Hour 22

Slide Preparation II

Preparation procedures of simple slides under different themes are discussed in Hour 21. This Hour is devoted on some more important topics on slide preparation, such as piece-wise presentation of items, BEAMER environments, hyperlinking, etc.

22.1 Piece-Wise Presentation (BEAMER Overlays)

It is always preferred to present (cover, uncover/show, or highlight) the contents of a frame piece-wise, instead of displaying the entire contents in one go only as in the cases discussed in Hour 22. There are many processes for preparing frames, so as to present the contents piece-wise sequentially or in other orders. If the contents of a frame are to be presented piece-wise, the BEAMER will automatically split the frame into that number of slides.

22.1.1 Table of Contents

If a presentation is prepared section-wise, the table of contents of the presentation can be generated using the **\tableofcontents** command. Moreover, unlike in standard LaTeX, the BEAMER class also allows **\tableofcontents** to take an optional argument for creating certain special effects through various options. The very first such option is **pausesections** or **pausesubsections** (only one of them is to be used in a presentation). The **pausesections** option, to be used as **\tableofcontents[pausesections]**, internally issues a **\pause** command (§22.1.2 discusses **\pause** in detail) before each section entry in the table of contents, so as to uncover them piece-wise sequentially. The **pausesubsections** option does the same thing but in the case of subsections.

The second useful option is **currentsection**. It uncovers only the current section and its subsections under the table of contents, making it useful before starting the presentation of a new section. To get the effect, the table of contents is to be

D. Datta, *LaTeX in 24 Hours: A Practical Guide for Scientific Writing*,
DOI 10.1007/978-3-319-47831-9_22

repeated in an additional frame, say as \frame[t]{\tableofcontents[currentsection]}, immediately after each \section[ ]{ } command. Similar to currentsection, BEAMER defines the currentsubsection option also, which uncovers only the current subsection in the current section under the table of contents. To get this effect, the table of contents is to be repeated in an additional frame, say as \frame[t]{\tableofcontents[currentsubsection]}, immediately after each \subsection[ ]{ } command.

22.1.2 Uncovering Sequentially Using the \pause Command

The easiest way to uncover the contents of a frame is to use the \pause command, which pauses the display of the remaining contents/slides of the frame once a \pause is encountered. If it is inserted in multiple places in a frame, the contents of the frame will be uncovered only up to the first \pause in the first slide, up to the second \pause in the second slide, up to the third \pause in the third slide, and so forth, i.e., the effect of a \pause is terminated upon reaching at the next \pause or \onslide (§22.1.4 on the next page discusses \onslide) or the end of the frame.

An illustrative frame with the use of \pause is shown in Table 22.1 (output is not shown). Since \pause is inserted four times, the compilation will automatically split the frame over five slides as follows: the first slide to uncover only \frametitle{ }, the second slide to uncover up to the first \item, the third slide to uncover up to the second \item, the fourth slide to uncover up to the third \item, and the fifth slide to uncover up to the fourth \item, which is also the end of the frame.

Table 22.1 Uncovering slide contents piece-wise using the \pause command

```
\begin{frame}[t]
  \frametitle{Seasons over the year} \pause
    \begin{enumerate}
    \item Summer \pause
    \item Autumn \pause
    \item Winter \pause
    \item Spring.
    \end{enumerate}
\end{frame}
```

22.1.3 Uncovering Sequentially Using the Incremental Specification <+->

As stated in §22.1.2, the \pause command can be used anywhere for pausing the display of further contents/slides of a frame. However, if all the items of a listing environment (such as enumerate or itemize) are to be uncovered sequentially, instead of \pause at the end of each \item, the optional incremental specification <+->[1] may be used only once at the starting of the environment as shown in the left column of Table 22.2.

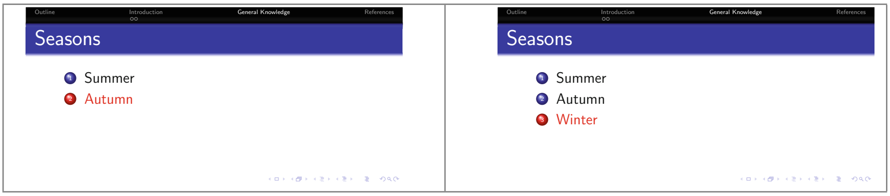
Table 22.2 Uncovering items piece-wise using the incremental specification <+->

\begin{frame}[t]
\frametitle{Seasons}
\begin{enumerate}[<+->]
\item Summer
\item Autumn
\item Winter
\item Spring.
\end{enumerate}
\end{frame}
	\begin{frame}[<+->][t]
	\frametitle{Animals}
	\begin{itemize}
	\item Cow
	\item Goat
	\end{itemize}
	%
	\begin{itemize}
	\item Lion
	\item Tiger
	\end{itemize}
	\end{frame}
		\begin{frame}[t]
		\frametitle{Seasons}
		\begin{enumerate}[<+-|alert@+>]
		\item Summer
		\item Autumn
		\item Winter
		\item Spring.
		\end{enumerate}
		\end{frame}

If a frame contains multiple number of listing environments and their all items are to be uncovered sequentially, instead of inserting the optional incremental specification <+-> in each environment, the same may be inserted only once in the frame as a whole as shown in the second column of Table 22.2. This is applicable not only to listed items, but also to other blocks, like theorem, proof, example, etc.

Further, as shown in the third column of Table 22.2, the incremental specification can be modified as <+-|alert@+> to alert (i.e., highlight) the current item by red color upon uncovering it. In this case, once the next item is uncovered, it will be highlighted quitting that of the previous item. The second and third slides of this frame under the Frankfurt presentation theme (refer §21.4.1 on page 209 for detail) is shown in Table 22.3.

Table 22.3 Slides with incremental overlay-specification under the Frankfurt presentation theme for the input frame of the third column of Table 22.2



Seasons

1. Summer
2. Autumn

Seasons

1. Summer
2. Autumn
3. Winter

22.1.4 *Other Piece-Wise Presentation Specifications**

When it is sought to present (i.e., cover, uncover, or highlight) the contents of a frame piece-wise, the BEAMER automatically splits the frame into that number of slides. A piece-wise presentation (overlay) specification in <> specifies the slide numbers

[1] Specific slides of a frame in which a particular item or block is to be shown/not shown are specified in <>, which in the BEAMER class is known as the overlay specification.

of a frame in which a particular component is to be presented. The rules of such specifications are given in Table 22.4 with some examples. While specifying a slide number in an overlay specification, care must be taken that the corresponding frame has been split at least up to that number of slides, otherwise the intermediate gap will be filled up by generating some identical slides (i.e., slides with the same contents) in between.

Table 22.4 Rules for piece-wise presentation (overlay) specification

Specification	Meaning
<3>	Slide 3 only
<1,2,4>	Slides 1, 2 and 4 only
<3-6>	Slides 3–6 (i.e, slides 3, 4, 5, 6) only
<3->	Slide 3 onward all sides of the frame
<-4>	All starting slides up to slide 4 (i.e., slides 1–4) only
<2,4-6,8,11->	Slides 2, 4–6, 8, 11- (i.e., slides 2, 4, 5, 6, 8, 11 and rest of the slides of the frame).

There are many commands which can take overlay specifications, as shown in Table 22.4, to instruct in which slides their contents are to be presented. Such commands[2] are listed in Table 22.5 with some explanatory texts stating the functions of the commands. Although only one slide number is specified in each command, they (except \alt<>{}{}) can take any type of specifications as shown in Table 22.4. The \alt<>{}{} command can take only one slide, in which its first argument will be shown and the second argument will be shown in all other slides of a frame. On the other hand, the \temporal<>{}{}{} command alternatively handles three arguments.

Table 22.5 Commands which can take overlay specifications

Command	Example stating the function
\textbf<>{}	\textbf<3>{It is boldfaced in slide 3, and in normal fonts in all other slides}
\textit<>{}	\textit<4>{It is in italic fonts in slide 4, and in normal fonts in others}
\textrm<>{}	\textrm<5>{It is in serif fonts in slide 5, and in normal fonts in others}
\textsf<>{}	\textsf<6>{It is in sans serif fonts in slide 6, and in normal fonts in others}
\textsl<>{}	\textsl<7>{It is in slanted shape in slide 7, and in normal fonts in others}
\alert<>{}	\alert<1>{It is shown in red color in slide 1, and in normal color in others}
\color<>[]{}{}	\color<2>[rgb]{0,0,1}{It is in blue color in slide 2, and in normal color in others}
\only<>{}	\only<1>{It is shown in slide 1 only, and the space is freed in others}
\onslide<>{}	\onslide<2>{It is shown in slide 2 only, and the space is kept blank in others}
\uncover<>{}	\uncover<3>{It is shown in slide 3 only, and kept covered or transparent in others}
\visible<>{}	\visible<4>{It is shown in slide 4 only, and the space is kept blank in others}
\invisible<>{}	\invisible<5>{It is not shown in slide 5 only, but the space is kept blank}
\alt<>{}{}	\alt<6>{It is shown in slide 6 only}{It is shown in all slides other than in 6}
\temporal<>{}{}{}	\temporal<7>{It is shown in slides 1–6}{It is shown in slide 7 only}{It is shown in slides 8-}
\item<>	\item<8> It is shown in slide 8 only, and the space is kept blank in others.

[2] In the case of a command having arguments, the overlay specification is put in <> in between the command and its arguments (refer Table 22.5 for examples).

The first argument is shown in all slides prior to the specified ones, the second argument is shown in the specified slides only, and the third argument is shown in all the slides appearing after the specified ones. If discontinuous slides are specified in \temporal<>{}{}{} (say, 2,4,7), the intermediate non-specified slides are also treated to be prior to the specified ones and hence the first argument will be shown in those non-specified intermediate slides. Note that the alert@ command, used in the third column of Table 22.2 for alerting/highlighting an item, can also be used in the commands of Table 22.5 for alerting an item in particular slides, e.g., \item<2-|alert@2> for alerting the item in slide 2, or \item<alert@2,4> for alerting the item in slides 2 and 4.

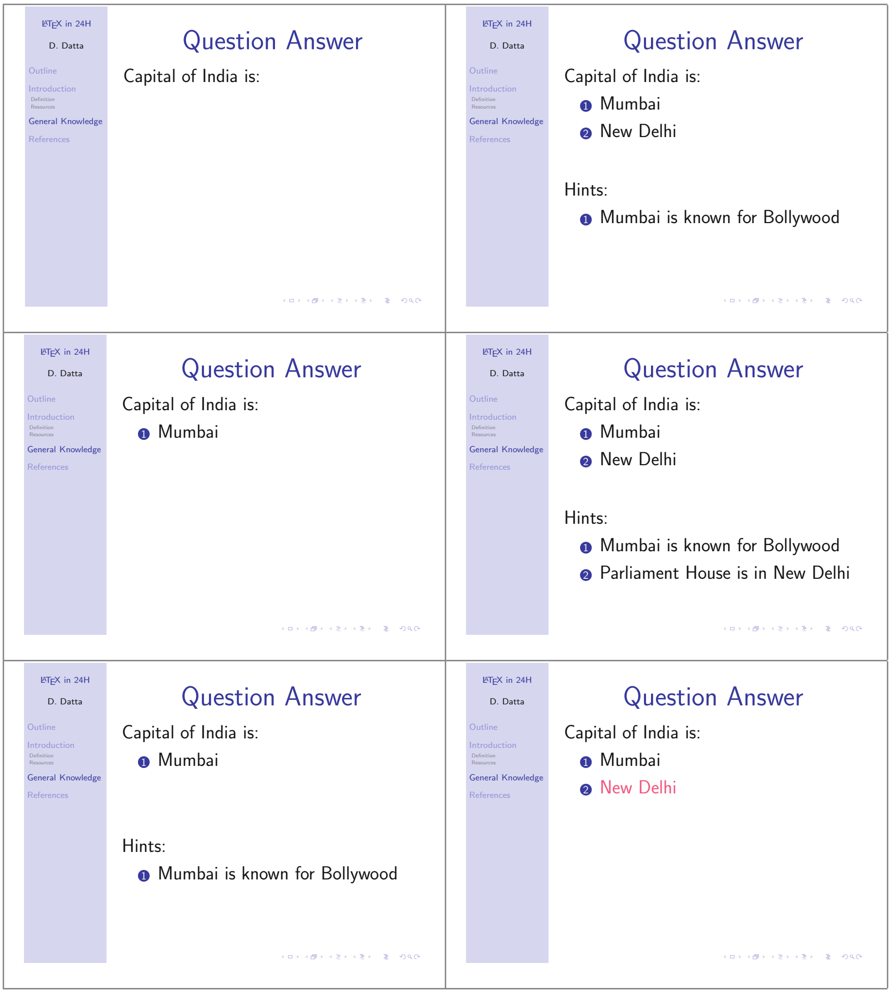
The overlay-specification supported commands shown in Table 22.5 can be used independently as well as in combination. Moreover, different commands can be used for similar effect, which is clear from their functions given in Table 22.5. Such applications through two frames under the Hannover presentation theme (refer §21.4.1 for detail) are shown in Table 22.6 (refer Table 21.1 for detail coding). The first frame will generate two slides, printing “Result: $x + y =$ **?** guess what” in the first slide and “Result: $x + y = \mathbf{z}$ $\checkmark$” in the second slide. On the other hand, the second frame will generate six slides, which are shown column-wise in Table 22.7 on the next page. Notice in these two frames that the slides, in which the items will be presented, are not specified in order. For example, in the second frame, the second \item of the first enumerate environment will be shown in slides 4–6, whereas the \item of the second enumerate environment will be shown in the earlier slide 3 and intermediate slide 5.

Table 22.6 Applications of some overlay-specification supported commands

```
\begin{frame}[t]
    \uncover<1-2>{Result: $x+y$ =}
    \alt<1>{\textbf{?}\quad guess what}{\textbf{$z$}\quad\checkmark}
\end{frame}
%
\begin{frame}[t]
    \frametitle{Question Answer}
    \uncover<1->{Capital of India is:}
    \begin{enumerate}
    \item<2-> Mumbai
    \item<4-> \color<6>[rgb]{0,0,1}{New Delhi}
    \end{enumerate}
    \vskip 10mm
    \only<3-5>{Hints:}
    \begin{enumerate}
    \item<3-5> Mumbai is known for Bollywood
    \item<5> Parliament House is in New Delhi
    \end{enumerate}
\end{frame}
```

However, even if the presenting slides are not specified in order in an input file, the positions of the items in any output slide will be in the same order in which they are inserted in the input file[3], which can be noticed in Table 22.7.

Table 22.7 Slides with overlay specification under the **Hannover** presentation theme for the second input frame of Table 22.6



LaTeX in 24H
D. Datta
Outline
Introduction
Definition
Resources
General Knowledge
References

Question Answer

Capital of India is:

LaTeX in 24H
D. Datta
Outline
Introduction
Definition
Resources
General Knowledge
References

Question Answer

Capital of India is:

1. Mumbai
2. New Delhi

Hints:

1. Mumbai is known for Bollywood

LaTeX in 24H
D. Datta
Outline
Introduction
Definition
Resources
General Knowledge
References

Question Answer

Capital of India is:

1. Mumbai

LaTeX in 24H
D. Datta
Outline
Introduction
Definition
Resources
General Knowledge
References

Question Answer

Capital of India is:

1. Mumbai
2. New Delhi

Hints:

1. Mumbai is known for Bollywood
2. Parliament House is in New Delhi

LaTeX in 24H
D. Datta
Outline
Introduction
Definition
Resources
General Knowledge
References

Question Answer

Capital of India is:

1. Mumbai

Hints:

1. Mumbai is known for Bollywood

LaTeX in 24H
D. Datta
Outline
Introduction
Definition
Resources
General Knowledge
References

Question Answer

Capital of India is:

1. Mumbai
2. New Delhi

[3] Even if the presenting slides are not specified in order in overlay specifications, the positions of the items in any output slide will be in the same order in which they are inserted in the input file.

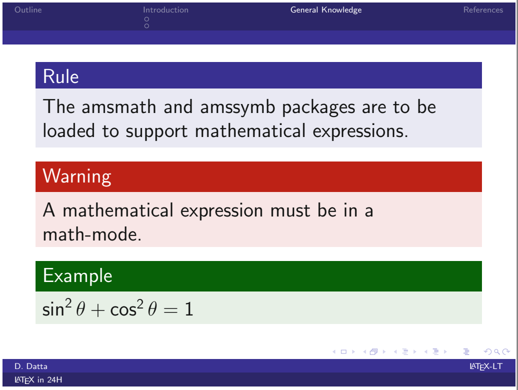
22.2 Environments in BEAMER Class*

The BEAMER class defines the **visibleenv**, **invisibleenv** and **uncoverenv** environments, whose effects are the same, respectively, with those of the **\visible{}**, **\invisible{}** and **\uncover{}** commands as stated in Table 22.5.

The BEAMER class also defines some block-type environments for producing a piece of texts with a user-defined heading. Such environments include **block**, **alertblock** and **exampleblock**, which are generally meant for a normal block of texts, an alerting message, and an example-like illustration, respectively. The **beamerboxesrounded** is another block-type environment, whose contents are framed by a rectangular area with rounded corners. A block-type environment takes its heading as an argument, and it is highlighted in the **alertblock** environment. A block-type environment also has the provision for taking overlay specifications, on either side of the heading argument, so as to uncover the environment in the specified slides only. Applications of the block-type environments under the **Berlin** presentation theme (refer §21.4.1 for detail) are shown in Table 22.8 (refer Table 21.1 on page 206 for detail coding). In this case the **\setbeamertemplate{blocks}[rounded][shadow=true]** command is also inserted in the preamble, where the argument **blocks** is used for defining the style of the background blocks of the environments as follows: the **rounded** option instructs to round off the corners of the blocks (if not by default), and the **shadow=true** option allows to draw shadows behind the blocks as clearly visible in the output slide shown in Table 22.8. In regard of the frame, it is split over three slides as per the used overlay specifications (only the last slide is shown in Table 22.8). Since the **block** environment will be shown in all the slides of the frame, no overlay specification is required to it. Also notice in the **alertblock** and **exampleblock** environments that an overlay specification can be inserted on either side of the heading argument of a block-type environment.

Table 22.8 Slides with block-type environments under the **Berlin** presentation theme

```
\begin{frame}[t]
\begin{block}{Rule}
The amsmath and amssymb ...
\end{block}
%
\begin{alertblock}<2->{Warning}
A mathematical expression ...
\end{alertblock}
%
\begin{exampleblock}{Example}<3>
$\sin^2\theta + \cos^2\theta = 1$
\end{exampleblock}
\end{frame}
```



Rule

The amsmath and amssymb packages are to be loaded to support mathematical expressions.

Warning

A mathematical expression must be in a math-mode.

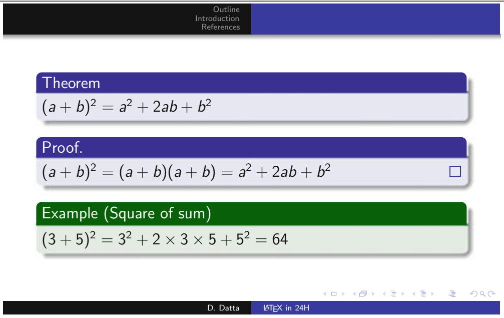
Example

$\sin^2\theta + \cos^2\theta = 1$

Apart from the above, the BEAMER class defines some theorem-like environments for producing a piece of texts with a default heading. Such environments include corollary, definition, definitions, example, examples, fact, proof, and theorem. The theorem-like environments work in a similar way with those of block-type environments, except that they do not need a mandatory heading argument but an optional argument may be provided as an additional heading. The additional heading is generally produced in a pair of parentheses after the default heading, except in the proof environment. In the proof environment, the default heading "Proof." is replaced by the optional heading. Moreover, by default a proof environment is ended by a right aligned \qed symbol (a small square). Applications of the theorem, proof and example environments under the Warsaw presentation theme (refer §21.4.1 for detail) are shown in Table 22.9 (refer Table 21.1 on page 206 for detail coding). In regard of the frame, it is split over three slides as per the used overlay specifications (only the last slide is shown in Table 22.9). Since the theorem environment will be shown in all the slides of the frame, no overlay specification is required to it. Moreover, an optional heading is provided to the example environment, which is produced in a pair of parentheses after the default heading.

Table 22.9 Slides with theorem-like environments under the Warsaw presentation theme

```
\begin{frame}[t]
\begin{theorem}
$(a+b)^2 = a^2 + 2ab + b^2$
\end{theorem}
%
\begin{proof}<2->
$(a+b)^2=(a+b)(a+b)=a^2+2ab+b^2$
\end{proof}
%
\begin{example}<3->[Square of sum]
$(3+5)^2=3^2+2\times3\times5+5^2=64$
\end{example}
\end{frame}
```



Outline
Introduction
References

Theorem

$(a+b)^2 = a^2 + 2ab + b^2$

Proof.

$(a+b)^2 = (a+b)(a+b) = a^2 + 2ab + b^2$ □

Example (Square of sum)

$(3+5)^2 = 3^2 + 2 \times 3 \times 5 + 5^2 = 64$

D. Datta LaTeX in 24H

22.3 Table and Figure in Presentation*

Tables in BEAMER presentation also can be prepared using the standard LaTeX environment table without any option for vertical positioning. Inside the table environment (even without the table environment), either tabular or tabularx environment can be used for generating tabular cells. Two such examples in a single frame are shown in Table 22.10 on the next page. The used overlay specifications split the frame over six slides. The first example is displayed in the first two slides and the second example in the next four slides. In the first example, the entire table is displayed in slide 2 as a whole, as it is inserted in \only<2>. Further, the table is put in \color<2>[ ]{ }{ } for producing it in given color. On the other hand, the table of the second example is inserted in \onslide<4-> for displaying it in all slides starting from slide 4, in such a way that the first row (after the heading row) will be uncovered in slides 5 and 6,

Table 22.10 Table in BEAMER presentation

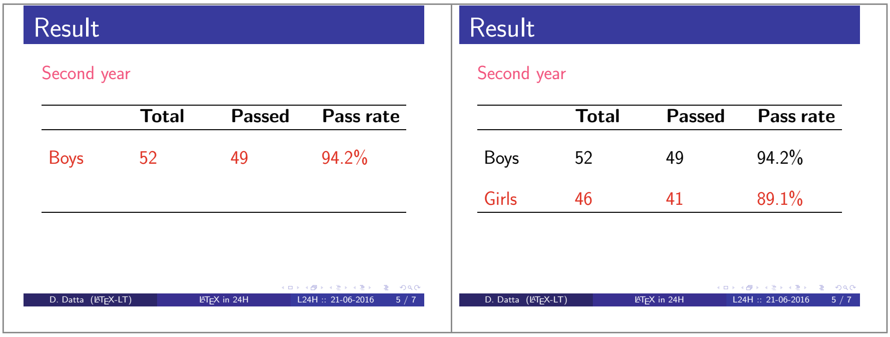
```
\begin{frame}[t]
   \frametitle{Result}
   % Example 1
   \only<1-2> {\color<1-2>[rgb]{1,0.3,0.5}{First year}}
   \only<2> {\color<2>[rgb]{1,0.3,0.5}{
      \begin{table}
      \flushleft
      \begin{tabular}{cccc}
        \hline         & {\bf Total} & {\bf Passed} & {\bf Pass rate}\\
        \hline  Boys   &      56     &      50      &      89.3\%\\
                Girls  &      38     &      36      &      94.7\%\\
        \hline
      \end{tabular}
      \end{table}  }}
   % Example 2
   \onslide<3-> {\color<3->[rgb]{1,0.3,0.5}{Second year}}
   \onslide<4-> {\color<4->[rgb]{0,0,0}{
      \begin{table}
      \begin{tabularx}{\linewidth}{XXXX}
      \hline   &   {\bf Total}&   {\bf Passed}&   {\bf Pass rate}\\
      \hline \uncover<5->{\alert<5>{Boys}}& \uncover<5->{\alert<5>{52}}&
                   \uncover<5->{\alert<5>{49}}& \uncover<5->{\alert<5>{94.2\%}}\\
               \uncover<6>{\alert<6>{Girls}}& \uncover<6>{\alert<6>{46}}&
                   \uncover<6>{\alert<6>{41}}& \uncover<6>{\alert<6>{89.1\%}}\\
      \hline
      \end{tabularx}
      \end{table}  }}
\end{frame}
```

while the second row in slide 6 only (these two slides under the **Madrid** presentation theme are shown in Table 22.11 on the next page). Since more than one cell of a table cannot be presented through a single overlay specification[4], the entry of each cell is inserted in separate `\uncover<>{}`, also in `\alert<>{}` for highlighting them in the specified slides. Note that apart from inserting in `\onslide<4->{}`, the table is also inserted in `\color<4->[rgb]{0,0,0}{}` for producing it in black color, otherwise it would have been produced in the same color specified for the previous slide, i.e., for slide 3.

Like tables, figures from external files also can be inserted in BEAMER presentation using the standard LaTeX environment **figure** without any option for vertical positioning. Inside the **figure** environment (even without the **figure** environment), a figure can be inserted through the standard LaTeX commands `\epsfig{}` and `\includegraphics[]{}`. The `\includegraphics[]{}` command can take overlay specifications also, in which case its form becomes `\includegraphics<>[]{}`, where overlay specifications as stated in Table 22.4 are inserted in `<>`. For example,

[4] More than one cell of a table cannot be presented through a single overlay specification, but each one through a separate specification.

Table 22.11 Table in slides with overlay specification under the **Madrid** presentation theme for the second example of Table 22.10



Result

Second year

	Total	Passed	Pass rate
Boys	52	49	94.2%

D. Datta (LATEX-LT) LATEX in 24H L24H :: 21-06-2016 5 / 7

Result

Second year

	Total	Passed	Pass rate
Boys	52	49	94.2%
Girls	46	41	89.1%

D. Datta (LATEX-LT) LATEX in 24H L24H :: 21-06-2016 5 / 7

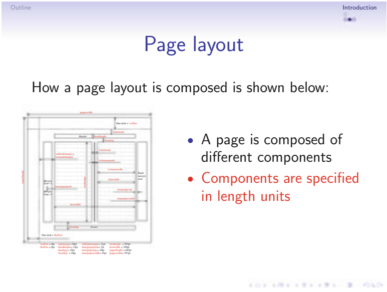
\includegraphics<2>[`width=5`**cm]{**`tiger`**}** for displaying the specified figure in slide 2 only, or **\includegraphics<**`2,5`**>[**`width=5`**cm]{**`tiger`**}** for displaying it in slides 2 and 5.

22.4 Dividing a Frame Column-Wise*

It is often required to present some materials in a frame side-by-side, e.g., showing a figure on the left side and explaining it on the right side. As the example shown in Table 22.12 along with a output slide under the **Singapore** presentation theme, such a frame can be produced through the **columns** environment, inside of which each column is created using a **\column{**`cwidth`**}** command with `cwidth` being the horizontal width of the column.

Table 22.12 Slides with side-by-side materials through the **columns** environment under the **Singapore** presentation theme

```
\begin{frame}
\frametitle{Page layout}
How a page … below:\pause\vskip 5mm
\begin{columns}
\column{0.4\textwidth}
\includegraphics[width=\textwidth]{layout}
%
\column{0.6\textwidth}
\begin{itemize}[<+-|alert@+>]
\item A page is composed of different …
\item Components are specified in length …
\item Length of a component can be …
\end{itemize}
\end{columns}
\end{frame}
```



Page layout

How a page layout is composed is shown below:

- A page is composed of different components
- Components are specified in length units

22.5 Repeating Slides in Presentation*

During a presentation, sometime it is required to go back to a particular slide of a previous frame for explanation purpose. The BEAMER class provides the **label** option and **\againframe<>{}** command for this purpose, avoiding the need to scroll back to the required slide. In this case, the previous frame is to be labeled with the **label** option, say as **\begin{frame}[label=**`stress`**]** to label the frame by `stress`. Then the **\againframe<>{}** command is to be inserted in the required location for reproducing the particular slide of the frame, say as **\againframe<**`3`**>{**`stress`**}** for reproducing slide 3 of the frame which is labeled by `stress`. Without any overlay specification, i.e., inserting as **\againframe{**`stress`**}**, the command will reproduce the entire frame (i.e., its all slides, if produced more than one using any overlay specification in the frame). Note that the **\againframe<>{}** command is to be inserted outside of any frame.

22.6 Jumping (Hyperlink) to Other Slides*

Provision is there in the BEAMER class to jump from the current slide (during presentation) to another slide for various purposes, such as referring to (showing) materials included in another slide, going back after referring, skipping some intermediate slides, etc. It can be done by first issuing a label-word to the frame containing the target slide where the jump is to be made (refer §22.5 for issuing a label-word), and then creating a hyperlinked button in the current slide to go to the target slide by clicking the hyperlinked button. Such a button is created through a button-type command with the texts, to be displayed in the button, as its argument. Then the button is activated (hyperlinked) by inserting it in a hyperlink-type command.

Some commonly used button-type commands are **\beamerbutton{}** (draws a simple button), **\beamergotobutton{}** (draws a button with a right-pointing arrow), **\beamerreturnbutton{}** (draws a button with a left-pointing arrow), and **\beamerskipbutton{}** (draws a button with a double right-pointing arrow). The argument of such a command is the name of the generated button (i.e., texts to be displayed in the button), e.g., `Go for detail`, `Return`, `Skip proof`, etc.

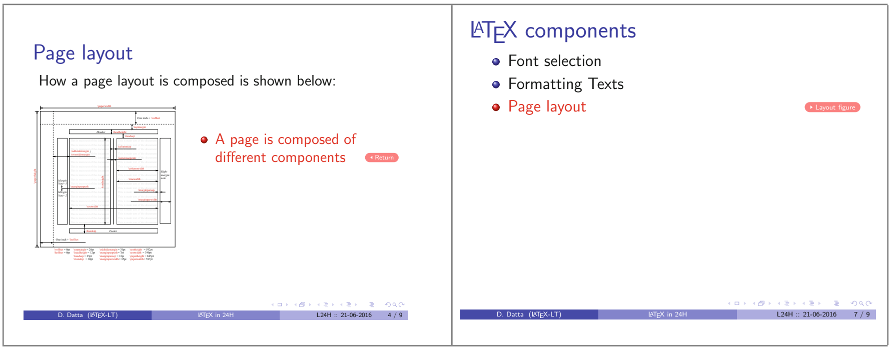
Similarly, the following are the commonly used hyperlink-type commands: **\hyperlink<>{}{}** (links the specified slide), **\hyperlinkframestart<>{}{}** (links the first slide of the current frame), **\hyperlinkframeend<>{}{}** (links the last slide of the current frame), **\hyperlinkframestartnext<>{}{}** (links the first slide of the next frame), **\hyperlinkframeendprev<>{}{}** (links the last slide of the previous frame), **\hyperlinkpresentationstart<>{}{}** (links the first slide of the presentation), and **\hyperlinkpresentationend<>{}{}** (links the last slide of the presentation). The overlay specification of these commands in **<>** is the slide in which the hyperlinked button is to be shown, the first mandatory argument in **{}** is the label-word of the target frame along with the overlay specification for the target slide in that frame (e.g., `stress` **<**`2`**>** for slide 2 of the frame having the label-word `stress`), and the second mandatory argument in **{}** is the button-type command as stated above. Note that, in the absence of an overlay specification in a hyperlink-type command, the hyperlinked button will be visible in all the slides of the frame in which it is inserted.

A hyperlinking example is shown in Table 22.13 along with its output in Table 22.14, where an earlier slide is hyperlinked from a latter slide, because of

Table 22.13 Hyperlinking slides in BEAMER presentation

```
\begin{frame}[t,label=layout]
  \frametitle{Page layout}
  How a page layout is composed is shown below:\pause \vskip 5mm
  \begin{columns}
    \column{0.4\textwidth}
    \includegraphics[width=\textwidth]{layoutpic}
    %
    \column{0.6\textwidth}
    \begin{itemize}[<+-|alert@+>]
    \item A page is composed of different components \hfill
                                 \hyperlink<2>{LaTeX<3>}{\beamerreturnbutton{Return}}
    \item Components are specified in length units
    \item Length of a component can be changed manually
    \end{itemize}
  \end{columns}
\end{frame}
%
\begin{frame}[t,label=LaTeX]
  \begin{itemize}[<+-|alert@+>]
  \frametitle{\LaTeX\ components}
  \item Font selection
  \item Formatting Texts
  \item Page layout \hfill      \hyperlink<3>{layout<2>}{\beamergotobutton{Layout figure}}
  \item Table, figure, equation, etc.
  \end{itemize}
\end{frame}
```

Table 22.14 Hyperlinked slides under the **Boadilla** presentation theme for the input frames of Table 22.13



Page layout

How a page layout is composed is shown below:

- A page is composed of different components

D. Datta (LATEX-LT) LATEX in 24H L24H :: 21-06-2016 4 / 9

LATEX components

- Font selection
- Formatting Texts
- Page layout

D. Datta (LATEX-LT) LATEX in 24H L24H :: 21-06-2016 7 / 9 which a return hyperlink is also added in the earlier slide (a return button seems essential since the presentation will not return automatically from the linked slide to the slide where it was linked). In this example, the combined command '**\hyperlink<**3**>{**layout**<**2**>}{\beamergotobutton{**Layout figure**}}**' is inserted to create a button with texts '`Layout figure`' in slide 3 of the frame `LaTeX`, so that a click on the button will lead the presentation to slide 2 of the frame `layout`. Similarly, the button with texts '`Return`', created using **\hyperlink<**2**>{**LaTeX**<**3**>}{\beamerreturnbutton{**Return**}}** in slide 3 of the frame `layout`, will return the presentation to slide 3 of the frame `LaTeX`.