

Elektronikus prezentáció készítése a \LaTeX BEAMER dokumentumosztályával

Tómacs Tibor

Eszterházy Károly Főiskola
Matematikai és Informatikai Intézet
Eger

<http://www.ektf.hu/tomacs>

Utolsó módosítás: 2012. február 5.



Till Tantau
a BEAMER megalkotója
(szül.: 1975)



A BEAMER szimbóluma

Ez a leírás [Tómacs Tibor: Bevezetés a \$\LaTeX\$ dokumentumkészítő rendszerbe](#) című prezentáció folytatása, így az abban található információkat ismertnek tekintjük.

A \LaTeX rendszeren belül a beamer (magyarul: kivetítő) osztállyal foglalkozunk, mely pdf alapú elektronikus prezentációk készítésére alkalmas.

Jelen prezentáció nem törekszik a teljességre. Az eredeti angol nyelvű teljes dokumentációt megtalálja a MiKTeX doc alkönyvtárában ([beameruserguide.pdf](#)).

A pdf alapú prezentációk előnye, hogy a végeredmény minden platformon levetíthető és ugyanúgy fog működni. Így nem kell attól tartani, hogy egy idegen gépen nem fog működni vagy más jelenik meg, mint a saját gépünkön.

Till Tantau az első BEAMER verziót 2003. februárjában készítette a doktori védésének bemutatására.

2010. áprilisában a karbantartást átvette Joseph Wright és Vedran Miletic.

```
\documentclass{beamer}  
\usepackage[utf8]{inputenc}  
\usepackage[T1]{fontenc}  
\usepackage[magyar]{babel}  
\begin{document}  
...  
\end{document}
```

Ha latin2 kódolású a forrásállomány utf8 helyett, vagy nem magyar szöveget ír, akkor értelemszerűen változtasson a kódon.

A beamer osztály jellemzői

- Oldalméret: 128 mm \times 96 mm (4 : 3 arány). Az `aspectratio=169` opció esetén 160 mm \times 90 mm (16 : 9 arány).
- Alap betűméret: 11 pt. Opcióban a következő további méretek adhatók meg: `8pt` `9pt` `10pt` `12pt` `14pt` `17pt` `20pt`.
- Alap betűtípus: álló, normál, groteszk.
- 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`.

Az alap betűtípus, az új bekezdés és a főszöveg sortörésének fenti választása a prezentáció gyorsabb áttekinthetőséget szolgálja.

- A BEAMER-ben a prezentáció keretek sorozatából, a keretek pedig diák sorozatából áll. Egy keretnek címet és alcímet is adhatunk.
- Ha egy keret több diából álló diasorozatot tartalmaz, akkor az adott keretben egymásután fognak megjelenni a diasorozat tagjai.
- Ha egy keret tartalma nem fér el egy dián, akkor az széttörhető több keretre is. Az eredeti keret címe és alcíme megjelenik minden „megtört” kereten. Az ilyen megtört keretekben csak egy-egy dia szerepelhet.

- Ez egy több diából álló keret. Amit most látunk, az a keret első diája. (Vegyük észre, hogy a keretszám 9.)

- Most jelent meg a második diája a keretnek. A keretszám a következő dia megjelenésével nem változik, továbbra is 9.

- Ez látható a 3. dián.

Végül most láthatjuk a keret 4. diáját. Mint látható, az előző három dia tartalma eltűnt, vagyis itt a diákat nem egymásra helyezzük, hanem azokat cserélgetjük a keretben.

Amit most látunk, az a „Minta” című keret, amiben olyan hosszú a tartalom, hogy nem fér ki a diára. Így a keret két keretre törik. Ezért lett a „Minta” címből „Minta I”.

Hosszú szöveg. Hosszú szöveg. Hosszú szöveg. Hosszú szöveg. Hosszú szöveg. Hosszú szöveg. Hosszú szöveg. Hosszú szöveg. Hosszú szöveg. Hosszú szöveg.

Hosszú szöveg. Hosszú szöveg. Hosszú szöveg. Hosszú szöveg. Hosszú szöveg. Hosszú szöveg. Hosszú szöveg. Hosszú szöveg. Hosszú szöveg. Hosszú szöveg.

Hosszú szöveg. Hosszú szöveg. Hosszú szöveg. Hosszú szöveg. Hosszú szöveg. Hosszú szöveg. Hosszú szöveg. Hosszú szöveg. Hosszú szöveg. Hosszú szöveg.

Amit most látunk, az a „Minta” című keret megtöréséből származó második keret, illetve annak a diája. Sem az előző „Minta I”, sem ez a „Minta II” megtört keret nem tartalmazhat csak egy-egy diát.

- fej- és lábléc
- bal és jobb oldali sáv
- navigációs szimbólumok
- logó
- keret címe és alcíme
- háttér, rajta a vászon
- tartalom

Természetesen nem kell mindegyik elemet használni.

A komponensek elhelyezkedését a következő ábra szemlélteti.

Keret címe

Keret alcíme

B A L
O L D A L I
S Á V

Ez a szöveg a keret tartalma.
A háttér vászna sárga színű.
A navigációs szimbólumok szürke színűek.

Minden keretet `frame` környezetbe kell rakni.

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

vagy

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

A frame környezet opciói

t, **b**, **c** → A keret tartalma függőlegesen felülre, alulra, középre igazított.
(A beamer alapopciói esetén a frame környezetnek c az alapopciója.)

plain → A keretben a fejléc, lábléc és az oldalsávok nem jelennek meg.

shrink=kicsinyítés → Aktiválja a t opciót és a keret tartalmát kicsinyítés % mértékben kicsinyíti. A kicsinyítés alapértéke 0.

fragile → Alapesetben verbatim szöveg vagy kód nem írható a keretbe. Ezt a korlátozást oldja fel ez az opció.

squeeze → Listák függőleges extra térközök nélkül jelennek meg.

allowframebreaks=kitöltés → A kitöltés egy 0 és 1 közötti szám, alapértéke 1. A keretet kitöltés arányú telítettség után több keretre töri. A keret ezen opció esetén a `\framebreak` paranccsal közvetlenül is megtörhető. Ez az opció nem támogatja a keretben több dia használatát.

A frame környezet opciói

Ha aktiváljuk az `allowframebreaks` opciót, akkor alapesetben a keret címe után megjelenik a megtört keret sorszáma nagy római számokkal. Pl. ha a keret címe „Példa”, akkor a megjelenő címek az egymást követő kereteken:

Példa I \rightarrow Példa II \rightarrow Példa III \rightarrow ...

Ennek átállítására nézzünk néhány példát.

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

Hatása: Példa \rightarrow Példa II. \rightarrow Példa III. \rightarrow ...

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

Hatása: Példa \rightarrow Példa 2. \rightarrow Példa 3. \rightarrow ...

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

Hatása: Példa \rightarrow Példa (folyt.) \rightarrow Példa (folyt.) \rightarrow ...

Új környezet definiálása frame környezettel

Ha frame környezettel szeretnénk új környezetet definiálni, akkor két megkötést figyelembe kell venni. Ha a frame környezet tartalmaz fragile opciót, akkor például az alábbi kód nem alkalmazható:

```
\newenvironment{slide}[1]
{\begin{frame}[fragile]{#1}{}}
{\end{frame}}
```

Ehelyett az alábbi fog helyesen működni:

```
\newenvironment{slide}[1]
{\begin{frame}[fragile,environment=slide]{#1}{}}
{\end{frame}}
```

Ezután a következő két kód ekvivalens:

```
\begin{frame}[fragile]{...}{
...
\end{frame}}
```

```
\begin{slide}{...}
...
\end{slide}
```

Új környezet definiálása frame környezettel

A másik megkötés, hogy az új környezet végét definiáló részben nem szerepelhet más `\end{...}` parancs, csak `\end{frame}`. Így az alábbi kód hibát fog jelezni:

```
\newenvironment{itemizeframe}[1]
{\begin{frame}{#1}}{\begin{itemize}}
{\end{itemize}\end{frame}}
\begin{itemizeframe}{Cím}} \item ... \end{itemizeframe}
```

Ehelyett az alábbi fog helyesen működni:

```
\newcommand{\startitemizeframe}{\begin{itemize}}
\newcommand{\stopitemizeframe}{\end{itemize}}
\newenvironment{itemizeframe}[1]
{\begin{frame}{#1}}{\startitemizeframe}
{\stopitemizeframe\end{frame}}
\begin{itemizeframe}{Cím}} \item ... \end{itemizeframe}
```

A `\pause` parancs

Emlékeztetünk arra, hogy a `frame` környezet `allowframebreaks` opcióval nem támogatja a kereten belüli több diá használatát.

A keret tartalmának több dián való megjelenítésére a legegyszerűbb megoldás a `\pause` parancs használata. Vigyázat, ez a parancs nem használható az `amsmath` csomag által definiált környezetekben, mint az `align`. Például

```
\begin{frame}{Példa}{}  
  Ez látható a keret 1. diáján.\par\pause  
  Ez látható a keret 2. diáján.\par\pause  
  Ez látható a keret 3. diáján.  
\end{frame}
```

Ennek hatása:

Példa

Ez látható a keret 1. diáján.

Példa

Ez látható a keret 1. diáján.

Ez látható a keret 2. diáján.

Példa

Ez látható a keret 1. diáján.

Ez látható a keret 2. diáján.

Ez látható a keret 3. diáján.

Ennél bonyolultabb diasorozatok is létrehozhatók az úgynevezett **overlay specifikációk** használatával. A BEAMER sok standard \LaTeX parancsot kiegészít overlay specifikációval. Például listák esetén az `\item` parancsot. A használata és működése megérthető a következő példán.

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

Ennek hatása:

Példa

- ▶ 1. listaelem

Példa

- ▶ 1. listaelem
- ▶ 2. listaelem

Példa

- ▶ 3. listaelem
- ▶ 4. listaelem

Példa

- ▶ 4. listaelem

Tehát az overlay specifikációt a `<` és `>` jelek közé rakjuk. Egyszerre több overlay specifikációt is beírhatunk, amiket vesszővel kell elválasztani.

Példák:

- `<0>` → Egyetlen dián sem látható.
- `<1>` → Az 1. dián látható.
- `<1-3>` → Az 1–3. diákon látható.
- `<1-3,5-6>` → Az 1–3. és 5–6. diákon látható.
- `<1,5>` → Az 1. és 5. diákon látható.
- `<3->` → A 3. diától az utolsóig látható.
- `<-3>` → Az 1–3. diákon látható.
- `<-2,4-6,8->` → A 3. és 7. dia kivételével minden dián látható.

Ún. léptető overlay specifikációk is írhatók a számok helyére. Ezek egy `beamerpauses` nevű számlálót használnak, melynek a kezdeti értéke a keret elején 1.

Az egyik léptető overlay specifikáció a `+(szám)`, ahol a `szám` bármilyen egész érték lehet, akár negatív is. Ennek hatása:

- A `+(szám)` helyére a `beamerpauses` + `szám` értékét írja.
- Az overlay specifikációt lezáró `>` jel után a `beamerpauses` értékét 1-gyel megnöveli. (Akkor is csak 1-gyel nő az érték, ha több `+` is szerepel az overlay specifikációk között.)

`+(0)` helyett írható egyszerűen csak `+` jel is.

A következő példák mindegyikében tételezzük fel, hogy az overlay specifikáció kifejtése előtt a `beamerpauses` értéke 2.

- $\langle+(1)\rangle = \langle 3 \rangle$
- $\langle+(-1)\rangle = \langle 1 \rangle$
- $\langle+(-2)\rangle = \langle 0 \rangle$
- $\langle+(-4)\rangle = \langle -2 \rangle = \langle -+ \rangle$ (cseles!)
- $\langle+(0)\rangle = \langle + \rangle = \langle 2 \rangle$
- $\langle+-+(2)\rangle = \langle 2-4 \rangle$

Ezen példák mindegyike után (még az utolsó példa után is) a `beamerpauses` értéke 3-ra nő.

Léptető overlay specifikációk

A másik ilyen léptető specifikáció a pont. Ennek használatánál ügyeljünk arra, hogy a `beamerpauses` értéke már legalább 2 legyen. Ez a következőt csinálja:

- A pont helyére a `beamerpauses` értékénél 1-gyel kisebbet ír.
- Az overlay specifikációt lezáró `>` jel után a `beamerpauses` értéke változatlan marad.

Például a következő két kód ekvivalens:

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

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

Az overlay specifikációval ellátott parancsoknak lehet alapspecifikációjuk is. Például az `\item`-nek van alapspecifikációja, az `<1->`, így `\item` ekvivalens az `\item<1->` paranccsal.

A többi parancs alapspecifikációját az adott parancs tárgyalásánál közöljük.

Diasorozat átlátszósága

Arra is lehetőség van, hogy a keret diáin halványan megjelenjen a kerethez tartozó minden más dia erre engedélyezett tartalma. Ezt a következő módon állíthatjuk be:

```
{\setbeamercovered{transparent=20}  
\begin{frame}{Példa}{  
  \begin{itemize}  
    \item<+> 1. listaelem  
    \item<+> 2. listaelem  
    \item<+> 3. listaelem  
    \item<+> 4. listaelem  
  \end{itemize}  
\end{frame}}
```

Itt az adott keretben a diákon 20% intenzitással látható a többi dia tartalma. Ha globálisan szeretnénk ezt a hatást, akkor a `\setbeamercovered{transparent=20}` parancsot a preambulumban kell kiadni (alapérték 15).

Példa

- ▶ 1. listaelem
- ▶ 2. listaelem
- ▶ 3. listaelem
- ▶ 4. listaelem

Példa

- ▶ 1. listaelem
- ▶ 2. listaelem
- ▶ 3. listaelem
- ▶ 4. listaelem

Példa

- ▶ 1. listaelem
- ▶ 2. listaelem
- ▶ 3. listaelem
- ▶ 4. listaelem

Példa

- ▶ 1. listaelem
- ▶ 2. listaelem
- ▶ 3. listaelem
- ▶ 4. listaelem

Overlay specifikációval rendelkező alapparancsok

```
\uncover<spec>{szöveg}
```

vagy

```
\begin{uncoverenv}<spec> szöveg \end{uncoverenv}
```

Csak a megadott diákon fog megjelenni a szöveg, a többin csak foglalja a helyet, illetve a transparent értékének megfelelően látjuk. (<spec> alapértéke <1->.) Például:

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

Példa

Ez az első dia, ez pedig a második.

Példa

Ez az első dia, ez pedig a második.

Overlay specifikációval rendelkező alapparancsok

`\visible<spec>{szöveg}`

vagy

`\begin{visibleenv}<spec> szöveg \end{visibleenv}`

Ugyanaz mint az `uncover`, csak a `transparent` pozitívrá állítása erre a parancsra nem hat. (`<spec>` alapértéke `<1->`.)

Az `\invisible` parancs illetve `invisibleenv` környezet a fentihez hasonlóan használható, de a hatása a fentivel ellentétes. Erre sem hat a pozitív `transparent` érték.

Például itt az első sornak nincs hatása:

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

Példa

Ez az első dia,

Példa

ez pedig a második.

Overlay specifikációval rendelkező alapparancsok

```
\only<spec>{szöveg} vagy \only{szöveg}<spec>  
vagy  
\begin{onlyenv}<spec> szöveg \end{onlyenv}
```

Ugyanaz mint a `visible`, de még a helyet se foglalja el a `<spec>`-en kívül eső diákon. A `transparent` pozitívrá állítása erre a parancsra nem hat. (`<spec>` alapértéke `<1->`.) Például:

```
\begin{frame}{Példa}{}  
  \only<1>{Ez az első dia, }  
  \only<2>{ez pedig a második.}  
\end{frame}
```

Példa

Ez az első dia,

Példa

ez pedig a második.

Overlay specifikációval rendelkező alapparancsok

```
\alt<spec>{szöveg1}{szöveg2}
```

vagy

```
\alt{szöveg1}{szöveg2}<spec>
```

A megadott diákon fog megjelenni a szöveg1, a többin a szöveg2. A transparent pozitívra állítása erre a parancsra nem hat. (<spec> alapértéke <1->.) Például:

```
\begin{frame}{Példa}{}  
  \alt<1>{Ez az első dia,}{Ez a második dia,}\\  
  \alt<2>{nem az első.}{nem a második.}  
\end{frame}
```

Példa

Ez az első dia,
nem a második.

Példa

Ez a második dia,
nem az első.

Overlay specifikációval rendelkező alapparancsok

```
\begin{altenv}<spec>{start1}{vége1}{start2}{vége2}  
szöveg  
\end{altenv}
```

vagy

```
\begin{altenv}{start1}{vége1}{start2}{vége2}<spec>  
szöveg  
\end{altenv}
```

A megadott diákon ez fog megjelenni: start1 szöveg vége1

A többin ez fog megjelenni: start2 szöveg vége2

A transparent pozitívrá állítása erre a parancsra nem hat. (<spec> alapértéke <1->.)

Overlay specifikációval rendelkező alapparancsok

`\temporal<spec>{szöveg-előtte}{szöveg}{szöveg-utána}`

A megadott diák előtt fog megjelenni a szöveg-előtte, a megadott diákon fog megjelenni a szöveg és a megadott diák után fog megjelenni a szöveg-utána. (<spec>-nek itt nincs alapértéke, kötelező megadni.)

Például:

```
\begin{frame}{Példa}{}  
\temporal<3-4>{1--2. dia}{3--4. dia}{5., 6., \dots dia}\\  
\temporal<3,5>{1., 2., 4. dia}{3., 5. dia}{6., 7., \dots dia}\\  
\temporal<2>{1. dia}{2. dia}{3., 4., \dots dia}  
\end{frame}
```

Példa

1–2. dia

1., 2., 4. dia

1. dia

Példa

1–2. dia

1., 2., 4. dia

2. dia

Példa

3–4. dia

3., 5. dia

3., 4., ... dia

Példa

3–4. dia

1., 2., 4. dia

3., 4., ... dia

Példa

5., 6., ... dia

3., 5. dia

3., 4., ... dia

Overlay specifikációval rendelkező parancsok

A következő betűtípusra és színre vonatkozó parancsok rendelkeznek overlay specifikációval (alapspecifikáció <1->):

`\textbf`, `\textit`, `\textsl`, `\textrm`, `\textsf`, `\textcolor`, `\color`

Használatukra itt van néhány példa:

```
\begin{frame}{Példa 1}{}  
  \textbf<1>{Ez félkövér az 1. dián, a többin normál.}\\  
  \textcolor<2>{red}{Ez a 2. dián piros, a többin fekete.}\\  
  \textcolor<3>[RGB]{43,52,223}{Ez a 3. dián kék, a többin fekete.}  
\end{frame}
```

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

Példa 1

Ez félkövér az 1. dián, a többin normál.

Ez a 2. dián piros, a többin fekete.

Ez a 3. dián kék, a többin fekete.

Példa 1

Ez félkövér az 1. dián, a többin normál.

Ez a 2. dián piros, a többin fekete.

Ez a 3. dián kék, a többin fekete.

Példa 1

Ez félkövér az 1. dián, a többin normál.

Ez a 2. dián piros, a többin fekete.

Ez a 3. dián kék, a többin fekete.

Példa 2

- ▶ 1. listaelem
- ▶ 2. listaelem
- ▶ 3. listaelem

Példa 2

- ▶ 1. listaelem
- ▶ 2. listaelem
- ▶ 3. listaelem

Példa 2

- ▶ 1. listaelem
- ▶ 2. listaelem
- ▶ 3. listaelem

Overlay specifikációval rendelkező parancsok

A frame környezetnek is van overlay specifikációja (alapspecifikáció <1->). Ezzel a keret diasorozatából néhány diát kihagyhatunk. Ha overlay specifikációt használunk, akkor az opciókat utána kell írni. Például az alábbi kód segítségével a diasorozatból az 1. dia kimarad.

```
\begin{frame}<2->[plain]{Példa}{}  
  \begin{itemize}  
    \item<+-> 1  
      \begin{itemize}  
        \item<+-> a  
        \item<+-> b  
      \end{itemize}  
    \item<+-> 2  
    \item<+-> 3  
  \end{itemize}  
\end{frame}
```

Példa

- ▶ 1
 - ▶ a

Példa

- ▶ 1
 - ▶ a
 - ▶ b

Példa

- ▶ 1
 - ▶ a
 - ▶ b
- ▶ 2

Példa

- ▶ 1
 - ▶ a
 - ▶ b
- ▶ 2
- ▶ 3

Overlay specifikációval rendelkező parancsok

A keretek címe és alcíme is kiegészíthető overlay specifikációkkal (alapspecifikáció <1->):

```
\frametitle<spec>{Cím}
```

```
\framesubtitle<spec>{Alcím}
```


Tartalom dinamikus cseréje a diákon

Próbáljuk ki a következő kódot:

```
\begin{frame}{}{}  
  \only<1>{Az 1. dián\\ ez a három sor\\ látható.}  
  \only<2>{A 2. dián ez az egy sor látható.}  
\end{frame}
```

Az 1. dián
ez a három sor
látható.

A 2. dián ez az egy sor látható.

Tartalom dinamikus cseréje a diákon

Láthatjuk, hogy diaváltásnál az első sor függőleges pozíciója változik. Ez amiatt van, hogy a dián a függőleges pozicionálás középre történik és a különböző diákon a sorok száma különböző.

Ha ez valami miatt gondot okoz, akkor a megoldást az `overprint` környezet és az `\onslide` parancs együttes használata jelentheti. Az `overprint` környezetnek egy opciója van, az általa létrehozott doboz szélessége, melynek alapértéke `\textwidth`. Az `\onslide` parancsot hasonlóan használhatjuk, mint az `\item`-et.

Az előző kódot változtassuk meg erre:

```
\begin{frame}{}{}
\begin{overprint}
\onslide<1> Az 1. dián\\ ez a három sor\\ látható.
\onslide<2> A 2. dián ez az egy sor látható.
\end{overprint}
\end{frame}
```

Az 1. dián
ez a három sor
látható.

A 2. dián ez az egy sor látható.

Opció használatával is nézzük meg az eredményt:

```
\begin{frame}{}{}  
  \begin{overprint}[5cm]  
    \onslide<1> Az 1. dián ez a három sor látható.  
        Amint látható ebben a dobozban sorkizárás van.  
    \onslide<2> A 2. dián ez az egy sor látható.  
  \end{overprint}  
\end{frame}
```

Az 1. dián ez a három sor látható. Amint látható ebben a dobozban sorkizárás van.

A 2. dián ez az egy sor látható.

Tartalom dinamikus cseréje a diákon

A probléma megoldására egy másik lehetőség az overlayarea környezet használata:

```
\begin{overlayarea}{szélesség}{magasság}  
...  
\end{overlayarea}
```

Ez egy dobozt hoz létre, melynek a szélességét és magasságát a környezet paramétereiként kell megadni. Ez a doboz minden dián lefoglalja a helyet, melynek tartalma diáról diára más lehet.

Próbáljuk ki a következő kódot is:

```
\begin{frame}{}{}  
  \begin{overlayarea}{5cm}{3cm}  
    \only<1>{Az 1. dián ez a három sor látható.  
            Amint látható ebben a dobozban sorkizárás van.}  
    \only<2>{A 2. dián ez az egy sor látható.}  
  \end{overlayarea}  
\end{frame}
```

Az 1. dián ez a három sor látható. Amint látható ebben a dobozban sorkizárás van.

A 2. dián ez az egy sor látható.

Overlay specifikációval rendelkező parancsok definiálása

Saját overlay specifikációval rendelkező parancsokat illetve környezeteket is definiálhatunk. Erre használhatjuk a `\newcommand<>`, `\renewcommand<>`, `\newenvironment<>`, `\renewenvironment<>` parancsokat. Ezek pontosan úgy működnek, mint a `<>` jel nélküli verziók, csak ha a definícióban n darab paraméter van, akkor az kibővül egy $n + 1$ -edikkel, melyben az overlay specifikáció adható meg. Az alábbi példákon ez érthetővé válik.

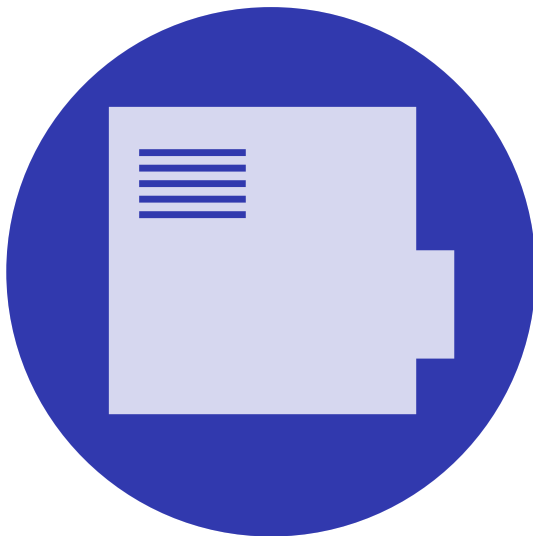
```
\newcommand<>{\makered}[1]{\color#2{blue}#1}}
```

```
\newenvironment<>{\boldornormal}
{\begin{altenv}#1{\begin{bfseries}}{\end{bfseries}}{}}
{\end{altenv}}
```

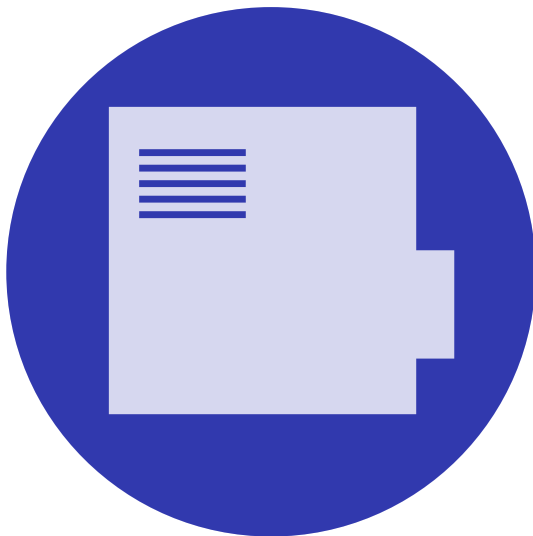
Amikor egy keret következő diájára, vagy a következő keret első diájára váltunk, akkor az eddigiekben csak annyi történt, hogy az előző dia képe „hirtelen” módosult az újra. Ezeket a váltásokat látványosabbá is tehetjük különböző effektekkel.

Sajnos nem minden pdf néző támogatja ezeket az effekteket, ezért idegen gépen nem biztos, hogy fog működni. Pl. az Adobe Reader esetén működnek, de csak akkor, ha teljes képernyős üzemmódban dolgozunk (ahogy ez egy prezentáció bemutatásánál el is várható).

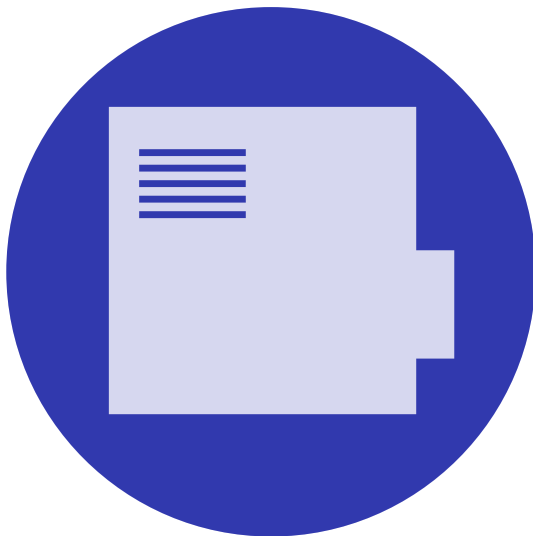
Lássunk néhány példát, de előtte váltson teljes képernyőre!



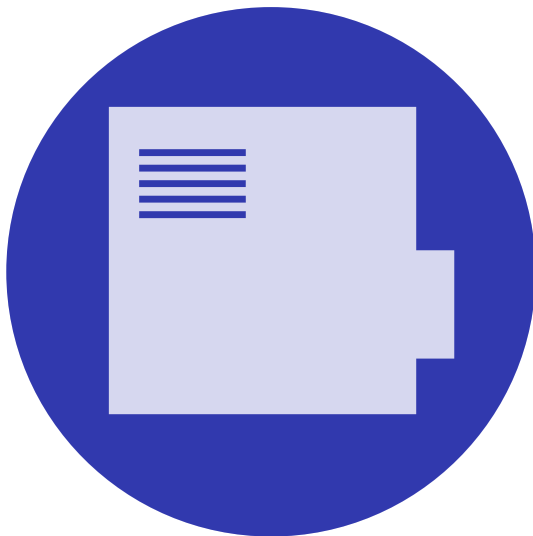




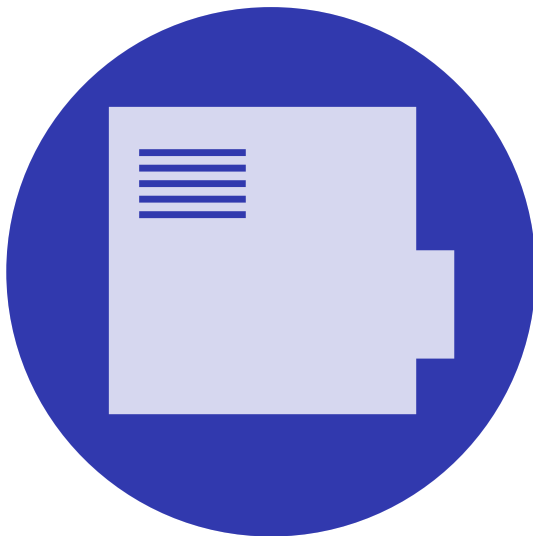




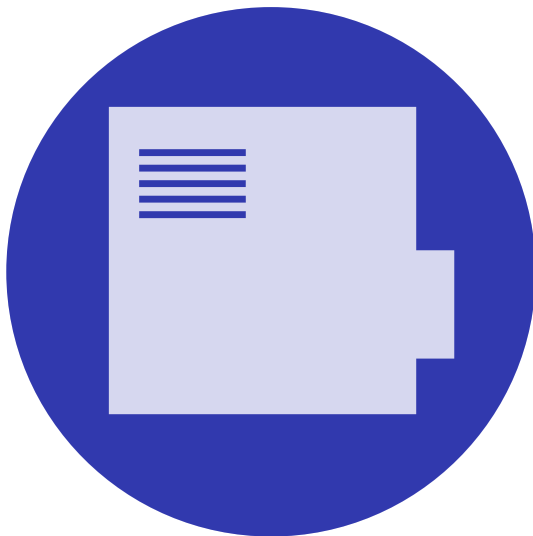












Ezeket az effekteket a következő parancsok keretbe írásával érhetjük el:

```
\transblindshorizontal<spec>[opciók]  
\transblindsvertical<spec>[opciók]  
\transboxin<spec>[opciók]  
\transboxout<spec>[opciók]  
\transdissolve<spec>[opciók]  
\transglitter<spec>[opciók]  
\transsplitverticalin<spec>[opciók]  
\transsplitverticalout<spec>[opciók]  
\transsplithorizontalin<spec>[opciók]  
\transsplithorizontalout<spec>[opciók]  
\transwipe<spec>[opciók]
```

Az overlay specifikáció alapértéke <1->.

Opciók:

`duration=idő` → Ennyi másodpercig tart az effekt (pl. `duration=1.5` esetén 1,5 másodpercig).

`direction=0,90,180,270` → Ennyi fokos szögben megy végbe az effekt (`\transglitter` esetén még lehet az érték 315 is).

Próbálja ki a következő kódot:

```
\begin{frame}{}{}  
\transwipe<1>[duration=2,direction=0]  
\transwipe<2>[duration=2,direction=180]  
\only<1>{AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA}  
\only<2>{\color{red}AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA}  
\end{frame}
```

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

Az eddigiekben diaváltás mindig gombnyomásra történt. Ez bizonyos idő megadásával automatizálható is, de ez is csak teljes képernyős üzemmódban lehetséges, a következő paranccsal:

```
\transduration<spec>{idő}
```

Az overlay specifikáció alapértéke <1->. Az idő helyére annyi másodpercet kell írni, ameddig a specifikációkkal megadott diákat látni akarjuk gomb megnyomása nélkül.

Próbáljuk ki a következő kódot:

```
\begin{frame}{}{}  
\transduration<-7>{1}  
\transduration<8-13>{0.3}  
\only<-7>{Visszaszámlálás:}  
\only<2>{5}%  
\only<3>{4}%  
\only<4>{3}%  
\only<5>{2}%  
\only<6>{1}%  
\only<7>{0}  
\only<8,10,12,14>{BUMM!!!!}  
\only<9,11,13>{\small BUMM!!!!}  
\end{frame}
```

Most nézzük meg a hatását, de előtte váltson teljes képernyőre!

Visszaszámlálás:

Visszaszámlálás: 5

Visszaszámlálás: 4

Visszaszámlálás: 3

Visszaszámlálás: 2

Visszaszámlálás: 1

Visszaszámlálás: 0

BUMM!!!!

BUMM!!!!!!

BUMM!!!!

BUMM!!!!!!

BUMM!!!!

BUMM!!!!!!

BUMM!!!!

Címoldal készítése

A prezentáció első oldala a címoldal, melynek elkészítéséhez szükséges adatokat a következő parancsokkal adhatjuk meg.

```
\title[rövid cím]{cím}  
\subtitle[rövid alcím]{alcím}  
\author[rövid név]{név}  
\institute[intézet rövid neve]{intézet}  
\date[rövid dátum]{dátum}  
\titlegraphic{\includegraphics{...}}
```

Ezután a címoldal a következőképpen készül el:

```
\begin{framed}[plain]  
  \titlepage  
\end{framed}
```

vagy `\maketitle`, ami a következővel ekvivalens:

```
\begin{framed}  
  \titlepage  
\end{framed}
```

A szerző és a cím automatikusan bekerül a pdf dokumentum leírásába is, amit a pdf nézőben a dokumentum tulajdonságainál láthatunk. De más információkat is megadhatunk a következő parancsokkal:

```
\subject{a prezentáció tárgya}  
\keywords{kulcsszavak}
```

Amikor a cím illetve a szerző megadásánál \LaTeX -parancsokat is használunk, nem csak egyszerű szöveget, akkor ezek a beírások a pdf dokumentum leírásába nem a megfelelő módon kerülnek be. Ilyenkor inkább kapcsoljuk ki ezt az automatikus beírást a beamer osztály `usepdftitle=false` opciójával.

A BEAMER-ben a szöveg tagolása az `article` osztályhoz hasonlóan

rész \rightarrow szakasz \rightarrow alszakasz \rightarrow al-alszakasz

de nincs paragrafus és alparagrafus. Sok esetben már az al-alszakasz használata is zavaró lehet, ezért inkább kerüljük a használatát.

Ha nagyon hosszú prezentációt készítünk, akkor lehet szükség a több részre való bontásra. Új részt a

```
\part[a rész rövid címe]{a rész címe}
```

parancs kereten kívüli kiadásával indíthatunk. Az opcióban megadott cím alapesetben a rész címével egyezik meg. Ez így még nem jelenik meg sehol, csak a pdf néző könyvjelzői között (a rövid cím), ha az aktiválva van, illetve a navigációs sávban (legtöbbször a rövid cím), ha az úgy van beállítva.

Ha azt akarjuk, hogy az előző parancs kiadásakor egy külön keret generálódjon a címmel, akkor használhatjuk az `\AtBeginPart{...}`, `\insertpart`, `\insertpartnumber` és `\insertromanpartnumber` parancsokat.

Például írjuk be a következőket a preambulumba:

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

Ezután a `\part{A rész címe}` parancs kiadása a következő egy diából álló keretet generálja.

I. rész

A rész címe

Új szakaszt a

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

parancs kereten kívüli kiadásával kezdhetünk. Az opcióban megadott cím alapesetben a szakasz címével egyezik meg. Ez így még nem jelenik meg sehol, csak a pdf néző könyvjelzői között, ha az aktiválva van, illetve a navigációs sávban, ha az úgy van beállítva.

Ha azt akarjuk, hogy az előző parancs kiadásakor egy külön keret generálódjon a címmel, akkor használhatjuk az `\AtBeginSection{...}`, `\insertsection` és `\insertsectionnumber` parancsokat.

Például írjuk be a következőket a preambulumba:

```
\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.

Értelemszerű változtatásokkal hasonlóan járhatunk el az alszakasz és al-alszakasz esetében is.

A rész, szakasz, alszakasz, al-alszakasz tartalmi felosztást linkek formájában megjeleníthetjük egy külön keretben is. Ha nem használtunk `\part` parancsot, akkor például a címloldal után beírhatjuk a következő kódot:

```
\begin{framed}[plain]{Tartalomjegyzék}  
  \tableofcontents  
\end{framed}
```

Tartalomjegyzék készítése

Ha használtunk `\part` parancsot, akkor az előző kód csak akkor hatásos, ha a `\part` parancs kiadása után van (ellenkezőleg egy üres diát kapunk). Ekkor a hatása nem az egész tartalomjegyzék, hanem csak az adott részé.

Ha azt akarjuk, hogy minden rész tartalma még a `\part` parancs előtt megjelenjen például a címdal után közvetlenül, akkor a következőt tehetjük (itt feltesszük, hogy két rész van):

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

Ha a tartalomjegyzéket tartalmazó keretben az első szakaszt kivéve minden szakasz címe elé egy `\pause` parancs beírásának hatását akarjuk elérni, akkor használjuk a `\tableofcontents` **pausesections** opcióját.

Ha a **pausesubsections** opciót használjuk, akkor azt a hatást éadjuk el, mintha a tartalomjegyzéket tartalmazó keretben az első alszakaszt kivéve minden alszakasz és al-alszakasz címe elé egy `\pause` parancsot írnánk.

Ha a tartalomjegyzékben nem akarunk például al-alszakasz címeket, vagy az éppen nem aktuális címeket csak halványan akarjuk megjeleníteni, akkor lehet használni a `\tableofcontents` alábbi opcióit:

- `sectionstyle=stílus`
- `subsectionstyle=stílus`
- `subsubsectionstyle=stílus`

ahol a stílus lehet: `show` (mutat), `hide` (rejt), `shaded` (halványan).

Például

```
\tableofcontents[subsubsectionstyle=hide]
```

esetén a tartalomjegyzékben nem szerepelnek az al-alszakasz címek.

Ha egy adott szakaszhoz készítünk al-tartalomjegyzéket, akkor a stílusokat kombinálhatjuk is.

`sectionstyle=stílus1/stílus2`

- `stílus1`: aktuális szakasz címének stílusa
- `stílus2`: többi szakasz címének stílusa

`subsectionstyle=stílus1/stílus2`

- `stílus1`: aktuális alszakasz címének stílusa
- `stílus2`: többi alszakasz címének stílusa

`subsectionstyle=stílus1/stílus2/stílus3`

- `stílus1`: aktuális alszakasz címének stílusa
- `stílus2`: aktuális szakasz többi alszakasz címének stílusa
- `stílus3`: többi alszakasz címének stílusa (hide esetén nem csak ezen alszakasz címek, hanem azok al-alszakasz címei sem jelennek meg a tartalomjegyzékben)

`subsectionstyle=stílus1/stílus2`

- stílus1: aktuális al-alszakasz címének stílusa
- stílus2: többi al-alszakasz címének stílusa

`subsectionstyle=stílus1/stílus2/stílus3`

- stílus1: aktuális al-alszakasz címének stílusa
- stílus2: aktuális alszakasz többi al-alszakasz címének stílusa
- stílus3: többi al-alszakasz címének stílusa

`subsectionstyle=stílus1/stílus2/stílus3/stílus4`

- stílus1: aktuális al-alszakasz címének stílusa
- stílus2: aktuális alszakasz többi al-alszakasz címének stílusa
- stílus3: aktuális szakasz többi al-alszakasz címének stílusa
- stílus4: többi al-alszakasz címének stílusa

Például a következő kódot használva az adott szakasz al-tartalomjegyzékét kapjuk:

```
\section{...}  
...  
\section{...}  
\begin{frame}[plain]{}{}  
  \tableofcontents[sectionstyle=show/hide,subsectionstyle=show/show/hide]  
\end{frame}
```

Irodalomjegyzéket pontosan úgy készíthetünk egy kereten belül, mint a nyomtatott dokumentumok esetében. Annyi csak a különbség, hogy a `\bibitem` parancsnak itt lehet adni overlay specifikációt (alap `<1->`). Például

```
\begin{frame}[plain]{Irodalomjegyzék}
  \begin{thebibliography}{12}
    \bibitem<+>{Salomaa1973} A.~Salomaa, ...
    \bibitem<+>{Dijkstra1982} E.~Dijkstra, ...
    ...
  \end{thebibliography}
\end{frame}
```

A BEAMER automatikusan betölti az `enumerate` csomagot. Ez nem kompatibilis a `paralist` csomaggal, így azt ne töltsük be. Ezért nem használhatjuk a `compactenum` és `compactitem` listakörnyezeteket sem. Ha a listákat függőleges extra térközök nélkül akarjuk, akkor a `frame` környezetet `squeeze` opcióval töltsük be.

A standard környezetek használhatók: `itemize`, `enumerate`, `description`. Ezen környezeteknek nincs, de az `\item` parancsnak van `overlay` specifikációja, melynek alapértéke `<1->`.

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

► 1. listaelem

- ▶ 1. listaelem
- ▶ 2. listaelem

- ▶ 1. listaelem
- ▶ 2. listaelem
- ▶ 3. listaelem

Ha az alap overlay specifikációt egy adott listában át akarjuk állítani pl. $\langle + - \rangle$ értékre, akkor azt az alábbi módon tehetjük meg. (Ez ekvivalens az előző kóddal.)

```
\begin{frame}{}{}  
  \begin{itemize}[\langle + - \rangle]  
    \item 1. listaelem  
    \item 2. listaelem  
    \item 3. listaelem  
  \end{itemize}  
\end{frame}
```

Ha az alap overlay specifikációt egy adott keret minden listájára át akarjuk állítani pl. $\langle + - \rangle$ értékre, akkor azt az alábbi módon tehetjük meg.

```
\begin{frame}[\langle + - \rangle]{ }{ }  
  \begin{itemize}  
    \item 1. listaelem  
    \item 2. listaelem  
  \end{itemize}  
  \begin{enumerate}  
    \item 1. listaelem  
    \item 2. listaelem  
  \end{enumerate}  
\end{frame}
```

Ha a keretnek kell pl. `t` opció, akkor az előző kódban az első sort így módosítsuk: `\begin{frame}[<+>] [t] {} {}`

Ha ezen felül még a keretnek saját overlay specifikációja is van, pl. `<2->`, akkor következőt írjuk: `\begin{frame}<2->[<+>] [t] {} {}`

Ha az `\item` parancsban egyszerre használunk overlay specifikációt és opciót, akkor azt ebben a sorrendben tesszük. Pl.: `\item<+>[--]`

Ha egy adott számozott lista adott szintjének számozását akarjuk megváltoztatni, akkor használjuk a `\begin{enumerate}[stílus]` környezetnyitást, ahol a stílus helyére a következőket írhatjuk:

1 → arab szám

i → kis római szám

I → nagy római szám

a → latin ábécé kisbetűi

A → latin ábécé nagybetűi

Ha a címkébe szöveg is kerül, akkor az abban szereplő fenti öt karaktert kapcsos zárójelek közé kell tenni. Pl. a `\begin{enumerate}[I-{\a}]` hatására a sorszámok rendre I-a, II-a, III-a, IV-a, ...

Ha az enumerate környezetnél az opció túl még az alap overlay specifikációt is be akarjuk állítani pl. $\langle + - \rangle$ értékre, akkor azt így lehet megtenni: `\begin{enumerate}[\langle + - \rangle [(1)]]`

A tömbök a keret olyan részei, amelyek saját fejrésszel és címmel rendelkeznek. Létrehozásuk:

```
\begin{frame}{}{}  
...  
\begin{block}<1>{Tömb címe}  
  Tömb szövege  
\end{block}  
...  
\end{frame}
```


Az alap overlay specifikáció <1->. Ha ezt át akarjuk állítani egy adott keret tömbjeire vonatkozólag, akkor pontosan úgy kell eljárni, mint a listák esetében.

Két speciális tömb is van, melyek alapvetően a színezésben térnek el: `alertblock` és `exampleblock` környezetek, melyek használata a fentihez hasonló.

A BEAMER-ben a tételszerű környezetek tömbként viselkednek, ahol a cím a tételszerű környezet címe. Mivel az `amsthm` csomag alpból betöltődik, ezért a `proof` környezet is használható.

A tételszerű környezeteket definiálni és használni pontosan úgy kell, mint azt taglaltuk a normál esetben illetve az `amsthm` csomag tárgyalásánál, két különbséggel.

Tömbök, tételszerű környezetek

Az egyik különbség, hogy a definiált tételszerű környezetek overlay specifikációval is használhatók (alapérték <1->). Ha ezt át akarjuk állítani egy adott keret tételszerű környezeteire vonatkozólag, akkor pontosan úgy kell eljárni, mint a listák esetében.

A másik különbség, hogy a tétel számozása alapesetben nem jelenik meg. Ez átállítható a `\setbeamertemplate{theorems}[numbered]` preambulumba írásával, de magyar nyelv esetén ekkor nem kapunk jó eredményt, mert az erre vonatkozó angol tipográfiát a `magyar.ldf` nem állítja át.

A számozás hiánya azért lett alapesetként beállítva, mert ez prezentációban jóval gyakoribb eset. Ha magyar nyelv esetén mégis szeretnénk tételszámozást, akkor használhatjuk a következő példában található megoldást:

```
\setbeamertemplate{theorems}[default]
\newtheorem{tetel}{\inserttheoremnumber. tétel}
\begin{frame}{}{}
  \begin{tetel}<1>[Cauchy tétele]
    Tétel szövege
  \end{tetel}
  \begin{tetel}<2>
    Tétel szövege
  \end{tetel}
\end{frame}
```

1. tétel (Cauchy tétele)

Tétel szövege

2. tétel

Tétel szövege

Dobozok pontosan úgy használhatók a BEAMER-ben, mint normál esetben, de itt még kiegészül két bekezdésdobozzal. Ezek ismertetése előtt pár szót a BEAMER színkezeléséről. A BEAMER előre definiál saját elnevezésű színösszeállításokat, és mi is készíthetünk ilyet. Például

```
\setbeamercolor{sajat szín}{fg=blue,bg=yellow}
```

sajat szín néven definiál egy olyan színösszeállítást, amelyben a háttér sárga, az előtér, azaz a tartalom kék.

Ennek ismeretében az egyik BEAMER bekezdésdoboz a következő:

```
\begin{beamercolorbox}[opció]{színösszeállítás}  
Doboz tartalma  
\end{beamercolorbox}
```

Az opciók:

`wd=szélesség` → doboz szélessége (alap `\textwidth`)

`dp=mélység` → doboz mélysége (alapvonal alatti rész)

`ht=magasság` → doboz magassága (alapvonal feletti rész)

`left` → doboz tartalma balra zárt

`right` → doboz tartalma jobbra zárt

`center` → doboz tartalma középre zárt

`sep=távolság` → doboz tartalma körüli extra tér nagysága

`shadow=true,false` → doboz árnyékolt, nem árnyékolt

`rounded=true,false` → doboz sarkai kerekítettek, nem kerekítettek

Például:

```
\setbeamercolor{sajat szín}{fg=blue,bg=yellow}
\begin{frame}{}{}
\begin{beamercolorbox}[wd=6cm,shadow=true,
                      rounded=true,center]{sajat szín}
  Doboz tartalma, doboz tartalma, doboz tartalma
\end{beamercolorbox}
\end{frame}
```

Doboz tartalma, doboz tartalma,
doboz tartalma

A másik BEAMER által definiált bekezdésdoboz kerekített sarkú és adhatunk neki címet egy fejrészben:

```
\begin{beamerboxesrounded}[opció]{Cím a fejrészben}  
Doboz tartalma  
\end{beamerboxesrounded}
```

Az opciók:

`width=szélesség` → doboz szélessége (alap `\textwidth`)

`shadow=true,false` → doboz árnyékolt, nem árnyékolt

`lower=színösszeállítás` → doboz tartalmának színösszeállítása

`upper=színösszeállítás` → doboz fejrészének színösszeállítása (akkor hat, ha a fejrész címe nem üres)

Például:

```
\setbeamercolor{sajat szin1}{fg=white,bg=blue}  
\setbeamercolor{sajat szin2}{fg=black,bg=yellow}  
\begin{frame}{}{}  
\begin{beamerboxesrounded}[upper=sajat szin1,  
                           lower=sajat szin2,shadow=true]{Cím}  
Doboz tartalma\\ Doboz tartalma  
\end{beamerboxesrounded}  
\end{frame}
```

Cím

Doboz tartalma

Doboz tartalma

Többhasábos terület

```
\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

Például

```
\begin{columns}[t]
\begin{column}{5cm}
1. oszlop tartalma, 1. oszlop tartalma, 1. oszlop tartalma
\end{column}
\begin{column}{5cm}
2. oszlop tartalma, 2. oszlop tartalma, 2. oszlop tartalma
\end{column}
\end{columns}
```

eredménye

1. oszlop tartalma, 1. oszlop
tartalma, 1. oszlop tartalma

2. oszlop tartalma, 2. oszlop
tartalma, 2. oszlop tartalma

A háttér színe a következő kóddal állítható be például halványsárgára:

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

Lehetőség van „árnyékos” háttér készítésére is. Például

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

esetén a következő hátteret kapjuk:

A top, middle és bottom opciók mellett használható még a midpoint opció is, amivel azt lehet megadni, hogy hol legyen a függőleges pozíciója middle-ben megadott szín középszintjének. Ez egy 0 és 1 közötti arányszám. 0 jelenti a legalsó szintet, 1 pedig a legfelsőt. Például

```
\setbeamertemplate{background canvas}[vertical shading][midpoint=0.3,  
middle=yellow]
```

eredménye:

Ha háttérképet akarunk a diáknak, akkor a preambulumba írjuk a következőt:

```
\setbeamertemplate{background canvas}  
{\includegraphics[width=\paperwidth]{kep}}
```

Ekkor a `kep` nevű képfájl minden dia hátterén megjelenik. Ha ugyanezt egyetlen keret diáira akarjuk elérni, akkor ezt kell tenni:

```
{\setbeamertemplate{background canvas}  
  {\includegraphics[width=\paperwidth]{kep}}}  
\begin{frame}  
  ...  
\end{frame}}
```

A következő kóddal a háttér rácsvonalakkal jelenik meg.

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

A step a rácsvonalak távolságát adja meg, alapértéke 5 mm. A color a rácsvonalak színét adja meg, alapértéke a

```
\setbeamercolor{background}{fg=szín}
```

módon megadott szín 10%-a.

A fenti kód eredménye:

A képek beillesztése, hasonlóan a normál esethez, történhet az `\includegraphics` paranccsal, de itt van overlay specifikációja, melynek alapértéke `<1->`. Arra ügyelni kell, hogy például

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

nem az

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

kóddal ekvivalens, hanem az

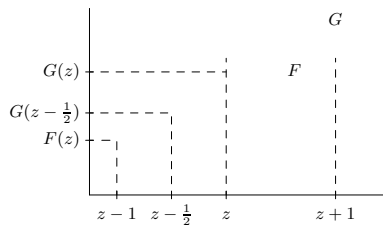
```
\only<2->{\includegraphics[width=5cm]{kep}}
```

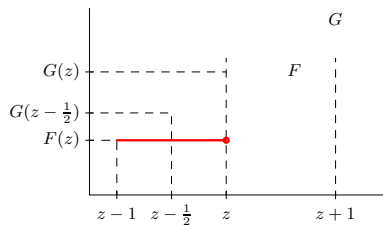
kóddal. Azaz a nem jelölt diákon nem foglalja a helyet a kép. Ez azért van így, hogy könnyebben lehessen egy képsorozatból animációt csinálni.

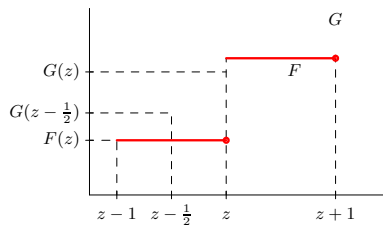
Például

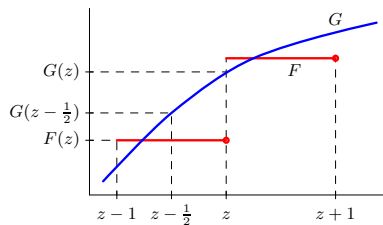
```
\includegraphics<+>[width=5cm]{figure0}  
\includegraphics<+>[width=5cm]{figure1}  
\includegraphics<+>[width=5cm]{figure2}  
\includegraphics<+>[width=5cm]{figure3}
```

eredménye









Ez már tekinthető animációnak, de ha sok képből áll, akkor a kód is sok sorból áll, ami kényelmetlen. Ez a probléma megoldható az `xmptmulti` csomaggal.

Tegyük fel, hogy az animáció a
`fig-0.jpg`, `fig-1.jpg`, `fig-2.jpg`, ..., `fig-20.jpg`
képekből áll. Ekkor a

```
\multiinclude[<+>][format=jpg,graphics={width=5cm}]{fig}
```

kód ekvivalens ezzel:

```
\includegraphics<+>[width=5cm]{fig-0}  
\includegraphics<+>[width=5cm]{fig-1}  
\includegraphics<+>[width=5cm]{fig-2}  
...  
\includegraphics<+>[width=5cm]{fig-20}
```

Ez a megoldás kódírás szempontjából már kényelmes, de a prezentáció használata még nem az, hiszen minden képváltáshoz léptetni kell a számítógépen. Ez a gond megoldható például a korábban már ismertetett `\transduration` paranccsal is, de sokkal „profibb” megoldás az `animate` csomag

```
\animategraphics[opciók]{sebesség}{alapnév}{első}{utolsó}
```

parancsa. Ez a képsorozatot videóként fogja lejátszani. Ezt a parancsot csak olyan keretben használjuk, ahol egyetlen dia van.

`sebesség` → Pozitív egész, ennyi kép/másodperc sebességgel játsza le.

`alapnév` → Pl. ha a képfájlok sorra `fig0.jpg`, `fig1.jpg`, ..., `fig20.jpg`, akkor ide `fig` kerül.

`első` → Az előző példában ide 0 kerül.

`utolsó` → Az előző példában ide 20 kerül.

Opciók:

`autoplay` → Az oldal megnyitásakor automatikusan indul a lejátszás.

`loop` → A lejátszás végén automatikusan újraindul.

`width=szélesség` → A képek szélessége.

`height=magasság` → A képek magassága.

`controls` → Lejátszó gombok jelenjenek meg.

`buttonsize=gombméret` → Lejátszó gombok mérete.

`buttonbg=szín` → Lejátszó gombok hátterének a színe.

`buttonfg=szín` → Lejátszó gombok vonalának a színe.

A szín megadása szürke skálával vagy rgb palettával történhet. Pl.

`buttonbg=0.8` vagy `buttonbg=0.36:0.08:0.88` (Fontos, hogy ha az új magyar.lidf-et használjuk, akkor a kettőspont aktívva tételét ki kell kapcsolni, különben a `buttonbg` és `buttonfg` opciók nem használhatók, csak szürke skálával.)

Például fig0.pdf, fig1.pdf, ..., fig8.pdf esetén

```
\animategraphics[autoplay,loop,controls]{4}{fig}{0}{8}
```

eredménye

Mint láttuk, animáció képsorozatból álló videóként is lejátszható. De arra is lehetőség van, hogy valódi videót játszunk le a prezentáció egy keretén belül a `multimedia` csomaggal.

Vigyázat, csak a lejátszás történik a pdf fájlban belül, a videó fájl nem épül be a pdf-be. Így vetítéskor a videó fájlt be kell másolni a pdf mellé.

A másik ami gondot jelenthet, hogy a pdf megjelenítő biztonsági kockázatnak tarthatja a videók lejátszását. Ezt külön be kell állítani vetítés előtt.

A betöltő parancs:

```
\movie[opciók]{poszter szövege}{videófájl neve}
```

Amíg nem indul el a videó, a „poszter szöveg” látható a videónak kijelölt területen, ha csak nem adtuk meg a poster opciót (lásd később). Erre kattintva indul a lejátszás. A „poszter szöveg” lehet `\includegraphics` paranccsal betöltött kép is.

Opciók:

`width=szélesség` → A videó szélessége (tanácsos megadni).

`height=magasság` → A videó magassága (tanácsos megadni).

`poster` → Amíg a videó nem indul el, nem a „poszter szöveg” látható, hanem a videó első képkockája. Erre kattintva indul a lejátszás.

`showcontrols` → Mutatja a videó alatt a navigációs sávot.

`start=idő` → A videó lejátszási kezdőpontjának megadása. Pl. `start=5s` azt jelenti, hogy a lejátszási kezdőpont az 5. másodperc.

`duration=idő` → A videóból milyen hosszú részt játszon le. Pl. `duration=25s` azt jelenti, hogy 25 másodpercnyi részt játszik le.

Például

```
\movie[width=8cm,height=6cm,showcontrols,poster]{}{video.avi}
```

Arra is lehetőség van a `label` opció és `\hyperlinkmovie` parancs együttes használatával, hogy a videónak különböző időintervallumait játszuk le egy-egy linkre kattintással. Ezeknek a linkeknek ugyanazon a dián kell lenniük, mint ahol a videó van.

Próbáljuk ki a következő kódot:

```
\begin{frame}{}{}  
\movie[label=cimke,width=8cm,height=6cm,showcontrols,poster]{}{video.avi}  
\par\medskip  
\hyperlinkmovie[start=5s,duration=10s]{cimke}{5--15\,sec}  
\par  
\hyperlinkmovie[start=20s,duration=25s]{cimke}{20--45\,sec}  
\end{frame}
```

Ha a videót nem a pdf fájlban, hanem csak egy linkre kattintva, külső alkalmazással akarjuk lejátszani, akkor nincs szükség a multimedia csomagra:

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

Ez a parancs természetesen csak akkor működik, ha a gépen az avi-hoz külső alkalmazás van rendelve (pl. Winamp).

Figyeljünk arra, hogy videó lejátszása idegen gépen nem feltétlenül fog működni. Például ha a gépre a lejátszáshoz szüksége codec nincs telepítve, hibát fog jelezni.

Lehetőség van arra, hogy a dia egy adott területét kinagyítsuk a `\framezoom` paranccsal. Például a

```
\framezoom<1><2>[border=3](1cm,2cm)(4cm,3cm)
```

parancsot a keret elejére írva a következő történik.

Az 1. dián meg fog jelenni egy 3 pixel vastag keret egy $4\text{ cm} \times 3\text{ cm}$ méretű téglalap körül, melynek a bal felső sarka 1 cm távolságra van a szövegtükör bal oldalától és 2 cm-re a szövegtükör tetejétől.

A kijelölt terület linkként működik, rákattintva a 2. diához jutunk, melyen az előbbi kijelölt részt láthatjuk a teljes dia méretére kinagyítva. A 2. dia teljes területe is linkként működik, rákattintva visszajutunk az 1. diára. (A linkek akkor fognak helyesen működni, ha teljes képernyős üzemmódban van a pdf néző.)

Például

```
\begin{frame}{}{}  
\framezoom<2><3>[border=3](1cm,0.5cm)(5cm,3.75cm)  
\framezoom<2><4>[border=3](6.2cm,0.2cm)(4.5cm,3.375cm)  
\framezoom<2><5>[border=3](2cm,5cm)(4cm,3cm)  
\includegraphics[width=\textwidth]{pic}  
\end{frame}
```

létrehoz egy 5 diából álló keretet. Az első kereten betölt egy `pic.jpg` képet, majd a másodikon kijelöli a nagyítandó részeket. Ezekre kattintva megnézhetjük a nagyítást. Nézzük működés közben, de előtte váltson teljes képernyős üzemmódba.











Próbáljuk ki a következő kódot:

```
\begin{frame}{Példa}{  
  \begin{itemize}  
    \item<+> 1. listaelem  
    \item<+> 2. listaelem  
      \begin{equation}\label{egyenlet}  
        a^2+b^2=c^2  
      \end{equation}  
    \end{itemize}  
  \end{frame}  
  
\begin{frame}  
  \eqref{egyenlet}  
\end{frame}
```

Azt fogjuk tapasztalni, hogy az `\eqref{egyenlet}` által létrehozott egyenletszám linkre kattintva nem az egyenlethez ugrik a prezentáció, azaz nem a „Példa” című keret 2. diájához, hanem az 1. diájához.

Ennek a problémának a megoldására kapott a `\label` parancs is overlay specifikációt, melynek alapértéke `<1>` (ezért ugrik a link az előző esetben az 1. diára). Így az előző kód helyesen:

```
\begin{frame}{Példa}{  
  \begin{itemize}  
    \item<+> 1. listaelem  
    \item<+> 2. listaelem  
      \begin{equation}\label<2>{egyenlet}  
        a^2+b^2=c^2  
      \end{equation}  
    \end{itemize}  
  \end{frame}  
  
\begin{frame}  
  \eqref{egyenlet}  
\end{frame}
```

Egy diasorozat adott diájára vonatkozó kereszthivatkozás

Ha egy keret adott diájára akarunk hivatkozni, akkor használjuk a `frame` környezet `label` opcióját:

```
\begin{frame}[label=cimke]{Példa}{}  
\begin{itemize}  
\item<+> 1. listaelem  
\item<+> 2. listaelem  
\end{itemize}  
\end{frame}
```

```
\begin{frame}  
\ref{cimke<2>}  
\end{frame}
```

Ekkor a `\ref{cimke<2>}` létrehoz egy keretszámot tartalmazó linket, melyre kattintva a keret 2. diájára ugrik.

A `\ref` parancs kiadásakor létrehozott link függ attól, hogy a rávonatkozó `\label` parancs hol lett kiadva. Ha a `\ref` helyett a [\hyperlink](#) parancsot használjuk, akkor a link szövegét mi adhatjuk meg. Próbáljuk ki a következő kódot:

```
\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}
```

Linknek nem csak szöveg, hanem gomb is megadható.

`\beamerbutton{gomb szövege}` → 

`\beamergotobutton{gomb szövege}` → 

`\beamerskipbutton{gomb szövege}` → 

`\beamerreturnbutton{gomb szövege}` → 


Példaként az előző kódot egészítsük ki a következővel:

```
\begin{frame}
\hyperlink{cimke<2>}{\beamerreturnbutton{Előző keret 2. diája}}
\end{frame}
```


A nyomógombok szimbólumait az

```
\insertgotosymbol  
\insertskipsymbol  
\insertreturnsymbol
```

parancsok átdefiniálásával változtathatjuk meg. A gomb színeit és a szöveg betűtípusát is átállíthatjuk. Például a következő beállítás után a

`\beamergotobutton{gomb szövege}` eredménye: 

```
\renewcommand{\insertgotosymbol}{\${\ggg$}  
\setbeamercolor{button}{fg=black,bg=yellow}  
\setbeamercolor{button border}{fg=red}  
\setbeamerfont{button}{family=\rmfamily,shape=\itshape,series=\bfseries}
```

Ha egy keretet label opcióval töltöttünk be, akkor lehetőség van a keret tartalmát egy másik ponton is megjeleníteni, esetleg más overlay specifikációval, más opcióval. Erre a következő példa ad útmutatást:

```
\begin{frame}[<+>][label=cimke]{Példa}{}  
  \begin{itemize}  
    \item 1. listaelem  
    \item 2. listaelem  
    \item 3. listaelem  
  \end{itemize}  
\end{frame}  
  
\againframe<2->[<+>][t]{cimke}
```

Példa

- ▶ 1. listaelem

Példa

- ▶ 2. listaelem

Példa

- ▶ 3. listaelem

Példa

- ▶ 1. listaelem
- ▶ 2. listaelem

Példa

- ▶ 1. listaelem
- ▶ 2. listaelem
- ▶ 3. listaelem

A nyomtatott illetve elektronikus publikációk szerkesztésénél a tipográfiai munka jelentős részét a \LaTeX -re bíztuk. Ez itt is megoldható, ugyanis a BEAMER szerzői rengeteg ún. témát készítettek, melyek mindegyike egy-egy tipográfiai beállítást, stílust jelent.

A témákat célszerű valamilyen szempont szerint csoportosítani, hogy könnyebb legyen a választás. Ehhez először a prezentáció elemeit csoportosítjuk.

Belső elemek

- címloldal
- listák
- tömbök
- tételszerű környezetek
- képek, táblázatok
- lábjegyzetek
- irodalomjegyzék

Külső elemek

- fej- és lábléc
- oldalsávok
- logó
- keret címe

Belső szerkezeti témák

Belső elemekből mi jelenjen meg és milyen geometriával.

Betöltése: `\useoutertheme[opciók]{név}`

Külső szerkezeti témák

Külső elemekből mi jelenjen meg és milyen geometriával.

Betöltése: `\useinnertheme[opciók]{név}`

Színtémák

Belső és külső elemek színvilága.

Betöltése: `\usecolortheme[opciók]{név}`

Betűtípus témák

Belső és külső elemek betűtípusai.

Betöltése: `\usefonttheme[opciók]{név}`

Teljes témák

Szerkezeti, szín- és betűtípus témák összehangolása.

Betöltése: `\usetheme[opciók]{név}`

Célszerű először egy teljes témát választani. Ha ebben valamilyen részlet nem tetszik, akkor alkalmazhatunk még valamilyen belső vagy külső szerkezeti, szín- vagy betűtípus témát is.

Teljes témák

\usetheme[opciók]{név} (Csak preambulumban adható ki!)

név	opciók
<i>Oldalsáv nélkül</i>	
Bergen	nincs
Boadilla	<code>secheader</code> → fejléc bekapcsolása
Madrid	<code>secheader</code> → fejléc bekapcsolása
AnnArbor	nincs
CambridgeUS	nincs
Pittsburgh	nincs
Rochester	<code>height=magasság</code> → keretcím magassága
<i>Fa navigáció</i>	
Antibes	nincs
JuanLesPins	nincs
Montpellier	nincs

Teljes témák

`\usetheme[opciók]{név}` (Csak preambulumban adható ki!)

név	opciók
<i>Oldalsávval</i>	
Berkeley	<code>hideallsubsections</code> → oldalsávon nincs alszakasz cím <code>hideothersubsections</code> → oldalsávon csak az aktuális alszakasz címe van <code>left</code> → oldalsáv bal oldalon <code>right</code> → oldalsáv jobb oldalon <code>width=szélesség</code> → oldalsáv szélessége
PaloAlto	lásd Berkeley
Goettingen	lásd Berkeley
Marburg	lásd Berkeley
Hannover	lásd Berkeley, de nincs left és right

Teljes témák

\usetheme[opciók]{név} (Csak preambulumban adható ki!)

név	opciók
<i>Mini keret a fejlécben</i>	
Berlin	<code>compress</code> → egysoros a mini keret
Ilmenau	lásd Berlin
Dresden	lásd Berlin
Darmstadt	nincs
Frankfurt	nincs
Singapore	nincs
Szeged	nincs
<i>Fejlécben az aktuális szakasz és alszakasz címe</i>	
Copenhagen	nincs
Luebeck	nincs
Malmoe	nincs
Warsaw	nincs

Belső témák

`\useinnertheme[opciók]{név}` (Csak preambulumban adható ki!)

név	opciók
<code>circles</code>	nincs
<code>rectangles</code>	nincs
<code>rounded</code>	<code>shadow</code> → árnyékolt tömbök
<code>inmargin</code>	nincs

Külső témák

`\useoutertheme[opciók]{név}` (Csak preambulumban adható ki!)

név	opciók
<code>infolines</code>	nincs
<code>miniframes</code>	<code>footline=authorinstitute</code> → láblécben: szerző, intézet <code>footline=authortitle</code> → láblécben: szerző, cím <code>footline=institutetitle</code> → láblécben: intézet, cím <code>footline=authorinstitutetitle</code> → láblécben: szerző, intézet, cím <code>subsection=true,false</code> → alszakasz címet mutassa, ne mutassa a fejlécben
<code>smoothbars</code>	<code>subsection=true,false</code> → alszakasz címet mutassa, ne mutassa a fejlécben

Külső témák

\useoutertheme[opciók]{név} (Csak preambulumban adható ki!)

név	opciók
sidebar	<code>hideallsubsections</code> → tartalomban nincs alszakasz cím <code>hideothersubsections</code> → tartalomban csak az aktuális alszakasz címe van <code>left</code> → oldalsáv bal oldalon <code>right</code> → oldalsáv jobb oldalon <code>width=szélesség</code> → oldalsáv szélessége <code>height=magasság</code> → keretcím magassága
split	nincs
shadow	nincs
tree	<code>hooks</code> → „faágak” behúzása
smoothtree	nincs

Színtémák

`\usecolortheme[opciók]{név}` (Csak preambulumban adható ki!)

név	opciók
<code>structure</code>	<code>named=színnév</code> → struktúrális elemek előterének színe <code>RGB={RGB színkód}</code> → struktúrális elemek előterének színe
<code>sidebartab</code>	nincs

Teljes színtémák

<code>albatross</code>	<code>overlystylish</code>
<code>beetle</code>	nincs
<code>crane</code>	nincs
<code>dove</code>	nincs
<code>fly</code>	nincs
<code>seagull</code>	nincs
<code>wolverine</code>	nincs
<code>beaver</code>	nincs

Színtémák

`\usecolortheme[opciók]{név}` (Csak preambulumban adható ki!)

név	opciók
<i>Belső elemek színtémái</i>	
<code>lily</code>	nincs
<code>orchid</code>	nincs
<code>rose</code>	nincs
<i>Külső elemek színtémái</i>	
<code>whale</code>	nincs
<code>seahorse</code>	nincs
<code>dolphin</code>	nincs

Betűtípus témák

`\usefonttheme[opciók]{név}` (Csak preambulumban adható ki!)

név	opciók
<code>serif</code>	<code>stillsansserifmath</code> <code>stillsansserifsmall</code> <code>stillsansseriflarge</code> <code>stillsansseriftext</code>
<code>structurebold</code>	<code>onlysmall</code> <code>onlylarge</code>
<code>structureitalicserif</code>	lásd <code>structurebold</code>
<code>structuresmallcapserif</code>	lásd <code>structurebold</code>

`\setbeamercolor{sajat szín}{fg=szín}` egy saját szín nevű színösszeállítást definiál, melyben szín a neve az előtérszínnek.

`\setbeamercolor{sajat szín}{bg=szín}` egy saját szín nevű színösszeállítást definiál, melyben szín a neve a háttérszínnek.

`\setbeamercolor{sajat szín}{fg=szín1,bg=szín2}` egy saját szín nevű színösszeállítást definiál, melyben szín1 a neve az előtérszínnek és szín2 a háttérszínnek. Ez a kód ekvivalens a következővel:

```
\setbeamercolor{sajat szín}{fg=szín1}  
\setbeamercolor{sajat szín}{bg=szín2}
```

Színösszeállítások definiálása

Egy színösszeállítás definiálható korábban definiált színösszeállítással is.

Például

```
\setbeamercolor{sajat szín}{parent=beamerszín}
```

után a saját szín előtérshíne megegyezik a beamerszín előtérshínevel, ha az definiált, illetve a saját szín háttérshíne megegyezik a beamerszín háttérshínevel, ha az definiált.

Ha a beamerszín1 színösszeállításnak pl. az előtérshíne definiált, de a háttérshíne nem, illetve a beamerszín2 színösszeállításnak az előtérshíne nem definiált, de a háttérshíne igen, akkor

```
\setbeamercolor{sajat szín}{parent={beamerszín1,beamerszín2}}
```

után a saját szín előtérshíne a beamerszín1 előtérshínevel, illetve a háttérshíne a beamerszín2 háttérshínevel fog megegyezni.

A parent-ben megadott színösszeállítást bármikor átdefiniálva a saját szín is vele változik.

Színösszeállítások definiálása

A `\usebeamercolor{sajat szín}` parancs kiadása után a saját szín nevű színösszeállítás háttérszíne `sajat szín.bg` illetve az előtér színe `sajat szín.fg` néven lesz elérhető (részletesebben lásd később). Ezen nevek felhasználásával más színösszeállításokat is készíthetünk:

Például

```
\setbeamercolor{sajat szín halvány}{fg=sajat szín.fg!40,bg=sajat szín.bg!40}
```

Ezen kód esetén, ha a saját szín később változni fog, ez a saját szín halvány-ra már nem fog kihatni.

Ha azt akarjuk, hogy a saját szín halvány együtt változzon a saját szín-nel, akkor a következő kódot használjuk:

```
\setbeamercolor{sajat szín halvány}{use=sajat szín,fg=sajat szín.fg!40,  
                                     bg=sajat szín.bg!40}
```

A `use`-ban több színösszeállítás is felsorolható. Például:

```
\setbeamercolor{sajat szín}{use={beamerszín1,beamerszín2},  
                             fg=beamerszín1.fg!40!beamerszín2}
```

A `\setbeamercolor` parancs helyett használható a `\setbeamercolor*` is, amely pontosan ugyanúgy működik, mint csillag nélkül, de először a színösszeállítás elő- és háttérszínét üresre állítja. Például

```
\setbeamercolor{sajat szín}{bg=blue}  
\setbeamercolor*{sajat szín}{fg=red}
```

után a `sajat szín` háttérszíne már nincs definiálva, azaz a korábbi kékre állítása törlődik.

Színösszeállítások használata

A `\usebeamercolor` parancs kiadásával válnak elérhetővé a színösszeállításban megadott színek. Például a

```
\setbeamercolor{sajat szin}{bg=blue,fg=red}  
\usebeamercolor{sajat szin}
```

után a háttérszín (blue) a következő neveken lesz elérhető:

```
bg  
sajat szin.bg
```

Míg az előtérszín (red) a következő neveken lesz elérhető:

```
fg  
sajat szin.fg
```

A `bg` és `fg` nevek természetesen újabb `\usebeamercolor` parancs kiadásával átdefiniálódnak, míg a `sajat szin.bg` és `sajat szin.fg` nem, azok csak a színösszeállítás átdefiniálásával változnak.

Az előző kód definiál még további két színnevet:

`parent.fg` → Ezzel elérhetővé válik a korábbi fg szín.

`parent.bg` → Ezzel elérhetővé válik a korábbi bg szín.

Például

```
\setbeamercolor{sajat szin 1}{bg=blue,fg=red}  
\usebeamercolor{sajat szin 1}  
\setbeamercolor{sajat szin 2}{bg=yellow,fg=green}  
\usebeamercolor{sajat szin 2}  
\textcolor{parent.fg}{Ez a szöveg piros.}  
\textcolor{parent.bg}{Ez a szöveg kék.}
```

hatása: **Ez a szöveg piros.** Ez a szöveg kék.

Ha egy színösszeállításban nincs definiálva az előtérszín, akkor annak használatakor a korábbi fg színt veszi fel. Például

```
\setbeamercolor{sajat szin 1}{bg=blue,fg=red}  
\usebeamercolor{sajat szin 1}  
\setbeamercolor*{sajat szin 2}{bg=yellow}  
\usebeamercolor{sajat szin 2}  
\textcolor{sajat szin 2.fg}{Ez a szöveg piros.}  
\textcolor{fg}{Ez is,}  
\textcolor{parent.fg}{és ez is.}
```

hatása: Ez a szöveg piros. Ez is, és ez is.

Ha egy színösszeállításban nincs definiálva a háttérszín, akkor annak használatakor a korábbi bg színt veszi fel. Például

```
\setbeamercolor{sajat szin 1}{bg=blue,fg=red}  
\usebeamercolor{sajat szin 1}  
\setbeamercolor*{sajat szin 2}{fg=yellow}  
\usebeamercolor{sajat szin 2}  
\textcolor{sajat szin 2.bg}{Ez a szöveg kék.}  
\textcolor{bg}{Ez is,}  
\textcolor{parent.bg}{és ez is.}
```

hatása: Ez a szöveg piros. Ez is, és ez is.

Hogy egy színösszeállításban van-e definiálva hát- illetve előtérszín, le is ellenőrizhetjük:

`\ifbeamercoloreempty{sajat szín}{kód1}{kód2}` → Ha a saját szín nevű színösszeállítás egyetlen eleme sincs definiálva, akkor a kód1 aktiválódik, különben pedig a kód2.

`\ifbeamercoloreempty[bg]{sajat szín}{kód1}{kód2}` → Ha a saját szín nevű színösszeállítás háttérszíne nincs definiálva, akkor a kód1 aktiválódik, különben pedig a kód2.

`\ifbeamercoloreempty[fg]{sajat szín}{kód1}{kód2}` → Ha a saját szín nevű színösszeállítás előtérszíne nincs definiálva, akkor a kód1 aktiválódik, különben pedig a kód2.

A `\usebeamercolor` parancsnak van két opciója is.

```
\usebeamercolor[fg]{sajat szin}
```

ekvivalens ezzel:

```
\usebeamercolor{sajat szin}\color{fg}
```

```
\usebeamercolor[bg]{sajat szin}
```

ekvivalens ezzel:

```
\usebeamercolor{sajat szin}\color{bg}
```

A `\usebeamercolor` parancs helyett használható a `\usebeamercolor*` is, amely pontosan úgy működik, mint a csillag nélkül, de először kiadja a `\usebeamercolor[fg]{normal text}` parancsot. Azaz például

```
\usebeamercolor*[bg]{sajat szin}
```

ekvivalens ezzel:

```
\usebeamercolor[fg]{normal text}
```

```
\usebeamercolor[bg]{sajat szin}
```

A `normal text` egy előre definiált színösszeállítás.

Betűbeállítások definiálása

Betűbeállításokat a

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

paranccsal definiálhatunk. 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`

Például

```
\setbeamerfont{sajat betu}{size=\large,series=\bfseries}
```

ami ekvivalens a következővel:

```
\setbeamerfont{sajat betu}{size=\large}  
\setbeamerfont{sajat betu}{series=\bfseries}
```

Betűbeállítások definiálása

Már korábban definiált betűbeállításokkal definiálhatunk újakat is. Például

```
\setbeamerfont{betu A}{size=\large}  
\setbeamerfont{betu B}{parent=betu A,shape=\slshape}
```

ekvivalens ezzel:

```
\setbeamerfont{betu B}{size=\large,shape=\slshape}
```

Vagy

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

ekvivalens ezzel:

```
\setbeamerfont{betu C}{size=\large,series=\bfseries,shape=\slshape}
```

A `\setbeamerfont*` parancs pontosan úgy működik, mint a csillag nélküli verzió, de előtte töröl minden korábbi beállítást az adott nevű betűbeállításra vonatkozólag. Például

```
\setbeamerfont{sajat betu}{size=\large,series=\bfseries}  
\setbeamerfont*{sajat betu}{shape=\slshape}
```

után az első sor hatástalanná válik.

A definiált betűbeállítások használata

Például

```
\setbeamerfont{sajat betu}{size=\large,series=\bfseries,shape=\slshape}  
{\usebeamerfont{sajat betu} Szöveg}
```

hatása: ***Szöveg***

A `\usebeamerfont*` parancs pontosan úgy működik, mint a csillag nélküli verzió, de előtte kiadja a `\normalfont\normalsize` parancsokat, azaz például

```
\usebeamerfont*{sajat betu}
```

ekvivalens a következővel:

```
\normalfont\normalsize\usebeamerfont{sajat betu}
```

Vízszintes navigáció (szakaszok)

Példa:

Bevezetés Keretek Egy keretben több dia Diaváltás A prezentáció tagolása Tartalmi elemek Témák Stíluselemek

Ha ilyen navigációt tartalmaz a betöltött téma fej- vagy lábléce, akkor a színeket így lehet beállítani:

```
\setbeamercolor{section in head/foot}{fg=blue,bg=white}
```

Ekkor az aktuális szakasz címének színe blue, míg a nem aktuális szakaszok címeinek színe blue!50!white (= blue!50).

Ha az 50 helyére például azt akarjuk, hogy 30 kerüljön, akkor használjuk ezt a kódot:

```
\setbeamertemplate{section in head/foot shaded}[default][30]
```

A betűtípust például félkövérre az alábbi módon állíthatjuk be:

```
\setbeamerfont{section in head/foot}{series=\bfseries}
```

Minden ezektől eltérő beállítás csak a sablon (template) átírásával lehetséges. Például

1. Bevezetés 2. Keretek 3. Egy keretben több dia 4. Diaváltás 5. A prezentáció tagolása 6. Tartalmi elemek 7. Témák 8. Stíluselemek

esetén a beállítás kódja:

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

Természetesen ezen kód után, a „direkt” beállítások miatt, az előző szín- illetve betűbeállítási lehetőségek megszűnnek.

Vízszintes navigáció (szakaszok)

Ha a használt téma nem tartalmaz ilyen navigációt, akkor mi is beszúrhatjuk adott helyre az alábbi kóddal:

```
\insertsectionnavigationhorizontal{szélesség}{kód1}{kód2}
```

A navigáció ekkor szélesség széles dobozban lesz. A kód1 a szakaszcímsor elején, a kód2 pedig a végén aktiválódik. Például

```
\insertsectionnavigationhorizontal{\textwidth}{}{\hspace{\stretch{3}}}
```

esetén a szakaszcímek a `\textwidth` széles dobozban balra zártan helyezkednek el.

Ha csak például 14 cm széles dobozban fér el a navigáció, ami szélesebb a szövegtükrőnél, akkor használhatjuk a következő kódot:

```
\resizebox{\textwidth}{!}{\mbox{\insertsectionnavigationhorizontal{14cm}{}{}}}
```

Pontosan azt kell tenni, mint a „Vízszintes navigáció (szakaszok)” esetében, csak a „section” helyére „subsection”, illetve „szakasz” helyére „alszakasz” írandó.

Pontosan azt kell tenni, mint a „Vízszintes navigáció (szakaszok)” esetében, csak a „section” helyére „subsubsection”, illetve „szakasz” helyére „al-alszakasz” írandó.

Vízszintes navigáció (szakaszok, keretek)

Példa:

Bevezetés	Keretek	Egy keretben több dia	Diaváltás	A prezentáció tagolása	Tartalmi elemek	Témák	Stíluselemek
○ ○ ○	○○ ○○ ○○ ○○	○○ ○○○ ○○○ ○ ○○ ○○○○○○○○○○ ○○○○○ ○○○○○○○ ○	○○○ ○○○	○○ ○○○○○ ○○○○○○○ ○	○○○○○○○ ○○ ○○○○ ○○ ○○○○○○○ ○ ○○○○○○○ ○ ○○ ○○○ ○○ ○ ○○○	○○○ ○○ ○○ ○○ ○	○○ ○○ ○○○●○○ ○○

A kereteket jelképező kis körök alszakaszonként külön sorban vannak. Ha ilyen navigációt tartalmaz a betöltött téma fej- vagy lábléce, akkor a beállítási lehetőségek a következők:

Vízszintes navigáció (szakaszok, keretek)

A szakaszcímek beállításához pontosan azt kell tenni, mint a „Vízszintes navigáció (szakaszok)” esetében.

Ha a kereteket jelképező kis körök helyett négyzeteket szeretnénk, akkor használjuk a következő kódot:

```
\setbeamertemplate{mini frames}[box]
```

Ha pedig kis függőleges vonalakat, akkor ezt:

```
\setbeamertemplate{mini frames}[tick]
```

Visszatérni a körökre így lehet:

```
\setbeamertemplate{mini frames}[default]
```

Vízszintes navigáció (szakaszok, keretek)

A keretek jelének színeit így lehet beállítani:

```
\setbeamercolor{mini frame}{fg=blue,bg=white}
```

Ekkor az aktuális alszakasz sorában a keretek jeleinek színe blue, míg a nem aktuális alszakaszok sorában a keretek jeleinek színe blue!50!white. Ha az 50 helyére például azt akarjuk, hogy 30 kerüljön, akkor használjuk ezt a kódot:

```
\setbeamertemplate{mini frame in other subsection}[default][30]
```

A keretek jeleinek távolsága az alábbi módon állítható:

```
\setbeamersize{mini frame size=2mm,mini frame offset=1mm}
```

Ekkor a jelek bal oldalai 2 mm-re, míg a sorok 1 mm-re lesznek egymástól.

Ezekről eltérő beállítás a sablon (template) átírásával lehetséges. Például

1. Bevezetés	2. Keretek	3. Egy keretben több dia	4. Diaváltás	5. A prezentáció tagolása	6. Tartalmi elemek	7. Témák	8. Stíluselemek
◇	◇◇	◇◇	◇◇◇◇	◇◇	◇◇◇◇◇◇◇	◇◇◇	◇◇
◇	◇	◇◇◇	◇◇◇	◇◇◇◇◇◇◇	◇◇	◇	◇◇
◇	◇◇	◇◇◇		◇◇◇◇◇◇◇◇	◇◇◇◇	◇◇	◇◇◇◇◆◇◇◇
	◇◇	◇		◇	◇◇	◇◇	◇◇
		◇◇			◇◇◇◇◇◇◇◇	◇	
		◇◇◇◇◇◇◇◇◇◇◇◇			◇		
		◇◇◇◇◇◇◇			◇◇◇◇◇◇◇		
		◇◇◇◇◇◇◇◇◇			◇		
		◇			◇◇◇		
					◇◇◇◇		
					◇		
					◇◇◇		

kódja:

Vízszintes navigáció (szakaszok, keretek)

```
\setbeamertemplate{section in head/foot}{\tiny\color{black}  
                                         \insertsectionheadnumber. \insertsectionhead}  
\setbeamertemplate{section in head/foot shaded}{\tiny\color{gray}  
                                         \insertsectionheadnumber. \insertsectionhead}  
  
\setbeamersize{mini frame size=2mm,mini frame offset=1mm}  
  
\setbeamertemplate{mini frame}{\tiny\color{black}$\blacklozenge$}  
\setbeamertemplate{mini frame in current subsection}{\tiny\color{blue}$\lozenge$}  
\setbeamertemplate{mini frame in other subsection}{\tiny\color{gray}$\lozenge$}
```

Ha a használt téma nem tartalmaz ilyen navigációt, akkor mi is beszúrhatjuk adott helyre az

```
\insertnavigation{szélesség}
```

paranccsal.

Ha például csak 14 cm széles dobozban fér el a navigáció, ami szélesebb a szövegtükörnél, akkor használhatjuk a következő kódot:

```
\resizebox{\textwidth}{!}{\mbox{\insertnavigation{14cm}}}
```

Függőleges navigáció (szakaszok)

- Bevezetés
- Keretek
- Egy keretben több dia
- Diaváltás
- A prezentáció tagolása
- Tartalmi elemek
- Témák
- Stíluselemek

Ha a navigáció a fej- vagy láblécben van, akkor a szakaszcímek beállításához pontosan azt kell tenni, mint a „**Vízszintes navigáció (szakaszok)**” esetében. Ha az oldalsávon van a navigáció, akkor a beállítás csak annyiban változik, hogy

„section in head/foot” helyett „**section in sidebar**”, illetve „section in head/foot shaded” helyett „**section in sidebar shaded**” írandó.

Ha a használt téma nem tartalmaz ilyen navigációt, akkor mi is beszúrhatjuk adott helyre az `\insertsectionnavigation{szélesség}` paranccsal.

Ennek formázására pontosan azt kell tenni, mint az előbb, csak a „section” helyére „subsection” írandó.

`\insertsection` → Aktuális szakasz címe, ami link a szakasz elejére.

`\insertsubsection` → Aktuális alszakasz címe, ami link az alszakasz elejére.

`\insertsubsubsection` → Aktuális al-alszakasz címe, ami link az al-alszakasz elejére.

A FOLYTATÁS SZERKESZTÉS ALATT!