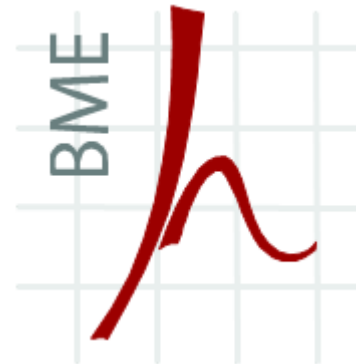




DOKUMENTUMSZERKESZTÉS

L^AT_EX

*A LaTeX kurzus előadás vázlatát elkészítette:
Dr. Jeney Gábor és Jakó Zoltán*

Gódor Győző
tudományos segédmunkatárs
BME Hálózati Rendszerek és Szolgáltatások Tanszék
godorgy@hit.bme.hu



2019. május 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 jelentkezett-e, egyébként nem kaphat jegyet a félév végén!!!**
- További kurzusok a tárgyon belül:
 - LaTeX kurzusok – L
 - (OpenOffice kurzus – O)
 - MS Word kurzus – W
- Tegezzetek! Én is így teszek!

Adminisztráció és első lépések 2.

■ Óra tartás:

- Az órák szerdán 12:15-től kezdődnek **ITT!** (R épület **R4J**),
- 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. – **Ez sajnos a GDPR miatt nem működik.**
- A jelenléti statisztika megtekinthető a tárgy oldalán.

Adminisztráció és első lépések 3.

- **Követelmények:**
 - A tárgy weboldala: <http://moodle.hit.bme.hu>
 - Dokumentumszerkesztés L2 2018/19/2
 - Az előadás anyaga (jelen diasor) szintén fellelhető az oldalon.
- A félév végén **nagy házi feladatot** kell beadni. A feladatot a 10. hét környékén osztom ki, és a 12. hét környékén kell beadni a fenti webcímen.
 - A nagy házi feladat leadásával késni nem lehet! Aki késik, az nem kap jegyet. – Érdemes időben elkészíteni.
- A félév során **9 darab órai feladatot** lehet megoldani, amelyeket az óra vége előtt kell feltölteni a fenti weboldalon. Csak a nem hiányzó feladatát értékelem. Ezekből a **legjobb 6 eredménye** játszik szerepet a végső jegyben.
- **A félév végi jegy két részből tevődik össze:** a nagy házi feladat 2/3 arányban, az órai feladatok 1/3 arányban számítanak be. Kétes jegy esetén a késések és hiányzások száma dönt.

- <http://moodle.hit.bme.hu/> – itt találhatók az előadás 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://moodle.hit.bme.hu>

Felhasznált irodalom

- 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*

Tárgy célja, küldetése

- 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 2016, LibreOffice stb. felhasználói szintű ismereteinek 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, dokumentumszerkesztési 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 IGÉNYYES 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 op. rendszert és nyelvet

- **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).

Magyar nyelv sajátosságai 1. – írásjelek

- Régen a Latin1 karakterkódolás miatt olyan betűket használtunk, ami hivatalosan nincs is magyar ABC-ben: Ő, û stb.
- *(A mondatok nagy betűvel kezdődnek és ponttal, felkiáltójellel, vagy kérdőjellel záródnak!)*
- Mondaton belül vesszőt, pontosvesszőt, kettőspontot, gondolatjelet és elválasztás esetén kötőjelet használunk.
- Magyarban a felsorolás esetén a felsorolt tagok közé vessző kerül (kivéve az „és” szó elé).
- Angolban más szabályok vannak érvényben; két tag közé csak egy „and” szócska kerül (Bee and Boe), három tagból az első kettőt vesszővel választjuk el (Lock, Stock and Two Smoking Barrels), négy, illetve több tag esetén mindenhova kell vessző (vodka, gin, whiskey, and tequila).

- A nyitó zárójel és záró zárójel közé kerülő szavakat szorosan kell bezárni, szóköz nélkül.
- Idézőjelek
 - **magyarban:** „99” alakúak,
 - **brit angolban:** ‘British English’,
 - **amerikai angolban:** “American English”,
 - **német nyelvben:** „Deutsch“,
 - **francia nyelvben:** « Français »
- *Magyarban az idézőjelek egymásba ágyazásánál a „normál” idézőjelen belül „a »hegyével befelé forduló jelpár« (lúdláb) megoldást használhatjuk”.*
- " jel (az angol hüvelyk – inch) távolságmérték.
- ' jel a percet, duplán " a másodpercet jelöli (főleg földrajzi koordinátákban), így a 20°11'12" egyértelműen nem idézőjelek.

- Számok kiírása: 100≠100,
- Továbbá, ha egyszerű számról van szó, célszerű kiírni betűvel, hiszen úgy jobban mutatnak:
 - *2 véleményen voltunk, pedig 100-szor elmondtam, hogy nincs igaza.*
 - *Két véleményen voltunk, pedig százszor elmondtam, hogy nincs igaza.*
- **Magyarban tizedesvesszőt, angolban tizedespontot használunk. – Ne keverjük!** $\pi=3,14$ vagy $\pi=3.14$
- A nagy számok – például 1 234 567,89 Ft – helyiértékeinek megtalálását segíthetjük térközök beszúrásával.
(Vigyázzunk, hogy a térközöknél ne törhessen a sor!)

- Angolban erre a célra a vesszőt használják, azaz \$1,234,567.89.
- Azonban ha a szám nem nagyobb 9999-nél, akkor nem szokás a helyiértéket jelölni (2 391 idétlenül mutat).
- **Pénznemek írásmódja:**
 - Magyarban az összeg után nem törhető szóközzel következik a pénznem jelölése (100 Ft), angolban a pénznem szóköz nélkül előzi meg az összeget (\$100).
 - Léteznek olyan valutanemek, amelyek pénzüpiaci rövidítésükkel szerepelhetnek (CHF) vagy speciális jellel (\$, £ stb.).
 - Magyarban a pénznemek rövidítve mindig nagy betűvel kezdődnek, kiírva viszont kis betűvel kell őket kezdeni.
 - A milliós tételeket nagy „M”-mel, a milliárdos tételeket „Mrd”-vel jelöljük.

Mértékegységek írása

- Minden esetben a mennyiséget jelölő szám után kell írni.
- Magyarban egy **nem törhető szóköz kerül a szám és a mértékegység közé**, angolban nem!
- **A mértékegységek ált. rövidítések, de nem szabad pontot tenni utánuk.**
- Rövidített alakban kis- és nagybetűk változatos formában lehetnek jelen, kiírva viszont mindig kis betűvel kell írni a mértékegységeket: 20 Hz, azaz húsz hertz.
- SI módosítók mindig a mértékegység elé: 20kHz

módosító	jelentése	tipikus példa
p	piko (10^{-12})	pF (pikofarad)
n	nano (10^{-9})	nH (nanohenry)
μ	mikro (10^{-6})	μ s (mikroszekundum)
m	milli (10^{-3})	mg (milligramm)
c	centi (10^{-2})	cl (centiliter)
d	deci (10^{-1})	dm (deciméter)
h	hekto (10^2)	hl (hektoliter)
k	kilo (10^3)	kA (kiloamper)
M	mega (10^6)	MV (megavolt)
G	giga (10^9)	GW (gigawatt)
T	tera (10^{12})	TJ (terajoule)

Időpontok/dátumok írása

- óra, perc, másodperc (utóbbi elhagyható) értékeket írjuk ki kettősponttal elválasztva, például: 21:16:34
- Ha a „de.” (délelőtt), vagy „du.” (délután) rövidítéseket – angolban „am” (ante meridiem) illetve „pm” (post meridiem) – akarjuk használni, **akkor a 12 órás formátumot kell alkalmaznunk**, például a 9:16:53pm helyes, viszont a du. 21:17:12 helytelen!
- „2015-01-02” a következő módokon írható:
 - 2015. január 2.
 - 2015. jan. 2.
 - 2015. I. 2.
 - 2015. 01. 02.
 - 2015-01-02

- Magyarban háromféle kötőjel szimbólum van:
 - **kötőjel** -,
 - **nagy kötőjel** --,
 - **gondolatjel** -- (nyomdatechnikailag ua., mint a nagy kötőjel).
- Kötőjelek alkalmazása: elválasztásnál, szókapcsolatoknál (héba-hóba), rövidítéseknél (kft.-nek), körülbelüli értékeknél (harminc-negyven), személyekhez köthető fogalmakkor (Jacobi-mátrix), földrajzi neveknel (Széchenyi-hegy) stb. használjuk.
- Nagy kötőjelek alkalmazása: intervallumokat (181–201. oldalig), ellentétpárokat (nyári–téli időszámítás), tulajdonnév jellegű szókapcsolatokat (B–52-es bombázó), vagy szerzőpárosokat (Bródy–Szörényi) jelöl.

- Vízszintes térköz beállítása **NEM** a space billentyű nyomogatásával!
- Függőleges térköz beállítása **NEM** az újsor karakterekkel.
- **nem törhető újsor karakter:** nem engedi a lapdobást a sortörés helyén.
- **nem törhető szóköz:** normál szóköz helyett használható, a két szó mindig ugyanabban a sorban marad.
- *Európán kívül:* a mondat végét jelző jel utáni szóköz (**mondatköz**) nagyobb, mint a szavak közöttiek.
- *Európában* (így a *magyarban* is) a tipográfiai hagyományok alapján **nem alkalmazunk a szóköztől eltérő méretű mondatközt.**

■ 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 fatincidunt sed, mollis sed.

Quisque sollicitudin ti ncidunt

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 facilisi. Nunc nulla neque, laoreet a, tincidunt sed, mollis sed, pe-

8

desol.

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

9

Felsorolások, listák

1. Számozott listák: *arab, és római számokkal.*
2. Számozatlan listák: *különleges karakterekkel.*
 - Amire figyeljünk:
 - hierarchia megtartása,
 - lista elemei legyenek jól elkülöníthetőek;
 - ha hivatkozunk listaelemre, akkor számozott listát használjunk.
3. **Leíró lista** esetében a felsorolás minden tagja kap egy szöveges címkét, amely valamilyen módon (esetünkben vastagítással) el van különítve a szövegtől.

- Nagyobb egyenleteket külön sorba kell írni.
- Csak azokat az egyenleteket számozzuk, amit később használunk is.
- Számozott képletek után a sor végén jobbra zárva egy szám található 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 táncos 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 $\underline{\underline{M}}$ mátrix dupla aláhúzással),
 - van, aki felülvonással
($\overline{v}$ vektor egy felülvonással és $\overline{\overline{m}}$ mátrix dupla felülvonással),
 - van, aki vastagítással
($\mathbf{v}$ vektor kisbetűvel és $\mathbf{M}$ mátrix nagybetűvel).
 - A lényeg, hogy egységes legyen a jelölés a teljes dokumentumban!

Gyakori nyelvtani hibák

- Visszautalás: „ebbe”, „erre”, „ez”, „ettől”, „ilyen”, „itt” stb.
- A magyarban két határozott névelő van: az „a” és az „az”. Azonban jól meghatározott szabályok írják elő, mikor melyiket kell alkalmazni, nem választhatunk közülük tetszőlegesen.
- A „-nak”, „-nek”, „-ra”, „-re”, „-ba”, „-be”, „-ban”, „-ben” stb. toldalékokat ugyanúgy kell használni számokhoz, rövidítésekhez kapcsolva, mint egyébként. 23-ban, 2.-ra.
- „-val”, „-vel”, „-vá”, „-vé” toldalékoknál elfelejtkeznek a hasonulásról, esetleg hagyják a „v” betűt.
 - **Helytelenül:** 55-el, Shannon-val, Totómix-el
 - **Helyesen:** 55-tel, Shannonnal, Totómix-szel

Gyakori nyelvtani hibák folyt.

- Sokszor nehéz elkerülni a **szóismétlést**, *azonban mindig ellenőrizzük*, hogy vajon nem azért maradt-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”, „legeslegmaximálisabb” szavaknak nincs sok értelme.
- Idegen szöveg tükörfordítása kapcsán jelentkező szórendhibák.

A dokumentum szerkezete

- A 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 készült 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.

A LaTeX története 1.

- 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).

A LaTeX története 2. – előnyök

▪ Különlegessége és egyben előnye is:

- Elég idős (elmúlt 30 éves).
- Minimális a hardverigénye.
- Ingyenes és nyílt forráskódú.
- 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:

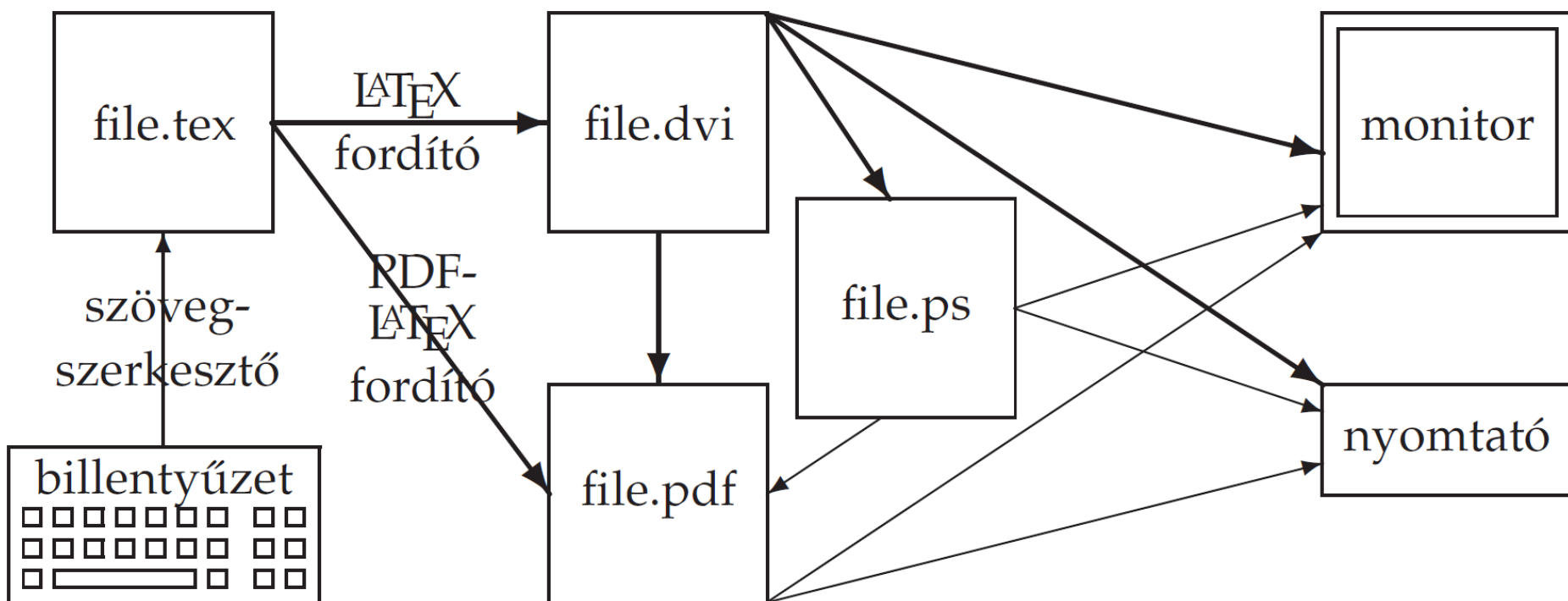
- Új stílus kialakítása bonyolult tud lenni.
- Hibaüzenetek megfejtése nehézkes.
- A LaTeX parancsokból és a szövegből álló dokumentum körülményesebb, nehezebben áttekinthető, mint a szokásos WYSIWYG rendszerek végeredménye.
- Leírás nélkül nehéz boldogulni, pl. egy új csomag használata.

Számos (szerkesztő és fordító) program fellelhető:

- Windows rendszerekben:
 - **MiKTeX** (<http://miktex.org>)
 - **TeXworks** (<http://www.tug.org/texworks/>) (összenőtt a MiKTeX-hel)
 - Texmaker (<http://www.xm1math.net/texmaker/>)
 - TeXstudio (<http://www.texstudio.org>)
 - WinEdt (<http://www.winedt.com>)
- Linux:
 - TeX Live (<https://www.tug.org/texlive/>)
 - Texmaker
- Mac OS X:
 - TeX Live
 - Texmaker
 - TeXworks

A dokumentum fordítás menete

- A billentyűzettől a kész műig:



Egyszerű példa

```
\documentclass{article}
%\usepackage[utf8]{inputenc} %Ékezetes karakterekhez

\begin{document}
    Hello World!!!
\end{document}
```

L^AT_EX ALAPOK

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

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

- Mint látható, minden LaTeX parancs „\”-lel kezdődik!
- Az [opciók] paraméter tartalmazza az eredeti alapbeállításoktól való eltérést. Később nézünk rá néhány példát!
- A {} közti részt a dokumentumosztály **argumentumának** nevezzük.
- A dokumentumosztályok típusai a következők:
 - article (cikkek írásához)
 - report (beszámolók írásához)
 - book (könyv írásához)
 - letter (levelek írásához)
 - slides (prezentációk készítéséhez)

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 parancs**nak nevezzük: `{\bfseries ...}`
- **Példa:** `{\bfseries Csak ez fog félkövéren megjelenni.}`
Ez már nem.
2. Végül pedig 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,a4paper]{article}
```

Dokumentumosztályok opciói 2.

II. Report dokumentumosztály:

- Állítható opciók:
 - Megegyezik az Article dokumentumosztály opcióival, **DE**
 - Ezenkívül van még egy opciója: `openright`, `openany`
- Az egyes 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: 10pt, letterpaper, oneside, titlepage, openany, final.

Példa:

```
\documentclass[12pt,a4paper]{report}
```

III. Book dokumentumosztály:

- Állítható opciók: Egyezik az Article és Report dokumentumosztályoknál látottakkal.

Alap beállítások: 10pt, letterpaper, twoside, titlepage, openright, final.

Példa:

```
\documentclass[12pt,a4paper]{book}
```

Magyar szöveg írása

- Az előző példánkban nem használtunk magyar ékezetes karaktereket.
- Ékezetes karakterek írására két lehetőség adódik:
 - Ékezetes karakterek megjelenítése az alábbi paranccsal lehetséges: \
 Ez minden utána álló karakterre egy vesszőt helyez, pl.: \e, \u, \s.
 - További lehetőségek:

ó \'{o}	õ \~{o}	ō \={o}	o \b{o}	ö \u{o}	ı \i
ò \'{o}	ô \^ {o}	ó \. {o}	o \c{o}	ő \v{o}	j \j
ö \"{o}	ő \H{o}	o \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 üűgyes kérem szépen
\end{document}
```

- Figyelem! Bizonyos szövegszerkesztők nem UTF-8-ban mentik a fájlokat, hanem közép-európai kódolással. Ha így lenne, akkor `utf8` helyett `latin2`-t kell írni.
- Szükséges lehet még a `\usepackage{t1enc}`, ha egyéb speciális nyelvi karakter megjelenítésére van szükségünk, pl.: Đurđevac, Pórshöfn.

- A `babel` csomaggal magyaríthatjuk a LaTeX-et:
`\usepackage[magyar]{babel}`
- Bár a csomag általában jól választ el, azonban van, amit nem tudhat, pl. karóra (karó-ra, vagy kar-óra). Ilyenkor két lehetőségünk van:
 - Lokális megkerülő megoldás: „*puha*” elválasztás kézzel „\” alkalmazásával:
(`kar\-ó\ -ra`)
 - Globális megkerülő megoldás: kiadhatunk egy `\hyphenation{el-vá-lasz-ta-ni-mer-kell-neki-tyuhajja}` parancsot a dokumentum tetszőleges részén, így a szerkesztő tudni fogja, hol kell elválasztani ezt a meglehetősen összetett szót.
Figyelem! Ha nincs betöltve a `tlenc` csomag, ékezetes karaktereket tartalmazó szavak esetén nem fog működni a `\hyphenation` parancs!
- Ha **többs nyelvű dokumentumot** szeretnénk írni:
`\usepackage[english,magyar]{babel}` paranccsal indulunk, és a
`\selectlanguage{nyelv}` paranccsal lehet kiválasztani az aktuális nyelvet.

Speciális karakterek, parancsok 1.

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

$\TeX$, $\LaTeX$ és $\LaTeX 2_{\epsilon}$

Speciális karakterek, parancsok 2.

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

Speciális karakterek használata

- A dokumentumokat bekezdésekkel strukturáljuk. 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 („kisel, nagyó, nagyó”) $\neq$ 100 (száz).
 - Magyarban tizedesvessző, angolban pont van! 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 angolban “different”.
Bár más tördelőkből a hüvelyk (” shift+2) jelet használjuk, itt nem jó! Ez nem idézőjel!!! Az idézőjelek helyes megjelenítéséhez szükséges a `tlenc` csomag!

Ligatúrák és mondatközök

- Ligatúrán a betűknek a szokásosnál szorosabb összekötését értik. A legismertebbek a következő, ún. f-ligatúrák: `ff fi fl ffi fl1`. 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, a `\frenchspacing` paranccsal kikapcsolhatjuk.
De például az 1. ábra nem mondatvég, hiába van pont, ezért 1.~ábra.
- Megjegyzés: LATEX alatt nem baj, ha több space-t ütött le az ember:
`három_holló_=_három_holló` = három holló

Kötőjel, nagy kötőjel, gondolatjel, mínusz jel

- Kötőjel (-): sima „-” jel. Használata: „Kérek 3-4 darab paradicsomot”.
- 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!!!).
- Mínuszjel (-): \$-\$, odakint -5°C van, odakint -5°C van.

1. órai feladat

- Készítsetek különféle dokumentumokat az egyes dokumentumosztályok összehasonlítására!
 - Vizsgáljátok meg a különféle opciók hatásait!
- Készítsetek ékezetes karaktereket tartalmazó szöveget!
 - Próbáljátok ki mindkét megoldást!
- A szöveg tartalmazzon speciális (LaTeX által foglalt) karaktereket!
- Vizsgáljátok meg a kötőjelek, gondolatjelek közötti különbségeket!
- Vizsgáljátok meg az ligatúrákat!
 - Írjatok példákat (pl. maffia, affinitás stb.) hagyományos betűkkel, illetve ligatúrákkal!

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 dokumentum címét, 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 ...}`,
 - Grotteszk: `\textsf{...}`,
 - Írógép: `\texttt{...}`, `{\ttfamily ...}`.

- 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, célszerű az `\emph{...}` parancsot használni.
- (Az `\emph{}` utasítás egymásba is ágyazható:
`\emph{Akár hiszed, \emph{akár} nem}`)
- Ez figyeli az aktuális betűváltozatot, és aszerint emel 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 alap betűváltozatot 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 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 sorkizárással
 - 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` paranccsal kettes sorköz állítható be).

Fejezetek, alfejezetek írása

- *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.**

```
\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}
```

Oldaltörés és új bekezdés

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

`\newpage` – új oldal (képek, táblázat átkerülhet az új oldalra)

`\clearpage` – új oldal

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

`\nopagebreak`

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

Bekezdések igazítása

- Alapesetben a bekezdések sorkizártak.
- Bekezdések balra zárása: `\raggedright`

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

- Bekezdések jobbra zárása: `\raggedleft`

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

- Bekezdések középre zárása: `\centering`

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

Láb- és széljegyzet írása

- **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 előlről kezdődjön a számozás:

```
\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 elhelyezni, adjuk ki a következő parancsot:
`\reversemarginpar`, visszatérés a külső margóra: `\normalmarginpar`

- A `\pagestyle{sty}` utasítással a dokumentum egészét, a `\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 elhelyezett fejlécbe írja az oldalszámot:

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

2. órai feladat

- Hozz létre egy Report dokumentumot, ami legalább öt oldal terjedelmű legyen!
- A dokumentumon belül használj egy általad kiválasztott betűtípust (nem az alapértelmezettet), a betűméret 12pt legyen!
- Készíts címloldalt, mely tartalmazza a szerzőt, a mű címét, valamint a mű elkészítésének dátumát!
- A mű tartalmazzon tartalomjegyzéket!
- Az oldalszámozás a tartalomjegyzékig bezárólag kisbetűs római, míg a tényleges szöveg arab számozással legyen megvalósítva!
- A létrejövő mű tartalmazzon legalább négy szakaszt, melyek közül legyen egy Bevezető és egy Összefoglaló nevű szakasz! A szakaszok tartalmazzanak további alszakaszokat is!
- Az alszakaszokban az írott szöveg másfeles sorközű legyen!
- Gyakorold a megtanult szövegformázási lehetőségeket (pl. új bekezdés, kiemelés, betűméret változtatása)!
- Illessz be a műbe néhány széljegyzetet a belső és a külső margóra egyaránt, esetleg szúrj be lábjegyzetet is!
- Illessz a dokumentumba élőfej és élőlábat, úgy, hogy az eltérő legyen a páros, illetve a páratlan oldalakon!

KÖRNYEZETEK, FELSOROLÁSOK

- A `\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}
```

Versek és idézetek írása

- A `quote`, a `quotation` és a `verse` környezetekkel! Próbáljuk ki!
- A `quote` környezetben nem kezdődnek behúzással a bekezdések.
- A `quotation` környezetben a bekezdések behúzással kezdődnek.
- A `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
figyelnünk kell
arra is, hogy egy új bekezdést is kezdjünk.
Ha elég hosszúak a sorok, akkor
megláthatjuk, miféle
különbséget rejt e két környezet, persze
figyelnünk 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 – `verbatim`-ot szeretnénk, akkor célszerű a `\begin{verbatim} ... \end{verbatim}` környezetet használni.
- Fontos: `verbatim` környezet esetén a hosszú sorok nem tördelődnek automatikusan, kézzel kell tördelni!
- A fenti utasítás és környezet csillagos változata megjeleníti a szóközöket is, pl. `\verb*-Helló világ!` hatására a `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 } }`
- `lstlisting` környezetben lehetőség van a hosszú sorok automatikus tördelésére is a `breaklines=true` kapcsolóval.
- Ha utf8 kódolással dolgozunk, akkor a `listingsutf8` csomagot használjuk!

ABAP	Euphoria	ML	Rexx
ACSL	Fortran	Modelica	RSL
Ada	GCL	Modula-2	Ruby
Algol	Gnuplot	MuPAD	S
Ant	Haskell	NASTRAN	SAS
Assembler	HTML	Oberon-2	Scilab
Awk	IDL	Objective C	Sh
Bash	Inform	OCL	SHELXL
Basic	Java	Octave	Simula
C#	JVMIS	Oz	SQL
C++	Ksh	Pascal	Tcl
C	Lisp	Perl	TeX
Caml	Logo	PHP	VBScript
Clean	Lua	PL/I	Verilog
Cobol	Make	Plasm	VHDL
Comal	Mathematica	POV	VRML
Csh	Matlab	Prolog	XML
Delphi	Mercury	Promela	XSLT
Eiffel	MetaPost	Python	
Elan	Miranda	R	
Erlang	Mizar	Reduce	

Programkód beillesztése 1.

- 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 kerettel

```
\begin{lstlisting}[language=matlab, frame=single]
% 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}
```

Programkód beillesztése fájlból

- Ha lusták vagyunk, bemásolhatjuk közvetlenül fájlból a programkódot:

```
\lstinputlisting{matlab_file.m}
```

```
\lstinputlisting[language=Python]{source.py}
```

További hasznos leírás

- Lehetőség van programkód színezésére és egyéb paraméterek módosítására is:
 - Hasznos link:
`https://en.wikibooks.org/wiki/LaTeX/Source\_Code\_Listings`
- Algoritmusok és pszeudo kódok beillesztésére:
 - Algorithms csomag:
`https://en.wikibooks.org/wiki/LaTeX/Algorithms`

URL megadása

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

- Háromféle 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}
```

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

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

Számozási stílus megváltoztatása

- Az enumerate számozása alapesetben arab számozás, 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{tlenc}
\usepackage[utf8]{inputenc}
\usepackage[magyar]{babel}
\usepackage{paralist}
\begin{document}
\begin{enumerate}[I-{\a}]
\item a busa husa rendkívül finom,
\item a süllő sült nagyon ízletes.
\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 a butaságok listába szedve sem válnak
tudományossá,
\item a jó gondolatok viszont szebben mutatnak
rendszerelve.
\end{compactenum}
```

Listák személyre szabása

- Ha szeretnénk módosítani a felsorolási jelek utáni térközt, esetleg a listaelemek behúzásának mértékét, akkor szükségünk van az `enumitem` csomagra.
- Lehetőségünk van akár listánként is módosítani az egyes paramétereket, de globálisan is beállíthatjuk azokat.
- Példa (listánként):

```
\begin{itemize}[labelindent=1.5em,labelsep=1cm,leftmargin=*]  
  \item \blindtext  
  \item Two  
  \item Three  
\end{itemize}
```
- Példa (globálisan):

```
\setitemize{labelindent=1.5em,labelsep=1cm,leftmargin=*}
```
- Ha az egyes listaelemek közötti térközt szeretnénk módosítani, azt például így tehetjük meg: `\setlength{\itemsep}{15pt}`
- Vagy ha globálisan, az egész dokumentumra szeretnénk beállítani, akkor az `enumitem` csomag használatával tehetjük meg. A preambulumba írva:

```
\setlist[1]{itemsep=-5pt}
```

3. órai feladat

- Egy .tex fájlba írd az alábbihoz hasonló verset a tanult környezetek felhasználásával (nem ezt a pár sort!!!):

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

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

1. Emlősök

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

2. Hüllők

- kígyó,
- gyík

- Illessz be pár LaTeX parancsot, akár több soros kódot is a szövegbe a tanult módon!
- Tegyéél a szövegbe forráskódokat is, legalább két különböző nyelvet használj!

TABULÁLÁS ÉS TÁBLÁZATOK SZERKESZTÉSE

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

```
\begin{tabbing}
```

```
Elment a szóköz \= arrébb\\
```

```
\> ez meg még beljebb \= ment\\
```

```
\> \> végül \\
```

```
Ez lett \=\\
```

```
\> de ez van \> így jártunk!
```

```
\end{tabbing}
```

Táblázatok 1.

- 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 meg jobbra igazított oszlop \\

akkor látszik & ha több elemű

```
\end{tabular}
```

- A környezet argumentuma határozza meg, hogy hány oszlopa legyen a táblázatnak, azokban milyen a szöveg igazítása, valamint hogy van-e köztük rácsvonal:

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

- Az oszlopok négy típusba sorolhatók: `left` (balra igazított), `right` (jobbra igazított), `center` (középre igazított) és `p{szélesség}`: adott szélességű.
- Új sort a `\\` paranccsal lehet kezdeni, míg a soron belüli elemeket az `&` speciális karakter választja el.

- Ha szeretnénk vízszintes vonalakat a táblázatunkba, akkor a `\hline` parancs szükséges.

- Példa (kétszer kettes keretezett táblázatra):

```
\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, akkor a deklaráció rövidíthető a `*` operátorral: a `{*{3}{r|l|c|}}` teljesen ekvivalens az `{r|l|c|r|l|c|r|l|c|}` deklarációval.

- Példa (nyolc vonatos vasúti menetrendre):

```
\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:
- A `\usepackage{array}` szükséges hozzá.

- Példa:

```
\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}
```

- Ha azt szeretnénk, hogy ne csak vertikálisan, hanem horizontálisan is középre rendezettek legyenek az egyes celláink:

```
\begin{tabular}{>{\centering\arraybackslash} m{4cm}  
>{\centering\arraybackslash} m{4cm} }  
... & ... \\  
\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:

```
\begin{tabular}{p{2cm}|p{1cm}}  
\begin{itemize}  
\item 1  
\end{itemize} & \begin{itemize}  
\item 1  
\end{itemize}  
\end{tabular}
```

- Ha csak adott oszlopokban szeretnénk a cellák fölé vonalat húzni, akkor a `\hline` helyett a `\cline{x-y}` utasítás kell, ahol x és y a vonal két vége (oszlopban). Az oszlopok számozása egytől kezdődik.
- „Sor a sorban”: a `\usepackage{multirow}` szükséges hozzá!
- Példa:

```
\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}
```

- Az @ paraméterrel a táblázat oszlopai közötti elválasztó átírható;
- a @ { } nullára csökkenti az oszlopok közötti távolságot,
- a @ { : } kettőspontot tesz két oszlop elemei közé.
- A `\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!

- Példa (menetrend szebben – a kettőspontok egymás alá kerülnek):

```
\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@{ , }l`

- Példa (táblázat számokkal):

```
\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}
```

- Példa (áruházi pénztárgépre):

```
\begin{tabular}{l|r@{,}l@{\,Ft\ \ }}
\multicolumn{1}{c|}{Áru neve} &
\multicolumn{2}{c}{Érték} \\
\hline
Krumpli & 24&40 \\
Kenyér & 106&20 \\
Tej & 110&80 \\
\cline{2-3}
Összesen & 231&40
\end{tabular}
```

Kiadói minőségű táblázatok 1.

- Az eddigi „hagyományos” táblázatokhoz képest a kiadói szintű táblázatok tipográfiája kicsit eltérő:
 - A 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 megoldhatók.
 - 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}
```

- A `\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.

- Példa:

```
\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` környezettel „be kell burkolni” a `tabular` környezetet.
- A `table` környezetnek van egy opcionális paramétere, mellyel befolyásolhatjuk a táblázat helyét az oldalon:
 - `t` (top) – lap tetejére,
 - `b` (bottom) – lap aljára,
 - `h` (here) – pont ide,
 - `p` (page) – külön lapra.
- Az opcionális paraméterben ezek tetszőleges sorrendben felsorolhatóak, mellyel a prioritásukat határozzuk meg.
 - Alapértelmezés: `[tbp]`
- Ha azt szeretnénk, hogy mindenképpen adott helyre tegye, akkor egy felkiáltójellel lehet szigorítani az utasítást:
 - `![h]` jelentése: mindenképpen ide tedd.
- `table*` környezet használatával többhasábos szedés esetén is egyhasábos lesz a táblázat szedése (teljes oldalszélességet kitölti)
- A `table` környezet belsejében a `\caption{szöveg}` paranccsal adhatunk feliratot a táblázathoz. (A `\label{címke}` parancs használatával hivatkozás címkét definiálhatunk.)
- A `\caption` pozíciójától függően a táblázat felett, vagy alatt jelenik meg a felirat.

- Példa (úsztatott táblázat):

```

\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

- Adott szélességű táblázat beillesztése a `tabularx` csomaggal lehetséges.
- 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 paraméter az oszlopdeklarációkat tartalmazza, azonban itt megjelenik egy speciális oszlop: a rugalmas oszlopot az `X` definíció jelöli.
- Példa:

```
\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}
```

4. órai feladat

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

KÉPEK, ÁBRÁK BEILLESZTÉSE, DOBOZOK

Képek beillesztése 1.

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

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

Képek beillesztése 2.

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

Képek beillesztése manipulálással

- 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 mértéke
 - angle – forgatás szöge fokban
 - origin – forgatás középpontja
 - width – a kép kívánt szélessége
 - height – a kép kívánt magassága
 - (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.
 - `figure*` környezet használatával többhasábos szedés esetén is egyhasábos lesz a kép (teljes oldalszélesség).
- Ez pont ugyanaz, mint a `tabular` és a `table` egymásba ágyazása, ezért mindenképp használjuk!
- Ugyanúgy léteznek opcionális paraméterei: `[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 1.

- Ha az ábrának nem megfelelő a mérete, vagy nem megfelelő szögben áll, a `\resizebox{vízszintes}{függőleges}{mit}` és `\rotatebox{fok}{mit}` parancsokkal módosíthatunk rajtuk.
 - Az előbbinél a két paraméter egyikénél használható a felkiáltójel, melynek jelentése: az arányokat tartsd.

- 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 tükrözése:

```
\begin{figure} [h!]  
\centering  
\reflectbox{%  
\includegraphics[width=0.5\textwidth]{ide jön a  
kép}}  
\caption{A kép címét itt adjuk meg}  
\end{figure}
```

Ábra manipulálás figure környezetben 3.

- 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)

Ábra mellé magyarázó szöveg 1.

- A `\usepackage{sidecap}` és a `\begin{SCfigure} ... \end{SCfigure}` környezet használatával valósítható meg.
- Néhány lehetséges opció:
 - `leftcaption`, `rightcaption` – bal vagy jobb oldalon jelenjen meg a szöveg.
 - `raggedright`: ha kevés a magyarázó szöveg, így szebben néz ki balra zárva.

Ábra mellé magyarázó szöveg 2.

■ 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.

Ábrák szövegbe ágyazása 1.

- `\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. Az afrikai elefánt

subfigure környezet

- Ha több ábrát szeretnénk megjeleníteni egymás mellett/alatt, a subfigure környezetre van szükségünk, de ehhez kellenek még a caption és a subcaption csomagok is.

- 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 négyféle dobozt használ:
 - **LR doboz** (balról jobbra töltjük fel tartalommal, tipikusan csak egy sor),
 - **Bekezdésdoboz** (valamilyen szélességgel bíró, akár több sort is tartalmazó doboz),
 - **Vonaldoboz** (gyámfázásra használjuk),
 - **Tárolódoboz** (szövegrészek tárolására szolgál).

LR dobozok 1.

- `\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 szövegtől való távolságát az `\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.
 - A *poz* paraméter értékei: l – balra zárt, r – jobbra zárt, s – interword space
- A doboz szélessége megadható tetszőleges hosszúság mértékben, vagy használhatóak 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.
 - **Példa:**

```
{\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 `\Ovalbox{szöveg}`
 - **lekerekítés sugarának változtatása:** `\cornersize{r}`
 - **Duplakeretes doboz:** `\doublebox{szöveg}`

- Több bekezdésnyi szöveget helyezhetünk vele egy dobozba:

`\parbox[poz][mag][szpoz]{szélesség}{szöveg}`

- `[poz]` – t (top) vagy b (bottom), különben középre igazítja a dobozt a szöveg vonalában.
- `[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
```

```
\fbox{\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 nélkül:
\fbox{szöveg}}`
- A doboz tartalmának kiíratása, ahol szükség van rá:
`\usebox{\prncs}`
- Vigyázat, a `\usebox`-ot új bekezdésben használjuk, 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: `\phantom{...}`
- 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}`

- Megfelelően formázott dokumentum elkészítése!
- Gyakorold az ábrák különféle módon történő beillesztését!
 - A szöveghez valamennyire kapcsolódó ábrák legyenek.
- Ne feledkezz meg az ábrák úsztatásáról sem!
 - Képaláírás elhelyezése(, valamint hivatkozások beszúrása).
- Gyakorold a dobozok használatát!

HIVATKOZÁSOK, TARTALOMJEGYZÉK, ÁBRA-, TÁBLÁZAT- ÉS IRODALOMJEGYZÉK KÉSZÍTÉSE

Tartalomjegyzék készítése

- A LaTeX egyik nagy előnye, kétségtelenül!
- Emlékszünk még a szakaszok, 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 használt számlálókról később lesz szó!)*
- Ha olyan tartalomjegyzékre van szükségünk, ami részletesebb (pl. 4 mélységű `\paragraph` részt is szeretnénk feltüntetni), akkor:
 - `\setcounter{tocdepth}{4}`

Tartalomjegyzék szerkesztése 1.

- 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, és szeretnénk megváltoztatni azt, írjuk a preambulumba a következőt:

```
\addto\captionsmagyar{\renewcommand{  
\contentsname}{Tartalom}}
```

```
\addto\captionsenglish{\renewcommand  
{\contentsname}{Table of contents}}
```

Babel-t használunk, így a magyar és az angol esetén más-más parancs szükséges!

Tartalomjegyzék szerkesztése 2.

- Könyvek és fontosabb művek esetén a tartalomjegyzék oldalszámozása római számmal történik, míg 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` – ugyanaz, csak ábrajegyzék
- Ha szeretnénk módosítani az adott jegyzék nevét, írjuk a preambulumba a következőt:

```
\addto\captionsmagyar{\renewcommand{
\listfigurename}{Ábrajegyzék}}
\addto\captionsmagyar{\renewcommand{
\listtablename}{Táblázatjegyzék}}
```

- Ha függelékét szeretnénk, írjuk 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 függelék készítésére

```
\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 függelék}
\end{document}
```

Tárgymutató készítése

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

```
\usepackage{imakeidx}
```

```
\makeindex[name=terms, columns=1]
```
- A tárgymutatót az `\index[terms]{tárgyszó}` paranccsal bővíthetjük.
- Ahová el szeretnénk helyezni a tárgymutatót, írjuk be a `\printindex[terms]` parancsot.
- Ha szeretnénk lecserélni a „Tárgymutató” szót „Szójegyzék”-re, a `printindex` elé írjuk be a következőt:

```
\renewcommand{\indexname}{Szójegyzék}
```

vagy:

```
\makeindex[name=terms, title=Szójegyzék, columns=1]
```

Irodalomjegyzék készítése 1.

- Az irodalomjegyzéket általában a dokumentum végére szokás 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`

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

A műben ezzel a címkével
hivatkozunk erre a műre

Irodalomjegyzék készítése 2.

- A `thebibliography` környezetnek van egy argumentuma, amely segíti a LaTeX-et az irodalomjegyzék tördelésében:
 - ha tíznél kevesebb művet sorolunk fel, akkor egy 9-et,
 - ha száznál kevesebbet, de tíznél többet, akkor 99-et stb. kell ide írni.
- Az irodalomjegyzék automatikusan generálódik.
- Alapbeállításként a sorszámok növekvő arab számok, szögletes zárójelek között.
- A `\bibitem[Jópofa]{cimke}` hatására azonban a tartalomjegyzék vonatkozó elemének Jópofa lesz a „száma”, és ha hivatkozunk rá a szövegben, akkor is a [Jópofa] jelenik meg.
- Név módosítás, ha szükséges, de a Babel magyar tudja:
`\renewcommand{\bibname}{Felhasznált irodalom}`

Az IEEE hivatkozás formátuma

- A szabályosan szedett irodalmi hivatkozások alakja a következő:
- Könyv: szerző, *cím dőlt betűkkel*, kiadó neve, kiadó városa, kiadás éve.
- Újságcikk: szerző, „cím idézőjelek között”, *újság neve dőlt betűkkel*, évfolyam, szám, oldalak, kiadás éve.
- Konferencia cikk: szerző, „cím idézőjelek között”, *konferencia neve dőlt betűkkel*, konferencia kiadványban oldalak, konferencia városa, éve.
- Diplomadolgozat/disszertáció: szerző, *cím dőlt betűkkel*, oktatási intézmény, város, végzés éve.
- Elektronikus forrás: teljes URL (átviteli protokoll megjelölésével), megtekintés időpontja percre pontosan.

Irodalomjegyzék példa 1.

...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.karinthy.hu/~{}NIKITA/szedes/Bevezet.htm>,
2002.~január~9., 19:14

```
\end{thebibliography}
```

Irodalomjegyzék példa 2.

- A `\cite` utasításnak létezik opcionális paramétere is, amellyel például az oldalszámot (vagy egyéb kiegészítő információt) adhatjuk meg.
- A kimeneten az irodalom címkéje mellett vesszővel jelenik meg a szögletes zárójelen belül [1, 160–165. oldal].

- Példa:

```

...zárójelen belül~\cite[160--165.\ oldal]{konyv}.
...lásd \cite{Bujd} és \cite[134. oldal]{Feri}.
...lásd \cite{Bujd,Feri}.
...lásd \cite[függelék]{Bujd,Feri}.
...
\begin{thebibliography}{9}
\bibitem{konyv}
Virágvolgyi Péter, {\it A tipográfia mestersége}, Osiris, Budapest,
2001.
\bibitem[Bujdosó1997]{Bujd}
Bujdosó Gyöngyi, Fazekas Attila, {\it \TeX\ kezdőlépések}, Tertia
Kiadó, Budapest, 1997.
\bibitem[Wettl2004]{Feri}
Wettl Ferenc, Mayer Gyula, Szabó Péter, {\it \LaTeX\ kézikönyv},
Panem Könyvkiadó, Budapest, 2004.
...
\end{thebibliography}

```

```
@article{greenwade93,  
  author   = "George D. Greenwade",  
  title    = "The {C}omprehensive {T}ex {A}rchive {N}etwork  
({CTAN})",  
  year     = "1993",  
  journal  = "TUGBoat",  
  volume   = "14",  
  number   = "3",  
  pages    = "342--351"  
}  
  
@book{goossens93,  
  author    = "Michel Goossens and Frank Mittlebach and  
Alexander Samarin",  
  title     = "The Latex Companion",  
  year      = "1993",  
  publisher = "Addison-Wesley",  
  address   = "Reading, Massachusetts"  
}
```

- Hivatkozás irodalomra ugyanúgy, mint eddig:

- ... `\cite{greenwade93}`
- ... `\cite[Sec. 2]{goossens93}`

- Irodalomjegyzék beszúrása:

- `\bibliography{pelda}`
- `\bibliographystyle{plain}`

- Latex fordítás:

- `pdflatex forras[.tex]`
- `bibtex forras[.aux] (-> .bbl)`
- `pdflatex forras[.tex]`
- `pdflatex forras[.tex]`

Hivatkozások készítése 1.

- Nagyon fontos! A LaTeX egyik alapköve!
- Minden számozott elemre lehet hivatkozni: irodalomra, fejezetre, ábrára, táblázatra, oldalra, lábjegyzetre, 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}` utasítással hívjuk elő a számot.
 - Ha pl. 3. szintű elemre szeretnénk hivatkozni, növelni kell a hivatkozási mélységet a `\setcounter{secnumdepth}{3}` paranccsal.

- Példa:

```
\section{Bélabácsí}
```

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

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

```
...
```

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

Hivatkozások készítése 2.

- Hivatkozás oldalra: adott oldalra a `\pageref{cimke}` utasítással hivatkozhatunk.

Bővebben `a~\pageref{sec:bela}.~oldalon...`

- Hivatkozás ábrára:

```
\begin{figure}
```

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

```
\includegraphics{ábra neve}
```

```
\caption{Ide az ábra címe}
```

```
\label{fig:cimke}
```

```
\end{figure}
```

Az eredmény `a~\ref{fig:cimke}.~ábrán` látható.

Határozott névelő a hivatkozás előtt

- Az `\az{valami}` és `\Az{valami}` utasításokkal a `valami` elé határozott névelőt tehetünk, a különbség: az első kisbetűvel írja a névelőt, a második nagybetűvel kezdi (mondat elején jó).
- Az `\aref{}` és `\Aref{}` utasításokkal névelővel hivatkozhatunk objektumokra (tehát gyakorlatilag `\aref{cimke} = \az{\ref{cimke}}`).
- Az `\acite{}`, `\Acite{}`, `\apageref{}` és `\Apageref{}` utasítások hasonlóan az előzőek kiegészítései.
- A korábbi példák tehát helyesen:
Később (vagy akár hamarabb is) elkezdek hivatkozni `\aref{sec:bela}~fejezetre`.
Bővebben a Bélabácsi fejezetben, `\apageref{sec:bela}~oldalon...`

6. órai feladat

- Hozz létre egy legalább 5 oldalból álló dokumentumot, mely tartalmaz minimum 3 fő fejezetet, valamint ezek is tartalmazzanak 2-2 alfejezetet, de tartalmazhat még ennél mélyebb al-alfejezeteket is! A szövegben hivatkozz ezekre a tanult módokon!
- A megformázott dokumentumba tegyél be legalább 3 ábrát, képaláírással, megfelelően formázva azt! Valamint hivatkozz ezekre a szövegben!
- Illessz be legalább 3 táblázatot, megfelelően formázva, továbbá hivatkozz rájuk a szövegben!
- A tanult módon készíts irodalomjegyzéket, mely bemutatja a szöveghez kapcsolódó irodalmat (az órai példáknak megfelelő típusok, nem hiányozhatnak a lényeges elemek) és a szövegben hivatkozz ezekre a tanult parancsokkal!
 - A hivatkozott művek (minimum 5) között legyen folyóirat, konferencia cikk és egy könyv is! A hivatkozott művek valódi irodalmak legyenek!
- Készíts ábra- és táblázatjegyzéket is a tanult módon! Ne feledkezz meg a kinézetéről sem!

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

Matematikai mód 1.

- A LaTeX egyik legnagyobb előnye más szövegszerkesztőkhöz képest a szép matematikai szedés.
- Három matematikai módot különböztetünk meg egymástól:
 - soron belüli matematikai mód
 - `$` dollárjelek `$`, vagy `\(` ilyenek `\)` közé kerül a matematikai kifejezés, vagy a `math` környezetbe
 - külön sorba kerülő, de nem számozott egyenlet
 - `displaymath` környezetbe, vagy `\[` ilyenek `\]`, esetleg `$$` ilyenek `$$` közé írjuk a matematikai kifejezést
 - számozott, külön sorba írt egyenlet
 - `equation` környezet
 - Az utóbbi címkézhető a `\label{cimke}` utasítással és hivatkozható.
- Fontos, hogy ha a szövegben matematikai képleteket, változókat írunk, soron belül is váltsunk matematikai módba, még ha csak egyetlen betűről is van szó, mert így könnyebb megérteni a szöveget:
Ha `$a$` kisebb, mint `$b$`, akkor ...

- Nézzünk néhány példát:

- Ha $a^2 + b^2 = c^2$, akkor ez a Pitagorasz-tétel.
- A Pitagorasz-tétel számozatlanul:

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

- A Pitagorasz-tétel számozottan:

```
\begin{equation}  
  a^2 + b^2 = c^2  
\end{equation}
```

- További példák:

Mindegy, hogy $a^2 + b^2 = c^2$, vagy

$(a^2 + b^2 = c^2)$, vagy

```
\begin{math}
```

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

```
\end{math},
```

ugyanazt láthatjuk a kimeneten.

Viszont a

```
\[ x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \]
```

 és a

```
\begin{displaymath}
```

```
  x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}
```

```
\end{displaymath}
```

is ugyanaz.

- Matematikai módban a szóközök száma nem számít:
 - $\$ab_ = a_ _ _ b\$$, ami így néz ki: $ab = ab$.
- Üres sort nem szabad írni, vagyis matematikai módban nincs új bekezdés ($\backslash \backslash$).
- Magyarban tizedes vessző, míg angolban tizedes pont van. Azonban matematikai módban a vessző felsorolást jelöl, melynek következtében túl nagy térköz marad a tizedes jegyek és az egész szám között: $\$3,14\$$ így néz ki: 3, 14.
- Ennek ellensúlyozása az ún. negatív térközzel ($\backslash !$) történik, mellyel: $\$3, \backslash ! 14\$ = 3,14$.
- A kapcsos zárójelekkel blokkosíthatunk ($\{...\}$), amire gyakran van szükség ($\$x^y + z \backslash neq x^{\{y + z\}}\$$).

Térközök matematikai képletekben

- Gyakran kell „odébb tenni” a magyarázó szöveget, illetve a feleslegesen összefolyó elemek „szétválasztásakor” is szükség lehet rájuk.
- A már ismert `\,` `( )` mellett használható a `\: ( )` és a `\; ( )`. A szintén ismert `\_` utasítás szóközt generál, a `\quad ( )` és `\qquad ( )` nagyobb térközök beszúrására alkalmas. A `\quad` az aktuális betűtípus „M” betűjének szélessége.
- Negatív térköz beillesztésére a korábban ismertetett `\!` utasítással van lehetőségünk.
- A `\phantom{valami}` utasítással a `valami` nem jelenik meg, hanem csak a helyét hagyja ki a LaTeX. Tipikus alkalmazása az alábbi:

$$x^{12} _ {6} C$$

Alapműveleti jelek, relációjelek

- Az összeadás, a kivonás és az osztás műveletek matek módban is a már megszokott $+$, $-$, $/$ jelekkel írhatók le. Azonban fontos megjegyezni, hogy ezek matek módban másképp mutatnak:
 - $(2+3)/5=3-2$ hagyományosan és $(2 + 3)/5 = 3 - 2$ matek módban
- A szorzásjelet nem $*$ -gal jelöljük! Mérnöki gyakorlatban a csillag jelentése: konvolúció. Helyette vagy nem írunk semmit, vagy középmagas pontot „ $\cdot$ ” (`\cdot`), vagy $\times$ -et (`\times`) teszünk.
- Az $=$, $<$, $>$ relációjelek ugyanúgy használhatóak matematikai módban is, ám ha kisebb egyenlő, nagyobb egyenlő, vagy nem egyenlő jelekre van szükség, akkor a `\le`, `\ge`, `\ne` parancsok használata szükséges.

- Alsó index az $_{\{valami\}}$, felső index a $^{\{valami\}}$ utasításokkal adható meg.
 - Vigyázat! Egy matematikai elemnek csak egy alsó és egy felső indexe lehet.
 - Figyelem! A „ $^$ ” operátor csak az utána következő első elemet emeli felső indexbe, vagyis az „ a^{24} ” az a^{24} kimenetet eredményezi. Ezért, ha az a^{24} kimenetet szeretnénk megjeleníteni, blokkosításra van szükség: $a^{\{24\}}$. Ugyanez igaz az alsó indexre is „ $_$ ”!
 - Indexek egymásba ágyazása is blokkosítással valósítható meg (lásd $\pi^{\pi^{\pi}}$).

Szöveg bevitel, képlet bekeretezése

- Ha az egyenlethez magyarázó szöveget szeretnénk írni, vissza kell térnünk a normál módba (ne legyenek dőlttel szedve a betűk).
- Szöveg bevitele a `\textnormal{szöveg}`, vagy a `\textrm{szöveg}` paranccsal lehetséges:
$$\left[\begin{array}{l} x \geq 0, \textrm{ ha } x \text{ nem negatív} \end{array} \right]$$
- Képletek bekeretezéséhez a `\boxed{ }` parancsra van szükségünk (`\usepackage{amsmath}` kell hozzá).

- Az `equation` környezet számozott, tehát címkézhető; figyeljük meg a hivatkozás formáját – zárójelben szerepel, tehát `(\ref{eq:valami})` jellegű –, de az `\aref({eq:valami})` utasításnál az argumentumot kell bezárni kerek zárójellel!

...mindenki tudja, hogy a π közelíthető `\aref({eq:pi})` kifejezéssel. Mégis megadjuk:

```
\begin{equation}
\label{eq:pi}
\pi = \lim_{n \rightarrow \infty} \sqrt{\sum_{k=1}^n \frac{6}{k^2}}.
\end{equation}
```

- Kiemelt képletek esetében az írásjelek (pont, vessző) mindig a környezeten belülre, a formula végére kerülnek.

Számozás megváltoztatása 1.

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

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

```
\begin{document}
```

```
\section{Első szakaszom}
```

```
\subsection{Itt meg az alszakasz}
```

```
\begin{equation}
```

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

```
\end{equation}
```

```
\end{document}
```

Számozás megváltoztatása 2.

- Ha nem kellene a pontok és elég a subsection száma, valamint az egyenlet sorszáma:

```
\numberwithin{equation}{subsection}
```

```
\renewcommand{\theequation}{\thesection\arabic{equation}}
```

- Ha pedig római számokat szeretnénk például használni:

```
\renewcommand{\theequation}{\thesection\roman{equation}}
```

```
\renewcommand{\theequation}{\roman{equation}}
```

Műveleti és relációs jelek

$+$	<code>+</code>	$\wedge$	<code>\wedge</code>	$=$	<code>=</code>	$\geqslant$	<code>\geqslant</code>
$-$	<code>-</code>	$\vee$	<code>\vee</code>	$:=$	<code>:=</code>	$\ll$	<code>\ll</code>
$/$	<code>/</code>	$\star$	<code>\star</code>	$\dot{=}$	<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>
$\backslash$	<code>\setminus</code>	$\odot$	<code>\odot</code>	$<$	<code><</code>	$\supseteq$	<code>\supseteq</code>
$\cap$	<code>\cap</code>	$\oslash$	<code>\oslash</code>	$\leq$	<code>\leq</code>	$:$	<code>:</code> (arány)
$\cup$	<code>\cup</code>	$\otimes$	<code>\otimes</code>	$\geq$	<code>\geq</code>	$ $	<code>\mid</code>
				$\leqslant$	<code>\leqslant</code>	$\parallel$	<code>\parallel</code>
						$\perp$	<code>\perp</code>
$\%$	<code>\%</code>	$\triangle$	<code>\triangle</code>				
$\bot$	<code>\bot</code>	$\square$	<code>\square</code>				
$\top$	<code>\top</code>	$\blacksquare$	<code>\blacksquare</code>				
$\neg$	<code>\neg</code>	$\angle$	<code>\angle</code>				
$\forall$	<code>\forall</code>	$\sphericalangle$	<code>\sphericalangle</code>				
$\exists$	<code>\exists</code>	$\flat$	<code>\flat</code>				
$\nexists$	<code>\nexists</code>	$\sharp$	<code>\sharp</code>				
$\Re$	<code>\Re</code>	$\natural$	<code>\natural</code>				
$\Im$	<code>\Im</code>	$\#$	<code>\#</code>				
∇	<code>\nabla</code>	1°	<code>1^\circ</code>				
∂	<code>\partial</code>	$1'$	<code>1'\prime</code>				
$\eth$	<code>\eth</code>	$1''$	<code>1''\prime\prime</code>				
$\emptyset$	<code>\emptyset</code>						
∞	<code>\infty</code>						

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>

■ Példa:

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

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

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

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

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

- Beépített parancsokkal hívhatók 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>								

- Az ékezetes betűk is beépített parancsokkal hívhatók elő:

$\hat{a}$ `\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}`

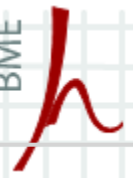
$\dot{a}$ `\dot{a}`

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

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

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

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



Betűtípusok matematikai módban

`\mathit{...}`

`\mathrm{...}`

`\mathbf{...}`

`\mathsf{...}`

`\mathtt{...}`

`\mathnormal{...}`

`\mathcal{...}`

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

`\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
⇐	\Longleftarrow	↑	\Uparrow
⇒	\Rightarrow	↓	\Downarrow
⇒	\Longrightarrow	↕	\Updownarrow
⇌	\Leftrightarrow	↗	\nearrow
⇌	\Longleftrightarrow	↘	\searrow
↦	\mapsto	↙	\swarrow
↦	\longmapsto	↘	\nwarrow

Zárójelek méret igazítása

- Ha egy képlet írásakor több, egymásba ágyazott zárójelet használunk, jó lenne megkülönböztetni őket, akár méretben is, hogy olvashatóbb maradjon a szövegünk:

$$2 \cdot (a^{(a+b)} - (c_i)) \cdot (q^2) - \text{ezt így nehéz,}$$
 ehelyette:
- Ha méretre szeretnénk igazítani a zárójeleket, akkor célszerű a `\left( \right)` kerek zárójeleket, esetleg a `\left[ \right]` szögletes, vagy a `\left\{ \right\}` kapcsos zárójeleket használni.
 - Jelentése: a `\left` és a `\right` közötti tartalom magasságához igazítsd a zárójelek méretét.
- Illetve manuálisan is beállíthatjuk a különféle zárójelek méreteit a `\big(`, `\Big(`, `\bigg(`, `\Bigg(` parancsok segítségével.

- $\frac{\text{ide jön a számláló}}{\text{ide meg a nevező}}$,
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.
- Ha zavar minket a zárójel, akkor lehet $\{n \text{ \atop k}\}$ is.

- Gyökjel az `\sqrt{valami}` parancs segítségével írható.
- Akár több tört is egymásba ágyazható:

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

- Ha a gyökjel kitevőjét meg akarjuk változtatni, akkor azt az opcionális paraméterrel tehetjük meg:

```
\sqrt[3]{2} \approx 1, \sqrt[26]{}
```

Norma és abszolút érték jelek

- Normaérték jele $\|$, míg az abszolút értéké $|$.
- Ne keverjük össze őket!

$$\sqrt{(a - b)^2} = |a - b|$$

- Ha két vektor (sík) párhuzamosságát akarjuk jelölni, akkor a `\parallel` parancsra van szükség: $e \parallel f$.

- Egy karakter alá és fölé írni a következő parancsokkal lehetséges:

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

- Példa:

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

- Kommentek készítése:

- `\overbrace{}` és `\underbrace{}` – kapcsos zárójel
- `\overbracket{}`, `\underbracket{}` – szögletes zárójel (mathtools csomag szükséges hozzá)

- Példa:

```
\underbrace{x}_{\text{valós rész}} +  
\underbrace{iy}_{\text{képzetes rész}}
```

Példa alá és fölé írásra

- Egy karakter alá és fölé az `\overset{}` és az `\underset{}` paranccsal tudunk írni.
- pl. az egyenlőség jelre:

```
\overset{\left[\frac{0}{0}\right]}{\underset{\mathrm{H}}{=}}
```

- Megjegyzés írására hasznosak lehetnek az `\overbrace{}` és `\underbrace{}` parancsok:

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

- A `\usepackage{amsmath}` csomag szükséges hozzá.

$\begin{matrix} a & b \\ c & d \end{matrix}$	<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

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

```
\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 & \ldots & 0 \\  
0 & 1 & \ldots & 0 \\  
\vdots & \vdots & \ddots & \vdots \\  
0 & 0 & \ldots & 1  
\end{pmatrix}
```

Beépített függvények 1.

arccos `\arccos`deg `\deg`sin `\sin`arcsin `\arcsin`dim `\dim`sinh `\sinh`arctan `\arctan`exp `\exp`tan `\tan`arg `\arg`hom `\hom`tanh `\tanh`cos `\cos`ker `\ker` $\lim$ `\varliminf`cosh `\cosh`lg `\lg` $\overline{\lim}$ `\varlimsup`cot `\cot`ln `\ln` $\lim_{\rightarrow}$ `\varinjlim`coth `\coth`log `\log` $\lim_{\leftarrow}$ `\varprojlim`csc `\csc`sec `\sec`

Beépített függvények 2.

det `\det`

gcd `\gcd`

inf `\inf`

inj lim `\injlim`

lim `\lim`

lim inf `\liminf`

lim sup `\limsup`

max `\max`

min `\min`

proj lim `\projlim`

Pr `\Pr`

sup `\sup`

Képletek tördelése 1.

- Ha túl hosszú az egyenletünk és nem fér ki egy sorba, el kell törni.
- Ezt a LaTeX sajnos nem végzi el helyettünk, mert nem tudja, hogy hol törheti az egyenletet.
- Azonban matematikai módban nincs új sor (`\\`), más megoldás szükséges (valami hasonló a mátrixok ábrázolásához).

- `multline` környezet:

```
\begin{multline}\label{cimke}
```

```
1+8+27+64=\\
```

```
=1+3+5+7+{\ }\\
```

```
+9+11+13+{\ }\\
```

```
+15+17+19
```

```
\end{multline}
```

Képletek tördelése 2a.

- Az `eqnarray` környezet segítségével igazított egyenleteket tudunk létrehozni.
- Gyakorlatilag egy `{rcl}` típusú tömb, melyben az oszlopokat `&` jellel, a sorokat pedig `\\` jellel választjuk el.
- Csillagos változata számozatlan egyenleteket szűr be, csillag nélküli változatában a `\nonumber` paranccsal lehet kikapcsolni a számozást egy adott sorban.

▪ Példa:

```
\begin{eqnarray}
1^3 + 2^3 & = & (1 + 2)^2 \nonumber \\
1^3 + 2^3 + 3^3 & = & (1 + 2 + 3)^2 \\
& & \vdots \nonumber \\
\sum_{i=1}^n i^3 & = & \left( \sum_{i=1}^n i \right)^2
\end{eqnarray}
```

Képletek tördelése 2b.

- Hasznos lehet a `\lefteqn{}` parancs, mellyel az argumentum hosszát virtuálisan `0pt`-re állíthatjuk.
- Ha az egyenlőségjel két oldalán lévő távolság nem tetszik, akkor használjuk a `\setlength\arraycolsep{2pt}` parancsot.

■ Példa:

```
\setlength\arraycolsep{2pt}
\begin{eqnarray*}
\lefteqn{f(x) = f(x)\Bigg|_{x=x_0} + (x - x_0)
f'(x)\Bigg|_{x=x_0} \setminus
& & {} + \{\frac{(x - x_0)^2}{2}\} f''(x)\Bigg|_{x=x_0} + \ldots +
\{\frac{(x - x_0)^n}{n!}\} f^{(n)}(x)\Bigg|_{x=x_0} + \ldots,
\end{eqnarray*}
vagy akár
\begin{eqnarray}
f(x) & = & f(x)\Bigg|_{x=x_0} + (x - x_0) f'(x)\Bigg|_{x=x_0}
\nonumber + \{\frac{(x - x_0)^2}{2}\} f''(x)\Bigg|_{x=x_0} \setminus
& & {} + \ldots + \{\frac{(x - x_0)^n}{n!}\}
f^{(n)}(x)\Bigg|_{x=x_0} + \ldots
\end{eqnarray}
```

Képletek tördelése 3a.

- `amsmath` csomagban még számtalan egyenlet tördelő megoldás létezik, ilyen például a `split` környezet:

- Példa 1:

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

- Példa 2:

```
\[
\begin{split}
\cos^2\alpha \quad &= 1-\sin^2\alpha=\\
&= 1+3+5+7+9+{\quad}\quad\\
&\quad + 11+13+15+17+19
\end{split}
\]
```

- Példa 3:

```
\begin{eqnarray*}
```

```
A&=&B, \\\
```

```
C&=&D, \\\
```

```
E&=&F
```

```
\end{eqnarray*}
```

```
\begin{align*}
```

```
A&=B, \\\
```

```
C&=D, \\\
```

```
E&=F
```

```
\end{align*}
```

Képletek tördelése 4.

- Új oldal: `\displaybreak\\` – csak `align` esetén.
- Több egyenlet esetén létrehozhatunk al-számozást is:

```
\begin{subequations}
```

Itt jönnek a felsorolt egyenletek:

```
\begin{align}
```

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

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

```
\displaybreak \nonumber \\
```

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

```
\end{align}
```

```
\end{subequations}
```

Képletek tördelése 5.

- Néha előfordul, hogy feltételekhez kötött az egyenletünk, ebben az esetben a `cases` környezetet alkalmazhatjuk:

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

Tételek, segédtételek, bizonyítások 1.

- Preambulumba helyezzük el a következők valamelyikét:
 - `\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.

Tételek, segéd tételek, bizonyítások 2.

- A dokumentum környezetben, ha létre szeretnénk hozni például egy *tétel* környezetet, ezt a következő módon tehetjük:

```
\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
```

Tételek, segéd tételek, bizonyítások 3.

- A matematikai képleteket olykor bizonyítani is kell!
- Az `amsthm` csomagban definiálva van egy környezet bizonyításokra, ez a `proof` környezet.

```
\begin{proof}
```

A bizonyítás szövege.

```
\end{proof}
```

...

```
\begin{tetel}\label{xy}
```

A tétel szövege.

```
\end{tetel}
```

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

A bizonyítás szövege.

```
\end{proof}
```

- Ha át akarjuk nevezni, például „Bizonyítás”-ra:

```
\renewcommand{\proofname}{Bizonyítás}
```

- Q.E.D. jel átdefiniálása:

```
\renewcommand{\qedsymbol}{\rule{1ex}{1ex}}
```

- Q.E.D. jel megszüntetése:

```
\renewcommand{\qedsymbol}{}{ }
```

7. órai feladat

- Hozz létre egy legalább 3 oldalból álló, lehetőség szerint műszaki vagy matematikai tartalmú dokumentumot, majd az alábbiaknak megfelelően készíts el tetszőleges képleteket, mátrixokat!
 - A képletek kapcsolódjanak a szöveghez!
 - A képletek mindhárom típust (szöveggel egy sorban, külön sorba írt számozatlan és számozott) fedjék le!
 - Legalább 5 összetettebb képlet a következő forrásokból:
 - vagy a Matek.pdf-ből a hozzá tartozó szöveggel együtt megformázva,
 - vagy más forrásból, de összetettség tekintetében a Matek.pdf-ben szereplő képletekhez hasonló.

- Gyakorold a tanult környezeteket, egysoros, többsoros egyenletek stb., minden típusra legalább 2-2 példát mutass!

- Mondj ki egy eléggé összetett tételt:

- Pl. Gauss–Osztrogradszkij-tétel:

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

$$\iint (A_x dydz + A_y dxdz + A_z dxdy) = \iiint \left(\frac{\partial A_x}{\partial x} + \frac{\partial A_y}{\partial y} + \frac{\partial A_z}{\partial z} \right) dxdydz.$$

- Illetve mutasd be a bizonyítását is (vagy annak egy részletét) a megfelelő környezet alkalmazásával!

- 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 öt (értelmes) táblázat,
 - van benne legalább öt kép/ábra (akár `subfigure` is),
 - van benne legalább öt képlet (de nem $a^2 + b^2 = c^2$ típusú!!!),
 - tartalmaz felsorolásokat,
 - tartalmaz irodalomjegyzéket, illetve azokra hivatkozást.
 - Továbbá hivatkozni kell a folyó szövegben az ábrákra, a táblázatokra és a képletekre is a tanultak szerint! Valamint jegyzékeket is készíts hozzájuk!
- A házi feladatban a bruttó (utasításokkal együtt számolt) karakterszám 25000–30000 legyen.
- Beadási határidő a 13. oktatási hét szerda (2019. május 8.) 23:55
 - A moodle.hit.bme.hu oldalon a NagyHF oldalon kell beadni az elkészített házi feladatot a fordításhoz szükséges minden egyes fájlal együtt!

Nagy HF-ben elkövethető súlyos hibák

- 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
- nem úsztatott táblák (`table` környezet elhagyása)
- tanult parancsok/paraméterek nem ismerete (pl. `eqnarray` környezet minden sorában `\nonumber` parancs)
- felesleges `\\` újsor utasítások

OLDALSZERKESZTÉS, MÉRET, ELFORGATÁS, HASÁBOK

- Ha nagy művet írunk, áttekinthetőbb, ha külön fájlokban tároljuk az egyes fejezeteket.
- A dokumentumok felszabdálását az `\include{}` parancs támogatja, melynek argumentumában a fájl nevét kell megadni, akár a `.tex` kiterjesztés nélkül.
- Az `\includeonly{}` paranccsal megmondhatjuk a LaTeX-nek, hogy mely fájlokat akarjuk valóban betölteni (akkor hasznos, ha egy fájlban dolgozunk és a többi fordítására nem akarunk időt pazarolni).
- Az `\include{}` parancs hatására új oldal kezdődik (mivel alapvetően chapter-ek beillesztésére tervezték). Ha ez nem tetszik, akkor az `\input{}` felhasználásával oldaltörés nélkül is beilleszthetünk fájlokat.
- Példa:

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

- Több fájlt is lehet egyetlen LaTeX forrásba fűzni. A `\documentclass` utasítás előtt a `filecontents` környezet jelöli a csatolt fájlt.
- Egyetlen argumentuma a fájl nevét tartalmazza, a környezetbe pedig a fájl tartalmát kell írni. Akár több fájlt is csatolhatunk a módszerrel.
- Példa:

```
\begin{filecontents}{maki.eps}
%!PS-Adobe-2.0 EPSF-1.2
%%Title: Maki majom ...
\end{filecontents}
\begin{filecontents}{magyar.ldf}
\ProvidesFile{magyar.ldf}
[1996/12/23 v1.3h Magyar support from ...] ...
\end{filecontents}
\documentclass[a4paper,10pt]{report} ...
```

- Előfordulhat, hogy többhasábos dokumentumot szeretnénk létrehozni.
- Ha két hasábot szeretnénk, akkor a dokumentumosztály definíciójánál a `twocolumn` ad erre lehetőséget.
(`\documentclass[11pt,twocolumn]{article}`)
 - Váltás `\twocolumn` és `\onecolumn` szedés között (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, ami a környezet argumentumában megadott számú hasábot alkotja
`\end{multicols}`
 - A hasábok közötti távolság 10 pt alapesetben, de ez állítható pl. 1 cm-re:
`\setlength{\columnsep}{1cm}`

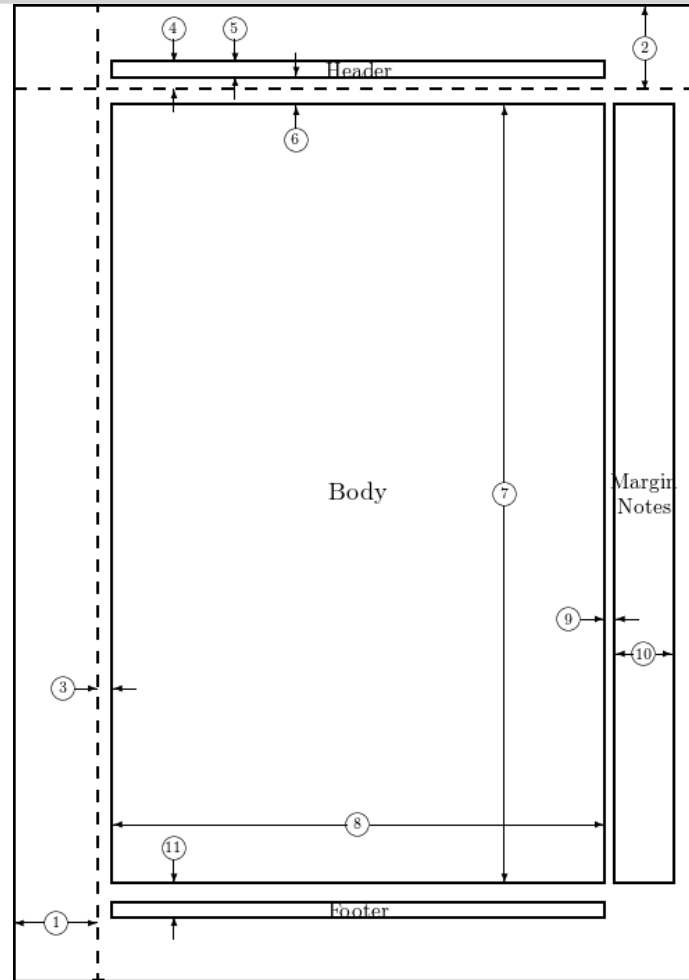
- Az oldal méret megadásoknál a LaTeX a nyomdászatban alkalmazott mértékegységeket használja:
 - Fix mértékegységek: (egybeírva!)
 - mm (milliméter)
 - cm (centiméter)
 - in (inch, 1 in = 25,4 mm)
 - pt (pont, 1 pt = 0,3515 mm)
 - Relatív mértékegységek:
 - ex (aktuális betűtípusban az „x” magassága)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

Mértékegységek táblázata

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



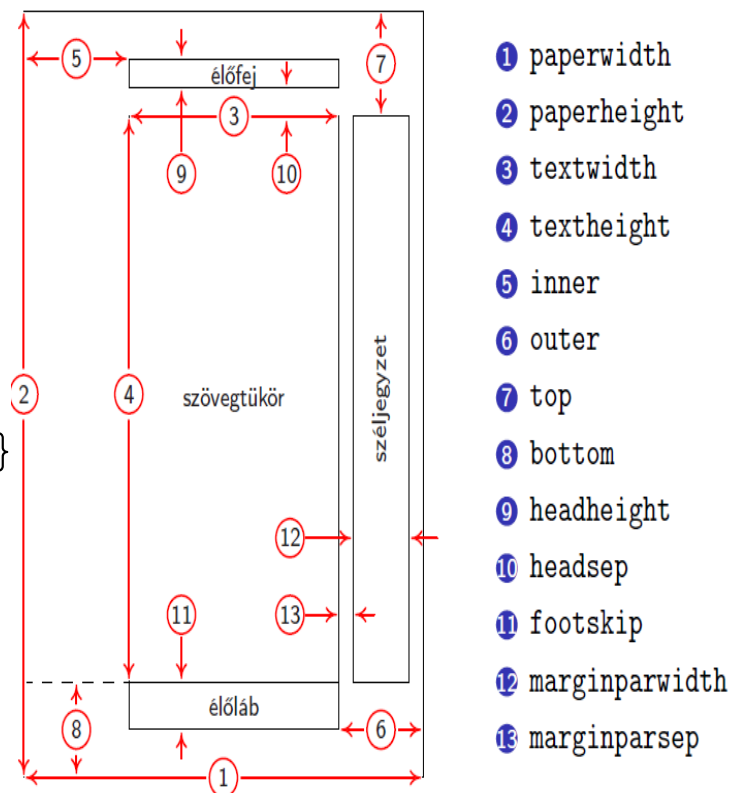
1	one inch + \hoffset	2	one inch + \voffset
3	\oddsidemargin = 13pt	4	\topmargin = -23pt
5	\headheight = 12pt	6	\headsep = 25pt
7	\textheight = 674pt	8	\textwidth = 426pt
9	\marginparsep = 10pt	10	\marginparwidth = 50pt
11	\footskip = 30pt		\marginparpush = 5pt (not shown)
	\hoffset = 0pt		\voffset =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

- Kisebbek 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} ==
\usepackage{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 `landscape` környezetével tudjuk megcsinálni:

```
\usepackage[landscape]{geometry}
```

- A forgatás hatására új oldal nyílik,
 - a szövegtükör és széljegyzet tartalma elfordul $+90^\circ$ -kal, de a fejléc és a lábléc nem,
 - a végén visszavált normál módra, de előtte új oldalt nyit.
- Nagyítás/kicsinyítés szintén a `geometry` csomaggal:

```
\usepackage[mag=500,truedimen,a4paper]{geometry}
```

 - A `mag=1500` opcióval másfélszeres nagyítás érhető el, míg a `mag=500` felére kicsinyít.
 - Ha valamely fix mértékegységgel megadott hossz méretet nem szeretnénk, hogy a nagyítás során megváltozzon, akkor a mértékegység elé tegyük a `true` szót.

Hosszméretek beállítása 1.

- LaTeX-ben létezik rugalmatlan és rugalmas hossz méret. Az előbbi egyszerűen `szám mértékegység` alakban (egybeírva!) kell megadni.
- A rugalmas méreteket `szám1[me1] plusz szám2[me2]`, vagy `szám1[me1] plusz szám2[me2] minusz szám3[me3]` alakban lehet megadni, ahol `[mex]` egy mértékegység.
- Az előbbi esetben a hossz `szám1[me1]` és `szám1[me1]+szám2[me2]` közötti tetszőleges értéket vehet fel, míg az utóbbi esetben az intervallum határait `szám1[me1]+szám2[me2]` és `szám1[me1]-szám3[me3]` jelöli.
- A `\setlength{\hp}{hossz}` paranccsal állíthatjuk be egy már 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.

Hosszméretek beállítása 2.

- `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 1.

- A `\vfill` és a `\hfill` utasításokkal illeszthetünk be vízszintes és függőleges végtelenül rugalmas térközt a dokumentumunkba.
- Ha például külön sorba írjuk, hogy `bal` széle `\hfill` jobb széle, akkor ezt láthatjuk:

bal szélejobb széle
- hiszen a `\hfill` rugalmasan kitöltötte a teret. 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 2.

- 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

- LaTeX-ben sajnos előfordul, hogy nem tud megfelelően elválasztani, így túllógó sorok keletkeznek. Ilyenkor be kell avatkozni, például manuális sortöréssel.
- A `\sloppy` utasítással nagyobb szóközöket is megengedünk a szavak között, mint például a fóliák szövege esetében.
- A `\fussy` utasítással kapcsolhatunk vissza az alapértelmezett üzemmódra.
- Azonos módon működnek a `sloppypar` és `fussypar` környezetek is.
- Keskeny helyeken célszerű a tömbös (justified) szedés helyett balra, vagy jobbra (esetleg középre) zárt szedést alkalmazni, mellyel elkerülhetjük a sorok egységnyi szélességbe kényszerítését.
- A balra zárt szedést a `flushleft` környezet, vagy a `\raggedright` parancs aktiválja.
- A szöveg jobbra zárásához a `flushright` környezetet, vagy a `\raggedleft` parancsot kell használni.
- Középre a `center` környezettel, vagy a `\center` utasítással lehet zárni a sorokat.
- Ha csak eseti jelleggel kell, a környezetek használatára van szükség.

- A draftwatermark csomag szükséges hozzá:
`\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.

- Példa:

```
\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}
```

Korrektúra és csapatmunka 1.

- 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}
```

Korrektúra és csapatmunka 2.

- `\usepackage[margins,adjustmargins]{trackchanges}`
 - `finalold` – A dokumentum minden módosításának elvetése.
 - `finalnew` – A dokumentum minden módosításának elfogadása.
 - `footnotes` – A módosítások megjelenítése lábjegyzetben.
 - `margins` – A módosítások megjelenítése széljegyzetként.
 - `inline` – A módosítások megjelenítése a folyó szövegben.
- `margins` opción belül további lehetőségek:
 - `movemargins`: balra tegye a szöveget (a `textwidth` marad)
 - `adjustmargins`: balra tegye a szöveget (a `textwidth` változik)

8. órai feladat

- Hozz létre egy legalább 5 fejezetből álló dokumentumot, ahol minden fejezet külön fájlban van tárolva!
- Az egyes fejezetek tartalmazzanak alfejezeteket, valamint értelmes szöveget, ahol a túllógó sorok megfelelő módon vannak kezelve!
- Formázd meg a dokumentumot (egyéni margóbeállítások stb.) a `geometry` csomag használatával!
- Készíts egyéni címoldalt! Esetleg használj vízjelet is!
- A dokumentum tartalmazzon rugalmas és rugalmatlan térközöket! (Lehet például nyilatkozatot készíteni, ahol ki kell hagyni az aláírásnak a helyet.)

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)

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

A Beamer sajátosságai 1.

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

A Beamer sajátosságai 2.

- Oldalméret: 4:3 (alapeset), de `[aspectratio=169]` opcióval 16:9 is választható.
- 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 szóelválasztások.
- 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`.

Első prezentáció létrehozása

- A kezdés a szokásos:

```
\documentclass[opciók]{beamer}
```

- Számos opció megadható (de nem kötelező!):

- `utf8` – pont ugyanaz, mint a
`\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 dokumentumtörzset a már jól ismert

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

 környezet tartalmazza!

KERETEK KÉSZÍTÉSE ÉS STRUKTURÁLÁSA

BEAMER

Keret (frame)

- A prezentáció keretekből áll, a keretek pedig diasorokból!
- A keret egy új környezet:

```
\begin{frame}
```

...

```
\end{frame}
```
- A keretnek van címe és megadható alcím is!
- Keret (lásd később) tartalmának függőleges pozíciója alapesetben 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.)
- Bonyolultnak 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).

- A keret kialakítására két megoldás létezik:

```
\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}
```

Egy keret példa

```
\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}
```

Címdia megadása

- A címdia tartalmazza a prezentáció címét, szerzőjét, a dátumot, helyszínt 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, illetve az oldalsávokat, akkor a `\begin{frame}[plain]` opciót kell használni!
- Ha `verbatim` szöveget akarunk a keretbe tenni, akkor azt a következő módon tehetjük:

```
\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 $\rightarrow$ Példa II $\rightarrow$ Példa III . . . kereteket kapunk

```
\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}
```

- Ha nem tetszik a keret tördelés, akkor így módosítható:

```
\setbeamertemplate{frametitle  
continuation}[from second]  
[\insertcontinuationcountroman.]
```

```
\setbeamertemplate{frametitle  
continuation}[from  
second][\insertcontinuationcount.]
```

```
\setbeamertemplate{frametitle  
continuation}[from second][(folyt.)]
```

Keret tartalmának több dián való megjelenítése

- Sajnos az `allowframebreaks` nem támogatja a több dia használatát a kereten belül, de sebj:
- 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}
```

Betűstílus megválasztása

- 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{betűbeállítás 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 betűstílus megválasztására

- 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 %félkövér  
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 dokumentumosztályhoz 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

Part (Rész) készítése 1.

- 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}
```

Part (Rész) készítése 2.

- 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!

Section (szakasz)

- Természetesen a prezentációnkat 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.

Tartalomjegyzék (dia) készítése 1.

- Használjuk a már megismert `\tableofcontents` parancsot:

```
\begin{frame}  
  \frametitle{Tartalomjegyzék} % keret neve  
  \tableofcontents  
\end{frame}
```
- *Vigyázat! Ha használunk `\part` parancsot, akkor utána kell 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}
```

Tartalomjegyzék (dia) készítése 2.

- 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` elrejtí, míg a `shaded` halvánnyá varázsolja az adott szakaszt, alszakaszt, alalszakaszt.
- Példa: Ha azt szeretnénk, hogy a tartalomjegyzékben csak a szakasz jelenjen meg, az alszakasz halványan, míg az alalszakasz ne jelenjen meg:

```
\tableofcontents[sectionstyle=show,subsectionstyle=shaded,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}
```

MEGJELENÉS ÉS KERET KOMPONENSEK

BEAMER

Keretek komponensei (ismétlés)

- Tartalom
- Fej- és lábléc
- Oldal sávok
- Navigációs szimbólumok
- Logó
- Keret címe és alcíme
- Háttér, rajta a vászon
- (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 nézzен 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:

Answered Questions

How many primes are there?

Open Questions

Is every even number the sum of two primes?

Ugye, hogy
szebb így!



- 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 automatikusan 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 jön a nev}
```

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

```
\end{alertblock}
```

ide jön 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)

Prezentáció tartalma – dobozok 1.

- A LaTeX-ben meglévő dobozok használhatóak természetesen.
- A Beamer további két újabb dobozt is bevezet:
 - `Beamercolorbox`
 - `Beamerboxesrounded`
- Fontos! A Beamer az egyéni színösszeállításnál két paramétert vár tőlünk:

```
\setbeamercolor{saját színeim}{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{saját színeim}{fg=red,bg=blue}
\begin{frame}{}{}
\begin{beamercolorbox}[wd=6cm,shadow=true,
rounded=true,center]{saját színeim}
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 doboz színezésre:

```
\setbeamercolor{egyen1}{fg=red,bg=blue}  
\setbeamercolor{egyen2}{fg=yellow,bg=blue}  
\begin{frame}{rovid keret cim}{hosszukeret cim}  
\begin{beamerboxesrounded}[upper=egyen1,  
lower=egyen2,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}`

- Példa – 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 Beamer-ben különféle 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!

Témák „felépítése”

- 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)
 - címoldal
 - listák
 - tömbök
 - tételszerű környezetek
 - képek, táblázatok
 - lábjegyzetek
 - Irodalomjegyzék
- Ha egy teljes témából nem tetszik valami, akkor másik témából választhatunk hozzá pl. fejléctet, színt stb.

Egyéni téma összeállítás

- Teljes témát a `\usetheme[opciók]{név}` parancs preambulumban történő kiadásával tölthetünk be.
- Például: `\usetheme{Singapore}` vagy `\usetheme{Warsaw}`
- Ez beállítja a témában foglalt szint, 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 (ismétlés)

- A címdia tartalmazza a prezentáció címét, szerzőjét, a dátumot, helyszínt 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}
```

Példák betűtípusra és színvilágra

■ Fontra:

- default (This is sans serif)
- professionalfonts
- serif
- structurebold
- structureitalicserif
- strucutresmallcapsserif

■ Színre:

- albatross
- beaver
- Beetle
- crane
- Dolphin
- dove
- Fly
- lily
- Orchid
- seahorse
- whale
- wolverine

- `\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:
 - Például: `\usetheme{Marburg}`
- Az egyetlen különbség az, hogy a `{section in head/foot}` 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}
```

Navigációs szimbólumok

- A legtöbb téma alapból 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}` parancsot használjuk.
- Ezek pedig a következők:
 - `\insertslidenavigationsymbol,`
 - `\insertframenavigationsymbol,`
 - `\insertsubsectionnavigationsymbol,`
 - `\insertsectionnavigationsymbol,`
 - `\insertdocnavigationsymbol,`
 - `\insertbackfindforwardnavigationsymbol`

Logó megadása

- A logó általában képből 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 a logó beillesztését.
- Példa:

```
\logo{\includegraphics[height=0.5cm]{logo.pdf}}
```

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

- Háttérszín beállítás 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]
```

9. órai feladat

- Hozz létre egy legalább 15 diából álló prezentációt (például az önlab beszámolót, vagy a beadott nagy házi feladatból egy rövid kivonatot)!
- A prezentáció tartalmazzon egyedi címdiát, valamint tartalomjegyzéket!
- A prezentáció tagolásához használd a part és secton/subsection parancsokat!
- Válassz egy számodra tetsző témát és alkalmazd a prezentációdra!
- Használj egyedi betűtípust, valamint a fej-, illetve láblécben add meg az aktuális dia sorszámát, a szerzőt, esetleg az épp aktuális szakasz címét!

LÁTVÁNYELEMENK

BEAMER

Overlays – \pause

- 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` parancsig 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}
```

Overlays – \only parancs

- 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>}
```

- 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}
```

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

Overlays – \alt

- A `\alt` parancs két szöveg környezetet tartalmaz, a `<szam>` opcióban megadott diákon az első szöveg környezet jelenik meg, a többin pedig a második:
- 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:
 - `<+(0)> = <3>`
 - `<+(1)> = <4>`
 - `<+(-1)> = <2>`
 - `<+-+(2)> = <3-5>`
 - `<-+> = <-3>`
 - `<.+>` - a `beamerpause` értékénél 1-gyel 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}
```

Transparency Effects 1.

- 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)

Transparency Effects 2.

- A `{\setbeamercovered{transparent=hány % az átlátsztság}}` parancs használható `\uncover`-nél is:

- Példa:

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

Gombok és linkek készítése 1.

- 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, a `\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 előző 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 az `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!

Animációk lejátszása 2.

- Töltsük be az `animate` csomagot!
- `\usepackage{animate}` (megjegyzés: `animate.sty` és az `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. ábráig 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 a következő parancs segítségével tudunk lejátszani:

```
\href{run:video.avi}{link szövege}
```

- ehhez természetesen kell 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 videófájl 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}`

<http://tinyurl.com/yxhefxap>

