

\LaTeX bevezetés

\LaTeX mint szövegszerkesztő

Vadon Viktória

2023/24/I. félév

1 Bevezetés: történet, motiváció

- Bevezetés
- Történeti áttekintő
- Miért \LaTeX ?
- Források

$\text{T}_{\text{E}}\text{X}$
(ejtsd „tech”/„tek”)
görög $\tau\epsilon\chi\nu\eta$ („techné”) szóból

és

$\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$

Mi az a \LaTeX ?

- kiadványszerkesztő:
 - tudományos cikkek (matematika és kapcsolódó területek, mint computer science, stb.)
 - diások tudományos előadásokhoz, tanórákhoz
 - nyomtatásra kész szakdolgozat
 - interneten megosztható pdf kereszthivatkozásokkal, linkekkel, stb.

Mi az a \LaTeX ?

- kiadványszerkesztő:
 - tudományos cikkek (matematika és kapcsolódó területek, mint computer science, stb.)
 - diások tudományos előadásokhoz, tanórákhoz
 - nyomtatásra kész szakdolgozat
 - interneten megosztható pdf kereszthivatkozásokkal, linkekkel, stb.
- jelölő nyelv (mint markdown, HTML, stb.)

Mi az a \LaTeX ?

- kiadványszerkesztő:
 - tudományos cikkek (matematika és kapcsolódó területek, mint computer science, stb.)
 - diások tudományos előadásokhoz, tanórákhoz
 - nyomtatásra kész szakdolgozat
 - interneten megosztható pdf kereszthivatkozásokkal, linkekkel, stb.
- jelölő nyelv (mint markdown, HTML, stb.)
- makró nyelv
 - saját makrókkal bővíthető

Mi az a \LaTeX ?

- kiadványszerkesztő:
 - tudományos cikkek (matematika és kapcsolódó területek, mint computer science, stb.)
 - diások tudományos előadásokhoz, tanórákhoz
 - nyomtatásra kész szakdolgozat
 - interneten megosztható pdf kereszthivatkozásokkal, linkekkel, stb.
- jelölő nyelv (mint markdown, HTML, stb.)
- makró nyelv
 - saját makrókkal bővíthető
- motorja: \TeX nyelv
 - „a szövegszedés assembly-je”
 - alapvető, ún. „primitív” tipográfiai (szedési) parancsok

A T_EX eredete

T_EX: (alacsony szintű) dokumentumkészítő rendszer, tipográfiai (nyomdai) tudással

- 1978, Donald E. Knuth
- T_EX nyelv: kb. 300 „primitív” parancs
- legújabb verzió: 3.141592653¹

¹a π jegyei szerint számozva; a diasor készülésekor legújabb 2021. januári verzió

A T_EX eredete

T_EX: (alacsony szintű) dokumentumkészítő rendszer, tipográfiai (nyomdai) tudással

- 1978, Donald E. Knuth
- T_EX nyelv: kb. 300 „primitív” parancs
- legújabb verzió: 3.141592653¹
- bővítés makrócsomagokkal (ez teszi „felhasználóbaráttá”):
 - *Knuth, plain T_EX, kb. 900 parancs*
forrás: „The T_EXbook”: forrás pdf
 - *A_MS-T_EX (American Mathematical Society)*
 - *1985, Leslie Lamport, L^AT_EX*

¹a π jegyei szerint számozva; a diasor készülésekor legújabb 2021. januári verzió

A \LaTeX története

\LaTeX : \TeX -re épülő, magas szintű (felhasználóbarátabb) kiadványszerkesztő

- 1985, Leslie Lamport, $\LaTeX 2.09$
- Lamport visszavonulása után: 1989 Stanford, $\LaTeX 3$ munkacsoport, cél: egységesítés, kiterjesztés, stb.
- elhúzódó fejlesztés, konkurens $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\LaTeX$; 1994-ben $\LaTeX 2_{\epsilon}$
- most: párhuzamosan $\LaTeX 2_{\epsilon}$ stabil verziók, $\LaTeX 3$ kísérleti verziók

Miért L^AT_EX? I

- nyomdakész pdf előállítás – pl. szakdolgozat, tudományos cikkek
- tudományos kiadványokhoz kifejezetten hasznos, formulák, táblázatok, kódrészletek stb. beillesztése
 - ugyanaz a kód újrahasznosítható diavetítésekhez
- nyílt forráskódú, ingyenes szoftver
 - bővítés csomagokkal, akárki hozzájárulhat
- jelölő nyelv, fordítás után rendszerfüggetlen, konzisztens eredmény pl. pdf-ben
- több forrásfájl kezelése, importálás
 - hosszú dokumentum (pl. könyv) áttekinthetősége; részletenként is fordítható
 - „stílusfájlok” vagy irodalomjegyzék újrahasznosítható!
 - több szerző külön fájlokban dolgozhat

Miért L^AT_EX? II

- tartalom/funkció és forma szétválasztása
 - mint HTML és CSS
 - cél: funkció jelölése a szövegben; munka közben nem kell a formázással foglalkozni
 - alapbeállításokkal is letisztult kinézet, nem kell tipográfusnak lenni (de újradefiniálhatók)
 - makrók segítségével egységes formázás definiálható egy egész dokumentumhoz, vagy adott elemeihez
 - makrók másolásával/importálásával több dokumentumra is átvihető az adott stílus
- jegyzékek (tartalomjegyzék, index, táblázatok, stb.)
automatikus generálása, kezelése – megfelelő inputtal (funkciót jelölni kell)

Hátrányok

- nem „WYSIWYG” (*what you see is what you get* – amit láatsz, azt kapod)
- különböző csomagok különböző felhasználóktól – kompatibilitási problémák
- meredek „learning curve” (tanulási görbe) – nyelv és logika elsajátítása, hibakezelés néha misztikus

LaTeX források

Magyarul

- röviden: Gyimesi Eszer könyve
- hosszabban: LaTeX 78 percben
- hosszan: Tómacs Tibor könyve

Angolul

- LaTeX wikibooks
- parancssorból `texdoc latex2e` (quick reference guide),
`texdoc source2e` (commentated source code)
- csomagok dokumentációi
- kérdések: Google > StackOverflow
- online reference manuals
- stb.

- 2 Dokumentum szerkezete
 - Minimális dokumentum
 - Documentclass
 - Preambulum
 - Csomagok

Minimális dokumentum

L^AT_EX

```
\documentclass{article}
% preambulumb (preamble):
% csomagok betöltése,
% stílus, saját makrók
\begin{document}
nemüres dokumentumtörzs:
tényleges tartalom
\end{document}
```

- mentsük fordítás előtt: valami.tex
- ha ok nélkül nem fordul: szóközt, ékezetes betűt irtsuk ki a fájlnevből és elérési útból

Minimális dokumentum

L^AT_EX

```
\documentclass{article}
% preambulom (preamble):
% csomagok betöltése,
% stílus, saját makrók
\begin{document}
nemüres dokumentumtörzs:
tényleges tartalom
\end{document}
```

HTML

```
<!DOCTYPE ...>
<HTML>
<HEAD>
fej: stílus, szkriptek, etc.
</HEAD>
<BODY>
törzs: tényleges tartalom
</BODY>
</HTML>
```

- mentsük fordítás előtt: valami.tex
- ha ok nélkül nem fordul: szóközt, ékezetes betűt irtsuk ki a fájlnevből és elérési útból

Documentclass

- pl. `\documentclass{article}`
- `documentclass` = dokumentumosztály
- \LaTeX fájl **kötelező első sora**
- megadja a dokumentum típusát
 - leggyakrabban használt: `article` = cikk
 - vagy pl. `report` = jelentés, `book` = könyv, `beamer` = diasor
- formázási alapbeállításokat is hoz magával
 - némelyik osztály (pl. `beamer`) extra parancsokat is (erről később)

Preambulum

preambulum:

- a `\documentclass` és `\begin{document}` közti szakasz
- HTML HEAD-hez hasonlóan „láthatatlan”
- céljai:
 - beállítások, testreszabás
 - csomagok betöltése (és beállítása)
 - saját makrók definiálása
 - stb.
 - (később látunk majd példákat)

Csomagok

- package = csomag
- gondoljunk rá úgy, mint egy programkönyvtárra
- adott célra szánt makrógyűjtemény; a \LaTeX funkcionalitását bővíti. a legtöbb csomagot felhasználók teszik közzé
 - szinte minden célra létezik csomag
 - kompatibilitás nem garantált
- csomagok (és dokumentumosztályok) böngészése:
CTAN (Comprehensive \TeX Archive Network)
- betöltés: `\usepackage{csomagnév}`

3 Szintaxis

- Parancsok típusai
- Argumentumok
- Parancsok hatásköre
- Hibakeresés
- Whitespace

Command sequence típusok

command sequence = vezérlő sorozat \approx parancs

- 1 ún. *vezérlő karakterek*: % \ { } ~ \$ ^ _ & #
 - legfontosabb %: komment, ezután a sor többi része „nem létezik” (a sor végi sortörés sem!)
- 2 \ + speciális karakter
 - egyes vezérlő karakterek kiíratása, pl. \\$ = \$
 - ún. lebegő ékezetek, pl. 'a = á
 - \\: sortörés
- 3 leggyakoribb *parancs/makró*: \ + betűsorozat
 - pl. \large: nagyobb betűméretre váltás
 - pl. \textit{dőltbetűs szöveg}
- 4 ún. *környezet*, \begin{valami} + \end{valami} parancspár
 - pl. már látott \begin{document}, \end{document}
 - pl. \begin{center}, \end{center}: középre zárt szöveg

Argumentumok

- argumentumok száma parancstól függ!
- kötelező argumentumok
 - minden egyes kötelező argumentum külön-külön kapcsos zárójelpárban {}, *szóköz nélkül!*
 - pl. `\textcolor{szín}{szöveg}`,
`\textcolor{red}{piros}` = piros
- opcionális argumentumok
 - közvetlen a parancs után, kötelező argumentumok előtt szögletes zárójelpárban []
 - ha nem használjuk, a zárójele is elhagyható!
 - több opcionális argumentum esetén egy zárójelben, vesszővel elválasztva!
 - pl. documentclass vagy csomag beállítása
`\usepackage[colorlinks,backref]{hyperref}`

Parancsok hatásköre I

- hatáskör szintén parancsonként változhat!
- egyes parancsoknál tárgytalan, mert csak helyileg kiír valamit, pl. `\$ = $`
- környezet, avagy `\begin{valami} + \end{valami}` parancspár esetén a kezdő- és záróparancs között érvényes a hatás, pl. `\begin{center} + \end{center}` között középre zár
- egyes parancsok csak az argumentumra hatnak

`\textit{dőlt betűs}` már nem

`\textcolor{blue}{kék}` már nem

dőlt betűs már nem

kék már nem

Parancsok hatásköre II

- DE! egyes parancsok *visszavonásig vagy felülírásig* érvényesek
 - pl. betűméret parancsok, `\large` stb., pl. `\centering`, középre igazítás (alternatív)
 - *környezeten belül* (`\begin{valami}`, `\end{valami}` között) kiadva a környezet végéig (`\end{valami}` parancsig) érvényes
 - ún. *csoportok* hozhatók létre, kapcsos zárójelek között `{}`; csoporton belül kiadott parancs is csoport végéig érvényes
 - `{` csoport kezdődik `\large` nagy betűk!
`\Large` felülírtuk, még nagyobb! `}`
csoport vége, ismét normál betűk
 - csoport kezdődik nagy betűk! felülírtuk, még nagyobb!
csoport vége, ismét normál betűk

Hibakeresés I

hibaüzenetek kezelése

- néha egyértelmű, ha tudunk angolul
- néha rejtélyesen fogalmaz és nem is jó helyen jelzi a hibát :(
- TIPP: fordítsunk gyakran, kevés módosításban könnyebb visszakeresni a hibát!
- TIPP: nem-e hagytunk bezáratlan zárójelet?
- TIPP: nem-e vezérlő karakter az a speciális karakter, amit kiírni szerettük volna, pl. \$, &? – szerkesztő parancsszínezése segíthet megtalálni!
- TIPP: szám után nem-e maradt le a mértékegység, vagy gépeltük el?

Hibakeresés II

gyakori hibák

- elgépelések
 - TIPP: használjuk a szerkesztő parancskiegészítését, hogy megelőzzük!
- ne hagyjunk szóközt parancs argumentumai között, mert parsing error-t (feldolgozási hibát) okoz!
 - avagy, a szóköznél azt hiszi, vége a parancsnak, és hiányzik az x darab kötelező argumentum
 - TIPP: ha túl sok az argumentum és átláthatatlanná válik, kommentezéssel át lehet verni:
% után nem regisztrál *a sortörés sem!* ezért

```
\documentclass%  
[12pt,twoside]%  
{article}
```

ugyanaz, mint `\documentclass[12pt,twoside]{article}`

Whitespace I

whitespace: láthatatlan karakterek összefoglaló neve, mint szóköz, tabulátor, sortörés, stb.

- \LaTeX dokumentum szövegtörzsében a whitespace feldolgozásra kerül – a forráskódbeli behúzás, stb. *nem* öröklődik!
 - előnye: forráskód tagolható és indentálható az átláthatóság kedvéért, és mindez nem zavarja a kész dokumentum oldalképét
 - hátránya: nem teljesen intuitív

forráskód	kész dokumentum
két sortörés (egy „üres”, avagy csak whitespace-t tartalmazó sor)	új bekezdés
szóközők, tabulátorok és max egy sor-törés kombinációja	szóköz

Whitespace II

- parancs megeszi az őt követő szóközt (vagy bármilyen whitespace karaktert)!
 - szóköz kényszerítése: `\ + szóköz`
 - pl. `\LaTeX nyelv = LATEXnyelv` VS `\LaTeX\ nyelv = LATEX nyelv`
 - vagy törhetetlen szóköz `~` – nem csak kikényszeríti a szóközt, de sortörés sem lehet itt; sorszámok után is hasznos

- 4 Szövegszerkesztés alapok
 - Nyelvek kezelése
 - Központozás
 - Betűstílusok
 - Méretezés
 - Szöveg színe
 - Dekoráció, transzformáció

Magyar nyelv I

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

- lokalizációs csomag, céljai:
 - automatikusan generált szövegek fordítása
 - dátum és formátuma
 - központosítás, pl. magyar idézőjelek
 - elválasztás szabályai
 - nem mindentudó, kézzel megadható neki elválasztás, pl. `\hyphenation{Ma-gyar-or-szág}`
 - bárhol szerepelhet a parancs; akár több helyen, a problémás szavak után; vagy közösen, egy vesszővel elválasztott listában

karakterkódolás magyar ékezetes betűkhöz

- `\usepackage{t1enc}` – a babel warning-ot (figyelmeztetést) dob, ha ez nincs betöltve
- ha a forrásban utf8-as karakterek szerepelnek:
`\usepackage [utf8] {inputenc}`

Babel több nyelvvel

- többnyelvű dokumentum esetén pl.
`\usepackage[english,magyar]{babel}`
- az *utolsónak* betöltött nyelv az alapértelmezett!
- egyes nyelvek sajnos nem kompatibilisek!
- váltás a *betöltött nyelvek között*: `\selectlanguage{nyelv}`,
adott ponttól az új nyelv szabályai érvényesek
- (rövid) idegen nyelvű szövegrészlet beszúrása (be nem töltött nyelvekre is!)
 - `\foreignlanguage{nyelv}{idegen nyelvi szöveg}`
 - `\begin{otherlanguage}{nyelv}`, `\end{otherlanguage}`
közé
 - érdemes extra csoportba (`{}` közé) zárni őket, mert
mellékhatásaik lehetnek

Központosítás

- mondatvégi térköz
 - magyarban: szokásos hosszúságú szóköz
 - \LaTeX alapértelmezés: angol konvenció, nagyobb térköz pont, pontosvessző stb. után!
 - babel nem állítja át! külön `\frenchspacing` parancs
 - eredeti visszaállítása: `\nofrenchspacing`
- idézőjel: *nem* a szokásos dupla idézőjel! "szó" = "szó", csúnya!
 - nyitó: „visszafelé” idézőjel ‘ (duplázva ha kell)
 - záró: szimpla idézőjel/aposztróf ’ (duplázva ha kell)
 - ‘ ‘szó’ ’ = „szó”
- kötőjel - = -; gondolatjel -- = -; angol em dash --- = —
- `\dots` = ... (... = ...)
- ligatúrák: betűtípus függvényében összeolvad ff, fi, fl;
elkerülés: `{ }` beillesztése a betűk közé

Betűstílusok I

- mint *dőlt*, **vastag**, KISKAPITÁLIS serif, *stb.*
- pl. `\textit{dőlt betűk}` *dőlt betűk*
- három kategóriába sorolhatók:

Family (betűtípus)

hatás	globális	lokális
roman	<code>\rmfamily</code>	<code>\textrm{szöveg}</code>
sans serif	<code>\sffamily</code>	<code>\textsf{szöveg}</code>
typewriter	<code>\ttfamily</code>	<code>\texttt{szöveg}</code>

Series (betűvastagság)

hatás	globális	lokális
normál (közepes)	<code>\mdseries</code>	<code>\textmd{szöveg}</code>
vastag	<code>\bfseries</code>	<code>\textbf{szöveg}</code>

Betűstílusok II

Shape (betűformák; roman family-vel)

hatás	globális	lokális
<i>italic</i>	<code>\itshape</code>	<code>\textit{szöveg}</code>
<i>slanted</i>	<code>\slshape</code>	<code>\textsl{szöveg}</code>
SMALL CAPS	<code>\scshape</code>	<code>\textsc{szöveg}</code>
upright	<code>\upshape</code>	<code>\textup{szöveg}</code>

- táblázatból látjuk, kétféle formájuk létezik:
 - „globális”: felülírásig, visszavonásig vagy csoport/környezet végéig érvényes
 - „lokális”: hatásköre csak az argumentum

Betűstílusok III

- a „lokális” parancsok egymásba ágyazhatók és halmozhatók:
 - `\textit{dőlt}` és `\textbf{vastag}`
dőlt és **vastag**
 - de kategóriánként csak 1 érvényesülhet, pl. *slanted* és SMALL CAPS felülírják egymást
 - más kombinációk betűkészlet hiánya miatt nem léteznek (pl. sans serif small caps)
- „globális” parancsok mind felülírják egymást (ős T_EX parancsok, teljes betűkészletet definiálnak)
- alapértelmezés változhat documentclass függvényében
 - pl. article-ben roman family
 - diasorban (beamer) sans serif family
- minden attribútum alapértelmezésbe állítása: `\normalfont` („globális”) vagy `\textnormal{szöveg}` („lokális”)

Betűstílusok IV

- kiemelés `\emph{kiemelt}` *kiemelt*, hatása *általában* dőlt betűk
 - egymásba ágyazható!
 - belső parancs ismét egyenes, harmadik szinten ismét dőlt, stb.²

```
\emph{kiemelt 1. szint \emph{kiemelt 2. szint  
  \emph{kiemelt 3. szint}}}
```

kiemelt 1. szint kiemelt 2. szint kiemelt 3. szint
 - `\textit{dőlt betűs}` környezetben szintén egyenes betűket eredményez
 - kis- és nagybetűk
 - `\MakeUppercase{Nagybetűsre!}` NAGYBETŰSRE!
 - `\MakeLowercase{KISBETŰSRE!}` kisbetűsre!
 - !! hibalehetőség: parancsokon kívül *mindenre* hatnak, pl. környezetek nevére is, (pl. `\begin{CENTER}`), így furcsa hibákat okozhatnak!

²csalok, mert beamerben pont nem működik

Betűméretek I

- dokumentum alapértelmezett betűmérete: `documentclass` opciójaként adjuk meg (opcionális argumentumban)
 - pl. `\documentclass[10pt]{article}`
 - alapért. 10pt, vagy 11pt, 12pt (egyéb méretezéshez extra csomagok szükségesek)
- dokumentumon belül: relatív méretezési parancsok, az alapértelmezetthez képest
- 2 verzió:
 - „globális”, felülírásig (vagy csoport végéig) érvényes parancsok, pl. `\normalsize`: normál (alapértelmezett) méret
 - környezetek, pl. `\begin{normalsize}` és `\end{normalsize}`, nyitó- és záróparancs között normál méret érvényes
 - környezet előtt és/vagy után extra térköz jelenhet meg, ezért ritkábban használjuk

Betűméretek II

táblázat: létező méretezési parancsok és hatásuk (kis- és nagybetűkre figyeljünk!)

parancs	hatás
<code>\tiny</code>	tiny
<code>\scriptsize</code>	scriptsize
<code>\footnotesize</code>	footnotesize
<code>\small</code>	small
<code>\normalsize</code>	normalsize
<code>\large</code>	large
<code>\Large</code>	Large
<code>\LARGE</code>	LARGE
<code>\huge</code>	huge
<code>\Huge</code>	Huge

Színek I

- töltsük be az `xcolor` csomagot:
preambulumban `\usepackage{xcolor}`
 - alapértelmezésben néhány alapszín érhető el az angol nevével: `black`, `blue`, `brown`, `cyan`, `darkgray`, `gray`, `green`, `lightgray`, `lime`, `magenta`, `olive`, `orange`, `pink`, `purple`, `red`, `teal`, `violet`, `white`, `yellow`
 - csomag opciókkal, mint pl. `dvips` csomag opcióval több szín érhető el
 - lista, hogy milyen opciókkal milyen színek (és milyen néven) érhetők el: dokumentáció
 - betöltés csomag opcióval: pl. `\usepackage[dvips]{xcolor}`
 - figyeljünk, hogy a korábbi csomag betöltést módosítsuk; ha ugyanazt a csomagot többször, különböző opciókkal töltjük be, hibát dob!

Színek II

- szöveg színezése: `\textcolor{szín}{szöveg}`, pl.
`\textcolor{red}{piros szöveg}` = **piros szöveg**, hatáskör:
csak az argumentum
- alternatív, felülírásig (vagy csoport végéig) érvényes parancs:
`\color{szín}`, pl. `\color{red}`
- oldal háttérszíne: `\pagecolor{szín}`
 - minden ezt követő oldalra érvényes. kikapcsolás:
`\nopagecolor`

Háttérszín, színes keret I

!! az alábbi műveleteket (egyelőre) *rövid* szövegen végezzük, mert a \LaTeX nem végez rajtuk automatikus tördelést!

- színes háttér (mintha szövegkiemelő lenne):

```
\colorbox{szín}{szöveg}, pl.
```

```
\colorbox{yellow}{szöveg sárga háttérrel} =  
szöveg sárga háttérrel
```

- színes háttér + színes keret:

```
\fcolorbox{keretszín}{háttérszín}{szöveg}
```

```
\fcolorbox{blue}{yellow}{szöveg kék kerettel %  
és sárga háttérrel}
```

```
szöveg kék kerettel és sárga háttérrel
```

```
\fcolorbox{blue}{white}{szöveg kék kerettel}
```

```
szöveg kék kerettel (fehér háttér)
```

Keretezés

!! az alábbi műveleteket (egyelőre) *rövid* szövegen végezzük, mert a \LaTeX nem végez rajtuk automatikus tördelést!

- keretezett szöveg: `\framebox{szöveg}`
- alapértelmezésben a szöveg szélességéhez igazodik a doboz
- első opcionális argumentum: szélesség, pl.
`\framebox[5cm]{szöveg}`
- alapértelmezésben középre igazít
- második opcionális argumentum: szöveg igazítása, pl.
`\framebox[5cm][c]{szöveg}`
 - `c` = közép, `l` = balra, `r` = jobbra, `s` = szöveg nyújtása, hogy kitöltse a dobozt – ha van benne nyújtható térköz

Tükrözés, forgatás I

- !! az alábbi műveleteket (egyelőre) *rövid* szövegen végezzük, mert a \LaTeX nem végez rajtuk automatikus tördelést!
- `\usepackage{graphicx}` csomag betöltése szükséges (preambulum)
 - forgatás: `\rotatebox{90}{szöveg}` – forgatás $+90^\circ$ -kal, avagy óramutatóval ellentétesen
 - forgatás középpontja alapértelmezésben a bal alsó sarok
 - átállítás: opcionális argumentumban, pl. középpont körül forgatás `\rotatebox[origin=c]{90}{szöveg}`, vagy jobb oldal közepe körül `\rotatebox[origin=rc]{90}{szöveg}`
 - lehetséges értékek: max 2 betű kombinációja a következőkből: 1 = left = bal, r = right = jobb, t = top = teteje, b = bottom = alja, c = center = közép

Tükrözés, forgatás II

- nagyítás és nyújtás: `scalebox{2.0}[0.5]{szöveg}` – vízszintesen 2-szeres, függőlegesen 0.5-szörös
 - függőleges argumentum opcionális, ha kihagyjuk, egyenlőnek veszi a vízszintessel és nyújtás nélkül nagyít
- negatív szám is lehet, akkor az adott irányban tükröz (is)! (mindkét irányban tükrözve 180°-os elforgatás)
- tükrözés: `\reflectbox{szöveg}` – függőleges tengely körül; tulajdonképpen `\scalebox{-1}[1]{szöveg}`