Izmir Jacksonville Juneau Jakarta Johannesburg Jeddah Jersey Joliet Juba Kansas Kyoto
 Kingston Knoxville Kampala Kobe Karachi Kabul London Lisbon Lagos Lima Lincoln
 Lansing Laredo Lubbock Miami Memphis Milwaukee Manila Melbourne Mumbai Mad-
 ison Mobile Nairobi Naples Nashville Newark Norfolk Norman Nagoya Niamey Orlando
 Omaha Oakland Osaka Oslo Odessa Ostrava Oxford Paris Portland Phoenix Philadel-
 phia Pittsburgh Perth Pretoria Peoria Queens Quito Quebec Quanzhou Qingdao Quetta
 Quilmes Quimper Rome Reno Richmond Raleigh Riverside Rochester Rockford Roswell
 Seattle Sydney Seoul Savannah Spokane Syracuse Springfield Tokyo Toronto Tampa Tu-
 cson Tulsa Toledo Tacoma Utah Utica Uppsala Ulaanbaatar Ulsan Ufa Ushuaia Vancou-
 ver Venice Vienna Victoria Vallejo Verona Varna Washington Wichita Waco Wilmington
 Winston Warsaw Wuhan Xiamen Xuzhou Xining Xian Xanthus Xalapa Xingu York Yoko-
 hama Yonkers Yakima Yantai Yuma Ypsilanti Zurich Zagreb Zhengzhou Zaria Zamboanga
 Zinder Zibo

Zibo Zinder Zamboanga Zaria Zhengzhou Zagreb Zurich Ypsilanti Yuma Yantai Yak-
 ima Yonkers Yokohama York Xingu Xalapa Xanthus Xian Xining Xuzhou Xiamen Wuhan
 Warsaw Winston Wilmington Waco Wichita Washington Varna Verona Vallejo Victoria
 Vienna Venice Vancouver Ushuaia Ufa Ulsan Ulaanbaatar Uppsala Utica Utah Taco-
 ma Toledo Tulsa Tucson Tampa Toronto Tokyo Springfield Syracuse Spokane Savannah
 Seoul Sydney Seattle Roswell Rockford Rochester Riverside Raleigh Richmond Reno Ro-
 me Quimper Quilmes Quetta Qingdao Quanzhou Quebec Quito Queens Peoria Pretoria
 Perth Pittsburgh Philadelphia Phoenix Portland Paris Oxford Ostrava Odessa Oslo Osa-
 ka Oakland Omaha Orlando Niamey Nagoya Norman Norfolk Newark Nashville Naples
 Nairobi Mobile Madison Mumbai Melbourne Manila Milwaukee Memphis Miami Lubbock
 Laredo Lansing Lincoln Lima Lagos Lisbon London Kabul Karachi Kobe Kampala Knox-
 ville Kingston Kyoto Kansas Juba Joliet Jersey Jeddah Johannesburg Jakarta Juneau
 Jacksonville Izmir Iquique Ibadan Incheon Ithaca Irvine Islamabad Indianapolis Hanoi
 Hobart Helsinki Hartford Havana Halifax Honolulu Houston Guangzhou Glendale Geor-
 getown Genoa Gary Galveston Glasgow Geneva Freetown Fukuoka Fremont Flint Fresno
 Fargo Florence Frankfurt Europe Entebbe Everett Evansville Erie El Paso Eugene Ed-
 monton Dover Dayton Darwin Dakar Dublin Detroit Denver Dallas Columbus Cincinnati
 Charlotte Colombo Calgary Cairo Cleveland Chicago Boise Buffalo Birmingham Berlin
 Belfast Brisbane Baltimore Boston Anchorage Albany Adelaide Accra Athens Atlanta
 Austin Amsterdam

Amsterdam Atlanta Accra Albany Boston Brisbane Berlin Buffalo Chicago Cairo Co-
 lombo Cincinnati Dallas Detroit Dakar Dayton Edmonton El Paso Evansville Entebbe
 Frankfurt Fargo Flint Fukuoka Geneva Galveston Genoa Glendale Houston Halifax Hart-
 ford Hobart Indianapolis Irvine Incheon Iquique Jacksonville Jakarta Jeddah Joliet Kansas
 Kingston Kampala Karachi London Lagos Lincoln Laredo Miami Milwaukee Melbourne
 Madison Nairobi Nashville Norfolk Norman Orlando Oakland Oslo Ostrava Paris Pho-
 enix Pittsburgh Pretoria Queens Quebec Qingdao Quilmes Rome Richmond Riverside
 Rockford Seattle Seoul Spokane Springfield Tokyo Tampa Tulsa Tacoma Utah Uppsala

Ulsan Ushuaia Vancouver Vienna Vallejo Varna Washington Waco Winston Wuhan Xiamen Xining Xanthus Xingu York Yonkers Yantai Ypsilanti Zurich Zhengzhou Zamboanga Zibo Austin Athens Adelaide Anchorage Baltimore Belfast Birmingham Boise Cleveland Calgary Charlotte Columbus Denver Dublin Darwin Dover Eugene Erie Everett Europe Florence Fresno Fremont Freetown Glasgow Gary Georgetown Guangzhou Honolulu Havana Helsinki Hanoi Islamabad Ithaca Ibadan Izmir Juneau Johannesburg Jersey Juba Kyoto Knoxville Kobe Kabul Lisbon Lima Lansing Lubbock Memphis Manila Mumbai Mobile Naples Newark Nagoya Niamey Omaha Osaka Odessa Oxford Portland Philadelphia Perth Peoria Quito Quanzhou Quetta Quimper Reno Raleigh Rochester Roswell Sydney Savannah Syracuse Toronto Tucson Toledo Topeka Tehran Vancouver Venice Victoria Verona Wichita Wilmington Warsaw Xuzhou Xian Xalapa Yokohama Yakima Yuma Zagreb Zaria Zinder

Tyukfalva Tyukmajor Tyukos Tyukod Tyukostanya Tyukospusztá

Lyukfalva Lyukos Lyukasd Lyukostanya Lyukospatak Lyukvolgy

Csongrád Csikszereda Csorna

Zsámbék Zsira Zsombolya

Szada Szarvas Szeged Szeghalom Szentendre Szekszárd Szentes Szigetvár Szolnok Szombathely

Dzsungelfalva Dzsunka-tó Dzeta-függvény-tó

A szórásnégyzet jele $D^2\xi$.

A Γ -eloszlás az egy Γ -eloszlás

Helységnevek

Accra, 13
Adelaide, 13, 14
Albany, 13
Amsterdam, 13
Anchorage, 13, 14
Athens, 13, 14
Atlanta, 13
Austin, 13, 14

Baltimore, 13, 14
Belfast, 13, 14
Berlin, 13
Birmingham, 13, 14
Boise, 13, 14
Boston, 13
Brisbane, 13
Buffalo, 13

Cairo, 13
Calgary, 13, 14
Charlotte, 13, 14
Chicago, 13
Cincinnati, 13
Cleveland, 13, 14
Colombo, 13
Columbus, 13, 14

Csikszereda, 14
Csongrád, 14
Csorna, 14

Dakar, 13
Dallas, 13
Darwin, 13, 14
Dayton, 13
Denver, 13, 14
Detroit, 13
 $D^2\xi$, 14
Dover, 13, 14
Dublin, 13, 14

Dzeta-függvény-tó,
14

Dzsungelfalva, 14
Dzsunka-tó, 14

Edmonton, 13
eloszlás
 Γ , 14
Elpaso, 13
Entebbe, 13
Erie, 13, 14
Eugene, 13, 14
Europe, 13, 14
Evansville, 13
Everett, 13, 14

Fargo, 13
Flint, 13
Florence, 13, 14
Frankfurt, 13
Freetown, 13, 14
Fremont, 13, 14
Fresno, 13, 14
Fukuoka, 13

Galveston, 13
 Γ -eloszlás
eloszlásfüggvénye, 14
Gary, 13, 14
Geneva, 13
Genoa, 13
Georgetown, 13, 14
Glasgow, 13, 14
Glendale, 13
Guangzhou, 13, 14

Halifax, 13
Hanoi, 13, 14
Hartford, 13
Havana, 13, 14
Helsinki, 13, 14
Hobart, 13
Honolulu, 13, 14
Houston, 13

Ibadan, 13, 14
Incheon, 13
Indianapolis, 13
Iquique, 13
Irvine, 13
Islamabad, 13, 14

Ithaca, 13, 14
Izmir, 13, 14

Jacksonville, 13
Jakarta, 13
Jeddah, 13
Jersey, 13, 14
Johannesburg, 13, 14
Joliet, 13
Juba, 13, 14
Juneau, 13, 14

Kabul, 13, 14
Kampala, 13
Kansas, 13
Karachi, 13
Kingston, 13
Knoxville, 13, 14
Kobe, 13, 14
Kyoto, 13, 14

Lagos, 13
Lansing, 13, 14
Laredo, 13
Lima, 13, 14
Lincoln, 13
Lisbon, 13, 14
London, 13
Lubbock, 13, 14

Lyukasd, 14
Lyukfalva, 14
Lyukos, 14
Lyukospatak, 14
Lyukostanya, 14
Lyukvolgy, 14

Madison, 13
Manila, 13, 14
Melbourne, 13
Memphis, 13, 14
Miami, 13
Milwaukee, 13
Mobile, 13, 14
Mumbai, 13, 14

Nagoya, 13, 14
Nairobi, 13
Naples, 13, 14
Nashville, 13
Newark, 13, 14
Niamey, 13, 14
Norfolk, 13
Norman, 13

Oakland, 13
Odessa, 13, 14
Omaha, 13, 14
Orlando, 13
Osaka, 13, 14
Oslo, 13
Ostrava, 13
Oxford, 13, 14

Paris, 13
Peoria, 13, 14
Perth, 13, 14
Philadelphia, 13, 14
Phoenix, 13
Pittsburgh, 13
Portland, 13, 14
Pretoria, 13

Qingdao, 13
Quanzhou, 13, 14
Quebec, 13
Queens, 13
Quetta, 13, 14
Quilmes, 13
Quimper, 13, 14
Quito, 13, 14

Raleigh, 13, 14
Reno, 13, 14
Richmond, 13
Riverside, 13
Rochester, 13, 14
Rockford, 13
Rome, 13
Roswell, 13, 14

Savannah, 13, 14
Seattle, 13
Seoul, 13
Spokane, 13
Springfield, 13
Sydney, 13, 14
Syracuse, 13, 14

Szada, 14
Szarvas, 14
Szeged, 14
Szeghalom, 14
Szekszard, 14
Szentendre, 14
Szentcs, 14
Szigetvar, 14
Szolnok, 14
Szombathely, 14

Tacoma, 13
Tampa, 13
Tehran, 14
Tokyo, 13
Toledo, 13, 14
Topeka, 14
Toronto, 13, 14
Tucson, 13, 14
Tulsa, 13

Tyukfalva, 14
Tyukmajor, 14
Tyukod, 14
Tyukos, 14
Tyukospuszt, 14
Tyukostanya, 14

Ufa, 13
Ulaanbaatar, 13
Ulsan, 13, 14
Uppsala, 13
Ushuaia, 13, 14
Utah, 13
Utica, 13

Vallejo, 13, 14
Vancouver, 13, 14
Varna, 13, 14
Venice, 13, 14
Verona, 13, 14
Victoria, 13, 14
Vienna, 13, 14

Waco, 13, 14
Warsaw, 13, 14
Washington, 13, 14
Wichita, 13, 14
Wilmington, 13, 14
Winston, 13, 14
Wuhan, 13, 14

Xalapa, 13, 14
Xanthus, 13, 14
Xiamen, 13, 14
Xian, 13, 14
Xingu, 13, 14
Xining, 13, 14
Xuzhou, 13, 14

Yakima, 13, 14
Yantai, 13, 14
Yokohama, 13, 14
Yonkers, 13, 14
York, 13, 14
Ypsilanti, 13, 14
Yuma, 13, 14

Zagreb, 13, 14
Zamboanga, 13, 14
Zaria, 13, 14
Zhengzhou, 13, 14
Zibo, 13, 14
Zinder, 13, 14
Zurich, 13, 14

Zsira, 14
Zsámék, 14
Zsombolya, 14