

\LaTeX : Diasorok beamer-rel

Vadon Viktória

2022/23/I. félév

- 1 Beamer bevezető
 - Motiváció
 - A beamer documentclass
 - Sajátosságok
 - Documentclass opciók
 - Struktúra áttekintő
 - Frame, slide, overlay
 - Frame, slide példa
 - Section és társai
 - Megjelenés, témák – áttekintő
 - Alapok, gyors testreszabás
 - Témák típusai

Motiváció

- beamer: documentclass diasorok készítésére
- miért készítsünk \LaTeX -kel diasort?
 - ha már írtunk \LaTeX -ben egy cikket vagy szakdolgozatot, viszonylag könnyen átültethető a kód
 - egyszer beállítva a megjelenés stílusát, ezután a \LaTeX gondoskodik az egységes formázásról, nekünk csak a tartalomra kell fókuszálni
 - \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. többoszlopos megjelenés, színes dobozok a kiemeléshez, színezhető verbatim jellegű környezet (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
 - alapértelmezés: középre, c opció
 - felülre, t opció

Documentclass opciók II

- `compress`: fejléc navigáció sűrítése (amelyik témánál releváns)
- `draft`: vázlat mód
- `handout`: nyomtatható verzió előállítása
 - ehhez az `overlay`-eket külön meg kell(het) adni a `handout` verzióhoz
- `formulák`
 - `leqno`: matematikai formulák számozása bal oldalon
 - `fleqn`: formulák balra zárva középre igazítás helyett

Struktúra: frame, slide, overlay I

- alapegység: frame
 - minden, ami megjelenik, egy frame-ben kell legyen
 - gondoljunk rá úgy, mint hagyományos szöveges dokumentum egy-egy oldalára
 - a diák számozása (ha az adott stílus része) a frame-ek alapján történik
- frame-en belül: slide
 - frame különböző verziói, nézetei
 - ha a frame egy gif, a slide-ok a különböző képkockái
 - a pdf-ben a slide-ok jelennek meg egy-egy oldalként

Struktúra: frame, slide, overlay II

- overlay: kb. „időzítés”
 - annak a megadása, hogy az adott tartalom (szöveg, listaelem, táblázat, akármilyen) a frame hanyadik slide-ján (vagy slide-jain) jelenjen meg
 - de formázás is időzíthető: ugyanaz a szöveg lehet piros a frame egyik slide-ján és fekete a másikon
 - a slide-okat 1-től számozza, ezzel a sorszámmal lehet utalni rájuk
 - minden frame-hez a legkevesebb szükséges slide-ot hozza létre
 - pl. ha az egyik overlay-hez a 2. slide kell, a másikhoz a 3. és 5. akkor 5 slide-ot hoz létre
 - overlay megadása: legtöbb parancshoz opcionális `<>` argumentum, még `[]` elé
 - vagy speciális overlay parancsok, ld. 4. szakasz
 - `<>` argumentumban vesszővel elválasztott slide sorszárok és/vagy tartományok, pl. `<1,3-4>`

Frame, slide példa – kód

```
\begin{frame}{Frame, slide 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 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 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 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

Struktúra: section és társai I

- beamerben part, section, subsection, subsection léteznek
- szokás szerint „egymásba ágyazhatók”, többszintű tartalomjegyzék
 - részt vesznek a navigációban (hyperref beépítve)
 - a part szerepe különbözik: az *újrakezdi* a tartalomjegyzéket!
- stílustól függően a diában fejlécben/oldalsávban megjelenik az aktuális section és subsection
 - automatikusan nem kapnak címsort(!)
- section és társai frame környezetben *kívül* használhatók(!)

Struktúra: section és társai II

```
\section{section}
\subsection{subsection}
\begin{frame}{cím}
tartalom...
\end{frame}
\subsection{újabb subsection}
...
```

Alapok, gyors testreszabás

Beamer theme matrix

- általában elég egy presentation theme-et (és egy color theme-et) betölteni
 - kombinációik megtekinthetők a beamer theme matrix-ban (link)
- presentation theme – teljes kinézetet meghatározza
 - használat: `\usetheme{Név}`
 - városokról kapták a nevüket
- color theme – színpaletta
 - használat: `\usecolortheme{név}`
 - teljes színtémák: repülő állatokról kapták a nevüket
 - *belső* színtémák: virágokról kapták a nevüket
 - frame tartalma (mint listák, block-ok) színei
 - *külső* színtémák: vízi állatokról kapták a nevüket
 - frame kerete (mint fejléc, lábléc, oldalsáv, navigáció) színei

Megjelenés – témák I

- beamer megjelenését, stílusát *témák* adják
 - sok előre definiált téma és komponens közül válogathatunk
 - de sajátok is definiálhatók
- témák típusokba sorolhatók
- egyesek a megjelenés egy-egy **komponensét** állítják be:
 - outer theme „külső téma”: a frame keretét adja, jelen diasorban a fejléc a navigációval, más témákban oldalsáv, stb.
 - inner theme „belső téma”: a frame tartalmi elemeit formázza, mint felsorolások, block-ok:

block címe

block tartalma

Megjelenés – témák II

- color theme „színtéma”: a színpalettát állítja be, amit a többi téma használ
 - létezik belőle inner color theme – inner theme által használt színek
 - outer color theme – outer theme által használt színek
 - teljes: mindkettő színpalettáját meghatározza
- font theme „betűtéma”: használt betűtípus
 - pl. váltás serif betűtípusra
 - vagy ún. *strukturális elemek* (mint szakaszcímek, dia címe, block-ok címe) kiemelése **bold**, *italic* vagy SMALL CAPS betűkkel
- presentation theme „diavetítés téma”: teljes téma, minden fenti komponenst meghatároz
 - általában csak előre összeválogatott, összeillő csomagok a komponensekből

Megjelenés – témák III

- ha nem állítunk be semmit, nagyon egyszerű, keret és navigáció nélküli, kevés színnel dolgozó alapértelmezéseket használ
- a különböző témák kombinálhatók – sőt, komponensek felülírhatók utólag!
 - pl. presentation theme után betölthető egy inner theme és egy color theme
 - minden komponensből külön-külön az utoljára betöltött marad érvényben
 - az alapértelmezést sokszor már nem lehet visszatölteni, ha felülírtuk
- témakomponensek listáját lásd a 6. szakaszban (a kiegészítő 2. részben)
- vagy még részletesebben a beamer csomag dokumentációjában ([link](#)), a III. part-ban (15. section-től)

2 Struktúra

- Frame
- Címsorok, tartalomjegyzék
 - Cím-dia
 - Tartalomjegyzék
 - Szakaszcímek
 - Section-önként ismétlődő kód

Frame I

- frame környezet

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

- 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
 - rövid verzió: környezet helyett parancs,

```
\frame[opciók]{tartalom}
```
 - [opciók] – megszokott módon vesszővel elválasztott lista
 - plain: plain oldalstílushoz hasonló, leveszi a frame keretét (mint fejléc, oldalsáv, címsor háttere)
 - fragile: ha *törékeny* a tartalom, például ha verbatim-ot használunk a frame-ben
 - rejtélyes hibákat tud produkálni, ha elfelejtkezünk róla

Frame II

- `noframenumbering`: adott frame számozatlan
- `squeeze`: csökkenti a függőleges térközt, hogy több férjen a frame-be
- `allowframebreaks`: ha túl sok tartalom kerül bele, automatikusan szétszítja több frame-re
 - például ez a frame is automatikusan lett felosztva – a római számozást a címben az generálta
 - 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: engedélyez törést, de a \LaTeX tud optimalizálni
 - ilyenkor nem lehet több slide, nem működik együtt `overlay-jel(!)`

Cím-dia

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

Tartalomjegyzék I

- emlékeztető: part, section, subsection, subsection használható
 - navigációban, fejléc-láblécben általában csak section és subsection
 - part újramezdi a tartalomjegyzéket
- kiírás: `\tableofcontents[opciók]` parancs – az opcionális argumentum újdonság beamerben!
- elérhető opciók:
 - `sections=1`, vagy `sections={2-3,5-6}`: csak adott section(ök és tartományok) tartalomjegyzéke
 - `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

Tartalomjegyzék II

- `hideallsubsections`: csak section mélységig megy a tartalomjegyzék
 - `hideothersubsections`: csak az aktuális section subsection-jei vannak „kibontva”

Szakaszcímek

- szöveges dokumentummal ellentétben a `\section`, stb. parancsok *nem* generálnak címsorokat!
 - a tartalomjegyzékbe és stílustól függően fejlécbe/oldalsávba kerülnek be
- `\maketitle` mintájára címsor parancsok: `\partpage`, `\sectionpage`, `\subsectionpage`
 - `\maketitle`-hoz hasonlóan tegyük bele egy frame-be!

Section-önként ismétlődő kód

- `\AtBeginSection{kód}` parancs: minden `\section` parancs után beilleszti kód-ot
 - `\AtBeginPart`, `\AtBeginSubsection`,
`\AtBeginSubsubsection` variánsok
- használható például `\sectionpage` automatikus beillesztésére

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

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

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

- emlékeztető: `section`, stb. számlálók: aktuális section (vagy egyéb egység) sorszáma
- `\value{számláló}`: számláló értékének lekérdezése, ha nem kiírni szeretnénk, hanem belső változóként használni

- 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**
- színes dobozok, fontos információ kiemelésére: `block` környezet
 - kötelező argumentum: cím
 - maradhat üres is, de furcsa hibákat tud produkálni, ha hiányzik
 - `exampleblock`, `alertblock` variáns: zöld és piros
- használat:

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

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
 - 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!
 - szélesség kötelező argumentum – `minipage`-en alapul
 - itt is használhatók \LaTeX hosszok, vagy pl. `0.5\linewidth` – sorszélesség fele
 - alternatív: `\column{szélesség}` parancs, új oszlopot kezd, ami a következő `\column` parancsig vagy a `columns` környezet végéig tart

Hasábok II

- az adott columns környezetben 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

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

Slide, overlay – emlékeztető I

- frame – kb. az oldal, amivel dolgozunk
- ezen belül az egymás után megjelenített nézetek: slide-ok, számozva 1-től addig, ameddig szükséges
 - ha a frame `allowframebreaks`-t használ, nem lehet több slide!
- overlay: adott elemhez megadjuk, hogy melyik slide-(ok)on jelenik meg
- új overlay argumentum: `<>`
 - `<>` argumentumba: vesszővel elválasztott slide sorszám(ok) és/vagy tartomány(ok), ahol az adott elem megjelenik
 - `<-3>` ekvivalens `<1-3>`-mal; `<2->` jelentése: második slide-tól mindig

Slide, overlay – emlékeztető II

- beamer sok létező parancshoz definiálja az `<>` overlay argumentumot
 - megelőz minden más `[]` és `{ }` argumentumot (kivéve környezet nevét a `\begin` parancsban), például
`\includegraphics<1,3-4>[width=.5\linewidth]{image}`
ezzel az `image` fájlnevű kép az 1., 3. és 4. slide-on jelenik meg
`\begin{theorem}<2->[Tétel neve]`
ezzel 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<>`

\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<>

- `\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<>

- `\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<>

- `\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
- elrendezéshez hasznos: `overlayarea` környezet
 - `\begin{overlayarea}{szélesség}{magasság}`
 - tartalomtól független 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
 - Automatikus léptetés
 - Média
 - Animáció

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 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
 - leírást lásd: beamer dokumentáció, 14.3. szakasz, 139. oldal
 - `\transreplace`: alapértelmezés, nincs animáció
 - `\transdissolve`
 - `\transfade`
 - `\transpush`
 - `\transcover`
 - `\transfly` – irány
 - `\transglitter` – irány
 - `\transwipe` – irány
 - `\transblindshorizontal`, `\transblindvertical`

Áttűnések (transitions) III

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

Automatikus léptetés

- automatikus léptetés
- `\transduration{2}`: automatikus léptetés 2 másodperc után
- ha 0-t adunk meg, a lehető legrövidebb ideig mutatja (felvillantja) a slide-ot – „animáció” készíthető vele
- automatikus léptetés kikapcsolása: `\transduration{}`
- kaphat overlay argumentumot: `\transduration<2-4>{0}`
- overlay nélkül a frame összes slide-jára vonatkozik
- az adott frame-re vonatkozik csak

Média I

- szükséges: multimedia csomag
- videó beillesztése pdf-be
- a videó fájlt mellékelni kell(!), de ha a pdf megjelenítű le tudja játszani a videót, meg tudjuk jeleníteni a diasorban
- szintaxis:

```
\movie [opciók] {előnézet}{fájlnév}
```

- a videófájl legyen elérhető – pl. azonos mappában a pdf-fel
- **előnézet**: tetszőleges \LaTeX kód, amit a lejátszás előtt megjelenít
 - az általa elfoglalt téglalapban jelenik meg a videó, az adott képarányban
 - ezért érdemes egy előnézeti képet beilleszteni! pl.
`\includegraphics-szal, vagy poster opcióval (ld. lentebb)`

Média II

- néhány hasznos **opció**:
 - `poster`: a videó első képkockája az automatikus előnézet
 - `width=hossz`, `height=hossz`: előnézet méreteinek 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
- jó tudni: a dia továbbléptetésével automatikusan leáll a lejátszás
- a parancs hangfájl beillesztésére is alkalmas(!)
 - hang lejátszására létezik külön `\sound` parancs, de Acrobat Reader bug-ok miatt meglehetősen macerás... ld. a beamer dokumentációban, 14.2. szakasz, 137. oldal

Animáció I

- \LaTeX kód és diák gyors egymásutánja alkalmas pl. elemek mozgására, színek 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 $\text{T}_{\text{E}}\text{X}$ szintaxissal kell változókat deklarálni
 - számláló: `\newcount\szamlalo`
 - hossz: `\newdimen\hossz`
 - `\animatevalue<kezdőslide-végslide>%
\{változó}\{kezdőérték}\{végérték}`
 - az adott slide-tartományban lineárisan interpolálja a változót a kezdő- és végréték között
 - nem feltétlen kell megegyeznie az átpörgetett tartománnyal(!)
- `\begin{frame}`
`\newdimen\behuzas`
`\animatevalue<1-11>\{behuzas}\{0cm}\{5cm}`
`\transduration<1-10>\{0.1}`
`\hspace{\behuzas}helló világ!`
`\end{frame}`

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_EX 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 komponenszt 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