

# 21. fejezet

## Prezentációk

Prezentáció készítéséhez a `beamer` dokumentumosztályt javasoljuk. Ennek jellemzői:

- Oldalméret: 128 mm × 96 mm (4 : 3 arány). Az `aspectratio=169` opció esetén 160 mm × 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 balra zárt, új bekezdés elején nincs behúzás.
- Keret (lásd később) tartalmának függőleges pozíciója: közép. Opcióban másik két lehetőség: `t` (fent), `b` (lent).
- Betöltődnek a `graphicx`, `amsthm`, `xcolor`, `enumerate`, `hyperref` csomagok.

### 21.1. Témák

A `beamer` sok ún. témát tartalmaz, melyek mindegyike egy-egy tipográfiai beállítást, stílust jelent. Ha az alapértelmezett témától el akar térni, akkor azok betöltése a preambulumban történik.

#### 21.1.1. Teljes témák

Célszerű először egy teljes témát választani, amely beállítja a szerkezetet, színeket és a betűtípust is. Ennek betöltése:

```
\usetheme[<opciók>]{<név>}
```

Ha ebben valamilyen részletet meg akar változtatni, akkor alkalmazhat egy később ismertetett belső vagy külső szerkezeti, szín- vagy betűtípus témát is. A *<név>* a következő listaelemek valamelyike lehet:

##### Oldalsáv nélkül

`Bergen` Nincsenek *<opciók>*

`Boadilla` Elérhető *<opciók>*: `secheader` (fejléc bekapcsolása)

`Madrid` Elérhető *<opciók>*: `secheader` (fejléc bekapcsolása)

`AnnArbor` Nincsenek *<opciók>*

`CambridgeUS` Nincsenek *<opciók>*

`EastLansing` Nincsenek *<opciók>*

`Pittsburgh` Nincsenek *<opciók>*

`Rochester` Elérhető *<opciók>*: `height=<magasság>` (keretcím magassága)

**Fa navigációval****Antibes** Nincsenek *<opciók>***JuanLesPins** Nincsenek *<opciók>***Montpellier** Nincsenek *<opciók>***Oldalsávval****Berkeley** Elérhető *<opciók>*:**hideallsubsections** (oldalsávon nincs alszakasz cím)**hideothersubsections** (oldalsávon csak az aktuális alszakasz címe)**left** (oldalsáv bal oldalon)**right** (oldalsáv jobb oldalon)**width=***<szélesség>* (oldalsáv szélessége)**PaloAlto** Elérhető *<opciók>*: lásd **Berkeley****Goettingen** Elérhető *<opciók>*: lásd **Berkeley****Marburg** Elérhető *<opciók>*: lásd **Berkeley****Hannover** Elérhető *<opciók>*: lásd **Berkeley**, de nincs **left** és **right****Mini keret a fejlécben****Berlin** Elérhető *<opciók>*: **compress** (egysoros a mini keret)**Ilmenau** Elérhető *<opciók>*: **compress** (egysoros a mini keret)**Dresden** Elérhető *<opciók>*: **compress** (egysoros a mini keret)**Darmstadt** Nincsenek *<opciók>***Frankfurt** Nincsenek *<opciók>***Singapore** Nincsenek *<opciók>***Szeged** Nincsenek *<opciók>***Fejlécben az aktuális szakasz és alszakasz címe****Copenhagen** Nincsenek *<opciók>***Luebeck** Nincsenek *<opciók>***Malmoe** Nincsenek *<opciók>***Warsaw** Nincsenek *<opciók>***21.1.2. Belső témák**

Azt szabályozza, hogy a belső szerkezeti elemekből (címoldal, listák, tömbök, tételszerű környezetek, képek, táblázatok, lábjegyzetek, irodalomjegyzék) mi jelenjen meg és milyen geometriával. Ennek betöltése:

`\useinnertheme[<opciók>]{<név>}`A *<név>* a következő listaelemek valamelyike lehet:**circles** Nincsenek *<opciók>***rectangles** Nincsenek *<opciók>***rounded** Elérhető *<opciók>*: **shadow** (árnyékolt tömbök)**inmargin** Nincsenek *<opciók>***21.1.3. Külső témák**

Azt szabályozza, hogy a külső szerkezeti elemekből (fej- és lábléc, oldalsávok, logó, keret címe) mi jelenjen meg és milyen geometriával. Ennek betöltése:

`\useoutertheme[<opciók>]{<név>}`

A *<név>* a következő listaelemek valamelyike lehet:

**infolines** Nincsenek *<opciók>*  
**miniframes** Elérhető *<opciók>*:  
     **footline=authorinstitute** (láblécben: szerző, intézet)  
     **footline=authortitle** (láblécben: szerző, cím)  
     **footline=institutetitle** (láblécben: intézet, cím)  
     **footline=authorinstitutetitle** (láblécben: szerző, intézet, cím)  
     **subsection=false** (alszakasz címet ne mutassa)  
**smoothbars** Elérhető *<opciók>*:  
     **subsection=false** (alszakasz címet ne mutassa)  
**sidebar** Elérhető *<opciók>*:  
     **hideallsubsections** (tartalomban nincs alszakasz cím)  
     **hideothersubsections** (tartalomban csak az aktuális alszakasz cím)  
     **left** (oldalsáv bal oldalon)  
     **right** (oldalsáv jobb oldalon)  
     **width=<szélesség>** (oldalsáv szélessége)  
     **height=<magasság>** (keretcím magassága)  
**split** Nincsenek *<opciók>*  
**shadow** Nincsenek *<opciók>*  
**tree** Elérhető *<opciók>*: **hooks** („faágak” behúzása)  
**smoothtree** Nincsenek *<opciók>*

#### 21.1.4. Színtémák

Belső és külső szerkezeti elemek színvilágát állítja be. Ennek betöltése:

```
\usecolortheme[<opciók>]{<név>}
```

A *<név>* a következő listaelemek valamelyike lehet:

**structure** Elérhető *<opciók>*: **named=<színnév>** (strukturális elemek előterének színe)  
**sidebartab** Nincsenek *<opciók>*

##### Teljes színtémák

**albatross** Elérhető *<opciók>*: **overlystylish**  
**beetle** Nincsenek *<opciók>*  
**crane** Nincsenek *<opciók>*  
**dove** Nincsenek *<opciók>*  
**fly** Nincsenek *<opciók>*  
**monarca** Nincsenek *<opciók>*  
**seagull** Nincsenek *<opciók>*  
**wolverine** Nincsenek *<opciók>*  
**beaver** Nincsenek *<opciók>*  
**spruce** Nincsenek *<opciók>*

##### Belső elemek színtémái

**lily** Nincsenek *<opciók>*  
**orchid** Nincsenek *<opciók>*  
**rose** Nincsenek *<opciók>*

##### Külső elemek színtémái

**whale** Nincsenek *<opciók>*

seahorse Nincsenek  $\langle opciók \rangle$

dolphin Nincsenek  $\langle opciók \rangle$

### 21.1.5. Betűtípus témák

Belső és külső szerkezeti elemek betűtípusait állítja be. Ennek betöltése:

```
\usefonttheme[ $\langle opciók \rangle$ ]{ $\langle név \rangle$ }
```

A  $\langle név \rangle$  a következő listaelemek valamelyike lehet:

**serif** Elérhető  $\langle opciók \rangle$ :

```
stillsansserifmath
stillsansserifsmall
stillsansseriflarge
stillsansseriftext
onlymath
```

**structurebold** Elérhető  $\langle opciók \rangle$ :

```
onlysmall
onlylarge
```

**structureitalicserif** Elérhető  $\langle opciók \rangle$ : lásd **structurebold**

**structuresmallcapsserif** Elérhető  $\langle opciók \rangle$ : lásd **structurebold**

## 21.2. Keretek

A beamer-ben a prezentáció keretek sorozatából, a keretek pedig diák sorozatából áll. Egy keretnek címet és alcímet is adhat. 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. Minden keretet **frame** környezetbe kell rakni:

```
\begin{frame}[ $\langle opció \rangle$ ]{ $\langle keret címe \rangle$ }{ $\langle keret alcíme \rangle$ }
 $\langle keret tartalma \rangle$ 
\end{frame}
```

vagy

```
\begin{frame}[ $\langle opció \rangle$ ]
\frametitle{ $\langle keret címe \rangle$ }
\framesubtitle{ $\langle keret alcíme \rangle$ }
 $\langle keret tartalma \rangle$ 
\end{frame}
```

Az első esetben a  $\{ \langle keret címe \rangle \}$  illetve  $\{ \langle keret alcíme \rangle \}$  elhagyható. A második esetben a  $\backslash frametitle \{ \langle keret címe \rangle \}$  illetve  $\backslash framesubtitle \{ \langle keret alcíme \rangle \}$  szintén elhagyható. A **frame** környezet opciói

**t, b, c** A keret tartalma függőlegesen felülre, alulra, középre igazított. (Alapopció **c**.)

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

**shrink**= $\langle kicsinyítés \rangle$  Aktiválja a **t** opciót és a keret tartalmát  $\langle kicsinyítés \rangle$ % mértékben kicsinyíti. A  $\langle kicsinyítés \rangle$  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.

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

Példa I → Példa II → Példa III → ...

Ennek átállítására nézzünk néhány példát:

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

Példa → Példa II. → Példa III. → ...

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

Példa → Példa 2. → Példa 3. → ...

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

Példa → Példa (folyt.) → Példa (folyt.) → ...

## 21.3. Egy keretben több dia

Emlékeztetünk arra, hogy a `frame` környezet `allowframebreaks` opcióval nem támogatja a kereten belüli több dia használatát. A keret tartalmának több dián való megjelenítésére a legegyszerűbb megoldás a `\pause` parancs használata. Vigyázat, ez a parancs nem használható az `amsmath` illetve `mathtools` csomagok által definiált környezetekben, mint például 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}
```

### 21.3.1. Overlay specifikációk

Ennél bonyolultabb diasorozatok is létrehozhatók az úgynevezett overlay specifikációk használatával. A `beamer` sok standard 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}
```

```
\end{itemize}
\end{frame}
```

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

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

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

```
<+(1)> = <3>
<+(-1)> = <1>
<+(-2)> = <0>
<+(-4)> = <-2> = <-+>
<+(0)> = <+> = <2>
<+-+(2)> = <2-4>
```

Ezen példák mindegyike után a `beamerpauses` értéke 3-ra nő.

A másik ilyen léptető specifikáció a pont. Ennek használatánál ügyeljén arra, hogy a `beamerpauses` értéke már legalább 2 legyen. Ennek hatása:

- 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}
```

és

```
\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` az alapspecifikációja `<1->`, azaz `\item` ekvivalens az `\item<1->` parancssal. A többi parancs alapspecifikációját az adott parancs tárgyalásánál közöljük.

### 21.3.2. 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 állíthatja be:

```
\setbeamercovered{transparent=<szám>}
```

Ezután a keretben a diákon `<szám>`% intenzitással látható a többi dia tartalma.

### 21.3.3. Overlay specifikációval rendelkező parancsok

```
\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->`.)

```
\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 az előbbihez hasonlóan használható, de a hatása azzal ellentétes. Erre sem hat a pozitív `transparent` érték.

```
\only<spec>{<szöveg>}
```

vagy

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

Ugyanaz mint a `visible`, de a helyet nem foglalja el a `<spec>`-en kívül eső diákon.

```
\alt<spec>{<szöveg1>}{<szöveg2>}
```

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

```

\begin{altenv}<spec>{<start1>}{<vége1>}{<start2>}{<vége2>}
<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->`.)

```
\temporal<spec>{<szöveg előtt>}{<szöveg>}{<szöveg után>}
```

A megadott diák előtt fog megjelenni a *<szöveg előtt>*, a megadott diákon fog megjelenni a *<szöveg>* és a megadott diák után fog megjelenni a *<szöveg után>*. (*<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}

\begin{overlayarea}{<szélesség>}{<magasság>}
\only<spec1>{<szöveg1>}
\only<spec2>{<szöveg2>}
...
\end{overlayarea}
```

A keret minden diáján lefoglal egy *<szélesség>* és *<magasság>* méretű dobozt, melyben a *<spec1>*, *<spec2>*, stb. overlay specifikációknak megfelelően kerül be a *<szöveg1>*, *<szöveg2>*, stb.

```
\begin{overprint}[<szélesség>]
\onslide<spec1><szöveg1>
\onslide<spec2><szöveg2>
...
\end{overprint}
```

A keret minden diáján lefoglal egy *<szélesség>* széles dobozt, melynek alapértéke a szövegtükör szélessége. A doboz magassága a *<szöveg1>*, *<szöveg2>*, stb. által meghatározott dobozok természetes magasságai közül a legnagyobb. A *<spec1>*, *<spec2>*, stb. overlay specifikációk között nem lehet átfedés. A dobozban a *<spec1>*, *<spec2>*, stb. overlay specifikációknak megfelelően kerül be a *<szöveg1>*, *<szöveg2>*, stb.

A következő parancsok is rendelkeznek overlay specifikációval: *\textbf*, *\textit*, *\textsl*, *\textrm*, *\textsf*, *\textcolor*, *\color*. Az alapspecifikáció *<1->*. Például

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

## 21.4. Diaváltás látványeffektekkel

Amikor egy keret következő diájára, vagy a következő keret első diájára váltunk, akkor az eddigiekben csak annyi történt, hogy az előző dia képe egyszerűen átváltott az újra. Ezeket a váltásokat látványosabbá is teheti különböző effektekkel. Sajnos nem



minden pdf néző támogatja ezeket az effekteket, ezért idegen gépen nem biztos, hogy fog működni. Például az Adobe Reader esetén működnek, de csak akkor, ha teljes képernyős üzemmódba váltunk, ahogy ez egy prezentáció bemutatásánál szokásos. Ezeket az effekteket a következő parancsok `frame` környezetbe írásával érheti el:

```
\transblindshorizontal<spec>[<opció>]
\transblindvertical<spec>[<opció>]
\transboxin<spec>[<opció>]
\transboxout<spec>[<opció>]
\transcover<spec>[<opció>]
\transdissolve<spec>[<opció>]
\transfade<spec>[<opció>]
\transglitter<spec>[<opció>]
\transreplace<spec>[<opció>]
\transsplitverticalin<spec>[<opció>]
\transsplitverticalout<spec>[<opció>]
\transsplithorizontalin<spec>[<opció>]
\transsplithorizontalout<spec>[<opció>]
\transwipe<spec>[<opció>]
```

Ezekben a parancsokban az overlay specifikáció alapértéke `<1->`. A lehetséges opciók:

`duration=<idő>` Ennyi másodpercig tart az effekt.

`direction=<szög>` Ennyi fokos szögben megy végbe az effekt. A `<szög>` lehetséges értékei 0, 90, 180, 270, illetve `\transglitter` esetén még lehet 315 is.

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

```
\transduration<spec2>{<idő>}
```

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

## 21.5. A prezentáció tagolása

### 21.5.1. Címoldal

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

```
\title[<rövid cím>]{<cím>}
\subtitle[<rövid alcím>]{<alcím>}
\author[<rövid név>]{<név>}
\institute[<intézet rövid neve>]{<intézet>}
\date[<rövid dátum>]{<dátum>}
\logo{\includegraphics[<opció>]{<képfájl>}}
\titlegraphic{\includegraphics[<opció>]{<képfájl>}}
```

Ezután a címoldal a dokumentumtestben a következőképpen hozható létre:

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

vagy

```
\maketitle
```

ami a

```
\begin{frame}
\titlepage
\end{frame}
```

kóddal ekvivalens.

## 21.5.2. A főszöveg tagolása

A `beamer`-ben a szöveg tagolása az `article` osztályhoz hasonló, de nincs paragrafus és alparagrafus. Ha nagyon hosszú prezentációt készít, akkor szükség lehet a több részre való bontásra. Új részt a

```
\part[<rész rövid címe>]{<rész címe>}
```

parancs kereten kívüli kiadásával indíthat. 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 akarja, hogy az előző parancs kiadásakor egy külön keret jöjjön létre a címmel, akkor használhatja az `\AtBeginPart`, `\insertpart`, `\insertpartnumber` és `\insertromanpartnumber` parancsokat. Például írja 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 egy keretet generál a rész sorszámával és címével. Új szakaszt a

```
\section[<szakasz rövid címe>]{<szakasz címe>}
```

parancs kereten kívüli kiadásával indíthat. Az opcióban megadott cím alapesetben a szakasz 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 akarja, hogy az előző parancs kiadásakor egy külön keret jöjjön létre a címmel, akkor használhatja az `\AtBeginSection`, `\insertsection` és `\insertsectionnumber` parancsokat. Például írja be a következőket a preambulumba:

```
\AtBeginSection{
\begin{frame}[plain]
\begin{center}
{\Large\insertsectionnumber. \insertsection\}
\end{center}
\end{frame}}
```

Ezután a

```
\section{Szakasz címe}
```

parancs kiadása egy keretet generál a szakasz sorszámával és címével. Értelemszerű változtatásokkal hasonlóan járhat el az alszakasz és al-alszakasz esetében is.

### 21.5.3. Tartalomjegyzék

A rész, szakasz, alszakasz, al-alszakasz tartalmi felosztást linkek formájában megjelenítheti egy külön keretben is. Ha nem használt `\part` parancsot, akkor például a címoldal után beírhatja a következő kódot:

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

Ha használt `\part` parancsot, akkor az előző kód csak akkor hatásos, ha a `\part` parancs kiadása után van. Ekkor a hatása nem az egész tartalomjegyzék, hanem csak az adott részé. Ha azt akarja, hogy minden rész tartalma még a `\part` parancs előtt megjelenjen például a címoldal után közvetlenül, akkor a következőt teheti:

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

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

Ha a tartalomjegyzéket tartalmazó keretben az első szakaszt kivéve minden szakasz címe elé egy `\pause` parancs hatását akarja elérni, akkor használja a `\tableofcontents` parancs `pausesections` opcióját. Ha a `pausesubsections` opciót használja, akkor azt a hatást érjük 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 akar például al-alszakasz címeket, vagy az éppen nem aktuális címeket csak halványan akarja 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 készít al-tartalomjegyzéket, akkor a stílusokat kombinálhatja 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.
subsubsectionstyle=<stílus1>/<stílus2>/<stílus3>
    <stílus1> Aktuális alszakasz címének stílusa.
    <stílus2> Aktuális szakasz többi alszakasz címének stílusa.
```

$\langle st\acute{il}us3 \rangle$  Többi alszakasz címének stílusa (`hide` esetén nemcsak ezen alszakasz címei, hanem azok al-alszakasz címei sem jelennek meg a tartalomjegyzékben).

`subsubsectionstyle= $\langle st\acute{il}us1 \rangle / \langle st\acute{il}us2 \rangle$`

$\langle st\acute{il}us1 \rangle$  Aktuális al-alszakasz címének stílusa.

$\langle st\acute{il}us2 \rangle$  Többi al-alszakasz címének stílusa.

`subsubsectionstyle= $\langle st\acute{il}us1 \rangle / \langle st\acute{il}us2 \rangle / \langle st\acute{il}us3 \rangle$`

$\langle st\acute{il}us1 \rangle$  Aktuális al-alszakasz címének stílusa.

$\langle st\acute{il}us2 \rangle$  Aktuális alszakasz többi al-alszakasz címének stílusa.

$\langle st\acute{il}us3 \rangle$  Többi al-alszakasz címének stílusa.

`subsubsectionstyle= $\langle st\acute{il}us1 \rangle / \langle st\acute{il}us2 \rangle / \langle st\acute{il}us3 \rangle / \langle st\acute{il}us4 \rangle$`

$\langle st\acute{il}us1 \rangle$  Aktuális al-alszakasz címének stílusa.

$\langle st\acute{il}us2 \rangle$  Aktuális alszakasz többi al-alszakasz címének stílusa.

$\langle st\acute{il}us3 \rangle$  Aktuális szakasz többi al-alszakasz címének stílusa.

$\langle st\acute{il}us4 \rangle$  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}
```

#### 21.5.4. Irodalomjegyzék

Irodalomjegyzéket pontosan úgy készíthet 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}
```

## 21.6. Tartalmi elemek

### 21.6.1. Listák

A `beamer` betölti az `enumerate` csomagot. Ez nem kompatibilis a `paralist` csomaggal, így azt ne töltsse be. Ezért nem használhatja a `compactenum` és `compactitem` listakörnyezeteket sem. Ha a listákat függőleges extra térközök nélkül akarja, akkor a `frame` környezetet `squeeze` opcióval töltsse be. A standard környezetek használhatók: `itemize`, `enumerate`, `description`. Ezen környezeteknek nincs, de az `\item` parancsnak van overlay specifikációja, melynek alapértéke `<1->`. Például

```
\begin{frame}
```

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

Ha az alap overlay specifikációt egy adott listában át akarja állítani például <+> értékre, akkor azt az alábbi módon teheti meg. (Ez ekvivalens az előző kóddal.)

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

Ha az alap overlay specifikációt egy adott keret minden listájára át akarja állítani például <+> értékre, akkor azt az alábbi módon teheti meg.

```
\begin{frame}[<+>]
\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 például `t` opció, akkor az előző kódban az 1. sort így módosítsa:

```
\begin{frame}[<+>][t]
```

Ha az `\item` parancsban egyszerre használ overlay specifikációt és opciót, akkor azt ebben a sorrendben tegye. Például

```
\item<+>[--]
```

Ha egy adott számozott lista adott szintjének számozását akarja megváltoztatni, akkor használhatja a

```
\begin{enumerate}[<stílus>]
```

környezetnyitást, pontosan úgy, mint például az `article` osztályban. Ha az `enumerate` környezetnél az opción túl még az alap overlay specifikációt is be akarja állítani például <+> értékre, akkor azt így lehet megtenni:

```
\begin{enumerate}[<+>][<stílus>]
```

### 21.6.2. 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.

### 21.6.3. Tömbök, tételszerű környezetek

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

```
\begin{block}<spec>{<tömb címe>}
<szöveg>
\end{block}
```

Az alap overlay specifikáció `<1->`. Ha ezt át akarja 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 az előzőhöz 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.

Az egyik különbség, hogy a definiált tételszerű környezetek overlay specifikációval is használhatók (alapérték `<1->`). Ha ezt át akarja 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. Ha magyar nyelv esetén mégis szeretne tételszámozást, akkor használhatja a következő megoldást:

```
\setbeamertemplate{theorems}[default]
\newtheorem{tetel}{\inserttheoremnumber. tétel}
```

### 21.6.4. Dobozok

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 szin}{fg=blue,bg=yellow}
```

sajat szín néven definiál egy olyan színösszeállítást, amelyben a háttér sárga, az előtér, azaz a tartalom pedig kék. Az egyik beamer bekezdésdoboz a következő:

```
\begin{beamercolorbox}[\langle opció \rangle]{\langle színösszeállítás \rangle}
\langle doboz tartalma \rangle
\end{beamercolorbox}
```

Az opciók:

`wd`=*szélesség* Doboz szélessége (alapérték `\textwidth`).  
`dp`=*mélység* Doboz mélysége.  
`ht`=*magasság* Doboz magassága.  
`left` doboz tartalma balra zárt.  
`right` Doboz tartalma jobbra zárt.  
`center` Doboz tartalma középre zárt.  
`sep`=*távolság* Doboz tartalma körüli extra tér nagysága.  
`shadow` Doboz árnyékolt.  
`shadow=false` Doboz nem árnyékolt.  
`rounded` Doboz sarkai kerekítettek.  
`rounded=false` Doboz sarkai nem kerekítettek.

Például

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

után

```
\begin{frame}
\begin{beamercolorbox}[wd=6cm,shadow,rounded,center]{sajat szín}
Doboz tartalma
\end{beamercolorbox}
\end{frame}
```

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

```
\begin{beamerboxesrounded}[\langle opció \rangle]{\langle cím \rangle}
\langle doboz tartalma \rangle
\end{beamerboxesrounded}
```

Az opciók:

`width`=*szélesség* Doboz szélessége (alapérték `\textwidth`).  
`shadow=true` Doboz árnyékolt.  
`shadow=false` Doboz nem árnyékolt.  
`lower`=*színösszeállítás* Doboz tartalmának színösszeállítása.  
`upper`=*színösszeállítás* Doboz fejrészének színösszeállítása.

Például

```
\setbeamercolor{sajat szín1}{fg=white,bg=blue}
\setbeamercolor{sajat szín2}{fg=black,bg=yellow}
```

után

```
\begin{frame}
\begin{beamerboxesrounded}[upper=sajat szín1,lower=sajat szín2]{Cím}
Doboz tartalma
\end{beamerboxesrounded}
\end{frame}
```

### 21.6.5. Háttér

A háttér színe a következő kóddal állítható be:

```
\setbeamercolor{background canvas}{bg=<színnév>}
```

Lehetőség van többszínű háttér készítésére is:

```
\setbeamertemplate{background canvas}
[vertical shading][top=<színnév>,middle=<színnév>,bottom=<színnév>]
```

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

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

Ha háttérképet akar a diáknak, akkor a preambulumba írja a következőt:

```
\setbeamertemplate{background canvas}
{\includegraphics[width=\paperwidth]{<kép>}}
```

Ekkor a `<kép>` minden dia háttérén megjelenik. Ha ugyanezt egyetlen keret diáira akarja elérni, akkor ezt kell tenni:

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

### 21.6.6. Képek

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

```
\includegraphics<<spec>>[<opció>]{<kép>}
```

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}
```

### 21.6.7. Animáció

Az előző kód 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.png`, `fig-1.png`, `fig-2.png`, ..., `fig-20.png` képekből áll. Ekkor a

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

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 szebb megoldást ad az

```
\animategraphics[<opció>]{<sebesség>}{<alapnév>}{<első>}{<utolsó>} ∈ animate
```

parancs. Ez a kód egy képsorozatot videóként fog lejátszani, feltéve, hogy ez a funkció a pdf nézőben támogatott. Az Adobe Reader ilyen, de a SumatraPDF vagy a TeXstudio beépített pdf-nézője nem. Ezt a parancsot csak olyan keretben használja, ahol egyetlen dia van.

*<sebesség>* Pozitív egész, ennyi kép/másodperc sebességgel játssza le.

*<alapnév>* Például ha a képfájlok sorra `fig0.png`, `fig1.png`, ..., `fig20.png`, akkor ide `fig` kerül.

*<első>* Az előző példában ide 0 kerül.

*<utolsó>* Az előző példában ide 20 kerül.

A lehetséges 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 háttérének a színe.

**buttonfg=<szín>** Lejátszó gombok vonalának a színe. A *<szín>* megadása szürke skálával vagy rgb palettával történhet. Például `buttonbg=0.8` vagy `buttonbg=0.36:0.08:0.88` (Ha a `magyar.ldf`-et használjuk, akkor a kettőspont aktívva tételét ki kell kapcsolni, különben a `buttonbg` és `buttonfg` opciók nem használhatók, csak szürke skálával.)

Az `\animategraphics` parancs természetesen nem csak a `beamer` dokumentumosztályban használható, de akkor az `animate` mellett töltsse be a `graphicx` csomagot is.

Animált gif közvetlenül nem építhető pdf-be. Ilyenkor a gif fájlt konvertálni kell png képekből álló sorozatba, amely már az előző módon megjeleníthető pdf-ben is. A konvertáláshoz használhatja például az [ImageMagick](#) programot. Telepítés után a következő parancssorral végezheti a konvertálást:

```
convert -coalesce <fájlnév>.gif <fájlnév>.png
```

### 21.6.8. Videó

Videó is lejátszható a prezentáció egy keretén belül a következő paranccsal:

```
\movie[<opció>]{<poszter>}{<videófájl>} ∈ multimedia
```

Ügyeljen arra, hogy a videó lejátszása idegen gépen nem feltétlenül fog működni (például, ha a lejátszáshoz szükséges codec nincs telepítve rá).

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

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

Amíg nem indul el a videó, a *<poster>* látható a videónak kijelölt területen, ha csak nem adta meg a `poster` opciót (lásd később). Erre kattintva indul a lejátszás. A *<poster>* lehet szöveg és `\includegraphics` paranccsal betöltött kép is.

A lehetséges opciók:

`width=<szélesség>` A videó szélessége.

`height=<magasság>` A videó magassága.

`poster` Amíg a videó nem indul el, nem a *<poster>* 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. Például `start=5s` azt jelenti, hogy a lejátszási kezdőpont az 5. másodperc.

`duration=<idő>` A videóból milyen hosszú részt játsszon le. Például `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`  $\in$  `multimedia` parancs együttes használatával, hogy a videónak különböző időintervallumait játssza le egy-egy linkre kattintással. Ezeknek a linkeknek ugyanazon a dián kell lenniük, mint ahol a videó van. Például

```
\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 akarja lejátszani, akkor nincs szükség a `multimedia` csomagra:

```
\href{run:<videófájl>}{<link szövege>}
```

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

### 21.6.9. Nagyítás

Lehetőség van arra, hogy a dia egy adott területét kinagyítsa 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 cm  $\times$  3 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 1. dián betölt egy `pic.jpg` képet, majd a másodikon kijelöli a nagyítandó részeket. Ezekre kattintva megnézhetjük a nagyítást.

### 21.6.10. Kereszthivatkozások, nyomógombok

Próbálja 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 fogja tapasztalni, hogy az `\eqref{egyenlet}` által létrehozott 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}
```

Ha egy keret adott diájára akar hivatkozni, akkor használja 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. Ha `\ref` helyett a `\hyperlink` parancsot használja, akkor a link szövegét mi adhatjuk meg. Például

```

\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 nemcsak szöveg, hanem nyomógomb is megadható:

```

\beamerbutton{<gomb szövege>}
\beamergetobutton{<gomb szövege>}
\beamerbutton{<gomb szövege>}
\beamerreturnbutton{<gomb szövege>}

```

Például az előző kód második keretét javítsa ki erre:

```

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

```

A nyomógombok szimbólumait az

```

\insertgotosymbol
\insertskipsymbol
\insertreturnsymbol

```

parancsok átdefiniálásával változtathatja meg. A gomb színeit és a szöveg betűtípusát is átállíthatja. Például

```

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

```

### 21.6.11. Keret ismétlése

Ha egy keretet `label` opcióval töltötte be, akkor lehetőség van az `\againframe` paranccsal a keret tartalmát egy másik ponton is megjeleníteni, esetleg más overlay specifikációval, más opcióval. Például

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