

DOKUMENTUMSZERKESZTÉS

L^AT_EX

Jakó Zoltán

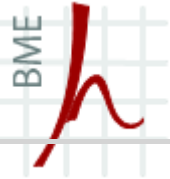
Tudományos segédmunkatárs

BME Hálózati Rendszerek és Szolgáltatások Tanszék
(Mobil Kommunikáció és Kvantumtechnológiák Labor)

BME I. épület IB 113

E-mail: jakoz@hit.bme.hu

2015. szeptember 8.,
Budapest



Adminisztráció és első lépések 1.

- *Dokumentumszerkesztés* c. szabadon választható tárgy
- 0/0/2 félévközi jegyes
- 2 kredit
- Mindenki ide jelentkezett?
- **Mindenki ellenőrizze, hogy a megfelelő kurzusra jelent-kezett-e, egyébként nem kaphat jegyet a félév végén!!!**
- További kurzusok a tárgyon belül:
 - LaTeX kurzusok - L
 - Open Office - O
 - MS Word kurzus -W
- Tegezzetek! Én is így teszek!

Adminisztráció és első lépések 2.

- *Óra tartás:*
 - Az órák kedden 12:15-től kezdődnek **ITT!** (R épület **R4K**),
 - Szünetet nem tartunk, ezért az órának 13:45-kor van vége!
 - 5 perc késést elnézek!
 - Aki 12:20 és 12:30 között ér be **Késés** minősítést kap, 12:30 után **Hiányzást** (ettől még az órára be lehet ülni).
- *Követelmények:*
 - A hiányzások száma maximum az órák 30%-a lehet!
 - Aláírást az kaphat, aki az órák maximum 30%-áról hiányzik (**maximum 4 óra**), a többin akár időben, akár késve, de ott van.
 - Minden órán kiosztok egy jelenléti ívet, melyet a megfelelő helyen kell aláírni. (Így nem is tudok neveket párosítani arcokhoz)
 - A jelenléti statisztika megtekinthető a tárgy oldalán

- *Követelmények:*
 - A tárgy weboldala: www.mcl.hu/~jakoz/dokszerk
 - Az előadás anyaga (jelen diasor) szintén fellelhető az oldalon
- A félév során **6 darab kis házi feladatot** lehet megoldani, amelyeket a következő gyakorlat napja előtti munkanap, reggel 8 óráig kell feltölteni az oldalra! („Kis házi feladat beadása” alatti link)
- Csak az időben leadott házi feladatokat tudom kijavítani!
- A kisHF-ok elkészítése **opcionális**, tehát nem kötelező. Ha kapok, kijavítom! – *Jó gyakorlási lehetőség!! Segít ha tanulsz a hibáidból és a nagy háziban ne követed el őket!*
- **A félév végi jegyet a nagy házi feladatra adom.** A feladatot a 10. hét környékén osztom ki, és a 12. hét környékén kell beadni a fenti webcímen.

Adminisztráció és első lépések 4.

- A nagy házi feladat leadásával késni nem lehet! Aki késik, az nem kap jegyet – Érdemes időben elkészíteni.
- Ha a jegy kétes, a késések és hiányzások számát figyelembe veszem. Egyéb esetben nincs jelentősége a késéseknek és hiányzásoknak.
- A kis HF-oknak abszolút semmi hatása nincs a félév végi jegyre.

- <http://www.mcl.hu/~jakoz/dokszerk/> - az itt található fóliák
- További irodalom nagyon érdeklődőknek:
 - Wettl, Mayer, Sudár, *LATEX kezdőknek és haladóknak*, Panem, 1998.
 - Wettl, Mayer, Szabó, *LATEX kézikönyv*, Panem, 2004 (*online letölthető az első két fejezete*)
 - LaTeX 78 percben (= LaTeX nem túl röviden) (ingyenesen letölthető)
 - Helmut Kopka, Patrick W. Daly, *Guide to LaTeX (4th Edition)*, ISBN-10: 0321173856, 2004
- Tipográfiai szabályszerűségek összefoglalása:
 - <http://mcl.hu/~jeneyg/kinezet.pdf>

Felhasznált irodalmak

- LaTeX 78 percben (= LaTeX nem túl röviden)
- Jeney Gábor LaTeX előadás fóliái
- Tómacs Tibor: *Bevezetés a LaTeX dokumentumkészítő rendszer használatába*
- Tómacs Tibor: *Elektronikus prezentáció készítése a LaTeX beamer dokumentumosztályával*

- A tárgy azért indult, mert a hallgatói dolgozatok (diploma, TDK, önlab beszámoló stb.) minősége meglehetősen alacsony volt.
- Szükséges beszélni az alapvető tipográfiai szabályokról, mert azokat sehol sem oktatják (középiskolában kellene). Másrészt a szövegtördelő programokat sem tudják használni a hallgatók (tartalomjegyzék kézzel, egyenletek szerkesztése stb.)
- A tárgy célja az, hogy
 1. A hallgatók igényesebb dokumentumokat alkossanak (pl. Diplomamunka)
 2. LaTeX2e, MS Word 2010, Libre office stb. felhasználói szintű ismereteket elsajátítása

A tárgy oktatásának módja

- Számítógépes tanórák keretében, heti két órában.
- Az elsajátítást segíti, hogy a tanult ismeretek azonnal kipróbálhatóak a számítógépen.
- Használd a teremben lévő gépeket, vagy hozd be a saját laptopodat, nekem mindegy!
- **Gyakorolj otthon is!** Ahogyan programozni sem lehetett megtanulni csak az egyetemi órákon, úgy a LaTeX sem elsajátítható otthoni gyakorlás nélkül.

A LaTeX kurzus tematikája

Hét	Dátum	Előadás címe
1.		Adminisztratív feladatok, dok.szerkesztési alapelvek, mi is az a LaTeX?, MikTeX/TexWorks/TexMaker megismerése
2.		A LATEX alapjai
3.		Dokumentumok kinézete és strukturálása
4.		Környezetek, felsorolások
5.		Tabulálás és Táblázatok szerkesztése
6.		Képek, ábrák beillesztése, dobozok
7.		Hivatkozások, tartalomjegyzék, ábra-, táblázat és irodalomjegyzék készítése
8.		Matematikai képletek,matematika mód 1
9.		Matematikai képletek,matematika mód 2
10.		Oldalszerkesztés, méret, elforgatás, hasábok, Nagy HF kiosztása
11.		A LaTeX programozása/csomagok/dokumentumosztályok használata
12.		Prezentáció készítés – A Beamer alapjai (Nagy HF beadási határidő)
13.		Prezentáció készítés – Prezentáció megjelenés és keret komponensek
14.		Prezentáció készítés – Látványeffektek (Beamer)

HOGYAN NÉZ KI EGY IGÉNYES DOKUMENTUM?

NÉHÁNY SZÓBAN AZ ALAPVETŐ TIPOGRÁFIAI SZABÁLYOKRÓL

Válasszunk megfelelő szövegszerkesztőt!

- A „✓” szimbólum azt jelenti, hogy a jellemzőt támogatja az alkalmazás.
- A „-” szimbólum azt mutatja, hogy az adott jellemző nem támogatott.
- Végül a „*” szimbólum azt jelzi, hogy az adott jellemző ugyan támogatott, de némi megjegyzésre szorul

Jellemző	MS Word	LaTeX
Grafikus felhasználói felület	✓	-
WYSIWYG	✓	✓
Beépített stílusok	✓	✓
Szerkeszthető más alkalmazással	*	✓
Hordozhatóság	*	✓
Hordozható formátumokba mentés	-	✓
Helyesírás-ellenőrzés magyarul	✓	-
Helyesírás idegen nyelven	✓	-
Elválasztási szabályok magyarul	✓	✓
Elválasztás idegen nyelven	✓	✓
Magyar jellegzetességek támogatása	-	✓
Irodalmi hivatkozások	-	✓
Irodalmi hivatkozások frissítése	-	✓
Hivatkozás irodalomra szövegben	-	✓
Hivatkozások frissítése	-	✓
Képek, ábrák, táblázatok	✓	*
Kép, ábra, táblázat feliratok	*	✓
Ezek automatikus számozása	*	✓
Képre, ábrára hivatkozás	✓	✓
Ezek automatikus frissítése	*	✓
Matematikai szedés soron belül	*	✓
Matematikai szedés külön sorban	✓	✓
Képletek automatikus számozása	-	✓
Képletekre hivatkozás	*	✓
Ezek automatikus frissítése	*	✓
Hivatkozás oldalakra	✓	✓
Ingyenesen elérhető	*	✓

Nézzük meg a LaTeX jellemzőit

- A LaTeX egy szkriptnyelv -> ezért alapból nincs GUI hozzá, de vannak olyan programok amik rendelkeznek ilyenekkel (pl. TexMaker lásd később.)
- Helyesírás ellenőrzés beállítása -> megoldható (lásd később)
- Nem WYSIWYG, cserébe szöveges formátumú, ezért egyszerű editálni.
- Irodalom hivatkozások fontosak főleg diplomamunkáknál! – ebből a szempontból a LaTeX jó választás lehet.

- **Válasszunk megfelelő operációs rendszert**, egy új operációs rendszer megtanulása sok időt vihet el.
- **Válasszunk megfelelő nyelvet** a diplomamunkához:
- Az idegen nyelven írt diplomának sok előnye és sok hátránya is van a magyar nyelvű verzióhoz képest.
 - idegen nyelvű dokumentummal **külföldön** is érvényesülhetsz,
 - Ugyanakkor nem szabad megfeledkezni arról, hogy mindenki az **anyanyelvén tudja a legjobban kifejezni magát**.
 - Ha nincs a környezetünkben született angol, német, francia stb. és/vagy a konzulensünk se beszéli az adott nyelvet, ne is fogjunk hozzá idegen nyelvű diplomamunkához.
 - **a szakzsargon, a szakkifejezések megfelelő alkalmazása** a legfontosabb követelmény egy szakmai dokumentumban
 - **Mindig olvassa át valaki a munkánkat**, mert sokszor átsiklik az ember a hibákon (stilisztikai és/vagy nyelvtani hiba).

Tördelési hibák - fattyúsor

- **Árva sor:**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse vel mauris. Cras quam tellus, interdum dapibus, mattis eu, pharetra vitae, neque. Nam semper mollis risus. Quisque sollicitudin pellentesque turpis. Aenean blandit, urna nec vehicula sollicitudin, ipsum justo condimentum dolor, vitae feugiat augue leo quis velit.

Nunc volutpat, est eget pharetra vulputate, orci nisi consequat felis, vel ultrices turpis pede at purus. Mauris nisi. Nulla facilis.

Quisque sollicitudin tincidunt

8

libero. Donec aliquet leo non tortor. Aenean vel massa quis nunc hendrerit vestibulum.

9

- **Özvegy sor:**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse vel mauris. Cras quam tellus, interdum dapibus, mattis eu, pharetra vitae, neque. Nam semper mollis risus. Quisque sollicitudin pellentesque turpis. Aenean blandit, urna nec vehicula sollicitudin, ipsum justo condimentum dolor, vitae feugiat augue leo quis velit.

Nunc volutpat, est eget pharetra vulputate, orci nisi consequat felis, vel ultrices turpis pede at purus. Mauris nisi. Nulla facilis. Nunc nulla neque, laoreet a, tincidunt sed, mollis sed, pe-

8

desol.

Quisque sollicitudin tincidunt libero. Donec aliquet leo non tortor. Aenean vel massa quis nunc hendrerit vestibulum.

9

- Nagyobb egyenleteket külön sorba kell írni,
- Csak azokat az egyenleteket számozzuk amit később használunk is.
- Számozott képletek után a sor végén jobbra zárva egy számtalálható zárójelben: ez az egyenlet sorszáma pl. (1), vagy fejezetenként (x .y)
- A matematikai képlet úgy viselkedik a szövegben, mint egy
- külön szó, vagy mondatrész, ezért **írásjelek szükségesek**

$$e^n = \lim_{x \rightarrow \infty} \left(1 + \frac{n}{x} \right)^x,$$

- matematikai szimbólumok más betűkkel legyenek,
 - *tancos x -et ne keverhessük össze a tancos x -szel.*
 - Változókat dőlt betűvel, állandókat (konstansokat) álló betűvel kell írni!
- Kivétel: differencia (d) álló, a természetes szám (e) ill. az imaginárius egység (j , vagy i) dőlt, minden más állandó álló.
- \cos , \sin , \tan , \lim , mod stb.) függvények álló, de az $f(x)$ dőlt.
- Vektorok, mátrixok jelölése:
 - A szakirodalomban eltérően kezelik:
 - Van aki aláhúzással jelöli, \underline{v} vektor egy aláhúzással és \underline{M}
 - Van aki felülvonással, \bar{v} vektor egyszer felülvonva, \bar{m} mátrix
 - Van aki vastagítással, \mathbf{v} vektor kisbetűvel és \mathbf{M} mátrix
 - A lényeg, hogy egységes legyen a jelölés a teljes dokumentumban!

- Sokszor nehéz elkerülni a **szóismétlést**, azonban mindig ellenőrizzük, hogy vajon nem azértmaradt-e a dokumentumunkban, mert lusták voltunk átfogalmazni az adott részt a dokumentumunkban.
- Az idegen szavak között több olyan is van, melyet nem szabad fokozni:
 - „optimális”, „minimális”, „maximális” szavak önmagukban is szélsőséges értékeket jelölnek, de a „legoptimálisabb”, „minimálisabb”, „legleslegmaximálisabb” szavaknak nincs sok értelme.
- idegen szöveg tükörfordítása kapcsán jelentkező szórendhibák

A dokumentum szerkezete

- legfontosabb dolog a strukturáltság,
- Az egymást követő részek gondolatmenete egymásra épüljön,
- A fejezetek, alfejezetek rendszere tükrözze a logikai egységek határait,
- Nem fordulhat elő az, hogy egy alfejezet nem kötődik a fejezethez és viszont,
- A fejezetek, alfejezetek rövid, tömör, egyszavas, vagy szókapcsolatos címeket kapjanak
- A strukturáltság leggyorsabban a tartalomjegyzékkel ellenőrizhető.
- Behúzások, kihúzások (ezt kerülni kell)

MI IS AZ A $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$?

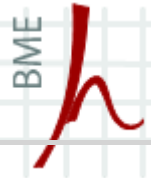
- Régen egy nyomtatott dokumentum a következő lépésekben készül el:
 1. A szerző megírta művét (kézirat),
 2. A művet szakértő lektorálta,
 3. bekerült egy kiadóba,
 4. tipográfus megtervezte a könyvet,
 5. a szedő a tervek alapján minden oldalt elkészített,
 6. nyomdába kerülve sokszorosították.

- A számítógépes programok megjelenése óta a szerző már nem csak a mű megírását hanem a szedést és tördelést is maga végzi.

- Őse a TeX (ejtsd: teh)
- (A szó a görög „techné” kifejezésből ered, melynek jelentése művészet, mesterség).
- A TeX egy betűszedő rendszer, amelyet Donald E. Knuth készített 1978-ban.
- A verziószáma aszimptotikusan közelít π -hez (jelenleg: 3,141592)
- A LaTeX (ejtsd: lateh) egy TeX-en alapuló szövegformázó rendszer melyet Leslie Lamport alkotott meg 1985-ben.
- A LaTeX mellett megjelent még az AMS-TeX (Michael Spivak, American Mathematical Society) mely a matematikusok körében volt elterjedt.
- 1994-ben megjelent a LaTeX2e amely magába foglalta az AMS-TeX-et is!
- A legújabb verzió LaTeX2e 20-as verziója (2011) – Mi is ezt használjuk!
- Készülőben a LaTeX3 (évek óta ígérik)

➤ **Különlegessége és egyben előnye is:**

- Elég idős (közelít a 30 évhez),
- Minimális a hardverigénye,
- Ingyenes és nyílt forrás kódú,
- Operációs rendszer független (Windows, Linux, MAC OS X),
- Nagyon szép dokumentumokat lehet vele létrehozni, mintha a nyomdában készült volna,
- Matematikai formulák megalkotása könnyedén, szépen
- Irodalomjegyzék, tartalomjegyzék, szójegyzék, lábjegyzet, kereszthivatkozások automatikus elkészítése és frissítése,
- Csekély méretű végdokumentumot (PDF) kapunk
- Nagy méretű dokumentumok esetén feltétlen javasolt a használata



A LaTeX története 3.- hátrányok

➤ **Megemlítendő azért a sötét oldal is:**

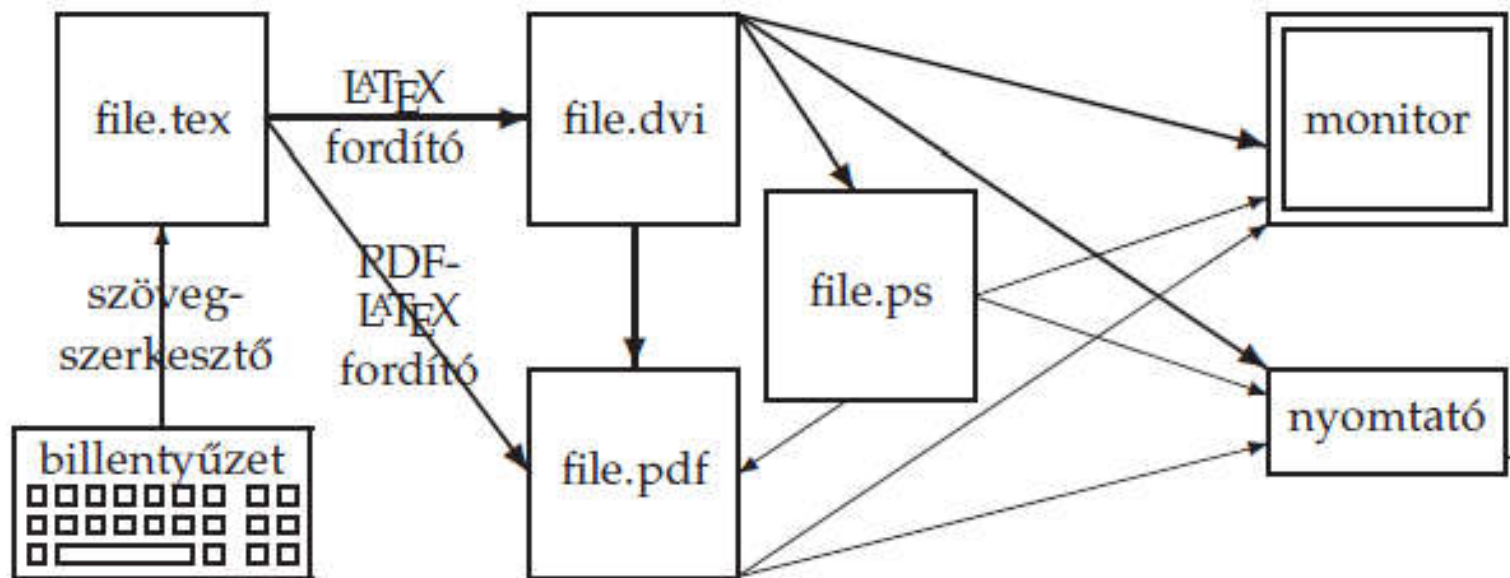
- Új stílus kialakítása bonyolult tud lenni,
- Hibaüzenetek megfejtése nehézkes,
- A LaTeX parancsok és a szövegből álló dokumentum körülményesebb, nehezebben áttekinthető, mint a szokásos WYSIWYG rendszerek végeredménye
- Leírás nélkül nehéz boldogulni, pl. egy új csomag használata

Számos (szerkesztő és fordító) program fellelhető:

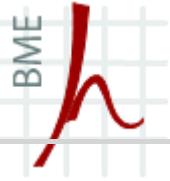
- Windows rendszerekben:
 - **MikTeX**,
 - **TeXworks** (összenőtt a MikTeX-hel),
 - TexMaker
 - TeXstudio,
 - WinEdt
- Linux:
 - TEXlive,
 - TEXmaker,
 - TeXstudio

A dokumentum fordítás menete

- A billentyűzettől a kész műig:



L^AT_EX ALAPOK



Kezdjük!

- Minden LaTeX dokumentum kötelezően kell, hogy tartalmazza az első sorában a következő utasítást:

```
\documentclass[opciók]{dokumentumosztály neve}
```

- Mint látható minden LaTeX parancs „\”-lel kezdődik!
- [opciók] paraméter tartalmazza, az eredeti alapbeállításoktól való eltérést. Később nézünk rá példát!
- { } közti részt a dokumentumosztály *argumentumának* nevezzük.
- Alapesetben 5 dokumentumosztály létezik:
 - Article (kisebb kutatási anyaghoz, cikk írásra),
 - Report (diplomamunkához),
 - Book (könyv írásához),
 - Slides (prezentációk készítéséhez),
 - Letter (levelek írásához).

1. A dokumentumosztályt követő részt **preambulum**-nak nevezik.
 - Ha a LaTeX képességét egyéb csomagokkal akarjuk bővíteni akkor itt lehet őket betölteni.
 - (megjegyzés: A csomagoknak is lehet opciója [] és argumentuma {} nem csak a dokumentum osztálynak!)
- Ha egy argumentum nélküli parancs az azt követő részre hat ki, akkor azt **deklarációs parancsnak** nevezzük: `\bfseries{}`
- **Példa:** `{\bfseries Csak ez fog félkövéren megjelenni.}`
Ez már nem.

2. Végül pedig a és a dokumentumtörzs:

```
\begin{document}
...Where do you want to go today?...
\end{document}
```

- Tehát az első dokumentumunk úgy néz ki, hogy:

```
\documentclass{article}
\begin{document}
Hogy vagyunk ma?
\end{document}
```

Dokumentumosztályok opciói 1.

- A dokumentumosztályokon belül léteznek opciók is:

I. Article dokumentumosztály:

- Állítható opciók:
 - Betűméret 10pt, 11pt, 12pt,
 - Papírméret a4paper, a5paper, b5paper, letterpaper,
 - Egy ill. kétoldalas szedés: oneside, twoside
 - Címlap: notitlepage, titlepage
 - Végleges verzió: final, draft

Alap beállítások: Betűméret:10pt, papírméret: letterpaper, egy ill. kétoldalas szedés: oneside, címlap van e: notitlepage, vázlat vagy végleges dokumentum: final

Példa:

```
\documentclass[12pt,b5paper]{article}
```

Dokumentumosztályok opciói 2.

II. Report dokumentumosztály:

- Állítható opciók:
 - Ugyanaz mint az előbbinél, **DE**
 - Ezenkívül van még egy opciója: `openright`, `openany`

A részek és fejezetek új oldalon kezdődnek. Ez az oldal `openright` esetén csak páratlan sorszámú lehet, s ennek érdekében akár üres oldalt is hagy,

Az `openany` opció esetén az új oldal száma bármilyen lehet.

Alap beállítások: Alapértékek: 10pt, letterpaper, oneseide, titlepage, final, `openany`

Példa:

```
\documentclass[12pt,b5paper]{report}
```

III. Book dokumentumosztály:

Lásd. Article dok. osztály

Alapbeállítások: 10pt, letterpaper, twoside, titlepage, openright, final.

Magyar szöveg írása

- Az előző példánkban nem használtunk magyar ékezetes karaktereket.
- Ékezetes karakterek írására két lehetőség adódik:

Ékezetes karakterek megjelenítése az alábbi paranccsal lehetések: \'
 Ez minden utána álló karakterre egy vesszőt helyez: pl. \'e, \'u, \'s.
 További lehetőségek:

ó \\'o	õ \~o	ő \=o	o \b{o}	ö \u{o}	i \i
ò \'o	ô \^o	ò \.o	o \c{o}	ö \v{o}	j \j
ö \"o	ő \H{o}	ô \r{o}	o \d{o}	oo \t{oo}	

- **Okosabb megoldás:** Ez mégiscsak a 21. század! Használjuk a csomagok adta lehetőséget:

```
\documentclass{article}
\usepackage[utf8]{inputenc}
\begin{document}
Hoppá nágyón üügyes kérem szépen
\end{document}
```

- Szükséges lehet még a `\usepackage{t1enc}` ha egyéb speciális nyelvi karakter megjelenítésére van szükségünk pl.: Đurdevac, Pórshöfn

- A babel csomaggal magyaríthatjuk a LaTeX-et:

```
\usepackage[magyar]{babel}
```

- A csomag általában jól választ el, de ebben az esetben pl. nem karóra (karó-ra, vagy kar-óra). Ilyenkor „puha” elválasztás kézzel: (kar\ -ó\ -ra),
- vagy megadhatunk egy `\hyphenation{el-vá-lasz-ta-ni-mer-kell-neki-tyuhajja}` így a szerkesztő tudni fogja hol kell elválasztani ezt a meglehetősen összetett szót.
- Ehhez azonban szükséges a t1enc csomag!
- Ha **többnyelvű dokumentumot** szeretnénk írni:

```
\usepackage[english,magyar]{babel}
```
- paranccsal indulunk, és a `\selectlanguage{nyelv}` paranccsal lehet kiválasztani az aktuális nyelvet.

Speciális karakterek, parancsok

- Vannak olyan karakterek melyeket a fordító parancsnak értelmez.
- Foglalt karakterek:
 - \ (backslash) parancsok ezzel kezdődnek,
 - % kommentek ezzel kezdődnek,
 - {} blokkok, illetve parancsok argumentumainak határai,
 - \$ matematikai mód határolójele,
 - & táblázatoknál kell,
 - # (hash mark) változót tartalmazó parancs definiálásához kell,
 - _ alsó index,
 - ^ felső index,
 - ~ törhetetlen szóköz.
- ezért, ha ilyen karaktert kell megjeleníteni a szövegbe: \ = \backslash , { = $\{$, } = $\}$, % = $\%$, ~ = $\sim\{\}$, \$ = $\$$, _ = $_$, ^ = $\^{\}$, & = $\&$, # = $\#$
- A LaTeX tartalmaz beépített logókat: \TeX , \LaTeX és \LaTeXe

\TeX , \LaTeX és \LaTeXe

Speciális karakterek, parancsok 2.

- Vigyázat az argumentum nélküli parancsok elnyelik az utána következő sorközt! Pl.: `\LaTeXe könyv`
- Szóköz kikényszerítése: `_` vagy `{\LaTeX} könyv`
vagy `\LaTeX{ } könyv` vagy `\LaTeX\ könyv`
- LaTeX-ben a parancsok `\command{param}` alakúak, és `{}` rész mindig kötelező (kivéve a fenti példán és még `\dots`)
- `[]` paraméterek opcionálisak

- A dokumentumokat bekezdésekkel struktúráljuk. Új bekezdés: két „rep” karakter `\\` (később részletesen)
- Nem törhető szóköz: tilde (`~`): `II.~Rákóczi~Ferenc`
- Blokkosítás: `\mbox{valami}`: `\mbox{II. Rákóczi Ferenc}` (később részletesen)
- *Néhány stilisztikai megjegyzés:*
 - `IOO□100`,
 - Magyarban tizedesvessző van, angolban pont! `3,14` vagy `3.14`.
 - Mértékegység, pénznemek: nem törhető szóközzel: `1~cm`
 - ... helyett `\dots`
 - Vessző után space jön,
 - Idézőjel magyarban „idézet”, Brit angolban ‘this’, USA angol “different”
- hüvelyk (`"` shift+2) bár más tördelőkben ezt használtuk, itt nem jó!) nem idézőjel!!! Az idézőjelek helyes megjelenítéséhez szükséges a `t1enc` csomag!

- Ligatúrán a betűknek a szokásosnál szorosabb összekötését értik. A legismertebbek a következő ún. f-ligatúrák: `ff fi fl ffi ffl`. A LaTeX ezt automatikusan megcsinálja, ha nem kérünk belőle, akkor a trükk: `ef{}féle`.

- Mondatköz: Alapértelmezésben a LaTeX a mondatok közé nagyobb térközt tesz, mint a szavak közé. Ettől olvashatóbbá válik a nyomtatott szöveg. Ha nem akarjuk `\frenchspacing` paranccsal kikapcsolhatjuk.

De például az 1. ábra nem mondatvég hiába van pont, ezért 1.~ábra.

Kötőjel, nagy kötőjel, gondolatjel, mínusz jel

- Kötőjel (-): sima „-” jel. Használata: „Kérek 3-4 darab paradicsomot”.

`-5°C`  `-5°C`

- Nagy kötőjel = gondolatjel (—): „- -” (mínusz mínusz) tartomány meghatározására szolgál pl.: „100—150 oldalas tankönyv”.
- Hosszú gondolatjel „---” (magyarban nem használjuk).
- Párbeszédet gondolatjellel kezdünk --\,szöveg alakban (nincs szóköz a --\, és a szöveg között!!!).

DOKUMENTUMOK KINÉZETE ÉS STRUKTURÁLÁSA

- Így nézett ki az első dokumentumunk:

```
\documentclass{article}
\begin{document}
Hogy vagyunk ma?
\end{document}
```

- Magyarosítva pedig így:

```
\documentclass{article}
\usepackage[utf8]{inputenc}
\usepackage[magyar]{babel}
\begin{document}
Hoppá nágyón üűgyes kérem szépen
\end{document}
```

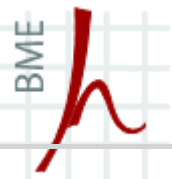

Cím, szerző és dátum megadása

- A `\title{...}`, `\author{...}`, `\date{...}` utasításokkal
- megadhatjuk a dokumentumcímét, a szerzőt és a dátumot.
Ha a `\date{}` utasítás argumentumát üresen hagyjuk nem tesz dátumot!
- A mai dátum beillesztése: `\date{\today}`
- Több szerző esetén az `\and` paranccsal sorolhatjuk fel őket.
- A cím, szerző és dátum beillesztése a `\maketitle` paranccsal lehetséges.

- Alapesetben arab számokkal történik az oldalszámozás,

<code>\pagenumbering{arabic}</code>	→	1, 2, 3, ...
<code>\pagenumbering{roman}</code>	→	i, ii, iii, ...
<code>\pagenumbering{Roman}</code>	→	I, II, III, ...
<code>\pagenumbering{alph}</code>	→	a, b, c, ...
<code>\pagenumbering{Alph}</code>	→	A, B, C, ...

- **Döntött** (slanted) betűket `\textsl{szöveg}`, `{\slshape szöveg}` parancsokkal lehet elérni.
- **Dőlt** (italics) betűket a `\textit{...}`, `{itshape ...}` parancs teremti.
- **Félkövér** betűkhöz: `\textbf{ide jön a szöveg}`, `{\bfseries vagy ide}`
- **Egyéb** betűkhöz:
 - Kiskapitális betűkhöz: `\textsc{...}`,
 - Antikva (normál): `\textrm{...}`, `{\rmfamily ...}`,
 - Grotész: `\textsf{...}`,
 - Írógép: `\texttt{...}`, `{\ttfamily ...}`



Betű típusok

- A dokumentum szintjén történik!
- (a nyomdászok szerint: egy betűcsaláddal kell szedni az egész művet, de a betűcsalád lehetséges alakjait is használni kell).
- Az Adobe által ingyenesen elérhetővé tett fontok a megfelelő csomag betöltésével használhatóak (pl. times, avant, avantgar, bookman, chancery, charter, courier stb.)
- Példa: `\usepackage{times}`

Kiemelés és betűméret 1.

- Amikor egy szót, vagy gondolatot ki akarunk emelni, használjuk az `\emph{...}` paranccsal lehet.
- (Az `\emph{}` utasítás egymásba is ágyazható. `\emph{Akár hiszed \emph{akár} nem}`)
- Ez figyeli az aktuális betűváltozatot, és aszerint emelnek ki. Álló alak esetén dőlt, nem álló alak esetén álló alakra vált.
- További kiemelési lehetőségek alá- illetve áthúzással:
 - `\underline{szöveg}`Ezekhez kell az `ulem` csomag. (`\usepackage{ulem}`)
 - `\uuline{szöveg}`
 - `\uwave{szöveg}`
 - `\sout{szöveg}`

Kiemelés és betűméret 2.

- Csupa nagybetűvel szedés: `\MakeUppercase{szöveg}`
- Csupa kisbetűvel szedés: `\MakeLowercase{szöveg}`
- Amikor nem alapbetűváltozatot használunk, de ideiglenesen vissza akarunk arra térni (álló, normál, antikva), akkor használjuk a `\textnormal{...}` vagy `{\normalfont ...}` parancsot.

- **Betűméret állítása:**

példa `{\tiny ...}`

példa `{\scriptsize ...}`

példa `{\footnotesize ...}`

példa `{\small ...}`

példa `{\normalsize ...}`

példa `{\large ...}`

példa `{\Large ...}`

példa `{\LARGE ...}`

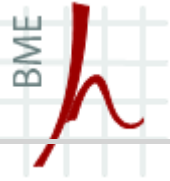
példa `{\huge ...}`

példa `{\Huge ...}`

Fix betűméret állítása és sortörés

- Fix betűméretet (pl. 20 pontosat 10 pontos sortávolsággal) a következő paranccsal érünk el:
`{\fontsize{20}{10}\selectfont ...}` utána a szöveg visszatér az „alap” stílusra.
- Sortörés:
 - `A \` új sort kezd sorkizárás nélkül
 - `\`* szintén új sort kezd sorkizárás nélkül, **DE** nem enged meg oldaltörést
 - `\`[2mm] szintén új sort kezd sorkizárás nélkül, **DE** A következő sor az aktuális sortávolság + 2 mm-re lesz.
 - `\`* [2mm]
 - `\linebreak` új sor sokizárással
 - Sortörés letiltása: `\nolinebreak`
- Sortávolság: A LaTeX automatikusan meghatározza, de ha változtatni szeretnénk: `\linespread{1.67}`
(megjegyzés: Ha használjuk a `setspace` csomagot, akkor `\onehalfspacing` paranccsal másfeles sorköz, míg a `\doublespacing` kettős sorköz állítható be).

- *Dokumentumosztály specifikus:*
 - `\part{Rész cím}` → csak book és report osztályokban
 - `\chapter{Alrész cím}` → csak book és report osztályokban
- *Közös:*
 - `\section{Fejezet címe}`
 - `\subsection{Alfejezet címe}`
 - `\subsubsection{Al-alfejezet címe}`
 - `\paragraph{Még kisebb egység címe}`
 - `\subparagraph{A legkisebb egység címe}`
- **számozatlan fejezetek:** (`\section*{Hello}` stb.)
- A `\tableofcontents` utasítás automatikusan generált tartalomjegyzéket illeszt be (később részletesen lesz róla szó), de csak a számozott fejezeteket tartalmazza.



Példa

```
\documentclass[a4paper]{book}
\usepackage{t1enc}
\usepackage[utf8]{inputenc}
\usepackage[magyar]{babel}
\begin{document}
\title{Az aranyhalak nemi élete \ \ és egyéb
cigerettatípusok}
\author{Dr.~Kiss Géza \ \ egyetemi tanár \and
Kovács Alajos \ \ nyugdíjas alezredes}
\maketitle
\part{Bevezető}
\section{Az aranyhal anatómiája}
Itt jön a szöveg maga\dots
\subsection{szaporítószervek}
\end{document}
```

- Szerencsére a LaTeX maga végzi az oldaltöréseket, de mi is kikényszeríthetjük:

`\newpage` – új oldal

`\pagebreak` - Az adott bekezdés után oldalt tör
oldalkitöltéssel.

`\nopagebreak`

- Új bekezdés a forrásállományban hagyni kell egy üres sort, vagy a `\par` parancsot kell kiadni.

- Alapestben a bekezdések sorkizártak,
- Bekezdések balra zárása: `\raggedright`

```
\begin{flushleft}  
... \\  
... \\  
\end{flushleft}
```

- Bekezdések jobbra zárása: `\raggedleft`

```
\begin{flushright}  
... \\  
... \\  
\end{flushright}
```

- Bekezdések középre zárás: `\centering`

```
\begin{center}  
... \\  
... \\  
\end{center}
```

- **Lábjegyzet:** `\footnote{}` paranccsal (előtte nem lehet szóköz!)

Például `\footnote{elegem van a példákból}` így.

- *Az `article` dokumentumosztályban a lábjegyzet sorszámozása folyamatos, míg `report` és `book` esetén fejezetenként 1-től kezdődik.*

Ha azt akarjuk, hogy oldalanként előről kezdődjön a szá-

mozás: `\footnotestyle{reset=page}`

- `\footnotestyle{reset=none}` Csak akkor kezdi újra a számozást, ha egyéb csomagok vagy az osztály ezt állítja be.

- *Ha `report` vagy `book` osztályban is folyamatos számozást szeretnénk, akkor írjuk a következőt:*

`\usepackage{remreset}`

`\makeatletter`

`\@removefromreset{footnote}{chapter}`

`\makeatother`

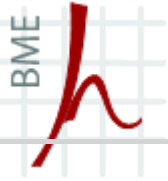
- **Széljegyzet:** A széljegyzetek alapértelmezésben a külső margóra

kerülnek: `\marginpar{széljegyzet}`, de ha a belső margóra szeretnénk adjuk ki a parancs:

`\reversemarginpar`, visszatérés a külső margóra: `\normalmarginpar`

- A `\pagestyle{sty}` utasítással a dokumentum egészét,
- `\thispagestyle{sty}` utasítással pedig az adott lap stílusát határozhatjuk meg.
- **fancyhdr csomag betöltésével** és a fancy stílus megadásával a fejléc és lábléc tartalma tetszőlegesen befolyásolható. A `\fancyhead[hely]{mit}` és `\fancyfoot[hely]{mit}` parancsokkal írhatunk a fejlécbe és láblécbe.
- A hely argumentum lehet L (left), R (right), C (center), E (even – páros oldalon), O (odd – páratlan oldalon). Egy utasításon belül több hely is megadható, ezeket vesszővel kell elválasztani. Az alábbi kód a lapok külső oldalára fejlécbe írja az oldalszámot:

```
\pagestyle{fancy} \fancyhead[LE,RO]{\thepage}
\fancyfoot{}
```



1. Kis HF

- Hozz létre egy Report dokumentumot, ami az első oldalon tartalmazza a szerzőt, a mű címét és a készítés dátumát.
- A dokumentum betűmérete 12pt legyen és Times betűtípust használj!
- A mű tartalmazzon tartalomjegyzéket!
- Az oldalszámozás kisbetűs római legyen!
- A létrejövő mű tartalmazzon egy Bevezető és egy Összefoglaló nevű szakaszt! A szakaszok tartalmazzanak alszakaszokat is! Az alszakaszokban írott szöveg legyen másfeles sorközű és gyakorold a megtanult szövegformázásokat (pl. új bekezdés, kiemelés, betűméret)
- *Opcionális gyakorlás:* egy széljegyzetet ill. élőfej és élőlábat illessz be a műbe!

KÖRNYEZETEK, FELSOROLÁSOK

- `\begin{valami}` és az `\end{valami}` közti részt nevezzük környezetnek.
- Már találkoztunk ilyesmivel: `\begin{document}` és `\end{document}`, ez nem más mint a dokumentum környezet!
- A környezetek egymásba ágyazhatóak:

Helyesen:

```
\begin{AAA}
\begin{BB}
\end{BB}
\end{AAA}
```

Helytelenül: =hiba

```
\begin{AAA}
\begin{BB}
\end{AAA}
\end{BB}
```


- `quote`, `a quotation` és `a verse` környezettel! Próbáljuk ki!
- A `quote` környezetben nem kezdődnek behúzással a bekezdések.
- A `quotation` környezetben a bekezdések behúzással kezdődnek,
- `verse` környezetben minden sor eleje azonos helyen kezdődik, a sorok végét `\\` karakterekkel jelöljük.

Íme az első vers:

```
\begin{verse}
Megy a juhász számaron,\\
Földig ér a lába;\\
Nagy a legény, de nagyobb\\
Boldogtalansága.
Gyepes hanton furulyált,\\
Legelészett nyája.\\
\dots
\end{verse}
Ez volt a versünk!
```

Próbáljuk ki mindkettőt (quote és quotation) hosszú sorokkal rendelkező szövegrészleten!

```
\begin{quote}
```

```
Ha elég hosszúak a sorok, akkor megláthatjuk,  
miféle
```

```
különbséget rejt e két környezet, persze figyelniük  
kell
```

```
arra is, hogy egy új bekezdést is kezdjünk.
```

```
Ha elég hosszúak a sorok, akkor megláthatjuk,  
miféle
```

```
különbséget rejt e két környezet, persze figyelniük  
kell
```

```
arra is, hogy egy új bekezdést is kezdjünk.
```

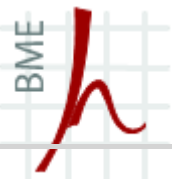
```
\end{quote}
```

LaTeX kód szövegbe illesztése (verbatim)

- Ha LaTeX kódot szeretnénk a szövegbe beilleszteni, úgy, hogy a fordító hagyja figyelmen kívül akkor a `verbatim` környezetet használhatjuk:
- A `\verb|szöveg\@%|` utasítással a `szöveg\@%` verbatim módban jelenik meg, azaz minden karakter megjelenik.
- Jelen esetben a `|` jellel jelöltük a verbatim széleit, de tetszőleges más karakter is használható (kivéve a betűket a szóközt és a csillagot)
- Ha hosszabb – több soros – verbatimot szeretnénk, akkor célszerű a `\begin{verbatim} ... \end{verbatim}` környezetet használni.
- A fenti utasítás és környezet csillagos változata megjeleníti a szóközöket is, pl. `\verb*-Helló világ!`-hatására `Helló világ!` kimenetet láthatjuk.

Kód szövegbe illesztése (listings)

- Különböző programnyelvek programkódjainak megjelenítésére a listings csomagot használhatjuk!
- `\usepackage{listings}` a preambulumba!
- `lstlisting` környezetet kell létrehozni, vagy a `\lstinputlisting{ }` argumentumába!
- Figyelem! Ilyen nincs: egymásba nem ágyazhatóak `\lstinputlisting{\lstinputlisting{hello } }`
- Ha utf8 kódolással dolgozunk akkor a `listingsutf8` csomagot használjuk!



Listings környezet által támogatott programnyelvek

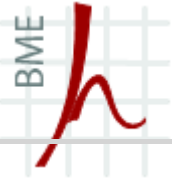
ABAP2,4	Fortran4	Modula-2	S4
ACSL	GCL	MuPAD	SAS
Ada4	Gnuplot	NASTRAN	Scilab
Algol4	Haskell	Oberon-2	sh
Ant	HTML	OCL4	SHELXL
Assembler2,4	IDL4	Octave	Simula4
Awk4	inform	Oz	SQL
bash	Java4	Pascal4	tcl4
Basic2,4	JVMIS	Perl	TeX4
C	ksh	PHP	VBScript
C++	Lisp4	PL/I	Verilog
Caml4	Logo	Plasm	VHDL4
Clean	make4	POV	VRML4
Cobol	Mathematica1,4	Prolog	XML
Comal	Matlab	Promela	XSLT
csh	Mercury	Python	
Delphi	MetaPost	R	
Eiffel	Miranda	Reduce	
Elan	Mizar	Rexx	
erlang	ML	RSL	
Euphoria	Modelica3	Ruby	

- Példa 1.: C++ kód beillesztése

```
\begin{lstlisting}[language=C++]  
    #include <iostream.h>  
  
    main()  
  
    {  
  
    cout << "Hello World!";  
  
    return 0;  
  
    }  
  
\end{lstlisting}
```

- **Példa 2.:** Matlab kód beillesztése

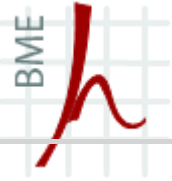
```
\begin{lstlisting}[language=matlab]
% ide jön a matlab kód
function [x,y] = csbinproc(xp, yp, n)
x = zeros(n,1);
y = zeros(n,1);
i = 1;
while i <= n
xt = rand(1)*cx + minx;
yt = rand(1)*cy + miny;
k = inpolygon(xt, yt, xp, yp);
if k == 1
% it is in the region
x(i) = xt;
y(i) = yt;
i = i+1;
end
end
\end{lstlisting}
```



Program kód beillesztése file-ból

- Ha lusták vagyunk bemásolhatjuk közvetlenül file-ból a programkódot:

```
\lstinputlisting{file1.m}
```

URL megadása

- `url` csomag: `\url{cím}`
- VAGY, ha azt akarjuk, hogy a link elektronikusan elérhető legyen a PDF-ből:
- `hyperref` csomag: `\url{cím}`

- Három féle felsorolási környezet létezik LaTeX-ben:
 1. számozott lista (enumerate),
 2. számozatlan lista (itemize),
 3. leíró lista (description).

Példa:

```
\begin{itemize}
\item A felsorolás első eleme
\item A felsorolás 2. eleme
\item A felsorolás 3. eleme
\end{itemize}
```

- Látható, hogy minden listaelemet egy `\item` előz meg függetlenül a felsorolási környezet típusától!

- A felsorolás környezetek négy mélységig egymásba ágyazhatóak!

Példa:

```
Felsorolások: \dots
\begin{itemize}
\item Állatok
\begin{enumerate}
\item madár ...
\item hal ...
\end{enumerate}
\item növények
\begin{description}
\item egyszikű
\item kétszikű
\end{description}
\end{itemize}
```

- Felsorolás jelek megváltoztatása:

```
\begin{itemize}
```

```
\item Első elem
```

```
\item[$\circ$] Második, kiemelt elem
```

(vagy pl. `\item[\dag]`, `\item[--]`) vagy a *pifont* csomag egyéb jeleivel is akár...

```
\end{itemize}
```

- Leíró lista esetén:

```
\begin{description}
```

```
\item[a butaságok] listába szedve sem válnak tudományossá,
```

```
\item[a jó gondolatok] viszont szebben mutatnak rendszerezve
```

```
\end{description}
```

Leíró lista módosítása

- Ha nem tetszik, hogy félkövérrel szedi a leírás „tárgyát” módosítható:
- a preambulumban adjuk ki a következőt:

```
\renewcommand{\descriptionlabel}[1]  
{\hspace\labelsep\normalfont\itshape#1}
```

- Az enumerate számozása alapesetben arab számzás, de ez megváltoztatható: (kell a paralist csomag hozzá!)
- `\begin{enumerate} [stílus]`
 - 1 - arab szám
 - i - kis római szám
 - I - nagy római szám
 - a - latin ábécé kisbetűi
 - A - latin ábécé nagybetűi

Példa:

```
\documentclass[a4paper]{book}
\usepackage{t1enc}
\usepackage[utf8]{inputenc}
\usepackage[magyar]{babel}
\usepackage{paralist}
\begin{document}
\begin{enumerate}[I-{a}]
\item listaba szedve sem valnak tudományossa,
\item viszont szebben mutatnak rendszerezve
\end{enumerate}
\end{document}
```

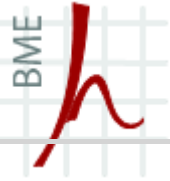
- Az enumerate számozása alapesetben nagyobb térközt hagy a felsorolások között.
- Ha ezt nem akarjuk, használjuk a `paralist` csomag `compactenum` környezetét:
- Példa:

```
\begin{compactenum}

\item listába szedve sem válnak tudományossá,

\item viszont szebben mutatnak rendszerezve

\end{compactenum}
```



2. Kis HF

- Egy .tex fájlba írd meg az alábbihoz hasonló verset a tanult környezetek felhasználásával:

Megy a juhász számaron,
Földig ér a lába;
Nagy a legény, de nagyobb
Boldogtalansága.

- Gyakorold a felsorolásokat (enumerate, itemize, description stb.) és formázásukat ilyesmi példákkal:

1. Emlősök

- Házi állatok,
 1. macska
 2. kutya
- Vadon élő állatok,
 - farkas,
 - oroszlán

2. Hüllők

- kígyó,
- gyík

TABULÁLÁS ÉS TÁBLÁZATOK SZERKESZTÉSE

- A `tabbing` környezettel valósítható meg, (ha valakinek szüksége van rá).
- Elég körülményes, inkább nézzünk rá példákat:

```
\begin{tabbing}
Element a szóköz \= arrébb\\
\> ez meg mégbeljebb \= ment\\
\> \> végül \\
Ez lett \=\\
\> de ez van \> így jártunk!
\end{tabbing}
```

- Sajnos elég bonyolult LaTeX-ben!
- A `tabular` környezetet kell használni,
- Nézzünk is gyorsan egy példát:

```
\begin{tabular}{|l|r|}
```

Ez egy balra igazított oszlop & ez megjobbra igazított oszlop

```
\\
```

akkor látszik & ha több elem

```
\end{tabular}
```

- A környezet argumentuma határozza meg, hogy hány oszlop legyen és azokban milyen igazítás van, vagy, hogy van e közte rácsvonal:

```
\begin{tabular}{|lrc|}
```

- Új sort a `\\` paranccsal lehet kezdeni, míg a soron belüli elemeket a `&` speciális karakter választja el.

- Ha szeretnénk vízszintes vonalakat a táblázatunkba akkor a

`\hline` parancs szükséges:

```
\begin{tabular}{|r|l|}
```

```
\hline
```

```
Egy & megérett a meggy \\
```

```
\hline
```

```
Kettő & csipkebokor vessző \\
```

```
\hline
```

```
\end{tabular}
```

- Dupla vonalakat a `|` és a `\hline` duplázásával lehet elérni.
- Ha több azonos rendű oszlopunk van a deklaráció rövidíthető: `{*{3}{|r||c|}}` ez teljesen ugyanaz mint a `{r||c|r||c|r||c|}`

- Példa:

```
\begin{tabular}{|l||*8{c|}}
\hline
A vonat száma: & 437 & ... \\
\hline\hline
Indul Bp.\ Keleti-pu.-ról & 8:00 & ... \\
Érkezik Hatvanba & 8:52 & ... \\
Indul Hatvanból & 8:55 & ... \\
Érkezik Miskolc Tiszai-pu.-ra & 10:22 & ... \\
\hline
\end{tabular}
```

- Megadható fix méretű táblázat is:

- (`\usepackage{array}`)

```
\begin{tabular}{|m{2cm}|m{4cm}|m{1cm}|m{3cm}|}
```

```
\hline
```

```
Éljen a fix méretű tábla! & Éljen a fix méretű  
tábla!&
```

```
Éljen a fix méretű tábla!& Éljen a fix méretű  
tábla! \\\
```

```
\hline
```

```
\end{tabular}
```

- Egyéb fix méretű táblázat:
 - `p{2cm}` – 2cm széles cella, ahol a cella tetejére kerül a szöveg (sorkizártan)
 - `m{2cm}` – 2cm széles cella, ahol a cella közepére kerül a szöveg (sorkizártan)
 - `b{2cm}` } – 2cm széles cella, ahol a cella aljára kerül a szöveg (sorkizártan)

- Példa:

```
\begin{tabular}{|l| p{2cm}| m{2cm}| b{2cm}}
```

```
\hline
```

```
középre igazítva & tetejére igazítva & középre igazítva  
& cella aljára igazítva\\
```

```
\hline
```

```
\end{tabular}
```

- Oszlopok közti távolság megváltoztatása (az alap 6pt)
`\setlength{\tabcolsep}{20pt}`
- Sorok közti távolság:
`\renewcommand{\arraystretch}{1.5}`
- Egyéb környezet táblába illesztése: `p{}` –segítségével:

- „**Sor a sorban**”:
- `\usepackage{multirow}` szükséges hozzá!
- `\cline{x-y}` cellák föle vonalat húz x és y az oszlop két sarka

```

\begin{tabular}{|l|c|}
\hline\multirow{2}*{Alma} & 1\\
& 2\\\hline
\multirow{3}{3cm}{3\,cm széles szöveg törve} &
  3\\\cline{2-2}
& 4\\\cline{2-2}
& 5\\\hline
\end{tabular}

```

- `@` paraméterrel a táblázat oszlopai közötti elválasztó átírható;
- a `@{}` nullára csökkenti az oszlopok közötti távolságot,
- a `@{:}` kettőspontot tesz két oszlop elemei közé.
- `\multicolumn{n} {oszldef} {szöveg}` paranccsal n cellát összevonva `oszldef` módon írjuk a szöveget.
- Arra kell ügyelni, hogy az `@` tényleg eltünteti mindent: ha az utolsó oszlop után, vagy az első oszlop előtt szerepel, megszünteti a távolságot a táblázat széle és a szöveg között → ezt manuálisan kell korrigálni.

- Vonat menetrend példa, szebben:

```

\begin{tabular}{l||*8{r@{:}l||}}
A vonat száma: & \multicolumn{2}{c|}{437} & ...
\\
\hline\hline
Indul Bp.\ Keleti-pu.-ról & 8&00 & ... \\
Érkezik Hatvanba & 8&52 & ... \\
Indul Hatvanból & 8&55 & ... \\
Érkezik Miskolc Tiszai-pu.-ra & 10&22 & ... \\
\hline
\end{tabular}

```

- A `\multicolumn{1}{oszldef}{szöveg}`-gel egyetlen cellában írhatjuk felül az oszlopokra vonatkozó definíciót. Az `@{}` nagyon hasznos, mert ismétlődő mintákat könnyedén beilleszthetünk az oszlopok elé és után.
- A tradicionális nyomdászatban számok írásakor az azonos helyiértékeknek egymás alá kell kerülniük. Ez csak akkor probléma, ha tizedestörteket kell egymás alá írni.
- A megoldás: `r@{,}`:

```
\begin{tabular}{c|r@{,}l}  
Kifejezés & \multicolumn{2}{c}{Érték} \\ \hline  
$\pi$ & 3&1415927 \\ $\pi^{\pi}$ & 36&46216 \\ $\pi^{\pi^{\pi}}$ & 80662&666  
\end{tabular}
```

- Az eddigi „hagyományos” táblázatokhoz képest a kiadói szintű táblázatok tipográfiája kicsit eltérő:
 - táblázat tetejére és aljára vastagabb vonal kell, mint a köztesek.
 - A táblázat két szélén ne legyenek extra térközök, melyek a formátumvezérlő két szélére írt egy-egy `@{}` paranccsal megoldható.
 - A legfontosabb és legkülönösebb szabály, hogy soha ne alkalmazzunk függőleges vonalat.
- Ha ilyen táblázatra van szükségünk, használjuk a `booktabs` csomagot.

Példa:

```

\begin{tabular}{@{}lrr@{}}
\toprule
&\multicolumn{2}{c}{Év}\\
\cmidrule{2-3}
&\multicolumn{1}{c}{2002} &
\multicolumn{1}{c}{2003}\\
\midrule
Jövedelem (Ft) & 775\,000 & 1\,166\,500\\
Adó (Ft) & 165\,000 & 194\,950\\
\bottomrule
\end{tabular}

```

- `\begin{tabular}[poz]{oszl}` környezettel hozhatunk létre tömböket.
- Az *oszl* paramétert már korábban megismertük, a *poz* paraméter a táblázat illesztésének módját adja meg.
- Ha *t* (top), akkor a táblázat első sora kerül az alapvonalra, ha *b* (bottom), akkor az alsó sor.
- Ha nincs megadva ez az opcionális paraméter, akkor a táblázat közepét igazítjuk az alapvonalhoz.

```
\hrulefill \begin{tabular}[t]{c} 1 \ 2 \ 3 \ 4 \end{tabular}%
```

```
\hrulefill \begin{tabular}{c} 1 \ 2 \ 3 \ 4 \end{tabular}%
```

```
\hrulefill \begin{tabular}[b]{c} 1 \ 2 \ 3 \ 4 \end{tabular}%
```

```
\hrulefill \
```

Táblázatok úsztatása (table)

- A táblázatok elég bután néznek ki szövegben, ezért célszerű „úsztatni” őket. Erre szolgál a `table` környezet.
- A `table` környezetnek van egy opcionális paramétere ami megmondja, hogy a táblázatot hova helyezze el az oldalon:
 - `t` – az oldal tetejére,
 - `b` – az oldal aljára,
 - `h` – pont ide,
 - `P` – külön lapra.

Alapértelmezés: `[tbp]`

- Ha azt szeretnénk, hogy mindenképpen adott helyre tegye, akkor egy felkiáltójellel lehet szigorítani az utasítást: `[!h]` jelentése: mindenképpen ide tedd.

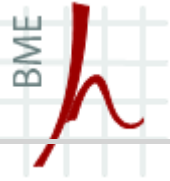


Table és Tabular egymásba ágyazása

```
\begin{table} [htb]
\caption{Nagy bölcsességek} %így lehet
  címet adni
\label{tab:fontos}
\center
\begin{tabular}{c|r@{,}l}
Kifejezés & \multicolumn{2}{c}{Érték} \\
\hline
 $\pi$  & 3&1415927 \\
 $\pi^{\pi}$  & 36&46216 \\
 $\pi^{\{\pi^{\pi}\}}$  & 80662&666
\end{tabular}
\end{table}
```

- Ha megadjuk egy táblázat oszlopdefiníciójában, hogy az oszlop adott szélességű legyen, akkor gyakran csúnya a kimenet, mivel a sorkizárt szedés miatt széles szóközök kerülnek a szavak közé.
- Megoldást jelent az array csomag használata:

```
\usepackage{array}
```

```
...
```

```
\newcommand*{\tabsovege}[1]{\let\tmp=\#1\let\=\tmp}
```

```
\begin{tabular}{c|>{\tabsovege{\raggedright}\mbox{}}p{25mm}}
```

```
\hline 1 & Az első sorban is látszik milyen hasznos kis  
segítség ez. \ \hline 2 & Hát még a második sorban,  
ahol
```

```
szintén egy hosszú szöveget láthatunk. \ \hline  
\end{tabular}
```

Fix szélességű táblázat

- A `tabularx` csomaggal illeszthetünk be adott szélességű táblázatot. Használatához a `\begin{tabularx}{szél}[poz]{oszl}` környezetet kell meghívni
- első argumentuma a táblázat szélességét határozza meg. Hasznos lehet a `\linewidth` méret alkalmazása, amely az aktuális vonalszélességet adja vissza.
- A következő paraméter opcionális; a korábban megismert `poz` pozícionáló paraméterrel ekvivalens. Az utolsó oszlopban az oszlopdeklarációk szerepelnek, azonban itt megjelenik egy speciális oszlop: a rugalmas oszlopot az `X` definíció jelöli.

```
\begin{tabularx}{\linewidth}{c|X} Akkora, amekkora kell  
& ez meg a maradékot kapja meg mindenképpen akármit is  
írok, legyen az akár nagyon hosszú szöveg \\ Rövidebb  
szöveg & második sor. \end{tabularx}
```

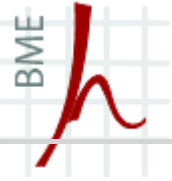
3. kis HF

- Egy táblázatokkal rendelkező dokumentum letöltése a webről és tördelése LATEX-hel.
- Legalább egy olyan táblázat is legyen, amelyikben tizedes törtek is vannak.
- Legalább egy kiadói minőségű táblázat készítése.
- A bruttó (utasításokkal együtt számolt) karakter szám 10 000–15 000 legyen.

KÉPEK, ÁBRÁK BEILLESZTÉSE, DOBOZOK

- Ha ábrákat szeretnénk beilleszteni a munkánkba, szükségünk van a `graphics` csomagra.
- A képeket célszerű elhelyezni a forrásállományt tartalmazó könyvtárba, mert akkor elég a nevükre hivatkozni (elérési úttal nem kell bajlódni).
- De ha egy grafikonok nevű alkönyvtárba helyezzük a képeket pl. a akkor a preambulumba írjuk be a következőt:

```
\graphicspath{{./grafikonok/}}
```
- Ábra beillesztése: `\includegraphics{abra}` utasítással történik,
- vagyis az adott helyre beilleszti a megnevezett képet.



Képek beillesztése

- Beilleszthető így: jpg, png, pdf és eps formátum.
- EPS beillesztése esetén :
`\usepackage{epstopdf}` csomag is kell!

- Az `\includegraphics {abra}` számos opcionális paraméterrel rendelkezik:
 - trim – képen meghatároz egy keretet (de levágás nem történik)
 - clip – ez már le is vágja
 - scale – nagyítás/kicsinyítés mérete
 - angle – forgatás szöge fokban
 - origin – forgatás középpontja
 - width – a kép kívánt szélessége
 - height – a kép kívánt hossza
 - (page - több oldalas PDF esetén hova melyik oldalra helyezze a képet)- nem ezt fogjuk majd használni!)

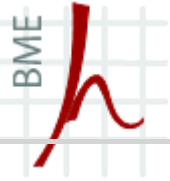
- Példák:

```
\includegraphics[trim=10mm 10mm 10mm 10mm, clip, scale=0.5] {abra}
```

```
\includegraphics[width=3cm, angle=90] {abra}
```


- Ábrák úsztatására a `figure` környezet szolgál.
- Ez pont ugyanaz, mint a `tabular`, `table` egymásba ágyazása, ezért mindenképp használjuk!
- Ugyanúgy létezik opcionális paramétere: `[tbph!]`
- Használata egyszerű:

```
\begin{figure}  
\centering % tegye középre az ábrát!  
\includegraphics{ábra neve}  
\caption{Ide az ábra címe}  
\label{cimke}  
\end{figure}
```



Ábra manipulálás figure környezetben

- A korábban bemutatott ábra manipulációk közül néhány itt is lehetséges:

Például:

```
\usepackage{graphics}
...
\begin{figure} [bp]
\center
\resizebox{80mm}{!}{
\rotatebox{-90}{
\includegraphics{maki.eps}}}
\caption{A \LaTeX\ órák mestere}
\label{fig:maki}
\end{figure}
```

- Ábra türközése:

```
\begin{figure} [h!]
```

```
\centering
```

```
\reflectbox{%
```

```
\includegraphics[width=0.5\textwidth]{ide jon a  
kep}}
```

```
\caption{A kep cimet itt adjuk meg}
```

```
\end{figure}
```

- Ábra mellé magyarázó szöveg: `\usepackage{sidecap}` és az `\begin{SCfigure} \end{SCfigure}` környezet segítségével

- Ha nem tetszik a Babel magyar által felkínált ábra kifejezés:

```
\usepackage[magyar]{babel}
\addto\captionsmagyar{%
\def\figurename{Képek}}
```

Caption módosítás:

- `\usepackage{caption}`
- `\captionsetup[figure]{labelformat=default,`
- `labelsep=space}`
 - labelsep = none (nincs)
 - labelsep = colon (:)
 - labelsep = period (.)
 - labelsep = space (üres karakter)
 - labelsep = quad (nagyobb space)
 - labelsep = newline (új sor)

- Példa:

```
\documentclass[letter,12pt]{article}
\usepackage[utf8]{inputenc}
\usepackage[magyar,english]{babel}
\usepackage[pdftex]{graphicx}
\usepackage{sidecap}
\begin{document}
\begin{SCfigure}
\centering
\includegraphics[width=0.55\textwidth]{elefant.jpg}
\caption{Az afrikai elefánt (Loxodonta africana) a Földön ma élő legerősebb és legnagyobb szárazföldi emlősállat. Eredendően szavannákon él, ám kitűnően alkalmazkodik ahhoz, hogy Afrika más, különböző éghajlatú területein is megélhessen, azonban élőhelyének közelében mindenképpen ivóvízforrás kell hogy legyen.}
\end{SCfigure}
\end{document}
```



Figure 1: Az afrikai elefánt (Loxodonta africana) a Földön ma élő legerősebb és legnagyobb szárazföldi emlősállat. Eredendően szavannákon él, ám kitűnően alkalmazkodik ahhoz, hogy Afrika más, különböző éghajlatú területein is megélhessen, azonban élőhelyének közelében mindenképpen ivóvízforrás kell hogy legyen.

- Néhány lehetséges opciók:
 - leftcaption, rightcaption – bal vagy jobb oldalon jelenjen meg a szöveg
 - raggedright: ha kevés a magyarázó szöveg így szebben néz ki balra zárva.

Ábrák szövegbe ágyazása

- `\usepackage{wrapfig}` kell hozzá!

- A környezet:

```
\begin{wrapfigure}{r}{0.5\textwidth}
\vspace{-20pt} %térközök beállítása
\begin{center}
\includegraphics[width=0.48\textwidth]{elefant.
jpg}
\end{center}
\vspace{-20pt}
\caption{Az afrikai elefánt}
\vspace{-10pt}
\end{wrapfigure}
```

1 Afrikai elefánt

Az afrikai elefánt (*Loxodonta africana*) a Földön ma élő legerősebb és legnagyobb szárazföldi emlősállat. Eredendően szavannákon él, ám kitűnően alkalmazkodik ahhoz, hogy Afrika más, különböző éghajlatú területein is megélhessen, azonban élőhelyének közelében mindenképpen ivóvízforrás kell hogy legyen.

2 Előfordulása

Az afrikai elefánt korábban a Szaharától délre egész Afrikában elterjedt volt, manapság elterjedése elsősorban a nemzeti parkokra és egyéb védett területekre korlátozódik.

Élőhelyük alapján két alfaját különböztetik meg. A *Loxodonta africana africana* erdős és füves szavannákon, félsivatagos területeken egyaránt előfordul, míg a kisebb termetű *Loxodonta africana cyclotis* alfaj Nyugat- és Közép-Afrika esőerdeiben él. Utóbbit, az erdei elefántot újabban különálló fajnak ismerik el.

Egyes források egy harmadik alfajt is megkülönböztetnek: ezek szerint a *Loxodonta africana africana* csak a Dél-Afrikában élő elefántokat foglalja magába, míg a kelet-afrikai populációt *Loxodonta africana kochi* néven említik.



Figure 1: Afrikai elefánt

- Ha több ábrát szeretnénk megjeleníteni egymás mellett/alatt a subfigure környezetre van szükségünk, de ehhez kell caption és **subcaption** csomagokra.

Példa:

```
\begin{figure*}
\centering
\begin{subfigure}[t]{0.3\textwidth}
\centering
\includegraphics[width=\textwidth]{elefant.jpg}
\caption{Afrikai elefánt}
\end{subfigure}%
\begin{subfigure}[t]{0.3\textwidth}
\centering
\includegraphics[width=\textwidth]{azsiai.jpg}
\caption{Ázsai elefánt}
\end{subfigure}
\begin{subfigure}[t]{0.3\textwidth}
\centering
\includegraphics[width=\textwidth]{dumbo.jpg}
\caption{Walt-Disney elefánt}
\end{subfigure}
\caption{A világ elefántjai}
\end{figure*}
```



(a) Afrikai elefánt



(b) Ázsai elefánt



(c) Walt-Disney elefánt

Figure 1. A világ elefántjai

- A doboz a dokumentum olyan része, melyet a TEX egyetlen egységként kezel, tartalma nem törhető el a sor végén vagy a lap alján.
- Ilyenek voltak például az úszó objektumok is.
- A LaTeX 4 féle dobozt használ:
 1. **LR doboz** (balról jobbra töltjük fel tartalommal, tipikusan csak egy sor)
 2. **Bekezdésdoboz** (Valamilyen szélességgel bíró, akár több sort is tartalmazó doboz)
 3. **Vonaldoboz** (gyámfázásra használjuk),
 4. **Tárolódoboz** (szövegrészek tárolására szolgál).

- `\mbox{szöveg}` – korábban is volt az elválasztás megakadályozására
- `\fbox{szöveg}` – itt már keret is megjelenik a szöveg körül.
- A keret vonalvastagságát az `\fboxrule` tárolja, míg a keret távolságát a szövegtől a `\fboxsep`.

- Példa:

```
\mbox{sima doboz keret nélkül}
\fbox{sima keretezett doboz}
{\setlength{\fboxrule}{3pt} \fbox{vastag keretes doboz}}
{\setlength{\fboxsep}{0pt} \fbox{térközmentes keretes doboz}}
```

Több lehetőséget kínál a:

`\makebox[szél][poz]{szöveg}` és a `\framebox[szél][poz]{szöveg}`
 LR doboz. poz l – balra, r – jobbra, s - interword space

- A doboz szélesség megadható tetszőleges hosszúság-mértékben, vagy használható a `\width`, `\height`, `\depth`, vagy `\totalheight` mértékek.

- Szöveg emelése: `\raisebox{emel}[mag][mély]{szöveg}`
Ez egy sor. `\`
Ez `\raisebox{2ex}{egy}` sor.
- Árnyékolt keret: *shadow* csomaggal `\shabox{szöveg}`
 - Stílusparaméterek:
 - `\sboxrule` – vonalvastagság,
 - `\sboxsep` – térköz,
 - `\sdim` – árnyékvastagság átállítása.

```
\setlength{\sboxrule}{3pt} \shabox{vastag keretes doboz}
\setlength{\sboxsep}{0pt} \shabox{térközmentes keretes doboz}
\setlength{\sdim}{10pt} \shabox{Nagy árnyékos doboz}
```
- További jópofa LR dobozok a *fancybox* csomagban:
 - Lekerekített sarkú dobozok: `\ovalbox{szöveg}` és `\lvalbox{szöveg}`
(lekerekítés sugarának változtatása: `\cornersize{r}`)
 - Duplakeretes doboz: `\doublebox{szöveg}`

- Több bekezdésnyi szöveget egy dobozba helyezhetünk vele,
- `\parbox[poz][mag][szpoz]{szélesség}{szöveg}`
 - [poz] – t (top) vagy b (bottom), különben középre igazítja a dobozt a szöveg vona,
 - [mag] – a szöveg magassága,
 - [szpoz] – a szöveg pozíciója a dobozban (t,b,c,vagy s),
- Természetesen ha keretet is akarunk látni, akkor pl. az fbox keretbe helyezhetjük!

Példa: Névjegykártya készítése:

```
\usepackage{pifont}
\usepackage{shadow}

\dots
\fbbox{\parbox[t][50mm]{90mm}{\vfill \raggedleft
\hrulefill \shabox{\scriptsize Kovács Alajos}
\\
\tiny nyugdíjas főtörzsőrmester \\[6pt]
\ding{38} +36-1-234-5678 \\
\ding{41} alajos@kovacs.hu \vfill}}
```

- `\rule[emelés]{szél}{mag}` paranccsal *szél* szélességű és *mag* magasságú kitöltött téglalapot rajzolhatunk.
- **Megjegyzés:** A téglalap lehet nulla vastagságú is, amit gyámfának nevezünk.
- Léteznek beépített gyámfák, melyek a sorvastagság méretét tárolják, ezek a `\strut` szöveg és `\mathstrut` matematikai módban:
- Példa:

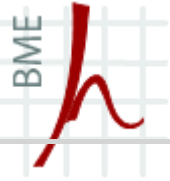
```
\fbox{\rule{0pt}{1cm}ez a doboz legalább 1\,cm magas}
\fbox{ez} \fbox{egy} \fbox{egyenetlen} \fbox{dobozsor}
\fbox{\strut ez} \fbox{\strut már} \fbox{\strut egyenletes}
\fbox{\strut dobozsor} ...  $(\sqrt{g} + \sqrt{h})$  csúnya, de
 $\left(\sqrt{\mathstrut g} + \sqrt{\mathstrut h}\right)$  szép.
```

Tárolódoboz (savebox)

- A dobozok alkalmasak szövegrészletek tárolására.
- Új tárolódoboz létrehozása: `\newsavebox{\prncs}`. Ezzel létrehoztunk egy `\prncs` nevű dobozt.
- Akkor most töltsük fel a következő paranccsal:
`\savebox{\prncs}[szél][poz]{keretesdobozt tegyünk bele?\fbox{szöveg} vagy keret nélkülit: \fbox{szöveg}}`
- A doboz tartalmának kiírása ahol szükség van rá:
`\usebox{\prncs}`
- Vigyázat a `\usebox`-ot használjuk új bekezdésben különben ráhelyezi a dobozt a folyószövegre!
- A mellékelt papíron egy példa látható, melyben lementett szövegdobozokkal egy A4-es lap két oldalára tíz kétoldalas névjegykártyát nyomtatunk. Vegyük észre a rugalmas méretek és a dobozok alkalmazását!

- Ha betöltjük a graphicx csomagot:
 - Dobozok forgatása:
 - `\rotatebox[origin=forgatáscentrum]{szög}{szöveg}`
 - Példa: `\rotatebox[origin=B]{90}{szöveg}`
 - Dobozok nyújtása:
 - `\scalebox{x}[y]{szöveg}`
 - Példa: `\scalebox{2}[1]{szöveg}`
 - Tükrözés:
 - `\scalebox{-1}[1]{szöveg}`
 - `\scalebox{-1}[-1]{szöveg}`
 - `\reflectbox{szöveg}`

- Láthatatlan doboz: ``
- Dobozok színezése: xcolor csomag kell hozzá:
 - `\colorbox{red}{szöveg}`
 - `\fcolorbox{red}{yellow}{szöveg}`
 - `\fcolorbox{declared-color-frame}{declared-color-background}{text}`
 - **Példa:** `\fcolorbox[RGB]{0,64,128}{192,192,192}{szöveg}`



4. kis HF

- Gyakorold az ábra beillesztéseket!
- Ne feledkezz meg az ábrák úsztatásról sem!
- Gyakorold a dobozok használatát!

HIVATKOZÁSOK, TARTALOMJEGYZÉK, ÁBRA- TÁBLÁZAT ÉS, IRODALOMJEGYZÉK KÉSZÍTÉSE

- A LaTeX egyik nagy előnye, kétségtelenül!
- Emlékszünk még a szakasz, alszakaszok létrehozására?
(`\chapter` (report és book osztálynál) `\section{}`, `\subsection{}` és társai)
- Tartalomjegyzék a `\tableofcontents` paranccsal hozható létre.
- A tartalomjegyzék mélységét (fejezet, alfejezet, al-alfejezet stb.) a `tocdepth` számláló beállított értéke határozza meg. (A LaTeX-ben lévő számlálókról lesz szó később!)
- Ha olyan tartalomjegyzékre van szükségünk ami részletesebb (pl. 4 mélységű- `\paragraph` részt is szeretnénk feltüntetni) akkor:
- `\setcounter{tocdepth}{4}`

- Ha így hozunk létre fejezetet: `\section*{Bevezetés}` akkor nem lesz sorszámozva és nem kerül be a tartalomjegyzékbe sem.
- Hogy mégis bekerüljön a tartalomjegyzékbe az oldalszámmal együtt, a `\section*{Bevezetés}` parancs után gépeljük be a következőt:

```
\addcontentsline{toc}{section}{Bevezetés}
```

- Ha egy jegyzék címe nem tetszik: írjuk a preambulumba, hogy `\addto\captionenglish{\renewcommand{\contentsname}{Tartalom}}`

Babel-t használunk magyar, angol!

- Könyvek és fontosabb művek esetén a tartalomjegyzék oldalszámozása római számmal történik és a mű érdemi részének számozása arab számozással.
- Ez könyv esetében egyszerű:
 - `\frontmatter`
 - `\tableofcontents`
 - `\mainmatter`
- Reportnál fapadosan:
 - `\setcounter{page}{1}`
 - `\pagenumbering{roman}`
 - `\tableofcontents`
 - `\newpage`
 - `\setcounter{page}{1}`
 - `\pagenumbering{arabic}`
 - `\chapter{Bevezetés}`

Táblázat- és ábrajegyzék készítése

- Természetesen nem csak tartalom, hanem táblázat és ábrajegyzékre is szükségünk lehet a mű megírásakor.
- Ebben az esetben:
 - `\listoftables` – Ilyenkor a táblázat száma, neve és oldalszáma kerül fel a listára.
 - `\listoffigures` – ua. csak ábrajegyzék

- Ha függelék szeretnénk írni be az `\appendix` parancsot.
- Ezt követően a fejezetek (chapterek) A, B, C számozást kapnak
- Sajnos a függelék nem kerül be automatikusan a tartalomjegyzékbe, de megoldható:

- Article osztály esetén:

```
\appendix
```

```
\section*{Függelék}
```

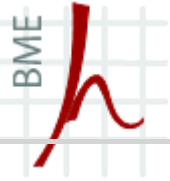
```
\addcontentsline{toc}{section}{Függelék}
```

- Book és report osztály esetén pedig:

```
\appendix
```

```
\addcontentsline{toc}{chapter}{Appendix}
```

```
\addtocontents{toc}{\protect\contentsline{chapter}{Appendix:}}}
```



Példa:

```
\documentclass[a4paper,10pt]{report}
\usepackage[utf8]{inputenc}
\usepackage[magyar,english]{babel}
  \begin{document}
    \tableofcontents
  \chapter{Első fejezet}
    \section{Első szakasz}
      \subsection{első alszakasz}
    \section{ második szakasz}
  \appendix
  \addcontentsline{toc}{chapter}{Appendix}
  \addtocontents{toc}{\protect\contentsline{chapter}{Appendix:}{}{}}
  \chapter{Első függelék}
  \chapter{Második}
  \end{document}
```

- Nagyobb mű esetén pl. könyv célszerű tárgymutatót készíteni a dokumentum végére, ahol a fontosabb fogalmak abc-ben vannak felsorolva.
- Írjuk be a preabulumba, hogy:

```
\usepackage{makeidx}
```



```
\makeindex
```
- A tárgymutatót a `\index{tárgyszó}` paranccsal bővíthetjük.
- Ahová elakarjuk helyezni a tárgymutatót, írjuk be a `\printindex` parancsot.
- (a `printindex` elé írjuk be a következőt: `\renewcommand{\indexname}{Szójegyzék}` – Tárgymutató szó lecserélése Szójegyzékre)

Irodalomjegyzék készítése

- Az Irodalomjegyzéket általában a dokumentum végére szokták tenni,
- LaTeX-ben a `thebibliography` környezettel lehet készíteni irodalomjegyzéket,
- Az egyes felsorolt műveket a `\bibitem` parancs után írjuk (ismerős a felsorolások fejezetből?):

```

\begin{thebibliography}{peldacimke}
\bibitem[szam]{cimke} elemleiras
...
\end{thebibliography}

```

Ide írjuk azt, hogy{9} vagy {99}

A mű neve

- **Lássuk részletesen:** `\bibitem[szam]{cimke} elemleiras`

Opcionális: milyen számmal jelenjen meg az adott hivatkozás (ha elhagyjuk automatikus lesz)

A műben ezzel a címkével hivatkozunk erre a műre

- Név módosítás ha szükséges, de a Babel magyar tudja:
`\renewcommand{\bibname}{Felhasznált irodalom}`
- A szabályosan szedett irodalmi hivatkozások alakja a következő:
 - Könyv: szerző, *címdőlt betűkkel*, *kiadó neve*, *kiadó városa*, *kiadás éve*.
 - Újságcikk: szerző, „címidézőjelek között”, *újság neve dőlt betűkkel*, *évfolyam*, *szám*, *oldalak*, *kiadás éve*
 - Konferencia cikk: szerző, „címidézőjelek között”, *konferencia neve dőlt betűkkel*, *konferencia kiadványban oldalak*, *konferencia városa*, *éve*
 - Diplomadolgozat/disszertáció: szerző, *címdőlt betűkkel*, *oktatási intézmény*, *város*, *végzés éve*.
 - elektronikus forrás: teljes URL (átviteli protokoll megjelölésével), megtekintés időpontja percre pontosan

Lásd `\cite{Bujd}` és `\cite[134. oldal]{Feri}`.

Lásd `\cite{Bujd,Feri}`.

Lásd `\cite[függelék]{Bujd,Feri}`.

...

```
\begin{thebibliography}{99}
```

```
\bibitem[Bujdosó1997]{Bujd} Bujdosó Gyöngyi,  
Fazekas Attila: \TeX kezdolépések,  
Budapest, 1997, Tertia Kiadó.
```

```
\bibitem[Wettl2004]{Feri} Wettl Ferenc, Mayer  
Gyula, Szabó Péter: LaTeX kézikönyv,  
Budapest, 2004, Panem Könyvkiadó.
```

```
\end{thebibliography}
```

...itt pedig hivatkozok a könyvre~\cite{konyv}.

...

```
\begin{thebibliography}{9}
```

```
\bibitem{konyv}
```

Virágvölgyi Péter,

{\it A tipográfia mestersége},

Osiris, Budapest, 2001.

```
\bibitem{cikk}
```

Claude E.\ Shannon,

„A Mathematical Theory of Communication”,

{\it Bell System Technical Journal}, vol.~27, no.~10,

pp.~379--423 és pp.~623--656, 1948.

```
\bibitem{url}
```

```
{\tt
```

```
  http://www.karinty.hu/~{ }NIKITA/szedes/Bevezet.htm},
```

```
2002.~január~9., 19:14
```

```
\end{thebibliography}
```

- Nagyon fontos! A LaTeX egyik alapköve!
- Lehet hivatkozni szinte mindenre, irodalomra, fejezetre, ábrára, táblázatra, oldalra, egyenletre stb.
- Irodalomjegyzékre hivatkozás kizárólag a már tanult `\cite{}` paranccsal lehetséges!
- A hivatkozás számozása automatikus!
- Módja egyszerű: a hivatkozni kívánt objektumot megcímkezzük a `\label{cimke}` utasítással,
- majd a hivatkozás helyén a `\ref{cimke}`, `\aref{}`, `\Aref{}` utasítással hívjuk elő a számot.

Példa:

```
\section{Bélabácsí}
```

```
\label{sec:bela}
```

```
Itt kezdem a fejezetet\dots
```

```
...
```

```
Később (vagy akár hamarabb is) elkezdek hivatkozni  
a~\ref{sec:bela}~fejezetre. (Vagy \aref{sec:bela})
```


- Hivatkozás oldalra: az oldalra a `\pageref{cimke}` kerül, majd másol a szövegben:

Bővebben `\apageref{sec:bela}.`~oldalón...

- Hivatkozás ábrára:

```
\begin{figure}
```

```
\centering % tegye középre az ábrát!
```

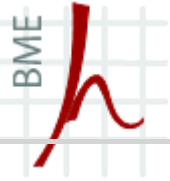
```
\includegraphics{ábra neve}
```

```
\caption{Ide az ábra címe}
```

```
\label{fig:címke}
```

```
\end{figure}
```

Az eredmény `\aref{fig:címke}.`~ábrán látható.



5. kis HF

- Készíts egy olyan folyószöveget amelynek a felhasznált irodalmat bemutatód és a szövegben hivatkozz ezekre a tanult parancsokkal.
- Ne feledkezz meg a tanult irodalomjegyzék kinézetéről sem!
- A hivatkozott művek között legyen folyóirat, konferenciacikk és egy könyv is!
- Ha ábrákat táblázatokat használsz csinálj róluk jegyzékeket.

MATEMATIKAI KÉPLETEK, MATEMATIKA MÓD TÉTELEK, SEGÉDTÉTELEK, BIZONYÍTÁSOK STB.

- A LaTeX legnagyobb előnye más szövegszerkesztőkhöz képest a szép matematikai képletek!
- Matematikai környezetre **három** lehetőség adódik:
 - `ide`, - szövegbe ágyazott matek mód dollárjelek segítségével
 - `\[\]`, - számozatlan képletek írása
 - `equation` környezet – számozott képletek írására
(a környezet „címkézhető” így a szövegben lehet hivatkozni az egyenletre)
- Fontos, hogy ha a szövegben matematikai változókat írunk akkor azok legyenek `$ $`-ben, mert így könnyebb megérteni a szöveget:
Ha `a` kisebb, mint `b` akkor ...

Nézzünk példákat:

- Ha $a^2+b^2=c^2$ akkor ez a Pitagorasz tétel.

- A Pitagorasz tétel:

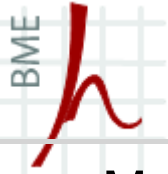
```
\[
  a^2+b^2=c^2
\]
```

- `\begin{equation}`

```
  a^2+b^2=c^2
```

```
\end{equation}
```

- **Vigyázzunk** a `^` karakter csak az utána következő első elemet emeli felső indexbe: a^{24} , helyett ilyenkor ún. blokkosításra van szükség: $a^{\{24\}}$
- Ugyanez igaz az alsó index `_`-re is! $R_{ui}=t^{a+b}\cdot r_{e2}$
- Alsó és felső indexben is lehet hatvány pl.: $R_{U^2_i}$



Matematikai mód

- Matematikai módban a szóközök száma nem számít,
- De **üres sort nem szabad hagyni**,
- nincs új bekezdés matematika módban (`\`)
- A magyarban tizedes vessző, míg az angolban tizedes pont van, de ha matek módban tizedes vesszőt írunk az felsorolásként értelmezi és csúnya nagy térközt hagy a tizedes jegyek és az egész szám között. Ennek ellensúlyozása az ún. negatív térközzel (`\!`) történik: $\$3,14\$$ helyett tehát $\$3,!14\$$
- `+`, `-`, `/` jelek használhatóak matek módban,
- A szorzást a mérnöki gyakorlatban nem `*` (az a konvolúció), helyette `\cdot` vagy `\times`
- Az `=`, `<`, `>` karakterek ugyanúgy megvannak matematika módban is, ám ha kisebb-,nagyobb egyenlő, nem egyenlő-re van szükség:
`\le`, `\ge`, `\ne`

- Ha az egyenletbe magyarázó szöveget kell betennünk vissza kell térni a normál módba (ne legyenek dőlt betűk):
- Szöveg bevitele a `\textnormal{szöveg}`, vagy a `\textrm{szöveg}` paranccsal lehetséges:

```
\[
```

```
x \geq 0, \textrm{ ha } \$x\$ \text{ nem negatív}
```

```
\]
```

- Képletek bekeretezésére a `\boxed{ }` parancsra van szükségünk (`\usepackage{amsmath}` kell hozzá)
- Hivatkozás képletekre:

```
\begin{equation}
```

```
\label{eq:emeletes}
```

```
A+b=c
```

```
\end{equation}
```

- `\aref{eq:emeletes}` ~képletben látható `\dots`
- `\aref({eq:emeletes})` képlet segítségével `\dots`

Számozás megváltoztatása 1.

- Ha az `equation` környezet által alkalmazott számozást szeretnénk megváltoztatni, pl. a chapter, szakasz, alszakasz számát is bele szeretnénk venni:

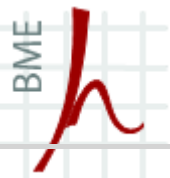
```
\numberwithin{equation}{subsection} % legyen
  benne szakasz és alszakasz is

\begin{document}
\section{Első szakaszom}

\subsection{Itt meg az alszakasz}
\begin{equation}
x \geq 0, \text{\textit{ ha } } \$x\$ \text{ nem negatív}
\end{equation}
\end{document}
```

Számozás megváltoztatása 2.

- Ha nem kellene a pontok és elég a subsection száma és az egyenlet sorszáma:
 - `\numberwithin{equation}{subsection}`
 - `\renewcommand{\theequation}{\thesection\arabic{equation}}`
- Római számokkal pl.:
 - `\renewcommand{\theequation}{\thesection\roman{equation}}`
 - `\renewcommand{\theequation}{\roman{equation}}`



Egyéb műveleti és relációs jelek

$+$	<code>+</code>	\wedge	<code>\wedge</code>	$=$	<code>=</code>	\geq	<code>\geqslant</code>
$-$	<code>-</code>	\vee	<code>\vee</code>	$:=$	<code>:=</code>	\ll	<code>\ll</code>
$/$	<code>/</code>	\star	<code>\star</code>	\doteq	<code>\doteq</code>	\gg	<code>\gg</code>
\pm	<code>\pm</code>	$*$	<code>*</code>	\equiv	<code>\equiv</code>	\in	<code>\in</code>
\mp	<code>\mp</code>	\circ	<code>\circ</code>	\sim	<code>\sim</code>	\ni	<code>\ni</code>
\cdot	<code>\cdot</code>	\bullet	<code>\bullet</code>	\simeq	<code>\simeq</code>	\subset	<code>\subset</code>
\times	<code>\times</code>	\oplus	<code>\oplus</code>	\approx	<code>\approx</code>	\supset	<code>\supset</code>
\div	<code>\div</code>	\ominus	<code>\ominus</code>	\cong	<code>\cong</code>	\subseteq	<code>\subseteq</code>
\setminus	<code>\setminus</code>	\odot	<code>\odot</code>	$<$	<code><</code>	\supseteq	<code>\supseteq</code>
\cap	<code>\cap</code>	\oslash	<code>\oslash</code>	$>$	<code>></code>	$:$	<code>:(arány)</code>
\cup	<code>\cup</code>	\otimes	<code>\otimes</code>	\leq	<code>\le</code>	$ $	<code>\mid</code>
				\geq	<code>\ge</code>	\parallel	<code>\parallel</code>
				\leqslant	<code>\leqslant</code>	\perp	<code>\perp</code>
$\%$	<code>\%</code>	\triangle	<code>\triangle</code>				
\perp	<code>\bot</code>	\square	<code>\square</code>				
\top	<code>\top</code>	\blacksquare	<code>\blacksquare</code>				
\neg	<code>\neg</code>	\sphericalangle	<code>\angle</code>				
\forall	<code>\forall</code>	\sphericalangle	<code>\measuredangle</code>				
\exists	<code>\exists</code>	\sphericalangle	<code>\sphericalangle</code>				
\nexists	<code>\nexists</code>	\flat	<code>\flat</code>				
\Re	<code>\Re</code>	\sharp	<code>\sharp</code>				
\Im	<code>\Im</code>	\natural	<code>\natural</code>				
∇	<code>\nabla</code>	$\#$	<code>\#</code>				
∂	<code>\partial</code>	1°	<code>1^\circ</code>				
\eth	<code>\eth</code>	$1'$	<code>1'\prime</code>				
\emptyset	<code>\emptyset</code>	$1''$	<code>1''\prime\prime</code>				
∞	<code>\infty</code>						

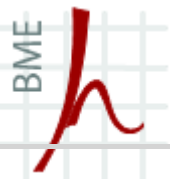
Operátorok

Σ	<code>\sum</code>	\int	<code>\int</code>	\oint	<code>\oint</code>
		\iint	<code>\iint</code>	\iiint	<code>\iiint</code>
\prod	<code>\prod</code>	\bigcap	<code>\bigcap</code>	\bigwedge	<code>\bigwedge</code>
\coprod	<code>\coprod</code>	\bigcup	<code>\bigcup</code>	\bigvee	<code>\bigvee</code>
\bigoplus	<code>\bigoplus</code>	\bigodot	<code>\bigodot</code>	\bigotimes	<code>\bigotimes</code>
\bigsqcup	<code>\bigsqcup</code>			\biguplus	<code>\biguplus</code>

```

\begin{equation}
\sum_{i=0}^{\infty} B_i \approx \int_0^{\infty} B \mathrm{d}x
\end{equation}

```



Deriválás, integrálás

$$\int f(x) \, dx \rightarrow \int f(x) dx$$

$$f'(x), f''(x) \text{ (' = shift 1)} \rightarrow f'(x), f''(x)$$

$$\frac{df(x)}{dx} \rightarrow \frac{df(x)}{dx}$$

$$\frac{\partial f(x,y)}{\partial y} \rightarrow \frac{\partial f(x,y)}{\partial y}$$

- A görög betűk beépített parancsokkal hívhatóak elő:

α	<code>\alpha</code>	ν	<code>\nu</code>	\digamma	<code>\digamma</code>	Γ	<code>\Gamma</code>	\varGamma	<code>\varGamma</code>
β	<code>\beta</code>	ξ	<code>\xi</code>	ε	<code>\varepsilon</code>	Δ	<code>\Delta</code>	\varDelta	<code>\varDelta</code>
γ	<code>\gamma</code>	π	<code>\pi</code>	ϑ	<code>\vartheta</code>	Θ	<code>\Theta</code>	\varTheta	<code>\varTheta</code>
δ	<code>\delta</code>	ρ	<code>\rho</code>	\varkappa	<code>\varkappa</code>	Λ	<code>\Lambda</code>	\varLambda	<code>\varLambda</code>
ϵ	<code>\epsilon</code>	σ	<code>\sigma</code>	ϖ	<code>\varpi</code>	Ξ	<code>\Xi</code>	\varXi	<code>\varXi</code>
ζ	<code>\zeta</code>	τ	<code>\tau</code>	ϱ	<code>\varrho</code>	Π	<code>\Pi</code>	\varPi	<code>\varPi</code>
η	<code>\eta</code>	υ	<code>\upsilon</code>	ς	<code>\varsigma</code>	Σ	<code>\Sigma</code>	\varSigma	<code>\varSigma</code>
θ	<code>\theta</code>	ϕ	<code>\phi</code>	φ	<code>\varphi</code>	Υ	<code>\Upsilon</code>	\varUpsilon	<code>\varUpsilon</code>
ι	<code>\iota</code>	χ	<code>\chi</code>			Φ	<code>\Phi</code>	\varPhi	<code>\varPhi</code>
κ	<code>\kappa</code>	ψ	<code>\psi</code>			Ψ	<code>\Psi</code>	\varPsi	<code>\varPsi</code>
λ	<code>\lambda</code>	ω	<code>\omega</code>			Ω	<code>\Omega</code>	\varOmega	<code>\varOmega</code>
μ	<code>\mu</code>								

Matematikai ékezetes betűk

- Matematikai ékezetek:

\hat{a} `\hat{a}`

\tilde{a} `\tilde{a}`

\bar{a} `\bar{a}`

\vec{a} `\vec{a}`

\acute{a} `\acute{a}`

\grave{a} `\grave{a}`

\breve{a} `\breve{a}`

\check{a} `\check{a}`

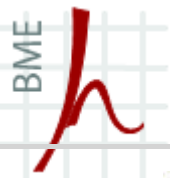
\dot{a} `\dot{a}`

\ddot{a} `\ddot{a}`

\dddot{a} `\dddot{a}`

\ddddot{a} `\ddddot{a}`

\mathring{a} `\mathring{a}`



Matematikai egyéb betűstílusok

`\mathit{...}`

`\mathrm{...}`

`\mathbf{...}`

`\mathsf{...}`

`\mathtt{...}`

`\mathnormal{...}`

`\mathcal{...}`

ABCDEFGHIJKLMNOPQRSTUVWXYZ

`\mathscr{...}` \in `mathrsfs`

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

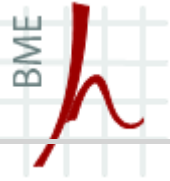
`\mathbb{...}`

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

`\mathfrak{...}`

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

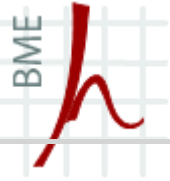


Zárójelek és nyilak

(())
[[vagy \lbrack]] vagy \rbrack
{	\{ vagy \lbrace	}	\} vagy \rbrace
<	\langle	>	\rangle
⌈	\lceil	⌋	\rceil
⌊	\lfloor	⌋	\rfloor
⌜	\ulcorner	⌝	\urcorner
⌞	\llcorner	⌟	\lrcorner
	vagy \vert		\ vagy \Vert
←	\leftarrow	↵	\leftharpoonup
⇐	\longleftarrow	↴	\leftharpoondown
→	\rightarrow	↶	\rightharpoonup
⇒	\longrightarrow	↷	\rightharpoondown
↔	\leftrightarrow	↑	\uparrow
⇔	\longleftrightarrow	↓	\downarrow
⇐	\Leftarrow	↕	\updownarrow
⇐	\Leftrightarrow	↑	\Uparrow
⇒	\Rightarrow	↓	\Downarrow
⇒	\Longrightarrow	↕	\Updownarrow
↔	\Leftrightarrow	↗	\nearrow
⇔	\Longleftrightarrow	↘	\searrow
↦	\mapsto	↙	\swarrow
↦	\longmapsto	↘	\nwarrow

Zárójelek méret igazítása

- Ha sok több zárójelet használunk jó lenne megkülönböztetni őket akár méretben, hogy olvasható maradjon a szövegünk: $2 \cdot ((a^{(a+b)} - c_i)^{q^2})$ – ezt így nehéz, helyette:
 - `\left (\right)` – zárójeleket használjunk, vagy akár szögletes `\left [\right]`, netalántán `\left \{ \right \}`
 - `\big(, \Big(, \bigg(, \Bigg(`



Tört és osztás művelete

- $\frac{\text{ide jön a számláló}}{\text{ide meg a nevező}}$, hát nem egyszerű?

- Természetesen emeletes törtet is lehet csinálni:

$$\frac{\frac{\pi}{2}}{\frac{1}{2}}$$

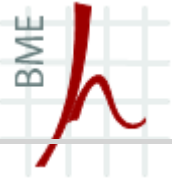
- Kombinatorikai kifejezésekben hasznos lehet az „n alatt a k”, amit $\{n \text{ \choose k}\}$ utasítással jelenítünk meg.

- `\sqrt{valami}` parancs segítségével.
- Akár több tört is egymásba helyezhető:

```
\begin{equation}
\label{eq:emeletes}
\lim_{n \to \infty} \left( \sqrt{ \left\{ \frac{1}{\frac{1}{n} + \sqrt{ \left\{ \frac{1}{\frac{1}{n} + \sqrt{ \left\{ \dots \right\} \right\} \right\} \right\} \right\} \right)} \right) = ?
\end{equation}
```

- Ha a gyökjel kitevőjét meg akarjuk változtatni akkor:

```
\sqrt[3]{2}
```



Norma és abszolút érték jelek

- Figyelem a norma jele $\|$, míg az abszolútértéké $|$.
- Ezért ne keverjük őket!
- Ha két vektor (sík) párhuzamosságát akarjuk jelölni, akkor a `\parallel` parancsra van szükség: $e \parallel f$.

- Bármilyen jel fölé/aláírás:

- `\overset{}`
- `\underset{}`

Példa:

`HCl\overset{\left[\frac{2}{3}\right]}\underset{\mathrm{B}}{\rightarrow}}H^+_2+Cl^-`

- Kommentek készítése:

`\overbrace{}` és `\underbrace{}`, `\underbracket{}`, `\overbracket{}`,
 Pl. `\underbrace{x}_{\text{valós rész}} +`
`\underbrace{iy}_{\text{képzetes rész}}`

`\usepackage{amsmath}` csomagot ne felejtsük el!

$\begin{matrix} a & b \\ c & d \end{matrix}$	<code>\begin{matrix} a&b \\ c&d \end{matrix}</code>
$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$	<code>\begin{pmatrix} a&b \\ c&d \end{pmatrix}</code>
$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$	<code>\begin{bmatrix} a&b \\ c&d \end{bmatrix}</code>
$\begin{Bmatrix} a & b \\ c & d \end{Bmatrix}$	<code>\begin{Bmatrix} a&b \\ c&d \end{Bmatrix}</code>
$\begin{vmatrix} a & b \\ c & d \end{vmatrix}$	<code>\begin{vmatrix} a&b \\ c&d \end{vmatrix}</code>
$\begin{Vmatrix} a & b \\ c & d \end{Vmatrix}$	<code>\begin{Vmatrix} a&b \\ c&d \end{Vmatrix}</code>

Nem teljes mátrixok

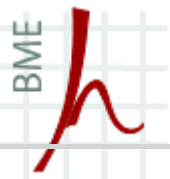
$$\begin{pmatrix} 1 & & \\ 0 & 2 & \\ & 0 & 3 \end{pmatrix} \quad \text{\code{\begin{pmatrix} 1\\ 0&2\\ &0&3 \end{pmatrix}}}$$

Szöveg közben az $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ helyett szebb az $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ mátrix. Ennek kódja:

```
\bigl(\!  
\begin{smallmatrix}  
a&b\  
c&d  
\end{smallmatrix}  
\!\bigr)$
```

$$\begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{pmatrix}$$

```
\begin{pmatrix}  
1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1  
\end{pmatrix}
```

Beépített függvények 1

`arccos` `\arccos`

`arcsin` `\arcsin`

`arctan` `\arctan`

`arg` `\arg`

`cos` `\cos`

`cosh` `\cosh`

`cot` `\cot`

`coth` `\coth`

`csc` `\csc`

`deg` `\deg`

`dim` `\dim`

`exp` `\exp`

`hom` `\hom`

`ker` `\ker`

`lg` `\lg`

`ln` `\ln`

`log` `\log`

`sec` `\sec`

`sin` `\sin`

`sinh` `\sinh`

`tan` `\tan`

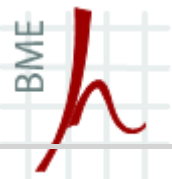
`tanh` `\tanh`

`\lim` `\varliminf`

`\overline{\lim}` `\varlimsup`

`\lim` `\varinjlim`

`\varprojlim`



Beépített függvények 2

det	<code>\det</code>	lim sup	<code>\limsup</code>
gcd	<code>\gcd</code>	max	<code>\max</code>
inf	<code>\inf</code>	min	<code>\min</code>
inj lim	<code>\injlim</code>	proj lim	<code>\projlim</code>
lim	<code>\lim</code>	Pr	<code>\Pr</code>
lim inf	<code>\liminf</code>	sup	<code>\sup</code>

Beépített függvények 3 - példa

- Rá és aláírás egy karakterre az `\overset{}` és az `\underset{}` paranccsal lehetséges:
- pl. az egyenlőség jelre:
- `\overset{\left[\frac{0}{0}\right]}{\underset{\mathrm{H}}{=}}`
- Megjegyzés írására hasznosak lehet a `\overbrace{}` és `\underbrace{}` parancsok:

```
\[
c^2 = \overbrace{
  \underbrace{a^2}_\text{Szöggel szemközti befogó} +
  \underbrace{b^2}_\text{Szög melletti befogó}
}^\text{Geometria alaptétele}
\]
```

```
\[
c^2 = \underbrace{
  \underbrace{a^2}_\text{Szöggel szemközti befogó} +
  \underbrace{b^2}_\text{Szög melletti befogó}
}_\text{Geometria alaptétele}
\]
```

- Ha a képletünk nem fér ki egy sorba el kell törni (ezt a LaTeX sajnos nem végzi el helyettünk), de mint az megtanultuk matematikai módban nincs `\` új sor ☹
- De vannak megoldások:
(korábban a mátrixok ábrázolása is valami hasonló volt)

- multiline környezet:

```
\begin{multiline} \label{címke}
1+8+27+64=\ \
=1+3+5+7+{\} \ \
+9+11+13+{\} \ \
+15+17+19
\end{multiline}
```

- Eqnarray környezet:
 - Hasznos lehet az `\lefteqn{}`

Képletek tördelése 2

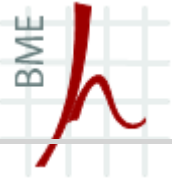
- *Amsmath* csomagban még számtalan egyenlet tördelő megoldás létezik pl. a `split`:

Példa 1:

```
\begin{equation} \label
  {címke}
\begin{split}
100 &= 1+8+27+64=\\
&= 1+3+5+7+9+{\ }\\
&\quad +11+13+15+17+19
\end{split} \tag*{A-1}
\end{equation}
```

Példa 2:

```
\[
\begin{split}
\cos^2\alpha &= 1-
\sin^2\alpha=\\
&= 1+3+5+7+9+{\ }\\
&\quad +11+13+15+17+1
9
\end{split}
\]
```



Képletek tördelése 2b

```
\documentclass{article}
\usepackage{eqnarray,amsmath}
\begin{document}
\begin{eqnarray*}
A&=&B,\\
C&=&D,\\
E&=&F
\end{eqnarray*}
\begin{align*}
A&=B,\\
C&=D,\\
E&=F
\end{align*}
\end{document}
```

Képletek tördelése 3

- Új oldal: `\displaybreak\\` - csak align esetén
- Több egyenlet esetén létrehozhatunk sub-számozást is:

```
\begin{subequations}
```

Itt jönnek a felsorolt egyenletek:

```
\begin{align}
```

```
x^2 + y^2 &= 1\\
```

```
y &= \sqrt{1 - x^2}.
```

```
\end{align}
```

```
\end{subequations}
```

Ha nem akarunk számozást valamelyik sorban: akkor írjuk a sor végére a `\nonumber` parancsot

- Néha előfordul, hogy feltételekhez kötött az egyenletünk, akkor a cases környezetet alkalmazhatjuk:

```
\[  
u(x) =  
\begin{cases}  
  \sqrt{x} & \text{Ha } x \geq 0 \\  
  1       & \text{Ha } x = 0 \\  
  0       & \text{egyébként}.  
\end{cases}  
\]
```


- Preambulumba helyezzük el valamelyiket :
 - `\newtheorem{tételnév}{tételcím}`
 - `\newtheorem{tételnév}{tételcím}[számlálóős]`
 - `\newtheorem{tételnév}[együttnév]{tételcím}`
- *tételnév*: Létrejön egy tételnév környezet és egy tételnév számláló, mely minden újabb ilyen környezet megnyitásakor növekszik eggyel.
- *tételcím*: Ez lesz a tételszerű bekezdés típus címe (pl. definíció, megjegyzés stb.). Ezen cím mellett megjelenik a tételnév számláló aktuális értéke is.
- *számlálóős*: Egy már korábban definiált számláló, általában valamelyik szint számlálója (fejezet, szakasz stb.). Ennek változásakor a tételnév nevű számláló 0-ra állítódik. A számlálóős és a tételnév számláló együtt jelenik meg (pl. 2.1. tétel).
- *együttnév*: Egy másik tételszerű környezet neve. A tételnév és együttnév környezetek számlálói együtt fognak növekedni.

- A dokumentum környezetben ha létre szeretnénk hozni egy tételt például:

```
\newtheorem{tetel}{tétel}
```

```
\begin{document}
```

```
...
```

```
\begin{tetel}
```

A tétel szövege.

```
\end{tetel}
```

```
\begin{tetel}[Cauchy]\label{cauchy}
```

A következő tétel szövege.

```
\end{tetel}
```

```
\aref{cauchy}.~tételből következően\dots
```

- A matematika képleteket olykor bizonyítani is kell!
- Az amsthm csomagban definiálva van egy környezet bizonyításokra: proof környezet.

```
\begin{proof}
```

A bizonyítás szövege.

```
\end{proof}
```

```
\begin{tétel}\label{xy}
```

A tétel szövege.

```
\end{tétel}
```

```
\begin{proof}[\Aref{xy}.~tétel bizonyítása]
```

A bizonyítás szövege.

```
\end{proof}
```

- Ha át akarjuk nevezni, pl. Bizonyításra:
 - `\renewcommand{\proofname}{Bizonyítás}`
- Q.E.D. jel átdefiniálása:
 - `\renewcommand{\qedsymbol}{\rule{1ex}{1ex}}`
- Q.E.D. jel megszüntetése:
 - `\renewcommand{\qedsymbol}{}{}`

- Egy .tex fájlba készíts el tetszőleges képleteket, mátrixokat!
- Gyakorold a tanult környezeteket, egysoros, többsoros egyenletek stb, mondj ki egy eléggé összetett tételt:
- Pl. Gauss–Osztrogradszkij-tétel:

$$\oint_F \mathbf{A} \cdot \mathbf{n} dF = \int_V \nabla \cdot \mathbf{A} d^3x,$$

$$\int \int (A_x dydz + A_y dx dz + A_z dx dy) = \int \int \int \left(\frac{\partial A_x}{\partial x} + \frac{\partial A_y}{\partial y} + \frac{\partial A_z}{\partial z} \right) dx dy dz.$$

- Levezetése?

OLDALSZERKESZTÉS, MÉRET ELFORGATÁS, HASÁBOK

Hosszabb dokumentum feldarabolása

- Ha nagy művet írunk áttekinthetőbb ha minden fejezetet külön .tex fájlba írunk, majd a végén összefűzzük őket!
- A dokumentumok felszabdalását az `\include{}` parancs támogatja,
- Argumentumában a fájl nevét kell megadni, akár a .tex kiterjesztés nélkül.
- Az `\include{}` parancs hatására új oldal kezdődik (mivel alapvetően chapter-ök beillesztésére tervezték). Ha ez nem tetszik, akkor az `\input{}` felhasználásával oldaltörés nélkül is beilleszthetünk fájlokat.

```
\documentclass[a4paper,10pt]{report} ...  
  \begin{document}  
... \includeonly{bevezeto} % most csak ezen  
   dolgozunk  
\include{bevezeto} \include{modell.tex} ...  
\include{befejezes} ... \end{document}
```

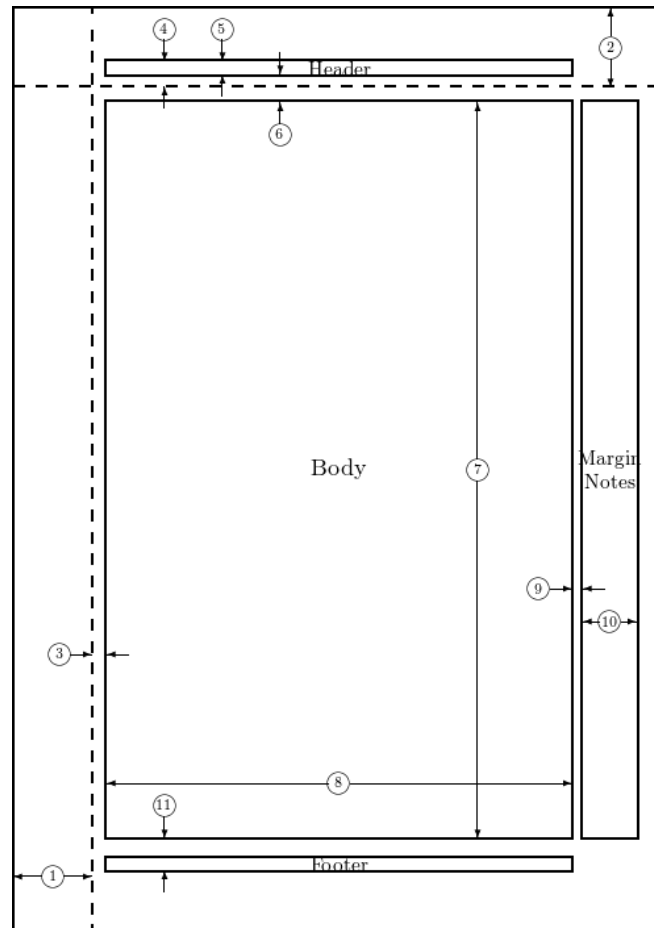
- Előfordulhat, hogy több hasábos dokumentumot szeretnénk
 - Ha két hasábot szeretnénk akkor : a dokumentumosztály definíciójánál a *twocolumn* ad erre lehetőséget.
(`\documentclass[11pt,twocolumn]{article}`)
 - Váltás `\twocolumn` és `\onecolumn` (egy új oldalon már 1 hasábos lesz a szedés)
 - Másik lehetőség:
 - A **multicol csomagban** található **multicols környezet** (javallott ezt használni!)
`\begin{multicols}{hasábok száma max 9 lehet!}`
Ide jön a sok szöveg, amit a környezet arg.-ban megadott számú hasábot alkotja
`\end{multicols}`
 - A hasábok közötti távolság 10 pt alapesetben, de ez állítható pl. 1 cm-re:
`\setlength{\columnsep}{1cm}`

- Az oldal méret megadásoknál a LaTeX a nyomdászatban alkalmazott mértékegységeket használja:
 - Fix mértékegységek: (egybeírva!)
 - mm (milliméter)
 - cm (centiméter)
 - in (inch, 1 in = 25,4 mm)
 - pt (pont, 1 pt = 0,3515 mm)
 - Relatív mértékegységek:
 - ex (aktuális betűalakzatban az „x” magassága)
 - em (aktuális betűfokozat mérete)

Mértékegységek összehasonlító táblázata

jele	jelentése	A méret nagysága más egységben							
		pt	pc	dd	cc	mm	in	bp	sp
pt	pont	1	0,0833	0,9346	0,07788	0,3515	0,0138	0,996	65536
pc	pica	12	1	11,215	0,93457	4,2175	0,166	11,955	786432
dd	Didót pont	1,07	0,0892	1	0,08333	0,3761	0,01481	1,066	70124
cc	cicero	12,84	1,07	12	1	4,5128	0,17767	12,792	841489
mm	miliméter	2,845	0,2371	2,6591	0,22159	1	0,03937	2,835	186468
in	hüvelyk	72,27	6,0225	67,5415	5,62846	25,4	1	72	4736287
bp	nagy pont	1,004	0,0836	0,938	0,07817	0,3528	0,01389	1	65782
cm	centiméter	28,45	2,371	26,591	2,2159	10	0,3937	28,346	1864680
sp	skálázott pont (1/65536 pt)								
ex	az aktuálisan használt betűtípusban „x” magassága, most éppen 7.22783pt								
em	az aktuálisan használt betűtípusban „M” szélessége, most éppen 13.82pt								
mu	matematikai mértékegység, csak matek módban létezik! (1/18 em)								
fill	végtelenül rugalmas mértékegység								

Oldal szerkezete és méretei



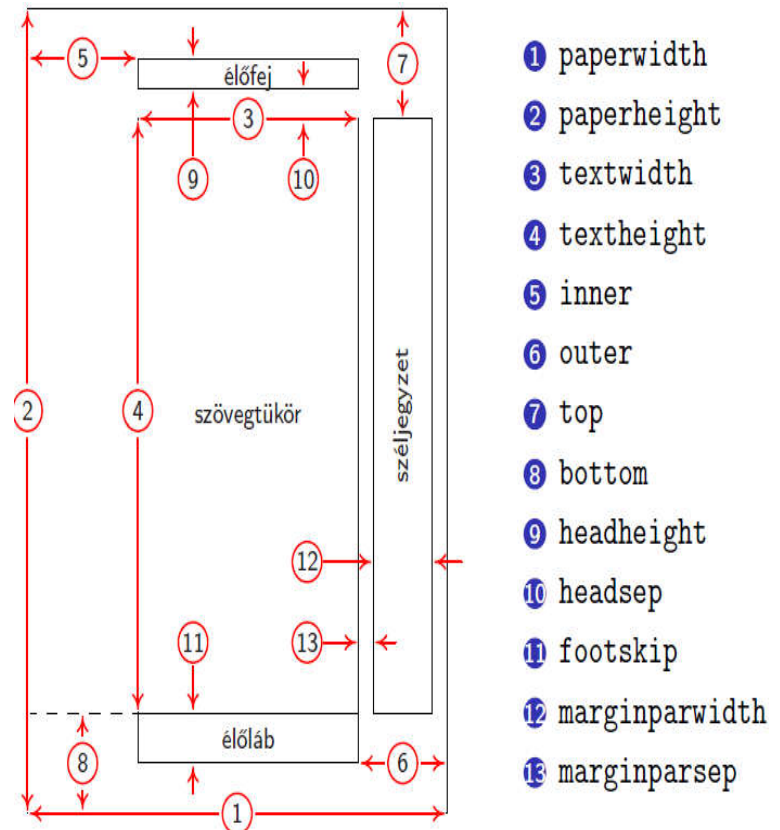
- | | |
|-------------------------|----------------------------------|
| 1 one inch + \hoffset | 2 one inch + \voffset |
| 3 \oddsidemargin = 13pt | 4 \topmargin = -23pt |
| 5 \headheight = 12pt | 6 \headsep = 25pt |
| 7 \textheight = 674pt | 8 \textwidth = 426pt |
| 9 \marginparsep = 10pt | 10 \marginparwidth = 50pt |
| 11 \footskip = 30pt | \marginparpush = 5pt (not shown) |
| \hoffset = 0pt | \voffset = 0pt |
| \paperwidth = 597pt | \paperheight = 845pt |

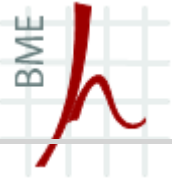
Oldal méret beállítási példák (kézzel)

```

\setlength{\oddsidemargin}{0in}
\setlength{\evensidemargin}{0in}
\setlength{\topmargin}{0in}
\setlength{\footskip}{10mm}
\setlength{\headheight}{0mm}
\setlength{\headsep}{0mm}
\setlength{\marginparwidth}{1.8cm}
\setlength{\unitlength}{1mm}
\setlength{\textheight}{230mm}
\setlength{\textwidth}{155mm}

```





Fullpage csomag használata

- Kisebbségek lesznek a margók tőle és több mint 100 karakter/sor (10pt betűméret esetén):
 - `\usepackage{fullpage}`
- Még kisebb margók (120 karakter/sor)
 - `\usepackage[cm]{fullpage}`

- Jobb a fullpage-nél, használata: `\usepackage{geometry}`
- Példák:
 - `\usepackage[margin=1in]{geometry} == \u.p{fullpage}`
 - `\usepackage[top=1in, bottom=1.25in, left=1.25in, right=1.25in]{geometry}`
 - `\usepackage{geometry}`
`\geometry{ a4paper,`
`total={210mm,297mm},`
`left=10mm,`
`right=2cm,`
`top=10mm,`
`bottom=50mm,`
`}`
 - `\usepackage[inner=4cm,outer=2cm]{geometry}` (külső/belső margó állítása)

- Ha az oldalt el szeretnénk **fordítani** a geometry csomag (`\usepackage[landscape]{geometry}`) landscape környezetével tudjuk megcsinálni:
- új oldalt nyit,
 - a szövegtükör és széljegyzet tartalmát elforgatja +90-kal, de a fejléctet és a lábléctet nem,
 - a végén visszavált normál módra, de előtte új oldalt nyit.
- **Nagyítás/kicsinyítés:** geometry csomaggal:
`\usepackage[mag=500,truedimen,a4paper]{geometry}`
 - `mag=1500` opcióval másfélszeres nagyítás érhetünk el, illetve a `mag=500` felére kicsinyít.
- Ha valamely fix mértékegységgel megadott hosszmeretet nem akarjuk, hogy a nagyítás során megváltozzon, akkor a mértékegység elé tegyük a `true` szót

- LaTeX-ben létezik rugalmatlan és rugalmas hossz méret.
 - 1. Rugalmatlan hossz mérték megadása:**
 - A `\setlength{\hp}{hossz}` paranccsal állíthatjuk be egymár létező `\hp` hosszúság parancs méretét.
 - Az `\addtolength{\hp}{hossz}` utasítással hossz mértékben növelhetjük `\hp` méretét.
 - A `\the\hp` utasítás írja ki a `\hp` aktuális értékét.
 - A `\newlength{\hp}` hoz létre egy új `\hp` hosszúságot.
 - A `\settowidth{\hp}{szöveg}` a szöveg szélességére állítja `\hp` értékét, a `\settoheight{\hp}{szöveg}` utasítás szöveg alapvonalától – a sor alapvonalától – mért távolságot adja vissza, míg a `\settodepth{\hp}{szöveg}` az alapvonalától mért mélységre állítja `\hp` értékét.

- A `\hspace{hossz}` és `\vspace{hossz}` parancsokkal vízszintes és függőleges térközt lehet kihagyni.
- Csillagos változataik sor-, illetve oldaltörés esetén is kihagyják a megfelelő méretű helyet.

- **Példa:**

```
\newlength{\gyalu}
```

A „gyulai gyalu” szélessége `\settowidth{\gyalu}{gyulai gyalu}`

`\the\gyalu\` (ide „`\hspace*{\gyalu}`” pont beférne), magassága

`\settoheight{\gyalu}{gyulai gyalu} \the\gyalu`, mélysége pedig

`\settodepth{\gyalu}{gyulai gyalu} \the\gyalu`.

Rugalmas térközök beállítása

- A `\vfill` és a `\hfill` utasításokkal illeszthetünk be vízszintes és függőleges végtelenül rugalmas térközt dokumentumunkba.
- A `\hfill` parancs ekvivalens a `\hspace{0pt plus 1fill}` paranccsal.
- Léteznek még `\hrulefill` és `\dotfill` parancsok is, melyekkel **végtelenül rugalmas vízszintes vonalat, illetve pontsorozatot** lehet beilleszteni.
- Aláírandó dokumentumokat ezért célszerű ilyes formán befejezni: `\vfill Budapest, 2002.\ november 19. \hfill \dotfill`

Rugalmas térközök beállítása

- Ha több `\hfill`, vagy `\vfill` követi egymást, akkor azok száma aránylik a távolságokhoz. Ha nem akarunk sokat ismételni, akkor a `\stretch{szám}` mértéket is alkalmazhatjuk a `\hspace{}`, vagy `\vspace{}` argumentumaként:

eleje `\hspace{\stretch{4}}` közepe
`\hspace{\stretch{1}}`

vége `\` ugyanaz, mint az `\`

eleje `\hfill\hfill\hfill\hfill` közepe `\hfill`
 vége `\`

és az `\` eleje `\hspace{0pt plus 4fill}` közepe
`\hspace{0pt plus 1fill}` vége `\` ugye?

Túllógó sorok kezelése

- Sajnos előfordul a LaTeX-ben, hogy nem tud elválasztani.
- Alkalmazható a `\sloppy` parancs, mely nagyobb sorközt is megenged.
- A `{\fussy}` utasítással kapcsolhatunk vissza az alapértelmezett üzemmódra.
- Azonos módon működnek a `sloppy`par és `fussy`par környezetek is.

- **csomag:** `\usepackage{draftwatermark}`
- `\usepackage[firstpage]{draftwatermark}`
- Szöveg ami a vízjelben legyen:
 - `\SetWatermarkText{Példa}`
- Lehet ábra is: (tikz csomaggal!)
 - `\usepackage{tikz}`
 - `\SetWatermarkText{`
 - `\tikz{\node[opacity=0.8]{\includegraphics{abra.jpg}};}}`
- Beállítások:
 - `\SetWatermarkAngle{0}`
 - `\SetWatermarkScale{0.45}`
 - `\SetWatermarkColor{red}` vagy `\SetWatermarkColor[rgb]{0,1,0}`
 - `\SetWatermarkFontSize{ 10cm}`
 - `\SetWatermarkLightness{1}`

Egyéni címoldal gyártása

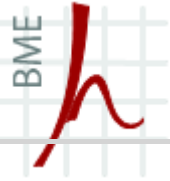
- Címoldalt a titlepage környezettel definiálhatunk. A címoldal tartalmát a LATEX azonnal megjeleníti, az oldalszámlálót pedig 1-re állítja. A címoldal környezetben akár több oldalnyi információt is megjeleníthetünk.
- `\usepackage{graphicx}` csomag kell hozzá:


```

\begin{titlepage} { \center
\resizebox{11cm}{!}{\includegraphics{kisbme.eps}} \
{\large \bf Budapest University of Technology and Economics}
\ \vfill {\Large Valami jópofa cím kell ide} \[20pt]
{\Large Nomeg egy szerző} \vfill {\bf Konzulens}: név \vfill
{\Large Budapest, 2002.} \clearpage \mbox{} \vfill \noindent
{\large Mégegyszer a szerző: cím}} \[10pt] Copyright
\copyright 2002. All rights reserved ... \[12pt] This
document was typeset in \LaTeXe. \vfill \end{titlepage}

```

- Egy tetszőleges saját, vagy internetről letöltött szöveget kell formázni, amely az alábbi követelményeket teljesíti:
 - van benne folyó szöveg (amelyben a hallgató értelmesen alkalmazza a tördelési, betűstílus és méretről tanultakat),
 - van benne legalább egy (értelmes) táblázat,
 - felsorolásokat tartalmaz,
 - van benne legalább egy kép/ábra (akár subfigure is),
 - van benne legalább egy képlet (de nem $a+b=c$ típusú!!),
 - tartalmaz irodalomjegyzéket, illetve azokra hivatkozást.
 - Hivatkozni kell a folyószövegben az ábrára a táblázatra és a képletre is a tanult módszerekkel!
- A házi feladatban a bruttó (utasításokkal együtt számolt) karakterszám 25 000–30 000 legyen.



Nagy HF beadás követelményei

- dolláros matek külön sorba írva (`\$. $` jellegű)
- frenchspacing nem alkalmazása, és pont utáni mondatvégek
- description környezet nem alkalmazása (`\item \textbf{.}` jellegű)
- gondolatjel, hosszú kötőjel nem ismerete (- mintázat)
- nincsenek meg a kért dolgok (lásd előző fólia)
- idézőjel hibásan (" = hüvelykjel)
- képletek végén nincs írásjel
- nemúsztatott táblák (table környezet elhagyása)
- tanult parancsok/paraméterek nem ismerete (pl. `eqnarray` környezet minden sorában `\nonumber` parancs)
- felesleges `\` újsor utasítások

- **Trackchanges** csomag segítségével:
`\usepackage{trackchanges}`
(hasznos továbbá: `\usepackage{color}`)
- 5 szerkesztési lehetőséget kínál:
 - `\note[initials]{note text}`
 - `\annote[initials]{note text}`
 - `\add[initials]{additional text}`
 - `\remove[initials]{removed text}`
 - `\change[initials]{original text}{new text}`
- Szerzők megadása:
 - `\addeditor{Bela.F}`
 - `\addeditor{Sanyi.G}`

- `\usepackage[margins,adjustmargins]{trackchanges}`
 - `finalold` - Reject all edits.
 - `finalnew` - Accept all edits.
 - `footnotes` - Display edits as footnotes.
 - `margins` - Display edits as margin notes.
 - `inline` - Display edits inline.

- Margin opción belül további lehetőségek:
 - `movemargins` : balra tegye a szöveget (a `textwidth` marad)
 - `adjustmargins` : balra tegye a szöveget (a `textwidth` változik)

A L^AT_EX PROGRAMOZÁSA, CSOMAG ÉS DOKUMENTUMOSZTÁLY KÉSZÍTÉSE

Új utasítás és környezet létrehozása 1

- Új utasítás létrehozása: `\newcommand` paranccsal,
- Az első argumentumba kell beírni az utasítás azonosítóját, a második argumentumban pedig a végrehajtandó parancsokat kell leírni:
- Példa: `\newcommand{\sebesseg}{~km/h}`

A vonat jelenlegi sebessége: `100\sebesseg` Itt a `\sebesseg` parancs helyére behelyettesítődik a `km/h` elválaszthatatlan szóközzel!

- Utasítás újradefiniálása:

`\renewcommand{\sebesseg}{~mérőöld/h}`

- Utasítás argumentumokkal:

`\newcommand{\prncs}[argsz][default]{def}`

Az általunk
létrehozott parancs

Hány darab argumentumot szeretnénk

A parancs
definíciója

Az első
Argumentum
alapértelmezett beállítása

- Az argumentumokat a #1, #2 stb. alakban hívhatjuk elő.
- Nézzünk egy példát:

```
\newcommand{\secref}[1]{\ref{sec:#1}~fejezet}
\section{Első fejezetünk}
\label{sec:egy}
Az \secref{egy} leírja a lényegét.
```
- Új környezet létrehozása (hasonlóan):

```
\newenvironment{körny}[argsz][default]{nyitódef}{záródef}
```

Példa:

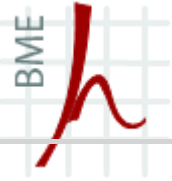
```
\newenvironment{bela}%
{Nyitottam egy környezetet, azt írtam bele, hogy}%
{Elég már ebből a környezetből, bezárom itt.}
\begin{bela}
Kacsacsőr
\end{bela}
```

Néhány (talán) hasznos utasítás

- `\newcommand{\ie}{i.\,e.}`
- `\newcommand{\eg}{e.\,g.}`
- `\newcommand{\dd}{\, \{\mathsf {d}\}}`
- `\newcommand{\MHz}{\ensuremath{\, \{\mathrm {MHz}\}}}`
- `\newcommand{\Mbps}{\ensuremath{\, \{\mathrm {Mbit/s}\}}}`
- `\newcommand{\kbps}{\ensuremath{\, \{\mathrm {kbit/s}\}}}`
- `\newcommand{\dB}{\ensuremath{\, \{\mathrm {dB}\}}}`
- `\newcommand{\dBm}{\ensuremath{\, \{\mathrm {dBm}\}}}`
- `\newcommand{\m}{\ensuremath{\, \{\mathrm {m}\}}}`
- `\newcommand{\mW}{\ensuremath{\, \{\mathrm {mW}\}}}`
- `\newcommand{\W}{\ensuremath{\, \{\mathrm {W}\}}}`
- `\newcommand{\km}{\ensuremath{\, \{\mathrm {km}\}}}`

- Korábban is számtalan alkalommal talákoztunk velük, csak nem vettük észre!
- Egész számokat tárolnak (nincs mértékegysége)
- A LaTeX beépített számlálói:
 - *part* rész sorszáma
 - *chapter* fejezet sorszáma
 - *section* szakasz sorszáma
 - *subsection* alszakasz sorszáma
 - *subsubsection* al-alszakasz sorszáma
 - *paragraph* paragrafus sorszáma
 - *subparagraph* alparagrafus sorszáma
 - *tocdepth* mi kerül a tartalomjegyzékbe
 - *secnumdepth* szintek számozásának mélysége
 - *page* oldalszám
 - *equation* egyenlet sorszáma
 - *figure* ábra sorszáma
 - *table* táblázat sorszáma
 - *enumi* lista 1. szintjének sorszáma
 - *enumii* lista 2. szintjének sorszáma
 - *enumiii* lista 3. szintjének sorszáma
 - *enumiv* lista 4. szintjének sorszáma
 - *footnote* lábjegyzet sorszáma
 - *mpfootnote* lábjegyzet sorszáma minipage környezetben

- Akár mi is létrehozhatunk számlálót: `\newcounter{szam}` parancs segítségével.
- Ez egy `szam` nevű számlálót hoz létre, természetesen 0 kezdőértékkel.
- Ha nem 0-tól szeretnénk kezdeni a számolást:
`\setcounter{szam}{17}`
- `\addtocounter{szam}{5}` a szám értékéhez hozzáad 5-öt.
- Számláló értékének kiíratása:
`\stepcounter{szamA}`
`\setcounter{szamB}{2}`
`\theszamB;`
`\stepcounter{szamA}`
`\theszamB`



Számláló értékének átadása

- `\value{szam}` parancs a *szam* értékének átadására használható:

`\setcounter{page}{\value{szam}}`
- a `page` értéke felveszi a *szam* aktuális értékét.

- Ide tartoznak a feltételes utasítások, eset szétválasztások és ciklus kezelések.
- Belső parancsnak nevezzük a @ karaktert tartalmazó parancsokat!
- Ezek egy egyszerű LATEX-forrásállományban (.tex) nem használhatók, csak az osztály- (.cls) és stílusfájlokban (.sty) (Előljáróban annyit, hogy a .sty-t a preambulumban elhelyezett \usepackage paranccsal hívjuk meg, míg osztály fájlt a \documentclass segítségével.)

While és for belső parancs

- A `\@whilenum feltétel\do{parancsok}` mindaddig végrehajtja a `\do` utáni parancsokat, amíg a numerikus feltétel fennáll.

Használatára egy példa:

```
\newcounter{szam}
```

```
\@whilenum\theszam<10\do
```

```
{\stepcounter{szam}\theszam\ }
```

- `\@for\parancs:={lista1,lista2,...}\do{parancsok}`

A listaelemeken végrehajtja a parancsokat.

Pl. `\@for\mitcsinal:={felkel, lenyugszik}\do{A Nap \mitcsinal. } → A Nap felkel. A Nap lenyugszik.`

- `\if@twoside A \else B \fi` – ha feltétel: Ha a dokumentumclass-ban *twoside* opciót adtunk meg az *A* parancs hajtódik végre, ha nem akkor a *B* parancs!
- `\@ifnextchar P{ha igaz}{ha hamis}` – Ha a következő karakter *P* akkor a „ha igaz” feltételt hajtja végre, ha következő karakter nem *P* akkor a „ha hamis” kódot:
- Példa:

`\@ifnextchar [{a}{b} [` → *a* kapott eredmény: *a*[(mivel *]* karakter jött)

`\@ifnextchar c{a}{b} c` → *ac* (*a* kapott eredmény, mert *c* karakter jött)

`\@ifnextchar c{a}{b} d` → *bd* (*a* kapott eredmény, mert *d* karakter jött *c* helyett ezért a *b* kód hajtódik végre!)

Ha feltétel (if) 2

- `\@ifstar{ha igaz}{ha hamis}`: A „ha igaz” kódot hajtja végre, ha a következő karakter a *. Ellenkező esetben a „ha hamis” kódot hajtja végre:
 - `\@ifstar{a}{b}* → a`
 - `\@ifstar{a}{b}x → bx`
- `\@ifundefined{parancs}{ha nincs}{ha van}`: A „ha nincs” kódot hajtja végre, ha a `\parancs` nincs definiálva. Ellenkező esetben a „ha van” kódot hajtja végre. Pl.
 - `\@ifundefined{section}{a}{b} → b`
 - `\@ifundefined{Section}{a}{b} → a`
- `ifnum` és `ifodd` használata számlálókra:
 - `\ifnum\value{szam}=1 A \fi` az eredménye: A ha a számláló értéke 1
 - `\ifnum\value{szam}<2 B\else C\fi` az eredménye: B
 - `\ifodd\value{szam}F\fi` az eredménye: F
 - `\ifodd\value{szam}G\else H\fi` az eredménye: H

Ha feltétel (if) 3

- `\@ifclassloaded{osztálynév}{ha betöltve}{ha nincs betöltve}`

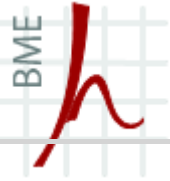
A „ha betöltve” kódot hajtja végre, ha az osztálynév osztály van betöltve. Ellenkező esetben a „ha nincs betöltve” kódot hajtja végre. (Csak preambulumban használható!)

- `\@ifpackageloaded{csomagnév}{ha betöltve}{ha nincs betöltve}`:

A „ha betöltve” kódot hajtja végre, ha a csomagnév csomag be van töltve. Ellenkező esetben a „ha nincs betöltve” kódot hajtja végre. (Csak preambulumban használható!)

Egyéb if utasítások (összefoglaló)

- `\iflanguage{english}{igaz}{hamis}`
- `\ifpdf A \else B \fi`
- `\ifdim\textwidth<\textheight A\fi`
- `\ifFileExists{pelda.tex}{igaz}{hamis}` → Ha a pelda.tex fájl létezik, akkor igaz, különben hamis.
- `\InputIfFileExists{foo.tex}{igaz}{hamis}` → Ha a foo.tex fájl létezik, akkor végrehajtódik az igaz kód, majd beolvassa a fájlt, különben hamis.



Ifthen csomag

- Be kell tölteni az ifthen csomagot!
- `\ifthenelse{feltétel}{igaz}{hamis}`: Ha a feltétel teljesül, akkor az „igaz” kódot, ellenkező esetben a „hamis” kódot hajtja végre.

- Ha a meglévő csomagok nem elégségesek készíthetünk sajátot is!
- Ha csomagot készítünk ügyeljünk rá, hogy több dokumentumosztállyal is működjön! (különben minek?)
- Szabályok csomag készítésére:
 - A csomag forrásfájlja legyen .sty végű és azonos mappában legyen a forrásállománnyal (.tex), különben a fordító nem találja meg a stílusfájlt!
 - Ékezetes karakterek ne szerepeljenek benne!
 - A csomag forrás fájljába minden olyan parancs írható, amely a főfájl preambulumban szerepelhet.
 - @-os belső parancsokat a .sty fájlba kell írni!
 - Csomagunk tartalmazhat opciókat is!

Csomagok készítése saját opciókkal

- Hozzuk létre az első saját csomagunkat:
 1. Új fájl, adjunk neki egy nevet pl.: csomagom.sty
(kiterjesztésre ügyeljünk most NEM .tex lesz!!)
 2. Írjuk bele milyen opciókat szeretnénk:


```
\DeclareOption{opciol}{kód1} % a csomag első opciója
\DeclareOption{opcioll}{kód2} % a csomag 2. opciója
\ExecuteOptions{opciol} % ilyenkor a kód1 fog lefutni
\ProcessOptions
```

Példa:

- `\DeclareOption{mikmukk}{\twocolumn}`
- `\DeclareOption{mik}{\onecolumn}`
- `\DeclareOption{onyside}{\@onesidetrue}`
- `\ProcessOptions`

- Ezután a csomag betöltése:

```
\documentclass{article}
```

```
\usepackage{csomagom}
```

```
\begin{document}
```

```
...
```

```
\end{document}
```

Csomag opció örökítése

- Egy már létező csomag opcióját felhasználhatjuk a saját csomagunkban is:
- saját.sty tartalma legyen az alábbi:

```
\usepackage[bookmarksopen]{hyperref}
```

```
\usepackage{kvoptions}
```

```
\SetupKeyvalOptions{family=sajat,prefix=sajat@}
```

```
\DeclareBoolOption{colorlinks}
```

```
\DeclareBoolOption{urlcolor}
```

```
\ProcessKeyvalOptions{sajat}
```

```
\ifsajat@colorlinks\PassOptionsToPackage{colorlinks}{hyperref}  
\fi
```

```
\ifsajat@urlcolor\PassOptionsToPackage{urlcolor}{hyperref}\fi
```

- Ekkor a saját csomag opciójaként használható a hyperref csomag colorlinks és urlcolor opciói.

Dokumentumosztály létrehozása

- Tegyük fel nem jó az 5 előre definiált dokumentumosztály.
- Ekkor létre lehet hozni egy sajátot is!
- Amire figyelni kell:
 - A dokumentumosztály kiterjesztése legyen .cls
 - Ha a saját dokumentumosztályban szeretnénk csomagokat meghívni akkor azt a `\RequirePackage` paranccsal tehetők meg és nem a `\usepackage`-el (DE ez csak a .cls-re igaz!)
 - Célszerű egy már meglévő dokumentumosztályt használni alapnak, ne nekünk kelljen minden apróságot megírni! Erre való a `\LoadClass` parancs!

- Példa:

```
\LoadClass[12pt,a4paper]{article}
\RequirePackage[T1]{fontenc}
\RequirePackage[utf8]{inputenc}
\PassOptionsToPackage{defaults=hu-min}{magyar.ldf}
\RequirePackage[magyar]{babel}
\RequirePackage{kvoptions}
\SetupKeyvalOptions{family=sajat,prefix=sajat@}
%\DeclareBoolOption[true]{opcion} % ide lehet opciókat adni a saját
doc.classnak
\ProcessKeyvalOptions{sajat}
```

- Majd a .tex fájlban a meghívása:

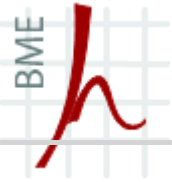
```
\documentclass{sajat}
\begin{document}
...
\end{document}
```

PREZENTÁCIÓ KÉSZÍTÉS

BEAMER

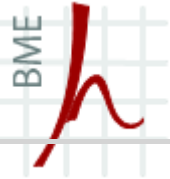
Slides document class

- A dokumentumosztályok közül a slide használható fapados prezentációk készítésére.
- Ugyanakkor mi a jóval okosabb Beamer nevű dokumentumosztályt fogjuk használni.
- Ez nem tartozik az alap 5 dokumentumosztály közé!
- Nem ez az egyetlen prezentációkészítésre használható dokumentumosztály.
- Régebbiek:
 - powerdot (Hendri Adriaens és Chris Ellison)
 - Prosper (Timothy Van Zandt)



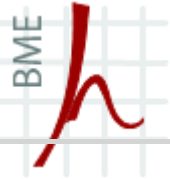
Beamer

- A Beamer (magyarul: kivetítő, projektor) Till Tantau alkotása (2003)
- Vedran Miletic és Joseph Wright intézi az ügyeket 2011 óta
- PDF alapú prezentációkat lehet vele létrehozni LaTeX alapon.
- A nagy előnye a PDF alapú prezentációknak a rendszerfüggetlenség (az idegen gépen se csúsznak el az ábrák, szövegek stb.), ppt-nél ez előfordulhat!
- Eltérő a „wysiwyg” prezentációs programoktól (Open office, PowerPoint)
- Felépítése ugyanúgy preambulumból és a dokumentumtörzsből áll!



A Beamer sajátosságai

- Elég sok már meglévő LaTeX parancsot használhatunk vele (pl. tartalomjegyzék, felsorolások, fejezetek stb.)
- Támogatja a pdflatex, latex+dvips, lualatex és xelatex, viszont a latex+dvipdfm nem támogatott!
- Átfedések és látványos effektek a diák között egyszerűen
- Témák használata, hogy változatos és szép prezentációt hozhassunk létre!



A Beamer sajátosságai

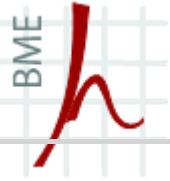
- Oldalméret 4:3 (alapeset), de [aspectratio=169] opcióval 16:9
- Betűméret: 11pt alapeset (opciók: 10pt, 12pt, 14pt, 17pt, 20pt)
- Alap betűtípus: álló, normál, groteszk.
- A főtörzs sortörése balra zárt és nincsenek szóelválasztások!
- Főszöveg sortörése: balra zárt, így nincsenek szó elválasztások.
- Új bekezdés elején nincs behúzás.
- Keret (lásd később) tartalmának függőleges pozíciója: közép.
- Opcióban a másik két lehetőség: t (fent), b (lent).
- Ezzel az osztállyal automatikusan betöltődnek a következő csomagok: graphicx, amsthm, xcolor, enumerate, hyperref.

- Kezdés a szokásos: `\documentclass[opciók]{beamer}`
- Számos opció megadható (de nem kötelező!):
 - utf8, - pont ua. mintha `\usepackage[utf8]{inputenc}`
 - utf8x, - `\usepackage[utf8x]{inputenc}`
 - ucs, - ucs csomag betöltése
 - draft - nincsenek figurák, láblécek stb.
 - handout – nincsenek látványeffektek
- *Ezt követően megadhatjuk a használni kívánt csomagokat.*
- A dokumentumtörzsét a jól ismert `\begin{document} ... \end{document}` környezet tartalmazza!

BEAMER

KERETEK KÉSZÍTÉSE ÉS STRUKTURÁLÁSA

- A prezentáció **keretekből** áll, a keretek pedig **diasorokból!**
- A keret egy új környezet: `\begin{frame} ... \end{frame}`
- A keretnek van címe és adható alcím is!
- Keret (lásd később) tartalmának függőleges pozíciója: közép. Opcióban a másik két lehetőség: t (fent), b (lent).
- A keret tartalma lehet szinte majdnem minden eddig tanult (felsorolás, táblázat, egyenletek, szöveg, ábra stb.)
- Bonyolultan látszik elsőre pedig nem az!



Keretek komponensei

1. fej- és lábléc
2. bal és jobb oldali sáv
3. navigációs szimbólumok
4. logó
5. keret címe és alcíme
6. háttér, rajta a vászon
7. tartalom

Természetesen nem kötelező mindet használni!

What Are Prime Numbers?

What Are Prime Numbers?

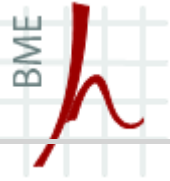
Definition

A **prime number** is a number that has exactly two divisors.

Example

- 2 is prime (two divisors: 1 and 2).
- 3 is prime (two divisors: 1 and 3).
- 4 is not prime (**three** divisors: 1, 2, and 4).

Euclid Az én első prezentációm



Keret környezet

```
\begin{frame} [opció] {Keret címe} {Keret alcíme}  
Keret tartalma ide jön  
\end{frame}
```

Vagy (a kettő megoldás ekvivalens):

```
\begin{frame} [opció]  
\frametitle{Keret címe}  
\framesubtitle{Keret alcíme}  
Keret tartalma ide jön  
\end{frame}
```

```
\begin{frame}
\frametitle{What Are Prime Numbers?}
\begin{definition}
A \alert{prime number} is a number that has exactly
two divisors.
\end{definition}
\begin{example}
\begin{itemize}
\item 2 is prime (two divisors: 1 and 2).
\item 3 is prime (two divisors: 1 and 3).
\item 4 is not prime (\alert{three} divisors: 1, 2,
and 4).
\end{itemize}
\end{example}
\end{frame}
```

- A címdia tartalmazza a prezentáció címét, szerzőjét, a dátumot, helyszín stb. (amit csak akarunk)
- A dokumentumtörzsbe (ne a keretbe) adjuk meg a mű címét, szerzőjét és a dátumot:

```
\title{Az én első prezentációm}
\subtitle[rövid alcím]{Az én első prezentációm
alcíme}
%\author{Tóth Géza} vagy
\author{Tóth Géza\\ \texttt{tothgeza@hszk.bme.hu}}
\institute[BME]{\inst{1} Budapesti Műszaki és
Gazd.tud. Egyetem \\ \inst{2} Villamosmérnöki és
Informatikai Kar \\ Hálózati Rendszerek és
Szolgáltatások Tanszék}
\date[ISPN '80]{27. Nemzetközi Beamer Találkozó,
Szeged}
```

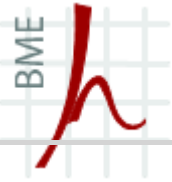
Ezt követően már létrehozhatjuk a címdiát:

```
\begin{frame}
\titlepage
\end{frame}
```

Keretek strukturálása 1

- Ha a keretben **nem** akarjuk megjeleníteni a fej- és lábléceket ill. az oldalsávokat, akkor `\begin{frame}[plain]` opciót kell használni!
- Ha Verbatim szöveget akarunk a keretbe tenni: `\begin{frame}[fragile]`
- majd `\begin{verbatim}...\end{verbatim}`
- Ha a listákat függőleges térközök nélkül akarjuk látni: `\begin{frame}[squeeze]`
- Kereteket a `\framebreak` paranccsal törhetünk vagy a `\begin{frame}[allowframebreaks=0..1]` környezettel.
- Itt a 0.5 például 50% keret töltöttség esetén automatikusan töri a keretet:

Példa I → Példa II → Példa III . . . kereteket kapunk



Példa:

```
\begin{frame}[allowframebreaks=0.5]{Title}
A\\ A\\ A\\ A\\ A\\ A\\ A\\ A\\ A\\ A\\ A\\ A\\ A\\
B\\ B\\ B\\ B\\ B\\ B\\ B\\B\\ B\\ B\\ B\\ B\\ B\\
\end{frame}
```

- Persze ha nem tetszik a keret tördelés így akkor módosítható:
- `\setbeamertemplate{frametitle continuation}[from second][\insertcontinuationcountroman.]`
- `\setbeamertemplate{frametitle continuation}[from second][\insertcontinuationcount.]`
- `\setbeamertemplate{frametitle continuation}[from second][(folyt.)]`

- Sajnos az `allowframebreaks` nem támogatja a több dia használatát a kereten belül, de sebaj:
- Használjuk a `\pause` parancsot:

```
\begin{frame} {Példa} {}  
  A keret 1.diájának tartalma\par\pause  
  A keret 2.diájának tartalma,  
  látható még mindig ua. a keret \par\pause  
  A keret 3.diájának tartalma,  
  továbbra is ua. a keret  
\end{frame}
```

- A prezentáció témájának megváltoztatása nélkül is lehet!
- A jól megszokott LaTeX fontok használhatóak itt is!
- Ha globálisan akarunk használni egy fontkészletet akkor a preambulumban kell elhelyezni:
- Pl. `\usepackage{times}`, `\usepackage{helvet}`

Lokálisan: Betűbeállításokat a következő paranccsal állíthatunk:

```
\setbeamerfont{betubeállítás neve}{beállítási lista}
```

A beállítási lista elemei (vesszővel elválasztva) a következők lehetnek:

- *size*=méret parancs
(méret parancsok: `\tiny`, `\scriptsize`, `\footnotesize`, `\small`, `\normalsize`, `\large`, `\Large`, `\huge`, `\Huge`)
- *shape*=alak parancs (Alak parancsok: `\upshape`, `\slshape`, `\itshape`, `\scshape`)
- *series*=testesség parancs (Testesség parancsok: `\mdseries`, `\bfseries`)
- *family*=család parancs (Család parancsok: `\rmfamily`, `\sffamily`, `\ttfamily`)

Példa: Előre definiálok betűtípus sablonokat:

- `\setbeamerfont{betu A}{size=\large}`
- `\setbeamerfont{betu B}{parent=betu A, shape=\slshape}`
- `\setbeamerfont{betu A}{size=\large}`
- `\setbeamerfont{betu B}{series=\bfseries}`
- `\setbeamerfont{betu C}{parent={betu A,betu B}, shape=\slshape}`

```
\setbeamerfont{betustilusom}{size=\large, series=
\bfseries
```

Majd a szövegben ahol kell meghívom:

```
\usebeamerfont{betustilusom} Szöveg %félköver
lesz
```

Hosszú prezentáció készítése

- Ha hosszú prezentációt készítünk célszerű több részre bontani.
- A PDF-ben így részekre lesz bontva a prezentációnk, ezért jobban átlátható lesz!
- Főleg ha tartalomjegyzéket is szeretnénk készíteni akkor elengedhetetlen!
- A Beamer (az article dokumentumoszályhoz hasonlóan) az alábbi felbontást támogatja:
- rész (part) → szakasz (section) → al-szakasz (subsection) → al-al szakasz (subsubsection)

```
\begin{columns} [opció]
\begin{column}{1. oszlop szélessége}
1. oszlop tartalma
\end{column}
\begin{column}{2. oszlop szélessége}
2. oszlop tartalma
\end{column}
...
\end{columns}
```

- **Az opciók:**

- `totalwidth`=szélesség → a többhasábos terület teljes szélessége
- `b` → az oszlopok alsó sorainak alapvonalát igazítja össze
- `c` → az oszlopok vertikális közepét igazítja össze
- `t` → az oszlopok felső sorainak alapvonalát igazítja össze
- `T` → az oszlopok felső sorainak tetejét igazítja össze

- Új részt a keret előtt ezzel a paranccsal kezdhetünk:
- `\part [rész rövid címe] {A rész teljes címe}`

- **Példa:**

```
\part [Bevezető] {Az első rész a Bevezetés}
  \begin{frame}
Első keret tartalma
\end{frame}
  \begin{frame}
második keret tartalma az Bevezető részben
\end{frame}
\part [Tárgyalás] {Az második rész a Tárgyalás}
  \begin{frame}
Első keret a Tárgyalás részben
\end{frame}
```

- Ha szeretnénk egy „címdiát” tenni minden rész elé akkor a következőt írjuk a preambulumba:

```
\AtBeginPart{  
  \begin{frame}  
  \begin{center}  
    {\Large\insertromanpartnumber. rész\}[10mm]}  
    {\large\insertpart\}  
  \end{center}  
  \end{frame}}
```

- Ekkor minden `\part{A rész címe}` kiadott parancs egy új címdiát hoz létre az épp aktuális keretnek!

- Természetesen a prezentációkat szakaszokra is tagolhatjuk a részeken belül:

```
\section[a szakasz rövid címe]{a szakasz címe}
```

- Ha minden szakaszra új „címdiát” szeretnénk akkor a preambulumba a következőt írjuk bele:

```
\AtBeginSection{  
  \begin{frame} [plain]  
  \begin{center}  
    {\Large\insertsectionnumber. \insertsection\\}  
  \end{center}  
  \end{frame} }
```

- Ezután a `\section{A rész címe}` parancs kiadása hasonló hatást eredményez, mint a `\part` esetében.

- Használjuk a már megismert `\tableofcontents` parancsot:

```
\begin{frame}
\frametitle{Tartalomjegyzék} % keret neve
\tableofcontents
\end{frame}
```
- **Vigyázat!** *Ha használunk `\part` parancsot akkor az utána írjuk a tartalomjegyzék keretet, mert különben üres diát kapunk!* Tartalomjegyzék **CSAK** section, subsection stb.-ből készül, a part-ból nem!
- Hosszú tartalomjegyzék több dián: (ha közte van egy `\part` akkor az első részén nem lesz rajta, csak a másodikon!)

```
\begin{frame}[plain]{I. rész tartalomjegyzéke}
\tableofcontents[part=1]
\end{frame}
\begin{frame}[plain]{II. rész tartalomjegyzéke}
\tableofcontents[part=2]
\end{frame}
```

- A tartalomjegyzéket lehet csinosítani is a beépített opciókkal!
 - *sectionstyle*= show, hide, shaded
 - *subsectionstyle*= show, hide, shaded
 - *subsubsectionstyle*= show, hide, shaded
- A show opció láthatóvá teszi, a hide elrejt, míg a shaded halványra varázsolja az adott szakaszt, alszakaszt, al-alszakaszt.
- **Példa:** Ha azt szeretnénk, hogy a tartalomjegyzékben csak a szakasz jelenjen meg, az alszakasz halványan, míg az al-alszakasz ne jelenjen meg:

```
\tableofcontents[sectionstyle=show,subsectionstyle=s  
haded,subsubsectionstyle=hide]
```


- A prezentációnk végén megemlíthetjük a felhasznált irodalmakat.
- Használata nem sokban tér el a már megszokott LaTeX-ben alkalmazottól:

```
\begin{frame}[plain]{Irodalomjegyzék}
\begin{thebibliography}{12}
\bibitem {Shannon1948} Claude E. Shannon, "A
  Mathematical Theory of Communication",
  \textit{Bell System Technical Journal},
  vol.~27, no.~10, pp.~379--423 és pp.~623--656,
  1948.
\bibitem {Dijkstra1982} E.~Dijkstra, ...
\end{thebibliography}
\end{frame}
```

- Csak egy példán keresztül mutatom be:

```
\begin{document}
\frame{\titlepage}
\section*{Outline}
\frame{\tableofcontents}
\section{Main Text}
\frame{Some text}
\section*{Summary}
\frame{Summary text}
\appendix
\section{\appendixname}
\frame{\tableofcontents}
\subsection{Additional material}
\frame{Details}
\frame{Text omitted in main talk.}
\subsection{Even more additional material}
\frame{More details}
\end{document}
```

BEAMER

MEGJELENÉS ÉS KERET KOMPONENSEK

1. Tartalom
2. Fej- és lábléc
3. Oldal sávok
4. Navigációs szimbólumok
5. Logó
6. Keret címe és alcíme
7. Háttér, rajta a vászon
8. (Témák)

What Are Prime Numbers?

What Are Prime Numbers?

Definition
A **prime number** is a number that has exactly two divisors.

Example

- 2 is prime (two divisors: 1 and 2).
- 3 is prime (two divisors: 1 and 3).
- 4 is not prime (**three** divisors: 1, 2, and 4).

Euclid Az én első prezentációm

- A keretben lévő prezentációs anyagot célszerű blokkosítani, hogy jól nézzen ki! Például blokkokkal, dobozokkal, felsorolásokkal stb.

Blokkosítás nélkül: Answered Questions
How many primes are there?
Open Questions
Is every even number the sum of two primes?

Blokkokba foglalva:

Ugye, hogy
szébb így!



Answered Questions
How many primes are there?
Open Questions
Is every even number the sum of two primes?

- Blokkokat a **block** környezettel lehet csinálni:

```
\begin{frame}
\frametitle{What's Still To Do?}
\begin{block}{Answered Questions}
How many primes are there?
\end{block}
\begin{block}{Open Questions}
Is every even number the sum of two primes?
\end{block}
\end{frame}
```

- A Beamer dokumentumosztály automatikus betölti az enumerate csomagot (itemize, enumerate, description)
- Használatuk a már megtanult módon:

```
\begin{frame}{}{}  
\begin{itemize}  
\item<+> 1. listaelem  
\item<+> 2. listaelem  
\item<+> 3. listaelem  
\end{itemize}  
\end{frame}
```

- Két speciális tömb is van, melyek alapvetően a színezésben térnek el: alertblock és exampleblock környezetek!
- exampleblock:

```
\begin{alertblock}{ide jon a nev}
```

```
$a^2+b^2=c^2$.
```

```
\end{alertblock}
```

ide jon a nev

$$a^2 + b^2 = c^2.$$

- alertblock:

```
\begin{exampleblock}{Példa}
```

```
$a^2+b^2=c^2$.
```

```
\end{exampleblock}
```

Példa

$$a^2 + b^2 = c^2.$$

- Az amsthm csomag is automatikus betöltődik, ezért a proof környezet azonnal használható: `\begin{proof} ... \end{proof}`

Proof.

ide jön a nev $a^2 + b^2 = c^2$.

- Tételeket a korábban tanult módon lehet megfogalmazni, de a következőkre oda kell figyelni:
 - A tételek overlay specifikációval is használhatók (később részletesen),
 - a tétel számozása alapesetben nem jelenik meg:
 - `\setbeamertemplate{theorems}[numbered]` preambulumba írásával (magyar nyelv esetén nem lesz jó)

- A LaTeX-ben meglévő dobozok használhatóak természetesen
- A Beamer 2 újabbal dobozt is bevezet:
 - Beamercolorbox
 - Beamerboxesrounded
- **Fontos!** A Beamer az egyéni színösszeállításnál két paramétert vár tőlünk:

```
\setbeamercolor{az en szineim most ezek}{fg=red,bg=blue}
```



A tartalom színe:
most piros



Háttér színe:
most kék

- Beamercolorbox:

```
\begin{beamercolorbox} [opció] {színösszeállítás}
```

A beamer colorboxban lévő tartalom

```
\end{beamercolorbox}
```

- A beamercolorbox opciói:

- wd=szélesség
- dp=mélység
- ht=magasság
- left
- right
- center
- sep=távolság
- shadow=true,false
- rounded=true,false

- **Példa:**

```
\setbeamercolor{az en szineim most ezek}{fg=red,bg=blue}
\begin{frame}{}{}
\begin{beamercolorbox}[wd=6cm,shadow=true,
rounded=true,center]{az en szineim most ezek}
Ez itt a dobozom hurrá!!!
\end{beamercolorbox}
\end{frame}
```

- A beamerboxesrounded doboznak cím is adható
- A doboz opciói:
 - width=szélesség
 - shadow=true,false
 - lower=színösszeállítás
 - upper=színösszeállítás

- **Példa:**

```
\begin{beamerboxesrounded}[opció]{Dobozom címe}  
Dobozom tartalma  
\end{beamerboxesrounded}
```

- Példa színezésre:

```
\setbeamercolor{egyeni1}{fg=red,bg=blue}
\setbeamercolor{egyeni2}{fg=yellow,bg=blue}
\begin{frame}{rovid keret cim}{hosszukeret cim}
\begin{beamerboxesrounded}[upper=egyeni1,
lower=egyeni2,shadow=true]{A dobozom címe}
Dobozom tartalma\ \ Doboz tartalma
\end{beamerboxesrounded}
\end{frame}
```

- képek beillesztése, hasonlóan a LaTeX-hez az `\includegraphics` paranccsal történik!

- Overlay effektek is rendelhetőek hozzá:

```
\includegraphics<2->[width=5cm]{kep}
```

- Pl. Animáció készítése képekből:

```
\includegraphics<+>[width=5cm]{figure0}
```

```
\includegraphics<+>[width=5cm]{figure1}
```

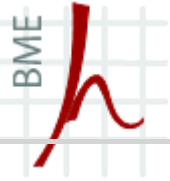
```
\includegraphics<+>[width=5cm]{figure2}
```

```
\includegraphics<+>[width=5cm]{figure3}
```

- Egy prezentáció megítélése nem kis részben a megjelenítésen múlik!
- A megjelenítést a Beamerben a témákkal tudjuk módosítani
- A Beamer készítői rengeteg teljes témát készítettek, amivel csinosíthatjuk a prezentációnkat:
- Néhány link ahonnan igényes témák szerezhetőek be:
 - <http://latex.simon04.net/>
 - http://deic.uab.es/~iblanes/beamer_gallery/
- Teljes témát a preambulumban a `\usetheme[opciók]{név}` paranccsal tölthetünk be!
- A témák tartalmazhatnak fej és lábléceket, oldalsávot stb., de ez téma függő!

- Egy teljes téma ún. külső és belső témákból áll össze!
- A külső téma a következőket tartalmazza:
 - *fej- és lábléc*
 - *oldalsávok*
 - *logó*
 - keret címe (már tudjuk, hogy kell)
- A belső téma a következőket tartalmazza: (ezeket is mind tudjuk)
 - címloldal
 - listák
 - tömbök
 - tételszerű környezetek
 - képek, táblázatok
 - lábjegyzetek
 - Irodalomjegyzék
- Ha egy teljes témából nem tetszik valami akkor másik témából választhatunk hozzá pl. fejléccet, színt stb.

- Teljes témát a `\usetheme[opciók]{név}` preambulumban történő kiadásával tölthetünk be.
- Pl.: `\usetheme{Singapore}` vagy `\usetheme{Warsaw}`
- Ez beállítja a témában foglalt színt, betű fontot és szerkezeti elemeket!
- Ha betűtípust akarunk változtatni:
`\usefonttheme[opciók]{név}`
- Ha színvilágot akarunk változtatni:
`\usecolortheme[opciók]{név}`
- Ha belső ill. külső témát akarunk változtatni:
`\useinnertheme[opciók]{név} \useoutertheme[opciók]{név}`



A Beamer téma „mátrix”-ja

- Érdeemes a teljes témák és a színvilág kínálatot megtekinteni:
- <http://www.hartwork.org/beamer-theme-matrix/>

Címdia megadása (ism.)

- A címdia tartalmazza a prezentáció címét, szerzőjét, a dátumot, helyszín stb. (amit csak akarunk)
- A dokumentumtörzsbe (ne a keretbe) adjuk meg a mű címét, szerzőjét és a dátumot:

```
\title{Az én első prezentációm}
\subtitle[rövid alcím]{Az én első prezentációm
alcíme}
%\author{Tóth Géza} vagy
\author{Tóth Géza\\ \texttt{tothgeza@hszk.bme.hu}}
\institute[BME]{\inst{1} Budapesti Műszaki és
Gazd.tud. Egyetem \\ \inst{2} Villamosmérnöki és
Informatikai Kar \\ Hálózati Rendszerek és
Szolgáltatások Tanszék}
\date[ISPN '80]{27. Nemzetközi Beamer Találkozó,
Szeged}
```

Ezt követően már létrehozhatjuk a címdiát:

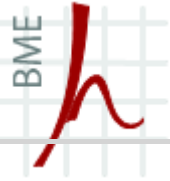
```
\begin{frame}
\titlepage
\end{frame}
```

- Fontra:
 - default (This is sans serif)
 - professionalfonts
 - serif
 - structurebold
 - structureitalicserif
 - strucutresmallcapsserif
- Színre:
 - albatross
 - beaver
 - Beetle
 - crane
 - Dolphin
 - dove
 - Fly
 - lily
 - Orchid
 - seahorse, whale, wolverine

Példák Inner és Outer témákra

- `\useinnertheme{rectangles}`
 - rectangles
 - circles
 - inmargin
 - rounded

- `\useoutertheme{infolines}`
 - infolines
 - miniframes
 - shadow
 - sidebar
 - smoothbars
 - smoothtree
 - split
 - tree



Fej és láblécek a keretben

- A téma betöltésével a fej és láblécek kinézete is adott lesz (már ha van a témában fej és lábléc)!
- A lábléc (általában) a szerzőt és a művet tartalmazza!
- Azért módosíthatunk rajta kicsit a `\setbeamertemplate` parancs segítségével (ez a témát módosítja):
 - Pl. színek: `\setbeamercolor{section in head/foot}{fg=red,bg=blue}`

- **Példa a szakaszok kiírására a fejlécben:**

```
\setbeamertemplate{section in head/foot}{\footnotesize\color{blue}
\insertsectionheadnumber. \insertsectionhead}
\setbeamertemplate{section in head/foot shaded}{\tiny\color{red}
\insertsectionheadnumber. \insertsectionhead}
```



proba dia intro

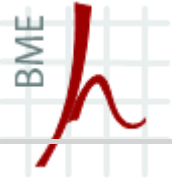


proba dia tárgyalásra

- Ha sok szakaszunk (section) van nem célszerű a fejrészben elhelyezni őket, hanem tegyük a bal vagy jobb oldali oldalsávba!
- Válasszunk egy olyan témát ahol van oldalsáv
- Pl. `\usetheme{Marburg}`
- Az egyetlen különbség `{section in head/footer}` helyett `{section in sidebar}`-t kell használni!

```
\setbeamertemplate{section in sidebar}{\footnotesize\color{blue}  
\insertsectionheadnumber. \insertsectionhead}  
\setbeamertemplate{section in sidebar shaded}{\tiny\color{red}  
\insertsectionheadnumber. \insertsectionhead}
```


- A legtöbb téma alapról beleteszi a prezentációnkba a navigációs szimbólumokat!
- Nagy előnye, hogy menet közben lehet ugrálni a fejezetek között, vagy keresni stb.
- Ha nincs minden szimbólumra szükségünk akkor a
- `\setbeamertemplate{navigation symbols} {ide írjuk ami kell}`
- Ezek pedig a következőek:
 - `\insertslidenavigationsymbol`,
 - `\insertframenavigationsymbol`,
 - `\insertsubsectionnavigationsymbol`,
 - `\insertsectionnavigationsymbol`,
 - `\insertdocnavigationsymbol`,
 - `\insertbackfindforwardnavigationsymbol`



Logó megadása

- A logó általában kép és szövegből áll.
- A logó pozícióját a téma állítja be és jelenleg ezen nem is lehet változtatni!
- Persze van olyan téma ami nem is támogatja ☹
- Példa:
`\logo{\includegraphics[height=0.5cm]{logo.pdf}}`

- Eddig alapból fehér (vagy az adott téma által definiált) volt a prezentáció háttérszíne.

- Háttérszínbeállítás parancs: (Például halvány pirosra)

```
\setbeamercolor{normal text}{bg=red!20}
```

- Vagy:

```
\setbeamercolor{background canvas}{bg=yellow!30}
```

- Ha nem elégszünk meg az egyszínű háttérrel készíthető színátmenetes is:

```
\setbeamertemplate{background canvas}[vertical shading][top=yellow,middle=blue,bottom=green]
```

- Rácsos hátteret a [grid] opcióval kapunk:

```
\setbeamercolor{background canvas}{bg=yellow}
```

```
\setbeamertemplate{background}[grid][step=3mm,color=black!15]
```

BEAMER

LÁTVÁNYELEMENK

- Az átfedések már korábban is előkerültek néhányszor (felsorolások, ábrák), most nézzük őket alaposabban!
- A Beamer bőségesen biztosít lehetőséget látványeffektek készítésére!
- `\pause` parancs megállítja a kiírást (pl. felsorolásban) majd ha továbbmegyünk a következő `\pause` parancsing folytatja a kiírást:

```
\begin{frame}
\begin{itemize}
\item Shown from first slide on.
\pause
\item Shown from second slide on.
  \begin{itemize} \item Shown from second slide on.
    \pause
    \item Shown from third slide on.
  \end{itemize}
\item Shown from third slide on.
\pause
\item Shown from fourth slide on.
\end{itemize}
\end{frame}
```

- Másik lehetőség:
- `<szam>` opció megadja, hogy a formázás melyik dián érvényesüljön:

- Példa 1:

```
\begin{frame}
\textbf{This line is bold on all three slides.}
\textbf<2>{This line is bold only on the second slide.}
\textbf<3>{This line is bold only on the third slide.}
\end{frame}
```

- Példa 2:

```
\begin{itemize}
\item<1-2> 1. listaelem
\item<2> 2. listaelem
\item<3> 3. listaelem
\item<3-4> 4. listaelem
\end{itemize}
```

- Ha azt szeretnénk, hogy bizonyos szöveg csak az első dián, vagy csak a másodikon vagy csak az első 5 dián szerepeljen akkor \only parancsot kell kiadni:

- `\only<overlay specification>{<text><overlay specification>}`

- **Néhány példa:**

```
\begin{frame}
```

```
\only<1>{Ez a tartalom csak az első dián látható.}
```

```
\only<2>{Ez a tartalom csak a 2. dián látható.}
```

```
\only<1,2,5,6>{Ez a tartalom csak az 1.,2.,5. és 6. dián látható.}
```

```
\only<1-3,6>{Ez a tartalom csak az 1.,2.,3. és a 6. dián látható}.
```

```
\only<-3,6-8 >{Tartalom csak a 1.--3. és a 6.--8. dián látható}.
```

```
\end{frame}
```

- Egy újabb lehetőségek a dián megjelenő szöveg manipulálására:
 - `\onslide<<overlay specification>>{<text>}`
 - Példa: `\onslide <1,3>{Same effect as the previous command.}`
 - `\uncover<<overlay specification>>{<text>}`
 - Példa: `\uncover<2>{ A 2. dián kívül mindenhol rejtve marad.}`
 - `\visible <<overlay specification>>{<text>}`
 - Példa: `\visible<2>{ A 2. dián kívül mindenhol rejtve marad.}`
 - `\invisible <<overlay specification>>{<text>}`
 - Példa: `\invisible<2>{ A 2. rejtve marad az itt lévő szöveg.}`

- A \alt parancs két szöveg környezetet tartalmaz, a <> megadott diákon az első szöveg környezet jelenik meg a többin pedig a második természetesen:
- Példa:
`\alt<1,3>{Ez itt csak az 1. és 3. dián jelenik meg}{ez meg a maradékon}`

- A léptető overlays az ún. beamerpauses számlálót használják, melynek kezdeti értéke 1.
- Az egyik léptető overlay specifikáció a +(szám), ahol a szám bármilyen egész érték lehet, akár negatív is:
- Vagyis ha a beamerpause számláló értéke 3 akkor a:
 - $\langle+(0)\rangle = \langle 3\rangle$
 - $\langle+(1)\rangle = \langle 4\rangle$
 - $\langle+(-1)\rangle = \langle 2\rangle$
 - $\langle+-+(2)\rangle = \langle 3-5\rangle$
 - $\langle-+\rangle = \langle -3\rangle$
 - $\langle.+ \rangle$ - a beamerpause értékénél 1-el kevesebbet ír a pont helyére

- Példa:

```
\begin{frame}{Példa}{}
\begin{itemize}
  \item<+> 1. listaelem
  \item<.> 2. listaelem
  \item<+> 3. listaelem
  \item<.> 4. listaelem
\end{itemize}
\end{frame}
```

- A felsorolásoknál jobban mutat, ha a többi pont is fel van sorolva halványan:

- Példa:

```
\setbeamercovered{transparent=25}
```

```
\begin{frame}{Példa}{}
```

```
\begin{itemize}
```

```
\item<+> 1. listaelem
```

```
\item<+> 2. listaelem
```

```
\item<+> 3. listaelem
```

```
\item<+> 4. listaelem
```

```
\end{itemize}
```

```
\end{frame}}
```

- Itt a többi pont is látszik halványabban (25% intenzitással)

- A `\setbeamercovered{transparent=hany % az átlásztósság}` parancs használható `\uncover`-nél is:
- Példa:

```
{\setbeamercovered{transparent=20}  
\begin{frame}{Példa}}  
\uncover<1>{Ez az elso dia, }  
\uncover<2>{ez pedig a második.}  
\end{frame}}
```

- Linkekkel és gombokkal is szebbé tehetjük a prezentációnkat:
 - A már korábban megtanult `\label{} \ref` páros itt is használható, de ha olyan linket akarunk amit mi nevezünk el:
 - `\hyperlink` parancs kell

- Példa:

```
\begin{frame}[label=cimke]{Példa}}
```

```
\begin{itemize}
```

```
\item<+>-> 1. listaelem
```

```
\item<+>-> 2. listaelem
```

```
\end{itemize}
```

```
\end{frame}
```

```
\begin{frame}
```

```
\hyperlink{cimke<2>}{Az elozo keret 2. diájára ugrás.}
```

```
\end{frame}
```

Gombok és linkek készítése 2.

- Gombok megadására több opció is kínálkozik:

`\beamerbutton{gomb szövege}` -

`\beamergotobutton{gomb szövege}`

`\beamerskipbutton{gomb szövege}`

`\beamerreturnbutton{gomb szövege}`

Elozo keret 2. diája

▶ Elozo keret 2. diája

▶▶ Elozo keret 2. diája

◀ Elozo keret 2. diája

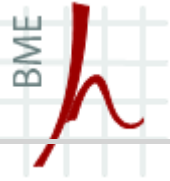
```
\begin{frame}
\hyperlink{cimke<2>}{\beamerbutton{Elozo keret 2. diája}}
\hyperlink{cimke<2>}{\beamergotobutton{Elozo keret 2. diája}}
\hyperlink{cimke<2>}{\beamerskipbutton{Elozo keret 2. diája}}
\hyperlink{cimke<2>}{\beamerreturnbutton{Elozo keret 2. diája}}
\end{frame}
```

- Animáció készíthető képekből a `<+>` overlay opció segítségével

```
\includegraphics<+>[width=5cm]{figure0}
\includegraphics<+>[width=5cm]{figure1}
\includegraphics<+>[width=5cm]{figure2}
\includegraphics<+>[width=5cm]{figure3}
```
- Ha sok képből álló animációt akarunk akkor a fenti megoldás kényelmetlen ezért helyette a `xmpmulti` csomag kínálja lehetőségekkel egy sorba is leírható.
- Ha az animáció ábrái `abra-1.jpg...abra-20.jpg` akkor így egyszerűbb:

```
\multiinclude[<+>][format=jpg,graphics={width=5cm}]{abra}
```
- Ez már egy fokkal jobb, de még mindig kézzel kell váltani a diák között!

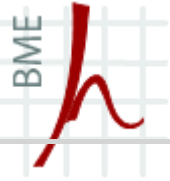
- Töltsük be a animate csomagot!
- `\usepackage{animate}` (megjegyzés: `animate.sty` és a `animfp.sty`-ra is szükség van)
- A parancs pedig:
`\animategraphics[opciók]{sebesség}{alapnév}{első}{utolsó}`
 - *sebesség* : pozitív egész szám ennyi képet játszik le másodpercenként,
 - *alapnév*: az ábrák alap neve: pl.`abra-1...abra-20` esetén: `abra`
 - *első* és *utolsó* : az ábrák közül az első és utolsó szám :
Pl. 5. ábrától a 15.-ig szeretnék egy animációt!
 - Opciók: (a teljesség igénye nélkül)
 - *autoplay* – automatikusan elkezdi lejátszani
 - *loop* – végtelen ciklusban lejátszás
 - *controls* – vezérlőgombok megjelenítése



Animációk példa

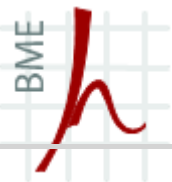
- `\begin{frame}`
- `\centering`
- `\animategraphics[loop,controls,width=3cm]{2}{monkey-}{0}{3}`
- `\end{frame}`

- Valódi videót játszunk le a `\href{run:video.avi}{link szövege}` parancs segítségével (ehhez kell természetesen egy hozzárendelt videó látszó program (pl. avi- windows media player))
- Ha prezentáció egy keretén belül szeretnénk videót lejátszani, akkor a **multimedia** csomag kell hozzá!
- **DE** vigyázzunk arra, hogy a videofájl a pdf mellett legyen!
- `\movie[opciók]{poszter szövege}{videófájl neve}`
- Példa:
`\movie[width=8cm,height=6cm,showcontrols,poster]{}{video.avi}`



Hang lejátszása prezentáció alatt

- Lehet a videóhoz hasonlóan linkkel: `\hyperlinksound`
- Vagy a multimedia csomaggal:
 - Csak egy példa:
 - `\sound[autostart,samplingrate=22050]{}{applause.au}`



Vége

Ennyi fért a félév anyagába, talán hasznos is volt. Remélem sikerült megismerni a LaTeX világát és alkalmatok nyílik később használni az itt tanultakat!

Köszönöm az egész féléves figyelmet!