

Alapok

Alkotó

Beamer tulajdonsága

Dokumentumosztály szerkezete

Beamer opciók

Egyszerű Szöveg megjelenése

Keretek

Blokkok

Listák

Kitakarások (overlay)

A Beamer alkotója



Till Tantau

2010. áprilisában a munkát Joseph Wright és Vedran Miletic vette át.

Beamer tulajdonsága

Előny:

- ▶ A Latex parancsokat ismeri.
- ▶ pdf alapú prezentációkat lehet készíteni, amit bármely operációs rendszer alatt meg lehet jeleníteni.
- ▶ Dinamikus effekteket és kitakarásokat lehet létrehozni.
- ▶ A diák kinézetét rugalmasan lehet változtatni, nagyon sok előre definiált stílus (témák) közül választhatunk.

Hátrány:

- ▶ Ismerni kell a Latex-et.
- ▶ A diák elkészítése munkaigényesebb a PowerPoint-nál.

További információk

- ▶ `https://www.hartwork.org/beamer-theme-matrix/`
- ▶ `http://deic.uab.es/~iblanes/beamer_gallery/`
- ▶ `http://tug.ctan.org/macros/latex/contrib/beamer/doc/beameruserguide.pdf`

A fájl felépítése

```
\documentclass{beamer}  
\usetheme{CambridgeUS}  
  
.....  
\begin{document}  
\begin{frame}{Fólia 1}  
.....  
\end{frame}  
  
\begin{frame}{Fólia 2}  
.....  
\end{frame}  
  
\end{document}
```

Magyar nyelvű dokumentum készítés

```
\documentclass{beamer}
....
\usepackage[latin2]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[magyar]{babel}
\def\magyarOptions{hyphenation=huhyphn}
.....
\begin{document}
....
\end{document}
```

utf8 kódolás esetén `\usepackage[utf8]{inputenc}`
parancsot kell használni.

Beamer opciók

- ▶ Alapértelmezett oldalméret $128mm \times 96mm$, ami (4 : 3) arányt jelent. Akinek a monitora (16 : 9) arányú, az használja a

`\documentclass[aspectratio=169]{beamer}`
paramétert.

- ▶ Font méret 11pt. A következő betűméretek adhatók meg:
8pt 9pt 10pt 12pt 14pt 17pt 20pt

pl: `\documentclass[14pt]{beamer}`

- ▶ Betűtípus: álló, normál, groteszk. Megváltoztatása:

`\usefonttheme{serif}`

`\usepackage{times}`

- ▶ Keret tartalma középre van igazítva. Lehetséges opciók: t (fent), b (lent)

pl: `\documentclass[t]{beamer}`

- ▶ Automatikusan betöltődő csomagok: graphicx, amsthm, xcolor, enumerate, hyperref.

Normál szöveg megjelenése

A szöveg balra zárt, így nincsenek szóelválasztások.

Ez egy hosszú szöveg, amely demonstrálja, hogy minden bekezdés a margótól azonos távolságra kezdődik, és nincs elválasztás. A sorok vége viszont összevissza végződik.

Persze ezt egy kis trükkel meg lehet szüntetni.

Sorkizárt szöveg elválasztással

```
\bgroup  
\begin{frame}  
\setlength{\rightskip}{20pt}  
  
\begin{frame}[fragile]{Szöveg elválasztása}  
babla ...  
\end{frame}  
\egroup
```

Szöveg elválasztása

A szöveg balra zárt, így nincsenek szóelválasztások.

Ez egy hosszú szöveg, amely demonstrálja, hogy minden bekezdés a margótól azonos távolságra kezdődik, és nincs elválasztás. A sorok vége viszont összevissza végződik. Persze ezt a egy kis trükkel meg meg lehet szüntetni.

Keretek

A kereteket a következőképpen lehet megadni:

```
\begin{frame}[opció]{cím}{alcím}
```

Keret tartalma

```
\end{frame}
```

vagy

```
\begin{frame}[opció]
```

```
\frametitle{cím}
```

```
\framesubtitle{alcím}
```

Keret tartalma

```
\end{frame}
```

Keret opciók

- ▶ Keret tartalmának az elhelyezése. Lehetséges értékek **[t]** (top), **[b]** (bottom), **[c]** (center).
- ▶ A keretben a fejléc, lábléc és az oldalsávok nem jelennek meg a **[plain]** hatására.
- ▶ Keret kicsinyítése **[shrink=20]** (százalékban). A kicsinyítés alapértéke 0%.
- ▶ Verbatim szöveg vagy kód nem írható a keretbe, ha nem adjuk meg a **[fragile]** opciót.
- ▶ **[squeeze]** opció megszünteti a listák térközeit.
- ▶ **[allowframebreaks=kitöltés]** A kitöltés mértéke $[0 - 1]$ érték, alapértéke 1.

Keretek

- ▶ A prezentáció keretek `frame`-k sorozatából áll.
- ▶ Egy `frame`-ből egy vagy több diakép is készülhet.
 - ▶ Ha nem fér rá egy diára, és használtuk a `\begin{frame}[allowframebreaks]{Cím}` opciót, akkor automatikusan több diakép keletkezik.
 - ▶ Használjuk a `\pause` parancsot vagy az `overlay` módot.

A következő két dia egy **[allowframebreaks]** opcióval megadott `frame`-ből keletkezett.

Gondolatok a programok megértéséről I

Életemben sok olyan programozási tanfolyamot láttam, amelyek hasonlatosak voltak az átlagos gépjárművezetői tanfolyamokhoz, ahol t.i. az ember nem azt tanulja meg, hogyan érhet el céljához a gépkocsival, hanem azt, hogyan kell a kocsival bánni.

Véleményem szerint a program önmagában sohasem végcél. A program célja, hogy bizonyos számításokat váltson ki, és ezeknek a számításoknak a célja valamely meghatározott eredmény elérése. Bár a programozó által előállított végtermék a program, foglalkozásának igazi tárgyai azok a lehetséges számítások, melyek a program hatására létrejöhetnek, és melyeknek az elvégzése már a gépre hárul. Például ha a programozó azt állítja, hogy programja helyes, akkor állítása

Gondolatok a programok megértéséről II

valójában a program által kiváltható számítások halmazára vonatkozik.

Az a tény, hogy a teljes tevékenységi lánc végső szakasza, az áttérés a statikus programszövegről a dinamikus programvégrehajtásra, a gép feladata marad, újabb bonyodalmat okoz. Bizonyos értelemben nehezebb programot írni, mint egy matematikai elméletet felállítani. Mind a matematikai elmélet, mind a program strukturált, időtlen objektumok. De míg a matematikai elméletnek önmagában is értelme van, addig a programnak csak a végrehajtása ad értelmet.

A `\pause` parancs

A kereteket a következőképpen lehet megadni:

```
\begin{frame}{Példa}{}  
Első sor.\par\pause  
Második sor.\par\pause  
Harmadik sor.  
\end{frame}
```

A példa kinézete:

A `\pause` parancs

Első sor.

A `\pause` parancs

Első sor.

Második sor.

A `\pause` parancs

Első sor.

Második sor.

Harmadik sor.

Blokkok

Theorem

```
\begin{theorem}  
...  
\end{theorem}
```

Corollary

```
\begin{corollary}  
...  
\end{corollary}
```

Definition

```
\begin{definition}  
...  
\end{definition}
```

Bizonyítás.

```
\begin{proof}  
...  
\end{proof}
```



Example

```
\begin{example}  
...  
\end{example}
```

Lemma

```
\begin{lemma}  
...  
\end{lemma}
```

Blokkok

Sima blokk

```
\begin{block}{Sima blokk}  
...  
\end{block}
```

Figyelmeztetés

```
\begin{alertblock}{Figyelmeztetés}  
...  
\end{alertblock}
```

Blokkok színeinek a megváltoztatása

```
% szinek nevének az elérése
\documentclass[xcolor=dvipsnames]{beamer}
. . .
\setbeamercolor{block title}{bg=Blue,fg=White}
\setbeamercolor{block body}{bg=Periwinkle!10,fg=Black}

\setbeamercolor{block title example}{bg=OliveGreen, fg=White}
\setbeamercolor{block body example}{fg= blue,bg= blue!5}

\setbeamercolor{block title alerted}{bg=BrickRed, fg=White}
\setbeamercolor{block body alerted}{fg= Black,bg= Red!5}
```

Lista környezetek

Itemize

```
\begin{itemize}  
  \item első tétel  
  \item második tétel  
\end{itemize}
```

- ▶ első tétel
- ▶ második tétel

Enumerate

```
\begin{enumerate}  
  \item első tétel  
  \item második tétel  
\end{enumerate}
```

1. első tétel
2. második tétel

Description

```
\begin{description}  
  \item[alma] első tétel  
  \item[körte] második tétel  
\end{description}
```

alma első tétel
körte második tétel

pause parancs

A `\pause` parancs nem nagyon rugalmas. Nemcsak listáknál, hanem szövegnél vagy matematikai képletnél is lehet használni.

```
\begin{enumerate}
\item Shown from first slide on.
\pause
\item Shown from second slide on.
\pause
\item Shown from third slide on.
\pause
\item Shown from fourth slide on.
\end{enumerate}
```

1. Shown from first slide on.
2. Shown from second slide on.
3. Shown from third slide on.
4. Shown from fourth slide on.

pause parancs

A `\pause` parancs nem nagyon rugalmas. Nemcsak listáknál, hanem szövegnél vagy matematikai képletnél is lehet használni.

```
\begin{enumerate}  
\item Shown from first slide on.  
\pause  
\item Shown from second slide on.  
\pause  
\item Shown from third slide on.  
\pause  
\item Shown from fourth slide on.  
\end{enumerate}
```

1. Shown from first slide on.
2. Shown from second slide on.
3. Shown from third slide on.
4. Shown from fourth slide on.

pause parancs

A `\pause` parancs nem nagyon rugalmas. Nemcsak listáknál, hanem szövegnél vagy matematikai képletnél is lehet használni.

```
\begin{enumerate}  
\item Shown from first slide on.  
\pause  
\item Shown from second slide on.  
\pause  
\item Shown from third slide on.  
\pause  
\item Shown from fourth slide on.  
\end{enumerate}
```

1. Shown from first slide on.
2. Shown from second slide on.
3. Shown from third slide on.
4. Shown from fourth slide on.

pause parancs

A `\pause` parancs nem nagyon rugalmas. Nemcsak listáknál, hanem szövegnél vagy matematikai képletnél is lehet használni.

```
\begin{enumerate}  
\item Shown from first slide on.  
\pause  
\item Shown from second slide on.  
\pause  
\item Shown from third slide on.  
\pause  
\item Shown from fourth slide on.  
\end{enumerate}
```

1. Shown from first slide on.
2. Shown from second slide on.
3. Shown from third slide on.
4. Shown from fourth slide on.

Overlay megadás

- ▶ Az overlay megadás `<` és `>` között történik. Ezzel tudjuk szabályozni, hogy mely szövegek vagy ábrák legyenek kitakarva, és melyek legyenek láthatóak.
- ▶ A `<` és `>` között lévő információval szabályozzuk, hogy a frame melyik eleme melyik dián vagy diákon jelenjen meg.
 - ▶ `<2>` csak a második dián látható
 - ▶ `<1->` első diától végig látható
 - ▶ `<1-3>` 1, 2 és a 3 dián látható
 - ▶ `<-4>` csak az első négy dián látható
 - ▶ `<-3,5-6,8->` a 4 és a 7 dián kívül minden dián látható.

Overlay megadás

Példa

```
\begin{enumerate}  
\item<1-> Első diától látható  
\item<2-> Második diától látható  
\item<4> Csak a 4. dián látható  
\item<3,5-> 3. dián és 5-től. végig látható  
\end{enumerate}
```

Eredmény

1. Első diától látható

Overlay megadás

Példa

```
\begin{enumerate}  
\item<1-> Első diától látható  
\item<2-> Második diától látható  
\item<4> Csak a 4. dián látható  
\item<3,5-> 3. dián és 5-től. végig látható  
\end{enumerate}
```

Eredmény

1. Első diától látható
2. Második diától látható

Overlay megadás

Példa

```
\begin{enumerate}  
\item<1-> Első diától látható  
\item<2-> Második diától látható  
\item<4> Csak a 4. dián látható  
\item<3,5-> 3. dián és 5-től. végig látható  
\end{enumerate}
```

Eredmény

1. Első diától látható
2. Második diától látható
4. 3. dián és 5-től. végig látható

Overlay megadás

Példa

```
\begin{enumerate}  
\item<1-> Első diától látható  
\item<2-> Második diától látható  
\item<4> Csak a 4. dián látható  
\item<3,5-> 3. dián és 5-től. végig látható  
\end{enumerate}
```

Eredmény

1. Első diától látható
2. Második diától látható
3. Csak a 4. dián látható

Overlay megadás

Példa

```
\begin{enumerate}  
\item<1-> Első diától látható  
\item<2-> Második diától látható  
\item<4> Csak a 4. dián látható  
\item<3,5-> 3. dián és 5-től. végig látható  
\end{enumerate}
```

Eredmény

1. Első diától látható
2. Második diától látható
3. Csak a 4. dián látható
4. 3. dián és 5-től. végig látható

Overlay megadás

A beamer-ben az overlay kezelésére a **beamerpauses** számláló szolgál, amelyre a $<$ és $>$ között a **+** jellel hivatkozni lehet.

- ▶ A keret elején az értéke 1
- ▶ Minden $>$ után eggyel nő.
- ▶ Egész értéket lehet hozzáadni $< +(\pm n) >$.

Példa

Ha a beamerpauses értéke 3, és a következő kifejezések egyike következi, akkor

1. $<+(3)> = <6>$
2. $<+(-1)> = <2>$
3. $<+(0)> = <+> = <3>$

Overlay megadás

A **beamerpauses** előző értékére a **.**-tal (ponttal) hivatkozhatunk. Ha a **beamerpauses** értékére ponttal hivatkozunk, akkor a **beamerpauses** értéke nem nő **>** jel után.

Példa

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

- ▶ 1. listaelem
- ▶ 2. listaelem

Overlay megadás

A **beamerpauses** előző értékére a **.**-tal (ponttal) hivatkozhatunk. Ha a **beamerpauses** értékére ponttal hivatkozunk, akkor a **beamerpauses** értéke nem nő **>** jel után.

Példa

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

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

Overlay megadás

Előző példával ekvivalens megoldás, ahol a láthatóság vezérlését számokkal adtuk meg. A nem látható sorok is szürkén megjelennek.

Példa

```
\begin{frame}[fragile]{Ovr}  
\setbeamercovered{transparent}  
\begin{itemize}  
\item<1-> 1. listaelem  
\item<1-> 2. listaelem  
\item<2-> 3. listaelem  
\item<2-> 4. listaelem  
\end{itemize}  
\end{frame}
```

▶ 1. listaelem

▶ 2. listaelem

▶ 3. listaelem

▶ 4. listaelem

Overlay megadás

Előző példával ekvivalens megoldás, ahol a láthatóság vezérlését számokkal adtuk meg. A nem látható sorok is szürkén megjelennek.

Példa

```
\begin{frame}[fragile]{Ovr}  
\setbeamercovered{transparent}  
\begin{itemize}  
\item<1-> 1. listaelem  
\item<1-> 2. listaelem  
\item<2-> 3. listaelem  
\item<2-> 4. listaelem  
\end{itemize}  
\end{frame}
```

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

Overlay megadás

A `\uncover<>` parancs hatására a megadott szöveg csak az előírt diákon fog megjelenni, máshol csak foglalja a helyet. A nem látható szövegre a `transparent` érték hatással van, tehát halványan látszik.

Példa

```
\begin{frame}[fragile]{Overlay megadás}  
\setbeamercovered{transparent=20}  
\uncover<1>{Ez az elso dia, }  
\uncover<2>{ez pedig a második.}  
\end{frame}
```

Eredmény

Ez az elso dia, ez pedig a második.

Overlay megadás

A `\uncover<>` parancs hatására a megadott szöveg csak az előírt diákon fog megjelenni, máshol csak foglalja a helyet. A nem látható szövegre a transparent érték hatással van, tehát halványan látszik.

Példa

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

Eredmény

Ez az első dia, ez pedig a második.

Overlay megadás

A `\visible<>` parancs hatására a megadott szöveg csak az előírt diákon fog megjelenni, máshol csak foglalja a helyet. A nem látható szövegre a `transparent` érték *nincs* hatással.

Példa

```
\begin{frame}[fragile]{Overlay megadás}  
\setbeamercovered{transparent=20}  
\visible<1>{Ez az elso dia, }  
\visible<2>{ez pedig a masodik.}  
\end{frame}
```

Eredmény

Ez az elso dia,

Overlay megadás

A `\visible<>` parancs hatására a megadott szöveg csak az előírt diákon fog megjelenni, máshol csak foglalja a helyet. A nem látható szövegre a *transparent* érték *nincs* hatással.

Példa

```
\begin{frame}[fragile]{Overlay megadás}  
\setbeamercovered{transparent=20}  
\visible<1>{Ez az elso dia, }  
\visible<2>{ez pedig a második.}  
\end{frame}
```

Eredmény

ez pedig a második.

Overlay megadás

A `\invisible<>` parancs hatására a megadott szöveg az előírt diákon *nem* fog megjelenni, csak foglalja a helyet. A nem látható szövegre a `transparent` érték *nincs* hatással.

Példa

```
\begin{frame}[fragile]{Overlay megadás}  
\setbeamercovered{transparent=20}  
\invisible<1>{Ez az elso dia, }  
\invisible<2>{ez pedig a masodik.}  
\end{frame}
```

Eredmény

ez pedig a második.

Overlay megadás

A `\invisible<>` parancs hatására a megadott szöveg az előírt diákon *nem* fog megjelenni, csak foglalja a helyet. A nem látható szövegre a *transparent* érték *nincs* hatással.

Példa

```
\begin{frame}[fragile]{Overlay megadás}  
\setbeamercovered{transparent=20}  
\invisible<1>{Ez az első dia, }  
\invisible<2>{ez pedig a második.}  
\end{frame}
```

Eredmény

Ez az első dia,

Overlay megadás

A `\only<>` parancs hatására a megadott szöveg csak az előírt diákon fog megjelenni, ahol viszont nem jelenik meg, ott *nem foglalja* a helyet. A nem látható szövegre a `transparent` érték *nincs* hatással.

Példa

```
\begin{frame}[fragile]{Overlay megadás}  
\setbeamercovered{transparent=20}  
\only<1>{Ez az elso dia, }  
\only<2>{ez pedig a masodik.}  
\end{frame}
```

Eredmény

Ez az elso dia,

Overlay megadás

A `\only<>` parancs hatására a megadott szöveg csak az előírt diákon fog megjelenni, ahol viszont nem jelenik meg, ott *nem foglalja* a helyet. A nem látható szövegre a transparent érték *nincs* hatással.

Példa

```
\begin{frame}[fragile]{Overlay megadás}  
\setbeamercovered{transparent=20}  
\only<1>{Ez az elso dia, }  
\only<2>{ez pedig a masodik.}  
\end{frame}
```

Eredmény

ez pedig a második.

Overlay megadás

A `\alt<>{szöveg1}{szöveg2}` parancs hatására a megadott diákra a `szöveg1`, a többire a `szöveg2` kerül. Csak az egyik szöveg fog megjelenni, a másik *nem foglalja* a helyet.

Példa

```
\alt<1>{Ez az 1. dia első sora.}{Ez a többi dián az első sor.} \\  
\alt<2>{Ez pedig a 2. dia második sora.}{Ez pedig a többi dia második sora.}
```

Eredmény

Ez az 1. dia első sora.

Ez pedig a többi dia második sora.

Overlay megadás

A `\alt<>{szöveg1}{szöveg2}` parancs hatására a megadott diákra a `szöveg1`, a többire a `szöveg2` kerül. Csak az egyik szöveg fog megjelenni, a másik *nem foglalja* a helyet.

Példa

```
\alt<1>{Ez az 1. dia első sora.}{Ez a többi dián az első sor.} \\  
\alt<2>{Ez pedig a 2. dia második sora.}{Ez pedig a többi dia második sora.}
```

Eredmény

Ez a többi dián az első sor.

Ez pedig a 2. dia második sora.

Overlay megadás

A `\temporal<spec>{szöveg-előtte}{szöveg}{szöveg-utána}` parancs hatására a megadott diák előtt a szöveg-előtte, a megadott diákra a szöveg, a megadott diák után pedig a szöveg-utána fog megjelenni.

Példa

```
\temporal<1>{Szöveg1  előtte}{Szöveg1}{Szöveg1  utána} \\
\temporal<2>{Szöveg2  előtte}{Szöveg2}{Szöveg2  utána} \\
\temporal<2>{Szöveg3  előtte}{Szöveg3}{Szöveg3  utána} \\
```

Eredmény

Szöveg1
Szöveg2 előtte
Szöveg3 előtte

Overlay megadás

A `\temporal<spec>{szöveg-előtte}{szöveg}{szöveg-utána}` parancs hatására a megadott diák előtt a szöveg-előtte, a megadott diákra a szöveg, a megadott diák után pedig a szöveg-utána fog megjelenni.

Példa

```
\temporal<1>{Szöveg1  előtte}{Szöveg1}{Szöveg1  utána} \\
\temporal<2>{Szöveg2  előtte}{Szöveg2}{Szöveg2  utána} \\
\temporal<2>{Szöveg3  előtte}{Szöveg3}{Szöveg3  utána} \\
```

Eredmény

Szöveg1 utána
 Szöveg2
 Szöveg3 előtte

Overlay megadás

A `\temporal<spec>{szöveg-előtte}{szöveg}{szöveg-utána}` parancs hatására a megadott diák előtt a szöveg-előtte, a megadott diákra a szöveg, a megadott diák után pedig a szöveg-utána fog megjelenni.

Példa

```
\temporal<1>{Szöveg1  előtte}{Szöveg1}{Szöveg1  utána} \\
\temporal<2>{Szöveg2  előtte}{Szöveg2}{Szöveg2  utána} \\
\temporal<2>{Szöveg3  előtte}{Szöveg3}{Szöveg3  utána} \\
```

Eredmény

Szöveg1 utána
Szöveg2 utána
Szöveg3

Overlay megadás

Figyelemfelkeltő sorokat az `\alert<spec>` parancs segítségével hozhatunk létre. A hatására a megadott diákon a felírat piros színnel, máshol normálisan fog megjelenni.

Példa

```
\alert{Alert az összes dián} \\  
\alert<2>{Alert a 2. dián} \\  
\alert<1,3>{Alert az 1-es és 3-as dián} \\  
\alert<-2,4>{Alert az 1,2 és 4-as dián} \\
```

Eredmény

Alert az összes dián

Alert a 2. dián

Alert az 1-es és 3-as dián

Alert az 1,2 és 4-as dián

Overlay megadás

Figyelemfelkeltő sorokat az `\alert<spec>` parancs segítségével hozhatunk létre. A hatására a megadott diákon a felírat piros színnel, máshol normálisan fog megjelenni.

Példa

```
\alert{Alert az összes dián} \\  
\alert<2>{Alert a 2. dián} \\  
\alert<1,3>{Alert az 1-es és 3-as dián} \\  
\alert<-2,4>{Alert az 1,2 és 4-as dián} \\
```

Eredmény

Alert az összes dián

Alert a 2. dián

Alert az 1-es és 3-as dián

Alert az 1,2 és 4-as dián

Overlay megadás

Figyelemfelkeltő sorokat az `\alert<spec>` parancs segítségével hozhatunk létre. A hatására a megadott diákon a felírat piros színnel, máshol normálisan fog megjelenni.

Példa

```
\alert{Alert az összes dián} \\  
\alert<2>{Alert a 2. dián} \\  
\alert<1,3>{Alert az 1-es és 3-as dián} \\  
\alert<-2,4>{Alert az 1,2 és 4-as dián} \\
```

Eredmény

Alert az összes dián

Alert a 2. dián

Alert az 1-es és 3-as dián

Alert az 1,2 és 4-as dián

Overlay megadás

Figyelemfelkeltő sorokat az `\alert<spec>` parancs segítségével hozhatunk létre. A hatására a megadott diákon a felírat piros színnel, máshol normálisan fog megjelenni.

Példa

```
\alert{Alert az összes dián} \\  
\alert<2>{Alert a 2. dián} \\  
\alert<1,3>{Alert az 1-es és 3-as dián} \\  
\alert<-2,4>{Alert az 1,2 és 4-as dián} \\
```

Eredmény

Alert az összes dián

Alert a 2. dián

Alert az 1-es és 3-as dián

Alert az 1,2 és 4-as dián

Overlay megadás

Overlay parancsokat nem csak a szöveg tartalmára, hanem a szöveg tulajdonságának megváltoztatására is lehet használni.

Példa

<code>\textbf<2>{Sample}\\</code>	Sample
<code>\textit<2>{Sample}\\</code>	Sample
<code>\textsl<2>{Sample}\\</code>	Sample
<code>\alert<2>{Sample}\\</code>	Sample
<code>\textrm<2>{Sample}\\</code>	Sample
<code>\textsf<2>{Sample}\\</code>	Sample
<code>\color<2>{green}{Sample}\\</code>	Sample
<code>\structure<2>{Sample}</code>	Sample

Overlay megadás

Overlay parancsokat nem csak a szöveg tartalmára, hanem a szöveg tulajdonságának megváltoztatására is lehet használni.

Példa

```
\textbf<2>{Sample}\\  
\textit<2>{Sample}\\  
\textsl<2>{Sample}\\  
\alert<2>{Sample}\\  
\textrm<2>{Sample}\\  
\textsf<2>{Sample}\\  
\color<2>{green}{Sample}\\  
\structure<2>{Sample}
```

Sample

Sample

Sample

Sample

Sample

Sample

Sample

Sample

Overlay megadás

Továbbá a blokkok láthatóságát is lehet vezérelni.

Példa

```
\begin{tetel}<1->
Létezik végtelen halmaz.
\end{tetel}
```

Tétel

Létezik végtelen halmaz.

```
\begin{proof}<3->
Következik a kiválasztási a
\end{proof}
```

Bizonyítás.

Következik a kiválasztási axiómából. ☐

```
\begin{example}<2->
A természetes számok halmaza végtelen.
\end{example}
```

Example

A természetes számok halmaza végtelen.

Overlay megadás

Továbbá a blokkok láthatóságát is lehet vezérelni.

Példa

```
\begin{tetel}<1->
Létezik végtelen halmaz.
\end{tetel}
```

Tétel

Létezik végtelen halmaz.

```
\begin{proof}<3->
Következik a kiválasztási a
\end{proof}
```

Bizonyítás.

Következik a kiválasztási
axiómából. ☐

```
\begin{example}<2->
A természetes számok halmaza végtelen
\end{example}
```

Example

A természetes számok
halmaza végtelen.

Overlay megadás

Továbbá a blokkok láthatóságát is lehet vezérelni.

Példa

```
\begin{tetel}<1->  
Létezik végtelen halmaz.  
\end{tetel}
```

Tétel

Létezik végtelen halmaz.

```
\begin{proof}<3->  
Következik a kiválasztási a  
\end{proof}
```

Bizonyítás.

Következik a kiválasztási
axiómából. ☐

```
\begin{example}<2->  
A természetes számok halmaza végtelen.  
\end{example}
```

Example

A természetes számok
halmaza végtelen.

Overlay megadás

Képletek megjelenítése:

```
\begin{theorem}<1->
  \[
    \int_{-\infty}^{\infty} e^{-x^2} dx = \sqrt{\Pi}
  \]
\end{theorem}

\begin{proof}<2->
\begin{align*}
\uncover<2->{\left(\int_{-\infty}^{\infty} e^{-x^2} dx = \sqrt{\Pi}\right)^2}
&= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-x^2+y^2} dx dy } \\

\uncover<3->{\&= \int_0^2 \Pi \int_0^{\infty} e^{-r^2} r
dr d\theta } \\

\uncover<4->{\&= 2 \Pi \int_0^{\infty} r e^{-r^2} dr = \Pi}
\end{align*}
\end{proof}
```

Overlay megadás

Theorem

$$\int_{-\infty}^{\infty} e^{-x^2} dx = \sqrt{\pi}$$

Overlay megadás

Theorem

$$\int_{-\infty}^{\infty} e^{-x^2} dx = \sqrt{\pi}$$

Bizonyítás.

$$\left(\int_{-\infty}^{\infty} e^{-x^2} dx = \sqrt{\pi} \right)^2 = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-x^2+y^2} dx dy$$

Overlay megadás

Theorem

$$\int_{-\infty}^{\infty} e^{-x^2} dx = \sqrt{\pi}$$

Bizonyítás.

$$\begin{aligned} \left(\int_{-\infty}^{\infty} e^{-x^2} dx = \sqrt{\pi} \right)^2 &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-x^2+y^2} dx dy \\ &= \int_0^{2\pi} \int_0^{\infty} e^{-r^2} r dr d\theta \end{aligned}$$

Overlay megadás

Theorem

$$\int_{-\infty}^{\infty} e^{-x^2} dx = \sqrt{\pi}$$

Bizonyítás.

$$\begin{aligned} \left(\int_{-\infty}^{\infty} e^{-x^2} dx = \sqrt{\pi} \right)^2 &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-x^2+y^2} dx dy \\ &= \int_0^{2\pi} \int_0^{\infty} e^{-r^2} r dr d\theta \\ &= 2\pi \int_0^{\infty} r e^{-r^2} dr = \pi \end{aligned}$$

Ugráló sorok

```

\only<1->{
  Adatok:
\begin{itemize}
\setlength\itemsep{0.1em}
  \item 1 merevlemez 3 db kétoldalas lemez
  \item 1 oldal 20000 sáv
  \item 1 sáv 1000 szektor
  \item 1 szektor 500 byte
  \item seek idő 4 ms
  \item 1 szektor átlagos olvasása 10ms
\end{itemize}
}

\only<2-4>{(a) Milyen gyorsan forog a lemez? \vskip 1ex}

\only<3-4>{ 4 + x / 2 + x / 1000 = 10ms }

\only<4-4>{ x = \frac{6000}{501}ms ; \frac{1}{x} =
  \frac{501}{6000} \frac{\text{fordulat}}{\text{ms}} =
  \frac{501}{6000} \frac{\text{fordulat}}{\text{ms}}
  \times 60000 \frac{\text{ms}}{\text{perc}} =
  5010 \frac{\text{fordulat}}{\text{perc}} }

```

Ugráló sorok - dia kinézet

Adatok:

- ▶ 1 merevlemez 3 db kétoldalas lemez
- ▶ 1 oldal 20000 sáv
- ▶ 1 sáv 1000 szektor
- ▶ 1 szektor 500 byte
- ▶ seek idő 4 ms
- ▶ 1 szektor átlagos olvasása 10ms

Ugráló sorok - dia kinézet

Adatok:

- ▶ 1 merevlemez 3 db kétoldalas lemez
- ▶ 1 oldal 20000 sáv
- ▶ 1 sáv 1000 szektor
- ▶ 1 szektor 500 byte
- ▶ seek idő 4 ms
- ▶ 1 szektor átlagos olvasása 10ms

(a) Milyen gyorsan forog a lemez?

Ugráló sorok - dia kinézet

Adatok:

- ▶ 1 merevlemez 3 db kétoldalas lemez
- ▶ 1 oldal 20000 sáv
- ▶ 1 sáv 1000 szektor
- ▶ 1 szektor 500 byte
- ▶ seek idő 4 ms
- ▶ 1 szektor átlagos olvasása 10ms

(a) Milyen gyorsan forog a lemez?

$$4 + x/2 + x/1000 = 10\text{ms}$$

Ugráló sorok - dia kinézet

Adatok:

- ▶ 1 merevlemez 3 db kétoldalas lemez
- ▶ 1 oldal 20000 sáv
- ▶ 1 sáv 1000 szektor
- ▶ 1 szektor 500 byte
- ▶ seek idő 4 ms
- ▶ 1 szektor átlagos olvasása 10ms

(a) Milyen gyorsan forog a lemez?

$$4 + x/2 + x/1000 = 10\text{ms}$$

$$x = \frac{6000}{501} \text{ms} ;$$

$$\frac{1}{x} = \frac{501}{6000} \frac{\text{fordulat}}{\text{ms}} = \frac{501}{6000} \frac{\text{fordulat}}{\text{ms}} \times 60000 \frac{\text{ms}}{\text{perc}} = 5010 \frac{\text{fordulat}}{\text{perc}}$$

Ugráló sorok megszüntetése

```

\begin{overlayarea}{\textwidth}{1\textheight}
\only<1->{
  Adatok:
\begin{itemize}
\setlength\itemsep{0.1em}
  \item 1 merevlemez 3 db kétoldalas lemez
  \item 1 oldal 20000 sáv
  \item 1 sáv 1000 szektor
  \item 1 szektor 500 byte
  \item seek idő 4 ms
  \item 1 szektor átlagos olvasása 10ms
\end{itemize}
}
\only<2-4>{(a) Milyen gyorsan forog a lemez? \vskip 1ex}

\only<3-4>{ $ 4 + x / 2 + x / 1000 $ = 10ms }

\only<4-4>{ $x = \frac{6000}{501}\text{ms}$ ; $\frac{1}{x} =
  \frac{501}{6000} \frac{\text{fordulat}}{\text{ms}} =
  \frac{501}{6000} \frac{\text{fordulat}}{\text{ms}}
  \times 60000 \frac{\text{ms}}{\text{perc}} =
  5010 \frac{\text{fordulat}}{\text{perc}}$ }
\end{overlayarea}

```

Ugráló sorok megszüntetése

Adatok:

- ▶ 1 merevlemez 3 db kétoldalas lemez
- ▶ 1 oldal 20000 sáv
- ▶ 1 sáv 1000 szektor
- ▶ 1 szektor 500 byte
- ▶ seek idő 4 ms
- ▶ 1 szektor átlagos olvasása 10ms

Ugráló sorok megszüntetése

Adatok:

- ▶ 1 merevlemez 3 db kétoldalas lemez
- ▶ 1 oldal 20000 sáv
- ▶ 1 sáv 1000 szektor
- ▶ 1 szektor 500 byte
- ▶ seek idő 4 ms
- ▶ 1 szektor átlagos olvasása 10ms

(a) Milyen gyorsan forog a lemez?

Ugráló sorok megszüntetése

Adatok:

- ▶ 1 merevlemez 3 db kétoldalas lemez
- ▶ 1 oldal 20000 sáv
- ▶ 1 sáv 1000 szektor
- ▶ 1 szektor 500 byte
- ▶ seek idő 4 ms
- ▶ 1 szektor átlagos olvasása 10ms

(a) Milyen gyorsan forog a lemez?

$$4 + x/2 + x/1000 = 10\text{ms}$$

Ugráló sorok megszüntetése

Adatok:

- ▶ 1 merevlemez 3 db kétoldalas lemez
- ▶ 1 oldal 20000 sáv
- ▶ 1 sáv 1000 szektor
- ▶ 1 szektor 500 byte
- ▶ seek idő 4 ms
- ▶ 1 szektor átlagos olvasása 10ms

(a) Milyen gyorsan forog a lemez?

$$4 + x/2 + x/1000 = 10\text{ms}$$

$$x = \frac{6000}{501} \text{ ms} ;$$

$$\frac{1}{x} = \frac{501}{6000} \frac{\text{fordulat}}{\text{ms}} = \frac{501}{6000} \frac{\text{fordulat}}{\text{ms}} \times 60000 \frac{\text{ms}}{\text{perc}} = 5010 \frac{\text{fordulat}}{\text{perc}}$$

Nem fér ki a szöveg, mert túl széles

A shrink opcióval a teljes keretet át tudjuk méretezni, de részeket nem. A problémát megoldhatjuk az adjustbox csomag segítségével.

LaTeX kód

```
\begin{block}{Példa}  
\begin{lstlisting}[escapeinside=]  
\alert<1>{Ez egy nagyon hosszú sor ami nem fér ki} \\  
\alert<2>{Ez egy másik nagyon hosszú sor ami nem fér ki} \\  
\alert<3>{És itt a harmadik nagyon hosszú sor ami nem fér ki}  
\end{lstlisting}  
\end{block}
```

Eredmény

```
\alert<1>{Ez egy nagyon hosszú sor ami nem fér ki} \\  
\alert<2>{Ez egy másik nagyon hosszú sor ami nem fér ki}  
\alert<3>{És itt a harmadik nagyon hosszú sor ami nem fé
```

Nem fér ki a szöveg, mert túl széles

A shrink opcióval a teljes keretet át tudjuk méretezni, de részeket nem. A problémát megoldhatjuk az adjustbox csomag segítségével.

LaTeX kód

```
\begin{block}{Példa}
\begin{adjustbox}{width=0.9\textwidth}
\begin{lstlisting}[escapeinside=II]
\alert<1>{Ez egy nagyon hosszú sor ami nem fér ki} \\
\alert<2>{Ez egy másik nagyon hosszú sor ami nem fér ki} \\
\alert<3>{És itt a harmadik nagyon hosszú sor ami nem fér ki}
\end{lstlisting}
\end{adjustbox}
\end{block}
```

Eredmény

```
\alert<1>{Ez egy nagyon hosszú sor ami nem fér ki} \\
\alert<2>{Ez egy másik nagyon hosszú sor ami nem fér ki} \\
\alert<3>{És itt a harmadik nagyon hosszú sor ami nem fér ki}
```

Feladat

- ▶ Készítsen egy 3 diából álló prezetációt.
- ▶ Az első dián használjon lista környezetet.
- ▶ A második dián használjon blokkokat.
- ▶ A harmadik diára írjon matematikai képleteket.
- ▶ A listaelemek láthatóságát adja meg.
- ▶ Változtassa meg a keret méretét
- ▶ Nézze meg hogyan változik a keret, ha használja a plain opciót.
- ▶ `https://www.hartwork.org/beamer-theme-matrix/` vagy
a
`http://deic.uab.es/~iblanes/beamer_gallery/`
honlapok alapján változtassa meg a prezentáció kinézetét.