

Introduktion till BEAMER

Robert Parviainen*
Matematiska institutionen
Uppsala Universitet

1 juni 2004

Innehåll

| | | |
|----------|--------------------------------------------|-----------|
| 1 | Introduktion | 3 |
| 1.1 | Ett första exempel | 3 |
| 1.2 | Kompilering | 4 |
| 1.2.1 | Utskrift | 5 |
| 1.2.2 | Hantering av PDF-filer | 5 |
| 2 | Struktur | 5 |
| 2.1 | Frames | 5 |
| 2.1.1 | Overlays | 6 |
| 2.2 | Kommandon för att skapa overlays | 7 |
| 2.3 | Miljöer för att skapa overlays | 7 |
| 2.3.1 | Pauses | 7 |
| 2.3.2 | overlayarea | 8 |
| 2.3.3 | overprint | 8 |
| 2.3.4 | Dold text | 8 |
| 2.3.5 | Verbatim-text | 8 |
| 2.4 | Part, section, subsection | 8 |
| 2.4.1 | Part | 9 |
| 2.4.2 | Titelsida | 9 |
| 2.4.3 | Innehållsförteckning | 10 |
| 2.5 | Referenslista | 10 |
| 2.6 | Kolumner | 10 |
| 2.7 | Boxar | 10 |
| 2.8 | Knappar och länkar | 10 |
| 2.8.1 | Knappar | 11 |
| 3 | Utseende | 11 |
| 3.1 | Teman | 11 |
| 3.2 | Mallar | 12 |
| 3.2.1 | Fördefinierade mallar | 12 |
| 3.3 | Egna teman och mallar | 13 |

*robert@math.uu.se

| | | |
|----------|--------------------------------------------------------------------------------------------|-----------|
| 4 | Grafik | 13 |
| 4.1 | Färg | 13 |
| 4.1.1 | Egna färger | 14 |
| 4.2 | Bilder | 14 |
| 4.2.1 | Skapa bilder | 14 |
| 4.2.2 | Bilder för <code>pdflatex</code> | 15 |
| 4.2.3 | Text i eps och ps-bilder | 15 |
| 4.3 | Importering i <code>L^AT_EX</code> med <code>pgf</code> | 15 |
| 4.3.1 | Transparenta bilder med <code>pgf</code> | 16 |
| 4.4 | Importering i <code>L^AT_EX</code> med <code>graphicx</code> | 16 |
| 4.4.1 | Ljud och film | 16 |
| A | Mer information | 16 |

1 Introduktion

BEAMER är en L^AT_EX-klass för att skapa presentationer. En BEAMER-presentation kommer bäst till sin rätt då den visas med dator och projektor (via PDF-läsare).

Att förbereda en presentation i BEAMER är nästan lika enkelt som att skriva ett vanligt L^AT_EX-dokument — ett par nya kommandon är allt man behöver lära sig.

Den här introduktionen ger en kort handledning i användandet av BEAMER, och de mest användbara finesserna. En utförlig manual för BEAMER finns på projektets hemsida¹.

I avsnitt 1 ges ett första exempel på hur BEAMER kan användas. I avsnitt 2 behandlas de grundläggande kommandona och finesserna — tillräckligt för de flesta tillämpningarna. Hur man enkelt kan styra utseendet beskrivs i avsnitt 3. I avsnitt 4 beskrivs användandet av färger och bilder.

Anmärkning. BEAMER är för närvarande under utveckling, och uppdateras för närvarande flera gånger om året. Den här handledningen beskriver version 2.20. Det mesta som går igenom här bör dock fungera även i framtida versioner.

1.1 Ett första exempel

För att använda BEAMER börjar man med att ange `beamer` som dokumentklass:

```
\documentclass{beamer}
```

En bild i BEAMER kallas `frame`. Presentationen ska delas in i `frames`. För detta används kommandot `\frame{text}`, som anger att en bild, med innehållet `text`, ska skapas.

Det här är ett exempel på hur en komplett (men kort) BEAMER-presentation kan se ut.

```
\documentclass{beamer}
\usepackage[latin1]{inputenc} % för svenska tecken
\begin{document}
  \frame{
    Det här är bild 1.
  }
  \frame{
    Det här är bild 2.
  }
\end{document}
```

Bild 1 består av texten “Det här är bild 1”, bild 2 av texten “Det här är bild 2”. Det första man kan göra för att snygga till presentationen är att lägga till titlar för varje bild. Det görs med kommandot `\frametitle{Bildtitel}`. T.ex. så här;

```
\documentclass{beamer}
\usepackage[latin1]{inputenc} % för svenska tecken
\begin{document}
  \frame{
    \frametitle{Bild 1}
    Det här är bild 1.
  }
  \frame{
    \frametitle{Bild 2}
    Det här är bild 2.
  }
\end{document}
```

¹<http://latex-beamer.sourceforge.net>

Dessa titlar kommer att visas som rubriker i bilderna. Som default är dessa blå. BEAMER använder samma färg (kallad `beamerstructure`) till alla rubriker som har med ”strukturen” att göra. (Utseendet på rubrikerna kan man naturligtvis detaljstyra, se avsnitt 3.) Man kommer också att få sidhuvud och sidfot med diverse information. Nästa steg för en prydlig presentation är att dela in presentationen i avsnitt.

```
\documentclass{beamer}
\usepackage[latin1]{inputenc} % för svenska tecken
\begin{document}
  \section{Avsnitt 1}
  \frame{
    \frametitle{Bild 1}
    Det här är bild 1.
  }
  \section{Avsnitt 2.}
  \frame{
    \frametitle{Bild 2}
    Det här är bild 2.
  }
\end{document}
```

Om man testkör kommer man att se att sidhuvudet nu inkluderar avsnittsrubrikerna. En poäng med detta är att man kan klicka på dessa för att hoppa till angivet avsnitt.

Det övergripande utseendet bestäms av vilket **tema** man använder. Enskilda beståndsdelars (som sidhuvud, sidfot — vilka som används bestäms av det valda temat) utseende bestäms av **mallar**. Hur dessa fungerar beskrivs i avsnitt 3.2.

Anmärkning. BEAMER ger ett dokument med pappersstorlek 128×96 mm. Detta är viktigt att komma ihåg då man använder avancerade kommandon, ändrar typsnittsstorlekar, eller vill skriva ut presentationen.

1.2 Kompilering

BEAMER-presentationer är avsedda att visas som PDF-filer, med Acrobat Reader, xpdf, eller annan läsare. Därför rekommenderas man att kompilera källkoden med `pdflatex`.

```
$pdflatex exempel1.tex
```

Ofta krävs att man upprepar detta, innan man får den avsedda PDF-filen (detta bland annat för att referenser inom dokumentet ska bli korrekta). Man måste också konvertera eventuella `eps`-bilder till PDF, t.ex. med `epstopdf`

```
$epstopdf bild.eps
```

Man kan också gå den långa vägen `tex` \rightarrow `dvi` \rightarrow `ps` \rightarrow `pdf` (med `dvips` och `ps2pdf`). Detta ger ibland dålig kvalitet (texten blir ”suddig”). Detta kan ofta lösas med att anropa `dvips` på följande sätt

```
$dvips -Ppdf exempel1.dvi -o exempel1.ps
eller
$dvips -Pamz -Pcmz exempel1.dvi -o exempel1.ps
```

Man kan också prova att använda paketet `lmodern`.

För att det ska fungera att köra \LaTeX på en BEAMER-fil, kan man behöva ange `dvips` som option i dokumentklassen:

```
\documentclass[dvips]{beamer}
```

1.2.1 Utskrift

Eftersom pappersstorleken är 128×96 mm, bör man ange alternativet “*expand small pages*” om man skriver ut från Acrobat Reader.

Om varje `frame` består av många `slides`, kommer utskriften att innehålla onödigt många sidor. Detta kan förhindras genom att ange `handout` som option.

```
\documentclass[handout]{beamer}
```

 Minimerar antalet `slides` i presentationen.

1.2.2 Hantering av PDF-filer

Ett \LaTeX -paket och ett program är värda att omnämnas. Paketet `pdfpages` används för att inkludera PDF-filer i ett \LaTeX -dokument. Programmet `pdfmerge` kan användas för att sätta ihop flera PDF-dokument till ett stort.

2 Struktur

En BEAMER-presentation är uppbyggd av `frames`. En `frame` kan ses som en bild i en klassisk presentation med OH-projektor. En av fördelarna med att visa presentationen med dator och projektor är att man enkelt kan skapa ”overlays” — en bild som är stegvis uppbyggd, som klassiskt görs med ett papper som täcker delar av bilden, eller med flera bilder som läggs ovanpå varandra. I BEAMER-termologi består varje `frame` av flera `slides`.

Att skapa overlays i BEAMER är enkelt, och möjligheterna är många; man kan inte bara få text att dyka upp, man kan byta typsnitt, ändra färg, och så vidare.

Liksom man strukturerar ett vanligt dokument i avsnitt och delavsnitt är det lämpligt att strukturera sin BEAMER-presentation så. Detta ger ytterligare en fördel gentemot OH-presentationer. Om man vill kan BEAMER lägga in navigeringsverktyg i bilderna. Dessa kan man klicka på för att direkt flytta sig till relevanta avsnitt — smidigt jämfört med att försöka bläddra sig fram till rätt bild.

2.1 Frames

En ny bild startas med kommandot `\frame{text}`, där `text` innehåller vanlig \LaTeX -text (med undantaget att de olika verbatim-miljöerna är speciella, se avsnitt 2.3.5), och speciella BEAMER-kommandon.

Varje `frame` består av 8 delar, av vilka endast 7 och 8 anges när bilden skapas.

1. Ett sidhuvud
2. En sidfot
3. En vänstermarginal
4. En högermarginal
5. Navigeringssymboler
6. En logotyp
7. En rubrik (`frametitle`)

8. Text

Hurvida de första 5 delarna visas, och hur de ser ut bestäms av det valda temat, och eventuella mallar. För att använda en logotyp, anges följande kommando.

`\logo{text}` `text` är lämpligen ett kommando för att inkludera en bild.

Rubriken anges med kommandot `\frametitle{rubrik}`. Den visas normalt högst upp på bilden.

2.1.1 Overlays

För att skapa effekten av bilder som läggs på varann används något av BEAMERS **overlay**-kommandon.

En overlayspecifikation är en kommaseparerad lista av delbilder (**slides**). Sammanhängande följder kan förkortas; 2,3,4 kan skrivas 2-4. BEAMER räknar själv ut hur många delbilder varje bild har. Man kan därför skriva 2- för att ange alla delbilder från och med 2 och framåt, och -3 för att ange alla delbilder fram till och med 3.

Många kommandon och miljöer vet hur overlayspecifikationer ska hanteras. För att ändra typsnitt i en eller flera **slides**, används t.ex. något av följande standardkommandon: `\textbf`, `\textit`, `\textsl`, `\textrm` eller `\textsf`. BEAMER definierar också `\color`, `\alert`, `\structure`, som förklaras i avsnitt 4.1.

Overlayspecifikationen anges direkt efter kommandot, inom `<>`-paranteser, som i det här exemplet.

```
\frame{
  \textbf{Rad som är i fetstil i alla slides}
  \textbf<1-2>{Rad som är i fetstil i slide 1 och 2}
  \textbf<2->{Rad som är i fetstil i slide 2 och framåt}
  \textbf<3>{Rad som är i fetstil i slide 3}
}
```

I listor (`itemize`, `enumerate`, `description`) kan `\item` ges med en overlayspecifikation — som `\item<x> ...`

Det finns fördefinierade miljöer för sats, bevis, osv, med engelska rubriker. Dessa är **theorem**, **proof**, **corollary**, **fact**, **example**, **examples**. (Exempel typsätts i en egen färg — som default blir exemplen gröna). En overlayspecifikation anger att t.ex. beviset endast ska visas i angivna delbilder, som i följande exempel.

```
\frame{ \frametitle{Main Theorem}
  \begin{theorem}
    The Theorem is always shown.
  \end{theorem}
  \begin{proof}<2->
    The Proof only in slide 2 and onwards.
  \end{proof}
}
```

Det finns också en generell miljö, **block**, som är ett enkelt sätt att t.ex. skapa motsvarande på svenska.

```
\begin{block}{titel}<x>
\end{block}
```

Detta skapar ett block, med titeln `titel`, som visas i delbilderna givna av `<x>`. Om man ska ha sin presentation på svenska, och har många, säg, satser och bevis, kan man givetvis skapa sin

egen miljö, precis som i ”vanlig” L^AT_EX. Man kan också få sina egna miljöer att hantera overlay-specifikationer, med kommandona `\newcommand<>`, `\renewcommand<>`, `\newenviroment<>` samt `\renewenviroment<>`. Dessa är utvidgningar av de vanliga L^AT_EX-kommandona, som dock har en extra parameter; overlayspecifikationen. Vi nöjer oss här med ett exempel, som definerar kommandot `\boldgreen`

```
\newcommand<>{\boldgreen}[1]{\color#2{green}\textbf#2{#1}}
```

2.2 Kommandon för att skapa overlays

`\only<x>{text}` Anger att `text` endast ska visas i delbilderna givna av `x`. I övriga delbilder ignoreras `text`. Speciellt tar `text` ingen plats.

`\uncover<x>{text}` Anger att `text` endast ska visas i delbilderna givna av `x`. I övriga delbilder tar dock `text` fortfarande plats — `text` är bara osynligt.

`\visible<x>{text}` Som `uncover`. Dock kan texten aldrig vara transparent (avsnitt 2.3.4).

`\invisible<x>{text}` Anger att `text` inte ska visas i delbilderna givna av `x`. Dock tar `text` fortfarande plats — `text` är bara osynligt.

`\alt<x>{text 1}{text 2}` Anger att `text 1` ska visas, utom i delbilderna givna av `x`, då `text 2` visas.

`\temporal<x>{text 1}{text 2}{text 3}` Anger att `text 1` visas före `x`, `text 2` i delbilder givna av `x`, och `text 3` efter `x`.

2.3 Miljöer för att skapa overlays

2.3.1 Pauses

Miljön `pauses` är ett enkelt, men föga flexibelt, sätt att göra overlays med. Mellan `\begin{pauses}` och `\end{pauses}` kan man ge kommandot `\pause`. I den första delbilden visas all text fram till det första `\pause`-kommandot. I den andra all text fram till det andra `\pause`-kommandot, osv. Man kan nollställa `pause`-räknaren med `\unpause`.

```
\frame{
  Visas alltid.
  \begin{pauses}
    Visas alltid.
    \pause
    Visas i delbild 2 och framåt.
    \pause
    Visas i delbild 3 och framåt.
    \unpause
    Visas alltid.
    \pause
    Visas i delbild 2 och framåt.
  \end{pauses}
}
```

De ovan nämnda kommandona kan ibland ge en oönskad bieffekt: text som är gemensam för två delbilder kan ibland flytta säg något — detta upplevs av de flesta som irriterande — och uppkommer då olika texter som visas på samma plats i olika delbilder har olika höjd. Det finns två miljöer som fixar detta.

2.3.2 overlayarea

```
\begin{overlayarea}{bredd}{höjd}
\end{overlayarea}
```

Allt inom miljön typsätts i en rektangel med angivna mått. Då rektangelns mått är fixt under alla delbilder, ger följande exempel önskat resultat.

```
\begin{overlayarea}{\textwidth}{3cm}
  \only<1>{Det här visas i delbild 1.\\
    Och är två rader.}
  \only<2>{I delbild 2 endast 1 rad.}
\end{overlayarea}
```

2.3.3 overprint

```
\begin{overprint}{bredd}
  \onslide<x>
  \onslide<y>
  ...
\end{overprint}
```

Inom miljön används `\onslide<x>` istället för `\only<x>`. Allt inom miljön typsätts i en rektangel med angiven bredd, och höjd vald så att den högsta delbilden får plats.

2.3.4 Dold text

För att uppnå effekten av att dold text inte är helt osynlig, utan bara transparent, finns några användbara mallar, som anges med kommandona

```
\beamertemplatecovered
\beamertemplatecoveredmedium
\beamertemplatecoveredhigh
\beamertemplatecovereddynamic
\beamertemplatecovereddynamicmedium
```

2.3.5 Verbatim-text

För att använda verbatim-text i en `frame`, krävs att den endast har en `slide`. Vidare måste detta anges explicit med `\frame[containsverbatim]{...}`

Vägar runt denna begränsning beskrivs i den utförliga manualen.

2.4 Part, section, subsection

Det är lämpligt att dela in presentationen med `\section` och `\subsection`. Notera dock att dessa kommandon **inte** ger en rubrik där dessa används. Syftet är istället att ge användaren möjligheter till snabb navigation vid själva presentationen. Både `\section` och `\subsection` klarar både långa och korta rubriker, och finns i *-form (tas inte med i innehållsförteckningen).

`\section[kort titel]{lång titel}` Om angiven, används den korta titel i eventuella navigationsverktyg.

2.4.1 Part

Riktigt långa presentationer kan delas in i delar med `\part{title}`. Varje `part` får sina egna innehållsförteckningar, navigationsverktyg, osv. Varje del kan få sin egen titelsida genom kommandot `\partpage`.

`\part{titel}` Startar en ny del.

2.4.2 Titelsida

En titelsida skapas enkelt med kommandot `\titlepage`. Kommandot hämtar information från användarens definitioner om författare, titel, datum, osv, som beskrivs nedan. Även om man inte tänker ha en titelsida, kan det vara lönt att definiera dessa variabler — de kan också användas i t.ex. sidhuvudet och sidfoten.

`\titlepage` Anger att en titelsida ska skapas i den aktuella bilden. `\titlepage` hämtar information från följande definitioner.

`\title[kort titel]{lång titel\\kan vara flera rader}` Den korta titeln används i sidhuvud och sidfot.

`\author[kort namn]{långt namn}` Flera namn separeras av `\and`. Om det är flera författare, kan dessas olika arbetsplatser eller adresser anges genom att skriva `\inst{1}`, `\inst{2}`, osv, efter namnet.

`\institute[kort namn]{långt namn}` För att definiera arbetsplats och/eller adress. Då det t.ex. finns flera författare med olika adresser separeras adresserna med `\and`, och adresserna skrivs **efter** `\inst{n}`.

`\date[kort datum]{långt datum}` Som vanligt kan man använda sig av `\today`.

`\titlegraphic{text}` text visas på titelsidan. Avsett för att visa en bild, se avsnitt 4.2.

Exempel:

```
\documentclass{beamer}
\title{Inclusions and non-inclusions}
\author[Parvianen, R. \and Wierman, J.~C.]
  {Robert Parviainen\inst{1}\and John C. Wierman\inst{2}}
\institute{
  \inst{1}Department of Mathematics\\
  Uppsala University
  \and
  \inst{2}Applied Mathematics and Statistics\\
  Johns Hopkins University
}
\date[040218]{18 februari 2004}

\begin{document}
  \frame{\titlepage}
  \frame{\tableofcontents}
  \frame{
    \frametitle{Introduction}
    The question...
  }
}
```

...

2.4.3 Innehållsförteckning

En innehållsförteckning skapas med kommandot `\tableofcontents`. Man kan, som i vanlig L^AT_EX, lägga till saker till innehållsförteckningen med kommandona `\addtocontents` och `\addcontentsline`.

2.5 Referenslista

Referenslistor skapas som vanligt med miljön `thebibliography`. En skillnad är att BEAMER som default använder en liten ikon i referenslistan istället för nycklar (som `[1]` eller `Wierman2002`).

Dock bör `bibtex` ej användas, utom för att få en första approximation av referenslistan.

2.6 Kolumner

BEAMER definerar miljöerna `columns` och `column` för att dela in en `frame` i kolumner — speciellt användbart för att infoga en bild sida vid sida med en text.

```
\begin{columns}
  \begin{column}{6cm}
    Första kolumnen, 6 cm bred.
  \end{column}
  \begin{column}{4cm}
    Andra kolumnen, 4 cm bred.
  \end{column}
\end{columns}
```

Båda miljöerna har en valfri option för vertikal alignering; man kan ange `b`, `c` eller `t` för ange alignering för alla kolumnerna (i `columns`) eller för en enskild kolumn (`column`).

2.7 Boxar

Förutom de vanliga L^AT_EX-boxarna (t.ex. `\fbox`) finns i BEAMER `beamerboxesrounded`.

```
\begin{beamerboxesrounded}[options]{titel}
  text
\end{beamerboxesrounded}
```

Detta skapar en box med färgad bakgrund. Om `titel` är angiven visas denna i annan färg. Med `options` kan man ange

- `width=bredd` Som default är bredden presentationens textbredd, `\textwidth`.
- `shadow=true` eller `shadow=false`.

2.8 Knappar och länkar

En fördel med PDF är att man kan använda länkar. En del länkar definieras automatiskt med BEAMER, såsom att rubriker blir klickbara. Det är också enkelt att definera egna länkar. En länk består av ett mål och en klickbar länk.

Målet definieras med

`\hypertarget<x>{namn}{text}` Anger att `text` (kan vara tomt) är målet för länkar till `namn`. Om `hypertarget` används i `frames` med fleras `slides`, **måste** overlayspecifikationen `x` anges.

En länk anges med

`\hyperlink{mål}{text}` Skapar en länk till målet `mål`: då `text` klickas hoppar man till den `slide/frame` där målet `mål` är definerat.

2.8.1 Knappar

BEAMER har 4 kommandon för att rita knappar. Om dessa placeras inom `hyperlink` blir dessa klickbara.

`\beamerbutton{text}` Knapp med texten `text`.
`\beamergetobutton{text}` Pil åt höger, framför `text`.
`\beamerskipbutton{text}` Dubbelpil åt höger, framför `text`.
`\beamerreturnbutton{text}` Pil åt vänster, framför `text`.

3 Utseende

Självklart vill man kunna ändra utseendet på sin presentation. Det görs lättast med att använda ett fördefinierat **tema**. Ett tema bestämmer bl.a, sidfot, sidhuvud, sidolister. Man kan också ändra utseendet på enskilda delar med **mallar**

3.1 Teman

De fördefinierade teman som ingår i BEAMER-paketet för tillfället listas nedan, med en kortfattad beskrivning.

`\usepackage{beamerthemebars}` Ger sidhuvud med avsnitts- och delavsnittsubriker, och sidfot med presentationens information.

`\usepackage{beamerthemeclassic}` Ger sidhuvud med avsnitts- och delavsnittsubriker.

`\usepackage{beamerthemelined}` Ger sidhuvud med avsnitts- och delavsnittsubriker, och sidfot med presentationsinformation.

`\usepackage{beamerthemeplain}` Inget sidhuvud, sidfot eller sidolist.

`\usepackage[width = w, dark, tab]{beamerthemesidebar}` Ger en sidolist med bredd `w`, med avsnitts- och delavsnittsubriker, samt presentationsinformation, `dark` och `tab` är valfria, och förändrar utseendet något (prova!).

`\usepackage{beamerthemeshadow}` Ger sidhuvud med avsnitts- och delavsnittsubriker, och sidfot med presentationsinformation.

`\usepackage{beamerthemesplit}` Ger sidhuvud med avsnitts- och delavsnittsubriker, och sidfot med presentationsinformation.

`\usepackage{beamerthemetree}` Ger sidhuvud med avsnitts- och delavsnittsubriker.

`\usepackage{beamerthemetreearrows}` Ger sidhuvud med avsnitts- och delavschnittsubriker.

De flesta av dessa använder sig av presentationsinformationen, som titel, författare, osv.

3.2 Mallar

3.2.1 Fördefinierade mallar

Nedan anges de fördefinierade mallarna, listade i kategorier, med kommandon för att använda mallen. Kommandonamnen är i stor grad beskrivande — och som vanligt är det lättast att prova sig fram. Vi beskriver dock de mest användbara först lite närmare.

Några användbara mallar:

`\beamertemplatenavigationssymbolseempty` Stänger av navigationslisten, som default visas i sidfoten.
`\beamertemplatesolidbackground{färg}` Anger att `färg` ska användas i bakgrunden.
`\beamertemplateheadempty` Anger att inget sidhuvud ska visas.
`\beamertemplatefootempty` Anger att ingen sidfot ska visas.

Dold text:

`\beamertemplatecovered`
`\beamertemplatecoveredmedium`
`\beamertemplatecoveredhigh`
`\beamertemplatecovereddynamic`
`\beamertemplatecovereddynamicmedium`

Titelsidor:

`\beamertemplatelargetitlepage`
`\beamertemplateboldtitlepage`
`\beamertemplatelargepartpage`
`\beamertemplateboldpartpage`

Bakgrund:

`\beamertemplatesolidbackground{färg}`
`\beamertemplateshadingbackground{färg 1}{färg 2}`
`\beamertemplategridbackground[spacing]`

Innehållsförteckning:

`\beamertemplateplaintoc`
`\beamertemplateballsectiontoc`
`\beamertemplatenumberedsectiontoc`
`\beamertemplatenumberedcirclesectiontoc`
`\beamertemplatenumberedballsectiontoc`
`\beamertemplatenumberedsubsectiontoc`

Referenslista:

`\beamertemplatetextbibitems`
`\beamertemplatearrowbibitems`
`\beamertemplatebookbibitems`
`\beamertemplatearticlebibitems`

Frametitle:

`\beamertemplateboldcenterframetitle`
`\beamertemplatelargeframetitle`

Sidhuvud och sidfot:

`\beamertemplateheadempty`
`\beamertemplatefootempty`
`\beamertemplatefootpagenumber`

Knappar:

`\beamertemplateoutlinebuttons`
`\beamertemplatesolidbuttons`

Navigation:

`\beamertemplatecircleminiframe`
`\beamertemplatecircleminiframeinverted`
`\beamertemplatesphereminiframe`
`\beamertemplatesphereminiframeinverted`
`\beamertemplateboxminiframe`
`\beamertemateticksminiframe`
`\beamertemplatenavigationsymbolsemt`
`\beamertemplatenavigationsymbolsframe`
`\beamertemplatenavigationsymbolsvertical`
`\beamertemplatenavigationsymbolshorizontal`

Figur- och tabellrubriker:

`\beamertemplatecaptionwithnumber`
`\beamertemplatecaptionownline`

Listor:

`\beamertemplatedotitem`
`\beamertemplateballitem`
`\beamertematetriangleitem`

Blockmiljön:

`\beamertemplateboldblocks`
`\beamertematelargeblocks`
`\beamertempleroundedblocks`
`\beamertempleroundedblocks`
`\beamertemplateshadowblocks`

3.3 Egna teman och mallar

Teman och mallar kan naturligtvis defineras av användaren också. Teman som följer Uppsala Universitets grafiska profil finns tillgängliga på <http://www.math.uu.se/robert/>. För information om hur man själv gör egna teman, se den utförliga manualen till BEAMER.

4 Grafik

4.1 Färg

Normalt klarar man sig utan att definera egna färger. BEAMER är konstruerat för att ge ett enhetligt utseende. Som default skrivs allt som har med strukturen att göra i färgen `beamerstructure`. Den är blå som default. För att använda sig av den färgen finns också ett kommando `\structure`.

`\structure<x>{text}`

Notera att `\structure` klarar overlayspecifikationer.

Det finns också ett kommando `\alert`, för att markera viktiga ord eller meningar.

`\alert<x>{text}`

Som default används rött av `alert`. Detta kan ändras genom att omdefiniera färgen `beameralert`. Alla exempel skrivs i grönt som default.

Dessa grundfärger kan ändras genom att ange en option i dokumentklassen. Fördefinierade färgteman är `red`, `brown`, och `blackandwhite`. T.ex. ger

```
\documentclass[brown]{beamer}
```

 en presentation som går i bruna toner.

4.1.1 Egna färger

BEAMER använder sig av paketeten `xcolor` och `xxcolor` för att hantera färger. Båda är utvidningar av `color`. Detta ger möjligheten till avancerade kommandon för att definera och blanda färger. Man kan dock fortfarande ange färger på det sätt som de flesta torde vara bekant med; med `rgb`-värden.

För att ange att en text ska vara t.ex. röd, skrivs

```
\color[rgb]{1,0,0}{Den här texten blir röd}.
```

Om man ska använda samma färg många gånger, kan det vara värt att definera den en gång för alla (de vanligast färgerna (`red`, `blue`, `green`, ...) är också fördefinierade). För att definera, säg, färgen `uured`, skrivs

```
\definecolor{uured}{rgb}{0.6,0,0}
```

Den kan nu användas genom

```
\color<2>{uured}{Den här texten blir uured, men bara i slide 2}.
```

Så för att ändra struktur- och alert-färgen kan man skriva t.ex.

```
\definecolor{beamerstructure}{rgb}{0.8,0.8,0.8}  
\definecolor{beameralert}{rgb}{0,0,0.6}
```

4.2 Bilder

BEAMER kräver att man också har paketet `pgf` — ett paket för att hantera figurer och bilder. Fördelen med `pgf` gentemot `graphicx` är att man kan använda transparenta bilder. (Dessutom ger `pgf` många kommandon för att rita figurer.)

4.2.1 Skapa bilder

Idealiskt bör man försöka skapa sina bilder i `eps`- eller `ps`-format. Desas kan skalas utan att förlora i kvalitet — till skillnad från bitmap-bilder (`jpg`, `gif`, `png`, osv.).

Diagram. För att rita diagram och enkla figurer är det lämpligt att använda `xfig` eller `tgif` (de är ganska lika, och det är mest en smaksak vilket man föredrar). De är program för att skapa vektor-grafik — bilderna beskrivs av enkla linjer och kurvor — och inte av vilken färg enskilda punkter har, och kan därför skalas.

Dessa program kan spara (exportera) bilderna i `eps`-format — idealiskt för att inkluderas i \LaTeX -dokument.

Man också använda ett \LaTeX -paket för att rita figurer för hand. Paketet `pgf` (som krävs för BEAMER) ger smidiga verktyg för detta.

Funktionsplottar, histogram, osv. Bilder som skapas i t.ex. matematikprogram som Matlab eller Maple kan naturligtvis också användas. Om det inte finns något alternativ för att spara som

ps eller eps, kan man alltid välja att skriva bilden till fil istället för till skrivare. Då skapas en ps-fil.

För den som vill ha extrasnygga plottar eller histogram rekommenderas `gnuplot` eller `gri`.

Foton. Digitaliserade foton bör sparas som jpg eller png. (Dessa format hanteras av `pdflatex`. Av dessa två är png att föredra, då jpg-bilder kan förlora mycket kvalitet vid komprimeringen.) Om bilden behöver retuscheras, är GIMP ett naturligt val. GIMP klarar nästan allt som den mer kända konkurrenten Photoshop, men har fördelen att finnas både i Linux- och Windowsversion, och är framförallt gratis.

4.2.2 Bilder för `pdflatex`

`pdflatex` klarar bilder i formaten PDF, jpg och png. PostScriptbilder (`eps`, `epsi` och `ps`) måste därför konverteras till PDF. För detta kan man använda `epstopdf`:

```
$epstopdf psbild.eps
```

4.2.3 Text i eps och ps-bilder

Om man gjort en snygg bild i t.ex. `xfig`, men är missnöjd utseendet på texten (vanligt om man använder matematiska formler i bilden), kan man använda `psfrag` för att få samma snygga typsnitt som i övriga texten. Det är ett \LaTeX -paket som byter ut text i en `eps`- eller `ps`-bild. Det används så här. paketet laddas med `\usepackage{psfrag}`. För att byta ut text används följande kommando

`\psfrag{pstext}{latextext}` Byter ut `pstext` mot `latextext`, i **alla** `eps`- och `ps`-bilder som innehåller texten `pstext`. Man kan också ange alignering och skalning av texten.

4.3 Importering i \LaTeX med `pgf`

För att använda en bild i \LaTeX med `pgf` krävs att bilden är deklarerad. Detta görs med kommandot `\pgfdeclareimage`, innan bilden används för första gången. Efter att bilden är deklarerad, används kommandot `\pgfuseimage` för att använda bilden.

I `pgf` anges bildernas namn **utan** filändelse! Detta är till för att underlätta då man både vill använda `pdflatex`, och \LaTeX och `dvips`. Då PDF ska genereras testas ändelserna `.pdf`, `.jpg`, `.jpeg`, `.png`. Då PostScript ska generas testas ändelserna `.eps`, `.epsi`, `.ps`.

`\pgfdeclareimage[options]{bildnamn}{filnamn}` Deklarerar att bilden `bildnamn`, med filnamn `filnamn.xxx` kan användas.

Med `options` kan man ange:

- `height=höjd` Bestämmer höjden.
- `width=bredd` Bestämmer bredden.
- `mask=masknamn` Anger att en mask ska användas. Detta används för att ge transparenta bilder (gäller `jpg` och `png`).

Om endast en av `height` eller `width` anges, behåller bilden sina ursprungliga proportioner.

`\pgfuseimage{bildnamn}` Inkluderar den tidigare deklarerade bilden `bildnamn`.

4.3.1 Transparenta bilder med pgf

Med hjälp av en mask kan `pgf` simulera bilder med transparenta områden. Masken är en gråskalebild i samma dimensioner som den ursprungliga. En vit pixel i masken anger att motsvarande pixel i bilden ska vara transparent, en svart att bildens ursprungliga pixelfärg ska användas, och gråa att en blandning av ursprungliga färgen och bakgrundsfärgen ska användas.

4.4 Importering i L^AT_EX med graphicx

Med paketet `graphicx` importeras bilder med kommandot `\includegraphics`.

```
\includegraphics[options]{filnamn}
```

Notera att till skillnad från `pgf` ska filändelsen anges! Det finns mängder med `options`. Den mest användbara är `scale=skala` som anger skalan.

Anmärkning. I BEAMER hanterar både `pgfuseimage` och `includegraphics` overlayspecifikationer.

4.4.1 Ljud och film

Från och med version 2.20 har BEAMER också ett tilläggspaket med kommandon för att inkludera film- och ljudklipp. Vilka format som kan användas beror på vilket operativsystem och med vilken PDF-läsare man ska visa presentationen. För mer information, se senaste BEAMER-manualen.

A Mer information

Beamer, pgf och xcolor

Se BEAMER-projektets hemsida, <http://latex-beamer.sourceforge.net>.

pdfpages

Se <http://www.ctan.org/tex-archive/help/Catalogue/entries/pdfpages.html>

pdfmerge

Hjälp fås med kommandot

```
$pdfmerge -h
```

psfrag

Se <http://www.ctan.org/tex-archive/help/Catalogue/entries/psfrag.html>.

graphicx

Se <http://www.ctan.org/tex-archive/help/Catalogue/entries/graphicx.html>.