

Hogyan használjuk a MultEdu csomagot

(Hogyan készítsünk érdekes
és vonzó tananyagot)



Végh János



L^AT_EX



Ez a Mű a "<http://creativecommons.org/licenses/by/4.0/> Creative Commons Nevezd meg! 4.0 Nemzetközi Licenc feltételeinek megfelelően felhasználható.

© Szerzői jogok © 2011-2016
Végh János (Janos.Vegh@unideb.hu)
Minden jog fenntartva

Kizárólag újra hasznosított elektronokkal nyomtatva

Kivonat

Kurzusaim oktatásához saját makrókészletet fejlesztettem, mivel az oktatandó anyag megjelenítéséhez különböző körülmények között különböző formákra van szükség. Az elméleti anyagot az előadásokon diasorozat alapján mutatom be, és a diákhoz fűzött magyarázatokat (természetesen tömörítve) jegyzet-szerű formában a hallgatóság számára is rendelkezésre bocsátom. A hallgatóság ezt az anyagot részben kinyomtatva, részben képernyőn olvasva (akár mobil eszközökön is) tanulmányozza. A terület folyamatos fejlődése miatt a tananyag is állandó fejlesztésre szorul, ezért feltétlenül szükséges, hogy az említett megjelenési formákat egymással szinkronban lehessen fejleszteni. Ennek legegyszerűbb megvalósítási formája, hogy egyazon forrásból, megfelelő formattálási utasításokkal készítem a tananyagokat. Számítógéppel alaposan megtámogatott, nagy felbontáshoz és vonzó grafikához szokott hallgatóság számára a fenti feltételeknek megfelelő tananyagot készíteni komoly kihívás.

Közös alapként a LaTeX nyelvet használtam, amely nyelven készült forrásból az elterjedten használt Beamer prezentáció készítő makró csomaggal állítom elő az előadáson bemutatandó diákat, és a memoir könyv készítő makró csomaggal a hallgatóság számára rendelkezésre bocsátandó "tananyagot". Ez utóbbi akár az "on demand printing" minőséget is elérheti. Vonzó grafikus megjelenéssel, a szokásoshoz képest sokkal több ábrával igyekszik felkelteni az anyag a hallgatóság figyelmét (de lehet belőle kevésbé "fancy", inkább "plain" stílusú, de még mindig könyv minőségű változatot is készíteni). A jegyzet-szerű változat az előadáson bemutatott ábrákat és szöveget teljes egészében tartalmazza, az előadás szövegének egy tömörített változatával kiegészítve. Ugyanez a könyv-szerű anyag jelenik meg a képernyős olvasásra szánt WEB-es formátumban, illetve az eBook kompatibilis (natív PDF) változatban. Ebben a változatban az anyag egy-képernyőnyi darabokra van tördelve, és (főként kisebb képernyőjű mobil eszközökre gondolva) nagyobb betűkkel, egy ábra/képernyő módon jelenik meg.

A többféle, egymásnak ellentmondó megjelenítési igény természetesen csak kompromisszumokkal oldható meg, így a tananyag megírása során a szöveg megjelenés formázására több gondot és időt kell fordítani. A makrócsomaggal akár egyidejűleg idegen nyelvre is lehet ugyanazon tananyagot fejleszteni. A LaTeX lehetőségeivel akár animáció, mozgófilm, WEB-lap, hang, stb. is beépíthető, természetesen gondolni kell a nyomtatott anyag ekvivalens megjelenítésére.

First edition: August 2016



Tartalomjegyzék

Tartalomjegyzék	i
1 Általános	1
1.1. Bevezetés	1
1.2. Beüzemelés	1
1.3. Szerkezet	1
1.3.1. common	2
1.3.2. Workstuff	2
1.4. Alapértelmezett	2
1.4.1. Alapértelmezett	2
1.5. Beállítások	3
1.5.1. Beamer	3
1.6. Ötletek	3
2 Fordítás	4
2.1. Kézi	4
2.2. Kötegelt	4
2.3. Beállítások	4
2.3.1. Verziók	4
2.3.2. Nyelvek	4
3 Tagolás	6
3.1. Egységek	6
3.1.1. Dia keretek	6
3.1.2. Fejezet	6
3.1.3. Szakasz és az alatt	6
3.2. Kétnyelvű forráskódok	6
3.2.1. Átváltás a nyelvek között	6
3.2.2. Dia keretek	6
3.2.3. Fejezet	6
3.2.4. Szakasz és az alatt	6
3.3. Fejezet illusztráció	6
4 Ábrák	8
4.1. Hagyományos	8
Tárgymutató	9
Ábrák jegyzéke	10

Általános információ

1.1. Bevezetés

Kurzusaim tartásához saját tananyagot fejlesztettem, különböző megjelenési formákban; a jelen csomag ennek mellékterméke. A jó kurzusok tananyaga gyorsan fejlődik, különösen akkor, ha maga a tudományág is naponta megújul. Az informatikában évről évre változik a technológia, a statisztikák, a termékek, a segédeszközök, stb.; és már csak emiatt is minden tanévre frissíteni kell a tananyagot.

Manapság a tananyagot a hallgatóság változatos formákban igényli: előadáson nagy méretű, jól áttekinthető, kivetíthető anyagot kell használni, amely képekkel gazdagon illusztrált és az előadó számára is jó sorvezetőként szolgál. A vizsgára készüléshöz pedig arra a magyarázatra is szükség van, amit az előadó a kivetített anyaghoz élő szóban hozzáfűz. Azaz, olyan magyarázó szöveggel ellátott anyagra is szükség van, amelyet kinyomtatva, asztali gép vagy mobil eszköz képernyőjén lehet elolvasni. Esetleg ugyanazt a változatot idegen nyelven is közzétenni, külföldi hallgatók számára. Bár sokszor lehet elérhető könyvekre és megvásárolható jegyzetekre hagyatkozni, a kicsit is speciálisabb anyagok esetén ez a segédlet lesz a felkészüléshez szükséges tananyag.

A jelen makró csomag olyan, amelyet saját kurzusaim készítéséhez fejlesztettem, és igyekeztem olyanná tenni, hogy tananyag fejlesztés közben már ne kelljen a megjelenítés technikájával foglalkozni, és ilyen módon mások is tudják hasonló célra használni, ha követik a fejlesztés logikáját. A csomag bizonyos vonatkozásokban egészen jó, néhol tudatosan kompromisszumot kellett kötni a sokféle igény között, néhol még nem tökéletes, és sok vonatkozásban még nem jutott időm további tulajdonságok fejlesztésére.

A jelen dokumentum egyúttal bemutató és tulajdonság tesztelő is. A dokumentum megkísérli bemutatni, mit hogyan kell és lehet használni, egyúttal azt is megvizsgálva, hogy tényleg működik-e az elvárt módon. A sokféle tulajdonság és különösen azok kölcsönhatása miatt sok munkát és időt igényel a fejlesztés, ezért a tényleges tulajdonságok nem mindig egyeznek meg a dokumentációval, különösen a kezdeti fázisban.

A makró csomag (legalább) három különböző felhasználói szinten alkalmazható. Már a legalacsonyabb szinten is szükségesek a \LaTeX -re vonatkozó elemi ismeretek. Az

alap szinten a felhasználó egyszerűen csak helyettesíti és módosítja a rendszert bemutató dokumentumokat. Haladó szinten (ehhez már el kell olvasni a felhasználói leírást is ☺) megtanulja a csomagban található makrók által biztosított lehetőségeket, és azokat aktívan használva fejleszti dokumentumait. Tapasztalt felhasználóként saját makrókat is készíthet (jó, ha azokat a letölthető anyaghoz hozzáadja), azaz aktívan részt vesz a fejlesztésben.

1.2. A MultEdu beüzemelése és használata

A MultEdu (mint minden \LaTeX alapú rendszer) feltételezi, hogy a felhasználó már rendelkezik tapasztalatokkal a \LaTeX használatában. Azaz, a felhasználó rendszerén már működnie kell valamilyen \LaTeX rendszernek.

Az egyszerű használat és a gyors elindulás érdekében célszerű a lentebb megadott módon saját projekt csoportjainak egy főkönyvtárat és azon belül az egyes projekteknek alkönyvtárakat létrehozni. A leggyorsabb magát a `./Workstuff` könyvtárat (a megfelelő átnevezésekkel és törlésekkel) lemásolni, és csekély módosításokkal elkészíteni saját 'Helló Világ' programját. Ezután érdemes legalább átlapozni a felhasználói kézikönyvet, ami után már elkezdheti saját fejlesztését. Eleinte csak szöveget, aztán sorjában megtanulni a használni kívánt tulajdonságok programozását. Ne feledje: a \LaTeX nehéz nyelv, pontos kódolást igényel, és ezért ilyen a MultEdu is. A gyakori mentések és a verziókövető rendszerek használata nagy segítséget jelentenek.

1.3. A MultEdu könyvtár szerkezete

A MultEdu rendszert az alábbi könyvtár szerkezetben célszerű használni. Két fő könyvtára: a `./common`, amely tartalmazza a MultEdu összes fájlját, és a `./Workstuff`, amely a felhasználói könyvtár szerkezetet modellezi.

```
.
|-- common
|-- WorkStuff
```

A felhasználói projekt csoportokat ilyen szerkezetben érdemes hozzáadni:

```
.
|-- Exams
|-- Labs
|-- Lectures
```

```
|-- Papers
```

amely könyvtáraknak a |-- `WorkStuff` könyvtárhoz hasonló belső alkönyvtárai vannak

1.3.1. A `common` alkönyvtár

A `./common` különleges célú al-alkönyvtárakat, valamint általános célú makró fájlokat tartalmaz. A Multedu megpróbál a lehető legbarátságosabb lenni: alapértelmezett beállításokat, fájlokat, képeket, stb használ, hogy gyorsan el lehessen kezdeni egy új fejlesztést.

```
.
|-- common
| |-- defaults
| |-- formats
| |-- images
```

A `./defaults` al-alkönyvtár olyan alapértelmezett szöveget tárol, mint a szerzői jogok. Alapértelmezetten, ha a felhasználó nem adja meg saját dokumentum elemeit, a MultEdu automatikusan az alapértelmezetteket használja helyettük (feltéve, hogy azok használata nincs megtiltva, lásd később).

A `./formats` al-alkönyvtár tartalmazza a formátumokat meghatározó makrókat; itt adhatja hozzá a felhasználó esetleges saját formátum leíró makróit.

Az `./images` al-alkönyvtár képeket tartalmaz, amelyek egy része alapértelmezett képként használatos.

1.3.2. A `Workstuff` alkönyvtár

A `./Workstuff` al-alkönyvtár tartalmazza (a példa programként is szolgáló) felhasználói leírás fájljait. Egy olyan `./Workstuff/Demo` projektet tartalmaz, amelyik (a saját főkönyvtárában) három fájlból áll.

```
|-- WorkStuff
| |-- Demo
| . |-- CMakeLists.txt
| . |-- Demo.tex
| . |-- Main.tex
```

A valódi főprogram `Main.tex`, és ehhez készült egy `Demo.tex` nagyon egyszerű boríték. Ha használja a UseLATEX csomagot, a `Main.tex` file használata (ezzel a névvel) kötelező, a boríték fájl nevét pedig a `CMakeLists.txt` fájlal egyeztetni kell.

A `./Workstuff` al-alkönyvtárai különböző célokat szolgálnak. Célszerű a felhasználói projekt könyvtárakat is hasonlóan berendezni.

```
|-- WorkStuff
| |-- Demo
| . |-- build
| . . . |-- build
| . |-- dat
| . |-- fig
| . |-- lst
| . |-- src
```

A fő `Main.tex` menet közben magába olvassa az alkönyvtárakban levő egyéb fájlokat.

```
| . |-- src
```

tartalmazza a felhasználó forráskód fájljait,

```
| . |-- fig
```

a képeit,

```
| . |-- lst
```

a programlisták forrás kódját,

```
| . |-- dat
```

a többi adatot (például táblázatok, adatok a pgfplot vagy kód a TikZ ábrák számára).

További alkönyvtárak is készíthetők, de azokat a felhasználónak kell kezelni, és módosítania kell a `CMakeLists.txt` fájlt is.

A CMake rendszeren keresztül a UseLATEX csomag is használható arra, hogy egy szerkesztés után, a kötegelt feldolgozási módot használva, egyetlen lépésben elő lehessen állítani a forrásnyelvi fájlból a különböző nyelvű és formátumú dokumentumokat; erre való a `CMakeLists.txt` fájl.

```
A
|-- build és
| . . |-- build
```

alkönyvtárak csak akkor kellenek ha a CMake rendszert használjuk; ezek a feldolgozás során szükséges átmeneti fájlokat tartalmazzák. A rendszer készíti a projekt könyvtárába (ami a `Demo` alkönyvtárnak felel meg egy saját másolatot a `common` alkönyvtárról. Ezek a fájlok bármikor törölhetők: amikor fordít, a CMaker újra generálja azokat.

1.4. A MultEdu csomag alapértelmezett fájljai

1.4.1. Alapértelmezett

A dokumentumokhoz tartozik néhány fejezet leíró definíció. Mintaként a felhasználói leírás `src/Heading.tex` fájlja szolgál.

A fejezetet olyan fázisban olvassa a program, amikor még nem használhatók a magyar ékezetes betűk, ezért azokat a szokásos L^AT_EX kódolással kell írni. A fejezet tartalma:

```
A \def\LectureAuthor{V\'egh J\'anos} sor adja meg a szerzőt, a \def\LectureTitle{Hogyan haszn\'aljuk\\ a MultEdu csomagot} a címét, a \def\LectureSubtitle{(Hogyan k\'esz\'i{ts}"unk \'erdekes\\ \'es vonz\'o tananyagot)}} pedig a dokumentum címét és alcímét. Megadhatunk egy \def\LecturePublisher{Egyetem neve vagy konferencia neve} meghatározást is. Javasolt egy \def\LectureRevision{V\Version\ (a \MERevision\ felhaszn\'al\'as\'aval) \at 2016.09.19} formájú sor használata is
```

```
Kétnyelvű dokumentumok készítéséhez a fentieket
\ifthenelse{\equal{\LectureLanguage}{magyar}}
{% in Hungarian
}% true
{% NOT magyar
}
```

blokkban kell elhelyezni.

Megadhatunk számítógépes címet is

```
\def\LectureEmail{Janos.Vegh\at unideb.hu}
```

Ugyancsak itt célszerű megadni a dokumentumban használt BibTeX fájlokat, akár a nyelv, vagy a fájl tényleges fellelhetősége alapján:

```
\IfFileExists{src/Bibliographyhu}
{\def\LectureBibliography{src/Bibliography,
src/Bibliographyhu}}
{\def\LectureBibliography{src/Bibliography}}
```

1.5. A MultEdu csomag beállítási lehetőségei

a

1.5.1. Beamer alapú formátum beállítások

A Multedu lehetővé teszi kétféle elterjedt formátum használatát. Egyre gyakoribb 16:9 arányú képformátum így az az alap beállítás. A 4:3 arányú képformátumot a `{\def\DisableWideScreen{YES}}`

definiálásával lehet beállítani.

Néha (főként rövid bemutatók esetén) egyáltalán nincs szükség tartalomjegyzékre. Ezt a `{\def\DisableTOC{YES}}`

definiálásával lehet elérni. Az is előfordul, hogy a fejezet-szintű tartalomjegyzék még szükséges, de a szakasz szintű már nem. Ezt a `{\def\DisableSectionTOC{YES}}`

definiálásával lehet elérni.

1.6. Ötletek a MultEdu csomag használatához

a

A dokumentum fordítása

2.1. Kézi fordítás

A MultEdu rendszer tökéletesen működik alapértelmezett beállításokkal is, de nem gondolatolvasó. A beállításokat `\def{\xxx}` formájú definíciókkal lehet megváltoztatni. A beállítások helye üzem módtól függ. A következő két szakasz az üzem módok használatát mutatja be, a harmadik pedig a beállításokat ismerteti részletesen.

A `Main.tex` fájl a közös és a két fordítási módban egyformán használt rész: ez tartalmazza a tényleges forráskódot. Az ebben a fájlban (továbbá az ide beolvasott fájlokban) szereplő bármely beállítás változtatás megváltoztatja a rendszer beállításait, azaz itt nem tanácsos bármiféle beállítást használni. Érdemes az összes beállítást egyetlen fájlba gyűjteni, amit aztán a fő fájl magába olvas.

A tananyag fejlesztést általában valamilyen szerkesztőbe integrált fejlesztő rendszerrel érdemes végezni. A szerkesztőbe be kell olvasni a boríték fájlt (a `Demo.tex` megfelelőjét) és azt gyöker dokumentumként megjelölni. A `Main.tex` fájlban érdemes hozzáadni a hivatkozásokat a tananyag fejezeteire, ami anyagokat természetesen a `src` alkönyvtárban célszerű elhelyezni, követve a demonstrációs anyag elrendezését.

A beállítások tárolására szolgáló fájl is a `src` alkönyvtárban érdemes elhelyezni, célszerűen `Defines.tex` néven. A burkolóként szolgáló `Demo.tex` feladata, hogy ezt és a fő fájlt beolvassa.

A kötegelt mód a konfigurálás során készít egy `Defines.tex` fájlt, de az a `build/build/src` alkönyvtárba kerül. (Onnét lehet puskázni, hogy mit és hogyan érdemes beállítani; miután egyszer már futott a kötegelt fordítás.) A kötegelt fordítás egy "minta" fájl is készít `Defines.tex.in` néven a `src` alkönyvtárba. Ennek a két fájlban a tartalma a kötegelt fordítás utolsó menetének felel meg.

2.2. Kötegelt fordítás

A kötegelt fordítás (főként) arra szolgál, hogy a közös forráskódból kényelmesen tudjuk előállítani a különféle formátumokban és nyelveken anyagainkat.

Technikai okokból a tényleges fordítás előtt a rendszer saját másolatot készít a MultEdu szükséges fájljairól a

projekt `common` alkönyvtárába. Ezzel a saját kópiával lehet kísérletezni, vagy akár törölni; a következő kötegelt fordítás majd helyreállítja. (azaz a következő fordítás előtt az értékes fejlesztést el kell menteni, akár a `../../common` alkönyvtárba, ha azt másutt is használni akarjuk.)

2.3. Az alapbeállítások megváltoztatása

2.3.1. A verziók kezelése

A MultEdu a standard háromszintű verzió számozást használja (fő és alszám, valamint folt). A MultEduval készült anyagoknak kétféle verziója van: a saját tananyagának verzióját a felhasználó tartja karban, a MultEdu változatát pedig a fejlesztő.

A Multedu verziószáma a `../../common/MEMacros.tex` fájlban található; célszerű változtatlanul hagyni. A saját kurzus anyag verzióját a `CMakeFiles.txt` file tartalmazza, az minden kötegelt fordítás alkalmával frissül a `Defines.tex` fájlban. A kézi fordításnak saját beállításai vannak, de célszerű azt átvenni a generált fájlból.

A saját verzió száma a generált kimeneti fájl nevében is szerepel, tehát érdemes következetesen használni azt. Használata: `\def\Version{nagy.kis.folt}`

2.3.2. Nyelvek kezelése

A MultEdu egy- és két-nyelvű dokumentumokat tud kezelni. A különböző nyelvekhez különböző helyesírás, fejezetcímek, feliratok tartoznak. A beállításoknál kell megadni a nyelvet: ezt pl. a `\LectureLanguage{magyar}` beállítással lehet megtenni (enélkül az alapbeállítás `\LectureLanguage{english}`).

A kiválasztott nyelv neve az eredmény file nevében is megjelenik.

A kétnyelvű dokumentumokban van egy első és egy második nyelv, amilyen sorrendben szerepelnek a nyelvi szövegek a dokumentumban. Ez lehetővé teszi, hogy az egymás alatt levő kétféle nyelvű kurzus anyagot összhangban tudjuk fejleszteni. A nyelv kiválasztásával a két anyag bármelyikéből tudunk eredmény fájlt generálni. Ha a `\UseSecondLanguage{}` definiálva van, a sorrendben második nyelvet fogja a csomag feldolgozni, és arra a `\LectureLanguage{}` által megadott szabályokat használja.

Kötegelt fordítás esetén meg kell adnunk a `FirstLanguage` és `SecondLanguage` értékét (azaz, hogy az elsőként és másodikként megtalált szöveg milyen nyelvű). Ha bekapcsoljuk a `NEED_BOTH_LANGUAGES` kapcsolót, a kötegelt feldolgozás során mindkét nyelvű kimenő fájrt előállítja a rendszer. Ha ez ki van kapcsolva, akkor a `USE_SECOND_LANGUAGE` kapcsoló dönti el, melyik nyelvet fogja a rendszer használni.

A dokumentum tagolása

3.1. Dokumentum egységek

A dokumentumot a 'beamer' csomag követelményeinek megfelelően kell szervezni. A nyomtatható formában való megjelenítéshez a MultEdu a 'beamerarticle' csomagot használja, és a tagolást is megfelelően változtatni kell. Ennek érdekében a MultEdu saját tagolási egységeket vezet be, amelyek valójában a 'book' formátumnak felelnek meg, és amelyeket dia készítéshez megfelelően átalakít.

3.1.1. Dia keretek

Használata:

```
\MEframe[keys]{subtitle}{content}
```

Értelmezett kulcsok

```
shrink=true|false and plain=true|false
```

Alapértelmezetten mindkettő false.

3.1.2. Fejezet

A dokumentum legnagyobb egysége a fejezet.

Használata:

```
\MEchapter[short title]{long title}
```

3.1.3. Szakasz és az alatt

A következő, kisebb egység a szakasz Használata:

```
\MEsection[r"ovid cím]{hosszú cím}
```

Hasonló

módon létezik `\MEsubsection[r"Ovid cím]{hosszú cím}` és `\MEsubsubsection[r"Ovid cím]{hosszú cím}`; ez utóbbi dia készítés esetén `\paragraph` alakot ölt.

3.2. Kétnyelvű forráskódok

Előfordul, hogy ugyanazt az anyagot saját nyelven oktatom hazai hallgatóknak, és angolul, külföldi hallgatóknak. A tananyag megegyezik, és együtt kell fejleszteni. Nyilván előnyös, ha a két anyag ugyanabban a forrásnyelvi fájlban, egymás mellett fejleszthető.

Erre szolgál a `\UseSecondLanguage`. A fent bevezetett makróknak van egy 'D' (Dual) taggal kibővített változata, amelyikben mind az elsődleges, mind a másodlagos nyelven megadjuk a szükséges tartalmakat.

3.2.1. Átváltás a nyelvek között

Használata:

```
\UseSecondLanguage{YES}
```

ahol az {} nem számít, csak az, hogy definiálva van-e ez a makro.

A kétféle makrókészlet keverhető, de csak a 'D' makrók reagálnak a nyelv változtatásra.

3.2.2. Dia keretek

Kétnyelvű dokumentumokban általában a

```
\MEDframe[keys]{subtitle, first language}
{content, first language } {subtitle, second
language} {content, second language}
```

keretet használjuk. Azaz a felhasználó megadja mindkét nyelven a címet és a tartalmat, majd fordítás előtt `\UseSecondLanguage` használatával kiválasztja az egyik nyelvet.

3.2.3. Fejezet

Hasonlóképpen, a kétnyelvű dokumentum legnagyobb egysége a 'Dchapter'. (Amint említettük, dia készítéskor ez átalakul 'Dsection' egységgé.) Használata:

```
\MEDchapter[r"Ovid cím1]{hosszú cím1}{r"Ovid
cím2}{hosszú cím2}
```

ami aztán átalakul

```
\MEchapter[r"Ovid cím1]{hosszú cím1} vagy
```

```
\MEchapter[r"Ovid cím2]{hosszú cím2}
```

attól függően, hogy `\UseSecondLanguage` definiált vagy sem.

3.2.4. Szakasz és az alatt

Teljesen hasonló a kisebb formázási egységek használata is.

3.3. Fejezet illusztráció

Néhány könyv stílus lehetővé teszi, hogy a fejezetek elején egy illusztrációt helyezünk el.

Használata:

`\MChapterillustration{file}`

Dia készítéskor, a kép egy 'plain' dián jelenik meg. Nyomtatható változatban a fejezet elején jelenik meg a kép.

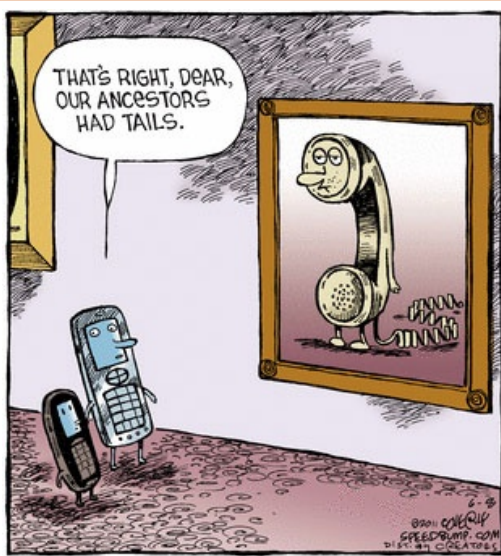
Ha a fájl név üres, a csomag a 'fig/DefaultIllustration.png' képet keresi. Ha a fájl nem található, nem készül illusztráció.

Ha definiáljuk a `\DisableChapterIllustration` makrót, a csomag nem generál képet.

Ábrák beszúrása

4.1. Hagyományos ábrák

©2011 <http://pinterest.com>



4.1. ábra. Amikor régi és új telefonok találkoznak

Az ?? ábra előállításához a

```
\Mefigure{fig/phone_ancestors} {{Regi es uj
telefonok ha talalkoznak}}
{fig:phonenancestors} {2011
http://pinterest.com}{.8}
```



Tárgymutató

`\MEchapter`, [6](#)

`\MEchapterillustration`, [7](#)

`\MEDchapter`, [6](#)

`\MEDframe`, [6](#)

`\MEDsection`, [6](#)

`\MEframe`, [6](#)

`\MEsection`, [6](#)

package

beamer, [6](#)

beamerarticle, [6](#)

`\UseSecondLanguage`, [6](#)



Ábrák jegyzéke

4.1. Amikor régi és új telefonok találkoznak	8
--	---

