

# L<sup>A</sup>T<sub>E</sub>X diáorok

## Beamer dokumentumosztály

Vadon Viktória

2023/24/I. félév

## 1 Beamer bevezető

- Motiváció
- A beamer documentclass
  - Sajátosságok
  - Documentclass opciók
- Animáció szintaxisa
  - Frame, slide, overlay
  - Frame, slide, overlay példa
- Megjelenés, témák
  - Bevezető
  - Gyors testreszabás

# Motiváció

- beamer: documentclass diasorok készítésére
- miért készítsünk  $\text{\LaTeX}$ -kel diasort?
  - ha már írtunk  $\text{\LaTeX}$ -ben egy cikket vagy szakdolgozatot, viszonylag könnyen átültethető a kód
  - egyszer beállítva a megjelenés stílusát, ezután a  $\text{\LaTeX}$  gondoskodik az egységes formázásról, nekünk csak a tartalomra kell fókuszálni
  - $\text{\LaTeX}$  programozási, automatizálási képessége – tartalomjegyzék, irodalomjegyzék generálása, saját makrók használhatók, stb.
  - tudományos körökben elfogadott letisztult megjelenés
  - haladó használattal animálható és még inkább személyre szabható a megjelenés

# Beamer documentclass

- példa: ezek az órai diasorok is beamerrel készültek
- beamer mint documentclass:
  - többek között: fekvő elrendezés, sans serif font, relatíve nagyobb betűméret (tulajdonképpen kisebb oldalméret)
  - egyes elemek megjelenése megváltozik, pl. listák színnel, tételek színes dobozban, stb.
  - plusz elemek definíciója, pl. helyi többoszlopos elrendezés, színes dobozok a kiemeléshez, stb. (ld. 3. szakasz)
- !! egyes csomagokat, mint xcolor, hyperref, előre betölt, és belsőleg használ
  - ezeknek a csomagoknak az opcióit documentclass opcióként lehet megadni:

```
\documentclass[hyperref={átadandó opciók},%  
xcolor={átadandó opciók}]{beamer}
```
- választott stílustól függően a kattintható navigációról automatikusan gondoskodik!

# Documentclass opciók I

- oldalarány
  - alapértelmezett: 4:3 (128mm\*96mm)
  - `\documentclass[aspectratio=169]{beamer}`: 16:9-es arányok (160mm\*90mm)
- betűméret
  - alapértelmezett: 11pt
  - `smaller`, vagy 10pt
  - `bigger`, vagy 12pt
  - ritka, csak `extsize` csomag telepítésével: 8pt, 9pt, 14pt, 17pt, 20pt
- függőleges igazítás (dia kereten kívüli tartalma)
  - alapértelmezés: középre, `c` opció
  - felülre, `t` opció

# Documentclass opciók II

- `compress`: fejléc navigáció sűrítése (amelyik témánál releváns, mint pl. ez a diasor)
- `draft`: vázlat mód – munka közben hasznos: képek helyett csak keret, túllógás mutatása, stb.
- `handout`: nyomtatható verzió előállítás\*
  - ha animáljuk a diákat, ehhez szükséges lehet külön definiálni az időzítést
- matematikai formulák
  - `leqno`: formulák számozása bal oldalon
  - `fleqn`: formulák balra zárva középre igazítás helyett

# Animáció szintaxisa: frame, slide, overlay I

- alapegység: frame  $\approx$  oldal
  - minden, ami megjelenik, egy frame-ben kell legyen
  - a diák számozása (ha az adott stílus része) a frame-ek alapján történik
- frame-en belül: slide  $\approx$  képkocka
  - frame különböző verziói, nézetei
  - ha a frame egy gif, a slide-ok a különböző képkockái
  - a generált pdf-ben a slide-ok jelennek meg egy-egy oldalként
- overlay  $\approx$  időzítés
  - objektumhoz megadható, melyik slide-(ok)on jelenjen meg

# Animáció szintaxisa: frame, slide, overlay II

- slide-ok azonosítása: sorszámmal, 1-től kezdve
- hány slide lesz egy frame-ben?
  - a legkevesebb szükséges
  - animáció nélkül csak 1
  - de ha mi valamit a 15. slide-ra időzítünk, akkor (legalább) 15 (ha kell, sok azonos slide-ot illeszt be elé)
- overlay megadása
  - legtöbb parancshoz opcionális `<>` argumentumban, még `[ ]` elé
  - `<>` argumentumban vesszővel elválasztott slide sorszámmal és/vagy tartományokkal, pl. `<1,3-4>`
  - használható objektumra, pl. kép vagy listaelem megjelenik vagy sem
  - formázásra is: egyes slide-okon kiemelés színnel/betűstílussal, a többi slide-on alapértelmezett formázás
  - alternatív: több objektum együttes időzítésére speciális overlay parancsok, ld. 4. szakasz



## Frame, slide, overlay példa – kód

```
\begin{frame}{Frame, slide, overlay példa}
  \begin{itemize}
    \item itt vagyunk most: \insertframenumber.\ frame,
    \insertslidenumbers.\ slide
    \item ez az elem minden slide-on megjelenik
    \item<1> ez az elem csak az első slide-on
    jelenik meg
    \item<2-> ez az elem csak a második slide-tól
    jelenik meg
    \item \alert<3>{ez az elem piros a 3.\ slide-on}
    \item összesen tehát 3 slide lesz
  \end{itemize}
\end{frame}
```

# Frame, slide, overlay példa

- itt vagyunk most: 10. frame, 1. slide
- ez az elem minden slide-on megjelenik
- ez az elem csak az első slide-on jelenik meg
  
- ez az elem piros a 3. slide-on
- összesen tehát 3 slide lesz

# Frame, slide, overlay példa

- itt vagyunk most: 10. frame, 2. slide
- ez az elem minden slide-on megjelenik
- ez az elem csak a második slide-tól jelenik meg
- ez az elem piros a 3. slide-on
- összesen tehát 3 slide lesz

# Frame, slide, overlay példa

- itt vagyunk most: 10. frame, 3. slide
- ez az elem minden slide-on megjelenik
- ez az elem csak a második slide-tól jelenik meg
- ez az elem piros a 3. slide-on
- összesen tehát 3 slide lesz

# Témák bevezető I

- beamer megjelenését, stílusát *témák* adják
- léteznek teljes témák: a megjelenés minden komponensét vezérli – ún. `presentation theme`
- részleges témák: a megjelenés egy-egy komponensét vezérik
  - `outer theme`, külső téma – dia kerete, elrendezése: fejléc és/vagy lábléc, esetleg oldalsáv; ezekben navigáció, szakaszcímek, diaszám, attribúció, stb.
  - `inner theme`, belső téma – dia tartalmi elemei, pl. listák, színes dobozok formázása
  - színek – színtémák a fenti típusok elemeihez
    - teljes `color theme`
    - külső `outer color theme`
    - belső `inner color theme`
  - `font theme` betűtémák
    - pl. teljes betűtípus
    - címsorok, dobozok fejléce, stb. dekorálható betűstílusokkal

# Témák bevezető II

- nyilván a részleges témák kombinálhatók
- de a teljes téma betöltése után is felülírhatók komponensek részleges témák betöltésével!
- a teljes téma is tulajdonképpen jól összeválogatott komponensekből áll össze
- ha nem töltünk be témát: `default presentation theme` + `default color theme`, üres keret (csak dia címe), minimális színek
  - az alapértelmezés nem mindig tölthető vissza, ha felülírtuk
- ez egyes komponensekre 6. szakaszban térünk ki részletesen („rejtett” szakasz a fájl végén)
- még részletesebben: [beamer csomag dokumentáció](#) (link), a III. part-ban (15. section-től)
  - témák felülírása, saját témák definiálása sem órai anyag, arról is a dokumentációban olvashatunk

# Megjelenés gyors testreszabása

- gyors testreszabás: teljes téma presentation theme + opcionálisan színtéma (vagy színtémák kombinációja)
- referencia, kombinációk megtekintése: beamer theme matrix (link)
- presentation theme (teljes téma)
  - `\usetheme{téma név}`
  - városokról elnevezve
- color theme (színtéma)
  - `\usecolortheme{téma név}`
  - teljes, inner és outer színtémákra is ugyanaz a parancs!
  - teljes színtémák: repülő állatok után elnevezve
  - inner (belső) színtémák: virágokról elnevezve
  - outer (külső) színtémák: vízi állatokról elnevezve
  - !! kombinációk használhatók itt is, vagy már betöltött komponens felülírható

## 2 Struktúra

- Frame
- Cím-dia
- Szakaszcímek
  - Szakaszcímek – szerep, szintaxis, kiíratás
  - Section-önként ismétlődő kód
- Tartalomjegyzék



# Frame I

frame  $\approx$  oldal létrehozása

- frame környezet

```
\begin{frame}[opciók]{Cím}{Alcím}  
tartalom  
\end{frame}
```

- nem csak az opcionális argumentum hagyható el, hanem a cím és alcím is(!)
- de alcím nyilván nem adható cím nélkül
- ne legyen üres frame, fordításkor error (mint az üres dokumentumtörzs)
- alternatív, rövid verzió: környezet helyett parancs,  
`\frame[opciók]{tartalom}`

# Frame II

- frame [opciók] – vesszővel elválasztott lista
  - plain: plain oldalstílushoz hasonló, leveszi a frame keretét/outer theme-et (mint fejléc, oldalsáv, címsor háttere)
    - pl. cím diához
  - fragile: ha ún. *törékeny* parancsokat tartalmaz a dia, például ha verbatim-ot használunk
    - rejtélyes hibákat tud produkálni, ha elfelejtkezünk róla
  - noframenumbering: adott frame számozatlan
  - alábbiakban: mit csinálhatunk, ha túl sok a tartalom a diában?
  - diavetítésben igyekezzünk ezeket kerülni...
  - squeeze: csökkenti a függőleges térközt, hogy több férjen a frame-be
  - shrink: fenti + szükség szerint csökkenti a betűméretet is – nem javasolt, különböző betűméretek nem néznek ki jól!

# Frame III

- `allowframebreaks`: automatikus tördelés több frame-re
  - például ez a frame is automatikusan lett felosztva
  - a keletkező frame-eket római számokkal számozza a címben
- !! `overlay`-jel együtt nem működik, ne legyen több slide!
- vetítésben csak automatikusan generált tartalom- vagy irodalomjegyzékhez használjuk!
- töréspont befolyásolása: `\framebreak[erősség]` parancs
- `\framebreak`, `\framebreak[4]` adott ponton kényszeríti a törést
- `\noframebreak`, `\framebreak[0]` adott ponton megtiltja a törést
- 1-3 erősség: engedélyezi/különböző erősséggel javasolja a törést, de a  $\text{\LaTeX}$  tud optimalizálni

# Cím-dia

- adatok bevitele
  - `\author{}` szerző
  - `\title{}` cím
  - `\date{}` dátum
  - új mezők:
  - `\subtitle{}` alcím
  - `\institute{}` szervezet
- kiírás: `\maketitle` vagy `\titlepage` paranccsal
  - *itt* nincs különbség köztük
  - tipikusan plain (keret nélküli) frame-ben: pl.  
`\frame[plain]{\maketitle}`

# Struktúra: szakaszcímek I

- !! beamer-ben a szakaszcímsorok elsődlegesen a tagolást és navigációt szolgálják
  - témától függően: fejlécben vagy oldalsávban navigáció/mini tartalomjegyzék
  - témától függően: dia címe felett aktuális section (és subsection)
- part, section, subsection, subsection létezik
  - !! part szerepe egyedi: csak az aktuális part részei jelennek meg a tartalomjegyzékben és navigációban
- parancsaik csak frame környezetben *kívül* érvényesek!
- !! a `\section` és hasonló szakaszcímek automatikusan *nem* generálnak címsorokat!
  - `\maketitle`-höz hasonlóan írathatók ki a szakaszcímsorok:
  - `\partpage`, `\sectionpage`, `\subsectionpage`
  - és foglaljuk frame-be `\frame[plain]{\sectionpage}`

# Section-önként ismétlődő kód

- `\AtBeginSection{kód}` parancs: minden `\section` parancs után beilleszti kód-ot
  - `\AtBeginPart`, `\AtBeginSubsection`,  
`\AtBeginSubsubsection` variánsok

- *mire használható:*

- például `\sectionpage` automatikus beillesztésére

```
\AtBeginSection{\frame{\sectionpage}}
```

- vagy adott section tartalomjegyzékének beillesztésére (mint ebben a diásban)

```
\AtBeginSection{\frame{%  
\tableofcontents[sections={\value{section}}]}}
```

- itt `\value{section}`: aktuális szakasz sorszáma, belső változóként használható formában
- tartalomjegyzék opciói: következő dia!

# Tartalomjegyzék I

- generálása: `\tableofcontents` [opciók] parancs – az opcionális argumentum újdonság beamerben!
- néhány gyakori [opció]:
  - `sections=1`, vagy `sections={2-3,5-6}`: csak adott section, ill. adott section-ök és/vagy tartományok tartalomjegyzéke
    - hasonlóképp `part=1`: adott part tartalomjegyzéke
  - `currentsection`: többi section félig átlátszó
    - akkor érdemes ilyet használni, ha minden section elején kiírjuk a tartalomjegyzéket, hogy hol járunk az előadásban
    - `currentsubsection` verzió
  - `hideallsubsections`: csak section mélységig megy a tartalomjegyzék
    - `hideothersubsections`: csak az aktuális section subsection-jei vannak „kibontva”
  - egyéb, komplex opciók: ld. a beamer dokumentációt (link)

### 3 Új elemek

- Más class-okban
- Színek és dobozok
- Hasábok
- Verbatim kiterjesztése



# Más class-okban

- `beamerarticle` csomaggal egyéb `documentclass`-okban is használhatóvá válnak az alábbi, beamer által definiált parancsok
- illetve egyéb csomagokat is betölt, beállításokat végez el(!)
  - pl. `hyperref`, `xcolor` betöltése
  - pl. `amsthm` betöltése és `theorem` (tétel) környezet definiálása

# Színek és dobozok I

- `\alert{}` parancs: `\emph{}` mintájára, kiemelés pirossal (témától függően)
- színes dobozok, fontos információ kiemelésére: `block` környezet

```
\begin{block}{block címe}
block tartalma
\end{block}
```

!! a cím kötelező argumentum!

- cím nélküli `block`: üres zárójellel
- (rendszeről függően) furcsa hibákat tud produkálni, ha hiányzik!
- variánsok: `exampleblock`, `alertblock`, zöld és piros
- példák a következő dián!

# Színek és dobozok II

block címe

block tartalma

- felsoroláselem

cím nélküli block

exampleblock címe

exampleblock tartalma

- felsoroláselem

alertblock címe

alertblock tartalma

- felsoroláselem

# Hasábok I

- `columns` környezet, `\begin{columns}`
  - környezet opcionális argumentuma: oszlopok függőleges igazítása, pl. `\begin{columns}[c]`
  - `c` – középre, `b` – oszlopok alja egy vonalban, `t` – oszlopok teteje (első alapvonala) egy vonalban, `T` – oszlopok (tényleges) teteje egy vonalban
- ezen belül tetszőleges számú oszlop
  - már amennyi a diába belefér
- oszlopok megadása: `\begin{column}{szélesség}` környezet
  - figyeljünk az `s`-re vagy hiányára!
  - !! a szélesség kötelező argumentum
  - lehet pl. `4cm`, `120pt`, vagy  $\LaTeX$  hosszváltozókkal:  
`0.5\linewidth` – sorszélesség fele

# Hasábok II

- alternatív: `\column{szélesség}` parancs, új oszlopot kezd, ami a következő `\column` parancsig vagy a `columns` környezet végéig tart

!! az adott `columns` környezeten belül *ne keverjük* az oszlopok környezet és parancs formájában való megadását!

- példa

```
\begin{columns}[c]
\begin{column}{.5\linewidth}
első oszlop tartalma
\end{column}
\begin{column}{.5\linewidth}
második oszlop tartalma
\end{column}
\end{columns}
```

# Verbatim kiterjesztése

- semiverbatim: verbatim kiterjesztése
- *részben* feldolgozatlan kód, de a `\`, `{` és `}` megtartja a jelentését(!)
  - így használhatók benne  $\LaTeX$  parancsok(!)
  - *többek között*: *színezés*, *overlay* (időzítés)
  - ha egy parancsot felismert, akkor már helyesen dolgozza fel hozzá a `[ ]` és/vagy `<>` argumentumot is
  - `{` és `}` kiírása: szokásos `\{` és `\}`
  - `\` kiírása `\\`-sel(!)
    - sortörés parancsra úgyszincs szükség, kódbeli sortörést megőrzi
- ```
\begin{semiverbatim}
\\begin{\alert{frame}}\{\cím\}
\end{semiverbatim}

\begin{frame}{cím}
```

## 4 Időzítés

- Az overlay argumentum
- Alap overlay parancsok
  - `\pause`
  - `\onslide<>`
  - `\only<>`
- Láthatatlanság
- Még több overlay parancs

# Az overlay argumentum I

- szemlélet: ld. 3. szakasz
  - röviden:
    - frame  $\approx$  oldal
    - slide  $\approx$  képkocka – legkevesebb szükségeset generál
    - overlay  $\approx$  időzítés – objektum mely slide-okon jelenik meg?
- overlay megadása: új overlay argumentum  $\langle \rangle$ 
  - még `[]{}` elé
  - $\langle \rangle$  argumentumba: vesszővel elválasztott slide sorszám(ok) és/vagy tartomány(ok), ahol az adott elem megjelenik
  - rövidítés:  $\langle -3 \rangle$  ekvivalens  $\langle 1-3 \rangle$ -mal,  $\langle 2- \rangle$ : második slide-tól mindig
  - sok létező parancshoz is definiálja!
  - pl.  
`\includegraphics<1,3-4>[width=.5\linewidth]{image}`  
az image fájlnevű kép az 1., 3. és 4. slide-on jelenik meg



# Az overlay argumentum II

!! overlay szintaxis környezet esetén:

```
\begin{környezetnév}<overlay>[]{}  
• pl.  
\begin{theorem}<2->[Tétel neve]  
a tétel a második slide-tól látható
```

- formázási parancsokkal, mint `\textbf`, `\alert` is használható
  - itt az adott slide-(ok)on érvényes a formázás, különben formázatlanul jelenik meg a szöveg
- nagyobb egységek együttes időzítése: beamer-specifikus overlay parancsokkal, pl. `\pause`, `\onslide<>`, `\only<>` – következő szakaszban

# \pause

- legegyszerűbb `\pause` parancs: „pillanatállj” a kiírásban
  - azaz, a `\pause` utáni tartalom csak a következő slide-ra kerül
  - több `\pause` is használható a frame-en belül
  - hasznos pl. ha fentről lefelé, szépen sorban fedjük fel a dia tartalmát, pl. egy lista elemeit

például

```
\begin{frame}{Pause példa}
\begin{enumerate}
\item első \pause
\item második \pause
\item harmadik
\end{enumerate}
\end{frame}
```

# Pause példa

1 első

# Pause példa

- 1 első
- 2 második

# Pause példa

- 1 első
- 2 második
- 3 harmadik

## \onslide&lt;&gt;

- `\onslide<overlay>`, vagy `\onslide<overlay>{tartalom}`
  - `\onslide<overlay>{tartalom}` verzió: csak a tartalom-ra vonatkozik
  - `\onslide<overlay>` *felülírásig* érvényes – következő `\onslide<overlay>` parancsig vagy `\pause`-ig
- csak az `<overlay>`-ben megadott slide-okon jeleníti meg a tartalmat
  - legtöbb `\parancs` esetén `\parancs<overlay>` ekvivalens `\onslide<overlay>{\parancs}`-csal
- ezzel nemlineáris sorrendben is felfedhető a frame tartalma

## \only&lt;&gt;

- `\only<overlay>{tartalom}`
- szintén az `<overlay>`-ben megadott slide-(ok)on jeleníti meg a tartalmat
- különbség `\onslide` és `\only` között:
  - `\onslide`-dal a tartalom láthatatlan, de elfoglalja a helyet a dián
  - `\only`-val a tartalom nem foglal el helyet a dián, mikor nincs jelen
- `\onslide` VS `\only` példa
  - 1. slide
    - `| \onslide<2>{onslide}`
    - `| \only<2>{only}`

## \only&lt;&gt;

- `\only<overlay>{tartalom}`
- szintén az `<overlay>`-ben megadott slide-(ok)on jeleníti meg a tartalmat
- különbség `\onslide` és `\only` között:
  - `\onslide`-dal a tartalom láthatatlan, de elfoglalja a helyet a dián
  - `\only`-val a tartalom nem foglal el helyet a dián, mikor nincs jelen
- `\onslide` VS `\only` példa
  - 2. slide
  - `onslide | \onslide<2>{onslide}`
  - `only | \only<2>{only}`



# Láthatatlanság

- `\onslide<>` paranccsal a tartalom láthatatlan, de kitölti a helyet a frame-ben
- ez a *láthatatlan tartalom* lehet:
  - teljesen láthatatlan – ez az alapértelmezés
  - áttetsző (transparent)
- beállítás `\setbeamercovered{kulcsszó}` paranccsal, preambulumban
- a kulcsszó lehet:
  - láthatatlan: `invisible`
  - áttetsző: `transparent`, vagy `transparent=20`
    - a számérték az opacity, *átlátszatlanság* mértéke %-ban, alapértelmezés 15%
  - `dynamic`: dinamikus, átlátszatlanság mértéke változik attól függően, hogy slide-dal ezelőtt vagy slide-dal ezután volt/lesz látható a tartalom
    - `highly dynamic` variáns: drasztikusabb hatással

# Még több overlay parancs

- `\visible<overlay>{tartalom}`
  - `\onslide<>`-dal egyező szintaxis
  - különbség: `\visible` független a láthatatlanság beállításától, a rejtett tartalom mindig *teljesen láthatatlan*
  - ellentéte: `\invisible` parancs – az overlay-ben megadott slide-okon láthatatlan, különben mindig látható
- elrendezéshez hasznos: `overlayarea` környezet
  - `\begin{overlayarea}{szélesség}{magasság}`
  - tartalomtól függetlenül mindig egy szélesség×magasság méretű téglalapot foglal el
  - tartalma tetszőleges parancsokkal cserélhető – de érdemes `\only<>`-t használni
  - sokszor `columns` környezet egyszerűbb, és betölti ugyanazt a funkciót

## 5 Áttűnések, média

- Áttűnések
- Automatikusan léptetés
- Média
- Animáció
  - Szintaxis
  - Színek animálása

# Áttűnések (transitions) I

- áttűnések: „animáció” frame vagy slide váltásakor
- kész pdf-et diavetítés módban megtekintve érvényesül (pl. Adobe Acrobat Readerben Ctrl+L)
- megadás: az adott frame vagy slide milyen áttűnéssel *jeljen* meg
  - azaz, pl. áttűnés az első és második frame között a *második* frame-hez tartozik
  - pl. „dissolve” áttűnés parancsa: `\transdissolve`, vagy `\transdissolve<overlay>`
  - `overlay` nélkül, vagy `<1>`: a frame-re vonatkozik
  - egyéb `overlay` értékek: slide-ok megjelenési animációja
  - de `\transdissolve<1,2>` is működik
  - a parancs a frame-ben bárhol szerepelhet

# Áttűnések (transitions) II

- áttűnésekhez tartozhat opcionális argumentum is, pl. `\transdissolve<2>` [opciók]
  - `duration=1`: animáció hossza, másodpercben mérve
  - ahol releváns és választható irány: `direction=90`, fokban mérve, megengedett 0, 90, 180, 270 (és glitter-nél 315)
- áttűnések parancsai
  - teljes leírás: beamer dokumentáció, 14.3. szakasz, 139. oldal
  - kísérletezéshez itt a lista:
  - `\transreplace`: alapértelmezés, nincs animáció
  - `\transdissolve`
  - `\transfade`
  - `\transpush`
  - `\transcover`
  - `\transfly` – irány
  - `\transglitter` – irány
  - `\transwipe` – irány

# Áttűnések (transitions) III

- `\transblindshorizontal`, `\transblindsvertical`
- `\transboxin`, `\transboxout`
- `\transsplitverticalin`, `\transsplitverticalout`,  
`\transsplithorizontalin`, `\transsplithorizontalout`

# Automatikus léptetés

- automatikus léptetés
- `\transduration<overlay>{2}`: overlay-ben megadott slide-ok automatikus léptetése 2 másodperc után
- ha 0-t adunk meg, a lehető legrövidebb ideig mutatja (felvillantja) a slide-ot – „animáció” készíthető vele
- automatikus léptetés kikapcsolása: `\transduration{}`
- overlay nélkül is használható – akkor a frame *összes slide-jára* vonatkozik
- az adott frame-re vonatkozik csak

# Média I

- szükséges: multimedia csomag
- cél: videó beillesztése pdf-be
- !! képpel ellentétben nem fordítja bele a fájlba, a videó fájlt mellékelni kell(!)
  - de ha a pdf megjelenítő le tudja játszani a videót, meg tudjuk jeleníteni a diasorban
  - vagy beállíthatjuk, hogy külső lejátszót nyisson meg
- szintaxis:  
`\movie[opciók]{előnézet}{fájlnév}`
  - a videófájl legyen elérhető – pl. legegyszerűbb azonos mappában a pdf-fel, vagy adjuk meg a *relatív* elérési utat
- **előnézet**: tetszőleges L<sup>A</sup>T<sub>E</sub>X tartalom, amit a lejátszás előtt megjelenít, és erre kattintva indul a lejátszás



# Média II

- !! az általa elfoglalt téglalapban jelenik meg a videó, az adott képarányban
- ezért érdemes egy előnézeti képernyőképet beilleszteni! pl. `\includegraphics`-szal, vagy poster **opcióval** generáltatni
- néhány hasznos **[opció]**:
  - poster: a videó első képkockája az automatikus előnézet
  - width=hossz, height=hossz: előnézet (és vele együtt videó!) méreteinek beállítása/felülírása
  - repeat vagy loop: lejátszás ismétlése automatikusan
  - once: lejátszás csak egyszer (alapértelmezés)
  - externalviewer: külső lejátszó használata, ha a pdf megjelenítő nem tudja lejátszani a videót
    - open: a lejátszó maradjon nyitva a lejátszás után
  - showcontrols: húzható csúszka megjelenítése a videó alján
  - részletesebb leírás: beamer dokumentáció

# Média III

- jó tudni: a dia továbbléptetésével automatikusan leáll a lejátszás
- a parancs hangfájl beillesztésére is alkalmas(!)
  - nyilván `poster` opciónak nincs értelme, de minden más analóg módon működik
  - hang lejátszására ugyan létezik külön `\sound` parancs, de Acrobat Reader bug-ok miatt meglehetősen macerás... ld. a beamer dokumentációban, 14.2. szakasz, 137. oldal
  - ezért hangfájl beillesztésére is `\movie` parancs javasolt

# Animáció I

- a beamer gif jellegű animációra képes, slide-ok/képkockák gyors pörgetésével
- ehhez beilleszthetők kézzel a különböző tartalmak a megfelelő overlay-jel, pl. ha mozgóképet szeretnénk
- vagy  $\text{\LaTeX}$  kód segítségével interpolálhatók változók értékei pl. elemek mozgatására, színek fokozatos keverésére, stb.
- két parancs kell hozzá:
  - `\animatevalue` interpolálja nekünk a belső változó értékét
  - `\animate` megadja, hogy melyik slide-okat kell gyorsan pörgetni
- `\animate`
  - pl. `\animate<2-10>`, 2-10 slide-okat pörgessük gyorsan
  - gyakorlatilag ekvivalens `\transduration<2-10>\{0\}`-val
  - kiváltható `\transduration`-nel, finomabban kontrollálható, hogy mennyi ideig mutasson egy-egy slide-ot

# Animáció II

- `\animatevalue`:
  - kis nehézség: plain  $\TeX$  szintaxissal kell változókat deklarálni
    - számláló – egész számok: `\newcount\szamlalo`
    - hossz – hosszmértékek: `\newdimen\hossz`
    - itt változók neve `\szamlalo`, `\hossz`
    - formailag úgy néz ki, mint egy makró/parancs, azaz `\`-szel kezdődjön és angol ábécé kis-és nagybetűit tartalmazhatja
  - `\animatevalue<kezdőslide-végslide>%  
{\változó}{kezdőérték}{végérték}`
  - az adott slide-tartományban lineárisan interpolálja a változót a kezdő- és végérték között
  - az `\animate` és `\animatevalue` overlay tartományainak nem feltétlen kell azonosnak lennie!

# Animáció III

- példa egy „vándorló” szövegre:

```
\begin{frame}
\newdimen\behuzas
\animatevalue<1-11>{\behuzas}{0cm}{5cm}
\transduration<1-10>{0.1}
\hspace{\behuzas}helló világ!
\end{frame}
```

# Színek animálása I

- színkeverés emlékeztető:  
`\textcolor{red!30!blue}{színes}` – 30%-ban piros, maradék 70%-ban kék: `színes`

- színátmenet a keverési arány animálásával

```
\begin{frame}
\newcount\pirosszazalek
\animatevalue{\pirosszazalek}{100}{0}
\animate
\textcolor{red!\the\pirosszazalek!blue}%
{pirosból kékbe}
\end{frame}
```

- `\the\pirosszazalek` az érték kiírása számjegyekkel – itt így jó, mert a színkeverés parancsa ilyen formában várja az argumentumát!

# Színek animálása II

- színkeverés egy egész területre: `colormixin` környezet
  - egy adott színt képes a terület minden eleméhez azonos %-ban hozzákeverni
- pl. háttérszín belekeverésével az egész terület halványítása / „áttetszővé tétele”
  - `\begin{colormixin}{70!white}`  
70%-os átlátszatlanságú szöveg  
`\textcolor{blue}{színesben is}`  
`\end{colormixin}`  
70%-os átlátszatlanságú szöveg `színesben is`
  - a kapott színek `black!70!white`, `blue!70!white`
  - !! az argumentumban: `70!white`
    - az *eredeti színek* százalékos aránya kell
    - az első ! nem kell!
    - `white`: a hozzájuk kevert szín (fehér)

# Színek animálása III

- vagy mintha egy félig áttetsző, színes fóliát tennénk a tartalom fölé\*:
  - a háttérszínhez nem nyúl, de `\colorbox`-hoz igen!
  - `\begin{colormixin}{80!blue}`  
`\colorbox{white}{80%-os átlátszatlanságú szöveg%`  
- halványkék tónusban  
`\textcolor{red}{színesben is}`  
`\end{colormixin}`  
80%-os átlátszatlanságú szöveg – halványkék tónusban  
színesben is
- itt is animálhatjuk a keverési arányt a fenti szintaxissal
  - pl. 100-tól 0-ig animálva fehérbe fakulhat az egész dia
  - hasznos lehet: beamer-ben kulcsszó a(z átlagos) háttérszínre `averagebackgroundcolor`, úgy használható, mint bármely más elnevezett szín (red, blue, stb.)



## 6 Megjelenés: témák részletesen

- Témák típusai
- Presentation theme
- Font theme
- Inner theme
- Outer theme
- Color theme

# Témák típusai I

- diasor megjelenését, stílusát *témák* határozzák meg
  - elég sok előre definiált téma és komponens közül választhatunk
  - de sajátok is definiálhatók
- valamiféle téma kiválasztása, betöltése kötelező (T<sub>E</sub>X motortól függően nem is fordul nélküle, vagy alapértelmezettet használ)
- témák típusai: teljes megjelenés, vagy komponensenként
- presentation theme (prezentáció téma) = teljes téma, az alábbi négyből minden komponenst beállít
  - presentation theme után betölthetők komponensek külön-külön, ezzel felülírja a korábbi, módosítható az összeválogatott csomag

# Témák típusai II

- lehetséges kombinálható komponensek
  - color theme (színtéma)
  - font theme (betűtípus és -méret)
  - outer theme (külső téma): a frame keretét adja
    - van-e fejléc és/vagy lábléc, vagy oldalsáv
    - mi kerül ezekbe a sávokba, pl. prezentáció címe és szerzője, vagy tartalomjegyzék/navigáció
    - amiket automatikusan generál a frame környezet
  - inner theme (belső téma): frame-en belüli elemek megjelenése
    - mint felsorolások, színes dobozok (kiemelt tartalom, tételek), tartalomjegyzék dia
    - amiket mi kézzel helyezünk el tartalomként

# Presentation theme

- presentation theme: összeválogatott komponensek
  - utólag egyesével felülírhatók benne a komponensek
- példák megtekinthetők: beamer theme matrix ([link](#))
  - attól mátrix, hogy presentation theme + color theme kombinációkat mutat be
- `\setheme{Név}` paranccsal tölthető be
  - figyeljünk a kis-és nagybetűkre!
  - tulajdonképpen egy `\usepackage` parancsot használ
- adott témától függően módosító opciók:  
`\setheme[opciók]{Név}`
  - túl sok téma és opció, hogy mind felsoroljuk
  - lásd például: beamer csomag dokumentáció ([link](#))

# Font theme – betűstílus

- alapértelmezés: sans serif font
- különböző betűstílusok betöltése: `\usefonttheme{név}`
- váltás serif betűtípusra: serif téma
- többi téma: *structure* elemek kiemelésére
  - *structure*, „struktúrális elemek” = cím, frame-ek címe, tételek és dobozok címei
  - **vastag** betűkkel: `structurebold` téma
  - *italic (serif)* betűkkel: `structureitalicserif` téma
  - KISKAPITÁLIS (SERIF) betűkkel: `structuresmallcapserif` téma

# Inner theme

- inner theme, „belső” téma: frame belső, tartalmi elemeinek (mint felsorolás, dobozok és floatok, szakaszcímek) stílusa
- betöltés `\useinnertheme{név}`
- előre definiált belső témák
- default téma: alapértelmezett, egyszerű megjelenés
- circles: számozás színes körökben
- rectangles: számozás színes négyzetekben
- rounded: számozás árnyékolt színes körökben, dobozok lekerekítése (mint ebben a diasorban)
  - extra opció: dobozok árnyékolása, `\useinnertheme[shadow]{rounded}`

# Outer theme I

- outer theme, „külső téma”: a frame *keretét* határozza meg:
  - van-e oldalsáv
  - van-e fejléc és/vagy lábléc
  - van-e (kattintható) navigáció és hol (fejlécben vagy oldalsávban lehetséges)
- adott téma/stílus betöltése: `\useoutertheme{név}`
- elérhető témák
  - vizuális példák a beamer dokumentációban – vagy teszteljük magunk!
- default téma: alapértelmezett, „üres” keret
- infolines téma
  - fejlécben és láblécben egymás mellett a vetítés alapinformációi
  - navigáció nélkül
  - fejlécben aktuális section+subsection
  - láblécben cím, szerző, intézmény+oldalszám

# Outer theme II

- tree téma
  - fejlécben egymás alatt növekvő behúzással cím, section, subsection
  - extra opció: hooks, leágazó „kampók” a fejlécben, használat `\useoutertheme[hooks]{tree}`
- split téma
  - fejlécben tartalomjegyzék és navigáció: section-ök címei, és aktuális section subsection-jeinek címei
  - láblécben cím és szerző



# Outer theme III

- sidebar téma
  - színes háttérű oldalsáv tartalomjegyzékkel
  - section-ök címei
  - extra opciók: `\useoutertheme[opciók]{sidebar}`
    - `left`: bal oldalon (alapért.), `right`: jobb oldalon
    - ha túl hosszú a tartalomjegyzék, subsection-ök elrejtethők: mind elrejtése `hideallsubsections`, csak aktuális section kibontása `hideothersubsections`
    - szélesség `width=hossz`, cím magassága `height=hossz`

# Outer theme IV

- miniframes téma
  - navigáció a fejlécben: section-ök címei, alatta karikák jelzik a frame-eket (mint ebben a diáorban)
  - lábléc opcionális, 2 vagy 3 információs dobozzal:  
`\useoutertheme[footlines=kulcsszó]{miniframes}`,  
ahol a kulcsszó lehet
    - `empty` – kikapcsolás, alapértelmezett
    - `authorinstitute`, `authortitle`, `institutetitle`,  
`authorinstitutetitle`
    - ahol `author` = szerző, `institute` = intézmény, `title`=cím

# Color theme – színtéma I

- color theme (színtéma)
- az inner és outer theme-khez hasonlóan létezik ezekből is belső, külső, illetve teljes színtéma, és néhány kivétel
- betöltés: `\usecolortheme{név}`, vagy `\usecolortheme[opciók]{név}`
  - kombinációban is használhatók, egy belső + egy külső, egy teljes + részben felülírjuk egy külső/belső színtémával, stb.
- példákhoz, illetve teljes témákkal (presentation theme) való kombinációikhoz lásd szintén a beamer theme matrix-ot ([link](#))
  - hogy igazodjunk el köztük?
  - belső színtémák virágokról kapták a nevüket
  - külső színtémák vízi állatokról kapták a nevüket
  - teljes színtémák repülő állatokról kapták a nevüket

# Color theme – színtéma II

- kivételek, nem kategorizált színtémák
- default: alapértelmezett, egyszerű
- `\usecolortheme[színmodell=érték]{structure}`:  
*strukturális* elemek (pl. címsorok, tételek) színének megváltoztatása
  - legegyszerűbben szín nevével: `named=red` – amit `xcolor` ismer
  - de lehet pl. `RGB={255,0,0}`, vagy egyéb modellek amiket `xcolor` ismer
- `sidebartab` téma: sidebar külső témával az aktuális section kiemelése más színű háttérrel