Your Beamer Guide to Text Formatting

 By Admin August 26, 2021

This lesson teaches you how to format text in LaTeX Beamer. Firstly, we will highlight the use of Bold, Italic, and Underline commands. Then, we will present slanted and decorated text. After that, we will see how to change font size and text color. Finally, we conclude with text alignment and line spacing!

Contents :

1. Bold, italics and underlining

Simple text formatting is commonly used in documents to highlight special pieces of text, like the important concepts or definitions, making the document more readable.

In beamer we can print bold, italized and underlined text using the commands **\textbf**, **\textit** and **\underline**, respectively.

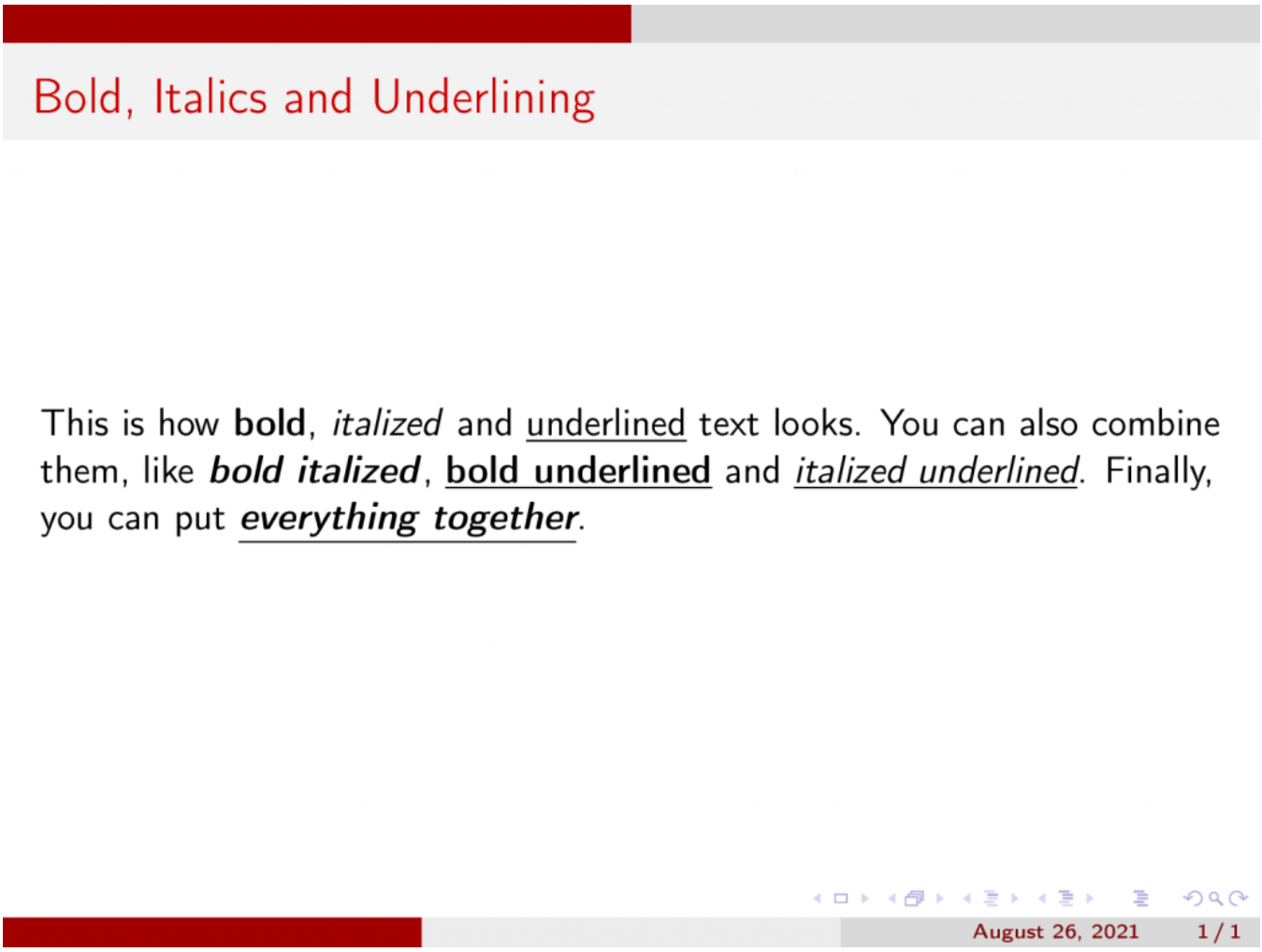
They can also be combined to produce bold italic text, or underlined bold text, etc. But in this case, we have to be sure that the combination exists for the font used. For instance, the combination of bold and italics is not available for the default font, but it can be used if you change the font encoding to T1 or load the lmodern package to use the Latin Modern font family.

In the following example, we show how to get all these combinations:

```
% Bold, Italics and underline text in Beamer
\documentclass{beamer}

% Theme choice
\usetheme{CambridgeUS}

% change the font encoding
\usepackage[T1]{fontenc}

% or either load the latin modern font
% \usepackage{lmodern}

\begin{document}

\begin{frame}{Bold, Italics and Underlining}

This is how \textbf{bold}, \textit{italized} and
\underline{underlined} text looks.
You can also combine them, like \textbf{\textit{bold
italized}}, \underline{\textbf{bold underlined}} and
\textit{\underline{italized underlined}}.
Finally, you can put
\textbf{\textit{\underline{everything together}}}.

\end{frame}

\end{document}
```

Copy

which yields the following output:



How bold, italized and underlined text looks in beamer, and different combinations between them

2. Slanted and emphasized text

2.1 Slanted text

Besides the bold and italic font shapes and the underlined decoration, beamer offers other font shapes which are not that common. The **`\textsl`** command creates slanted text, which has a similar slant to the right as an italic font, but it keeps the same lettering as the normal font family.

The `\textsc` command creates small capital text, which uses small forms of capital letters instead of lowercase letters.

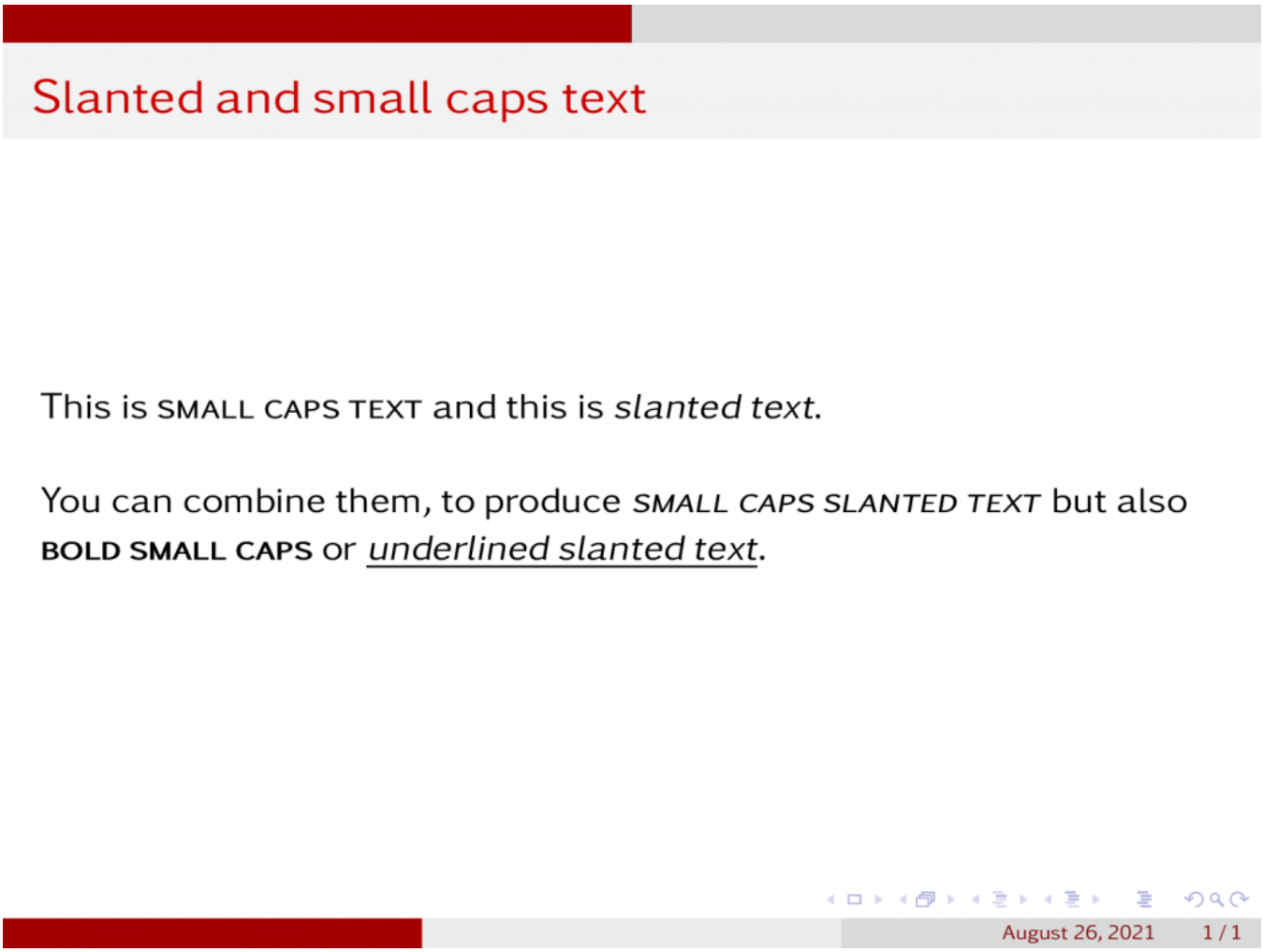
As with the bold and italized text, this font shapes can be all combined to produce all sorts of different outputs, but only if the selected font supports

them. In the following example, we use the KP Sans-Serif font to produce some combinations of the different shapes:

```
% Slanted and Small Cap text
\documentclass{beamer}

% Theme choice
\usetheme{CambridgeUS}

% select the KP Sans-Serif font
\usepackage[sfmath]{kpfonts}

\begin{document}

\begin{frame}{Slanted and small caps text}

This is \textsc{small caps text} and this is
\textsl{slanted text}.\\~\\

You can combine them, to produce \textsl{\textsc{small
caps slanted text}} but also \textsc{\textbf{bold small caps}} or
\textsl{\underline{underlined slanted text}}.

\end{frame}

\end{document}
```

Copy

Compiling this code yields:



2.2 Emphasized text

Finally, LaTeX offers the command **\emph** that can come very handy to emphasize text. This command will produce the correct shape to emphasize text in whichever context we use it.

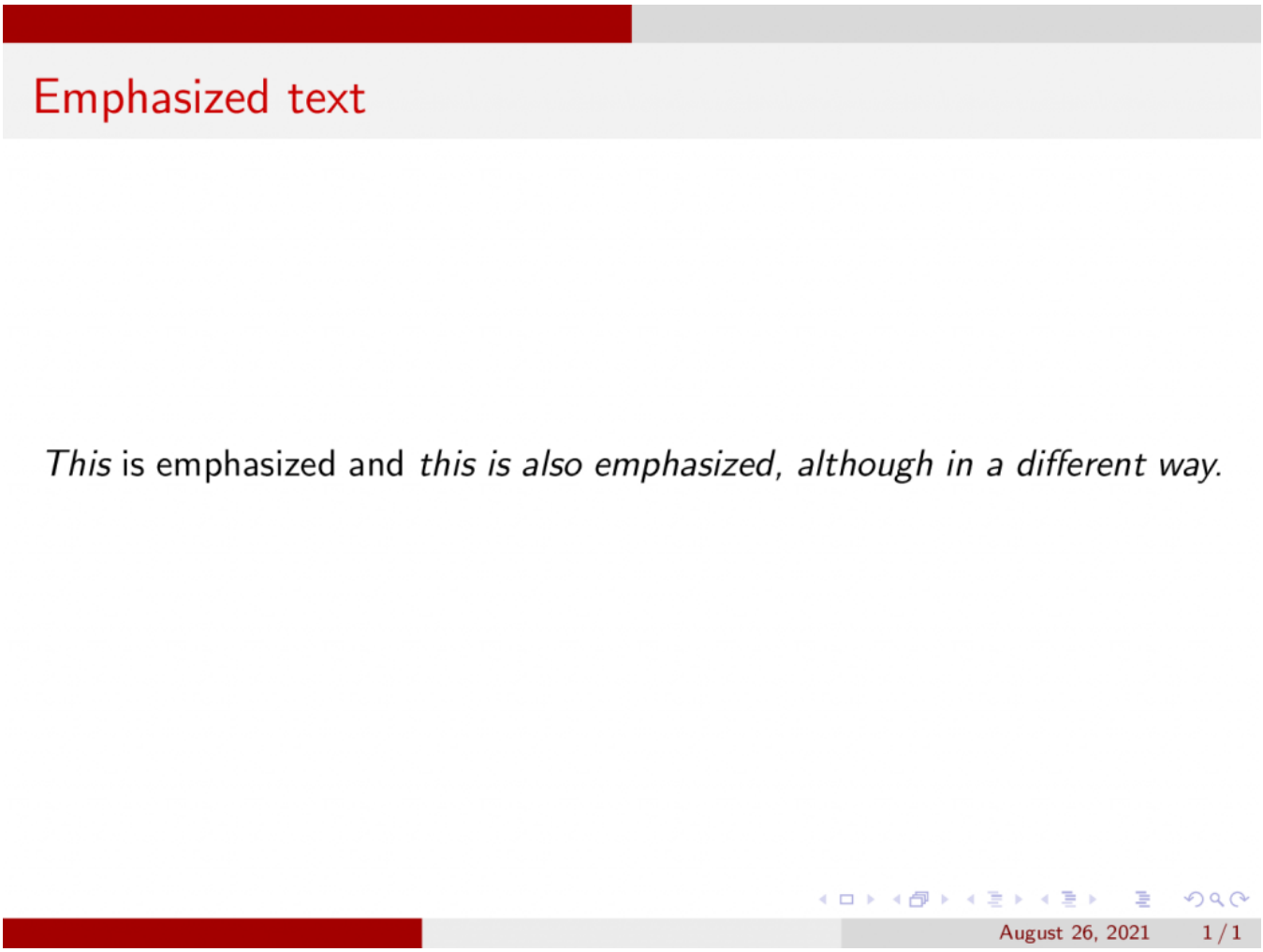
This means that **\emph** will produce italic text when writing in the usual upright font, and it will produce upright text when using it inside an italized environment. Check the following example:

```
% Emphasized text
\documentclass{beamer}

% Theme choice
\usetheme{CambridgeUS}

\begin{document}

\begin{frame}{Emphasized text}

```

Copy

```
11.   \emph{This} is emphasized and \textit{\emph{this} is
12.   also emphasized, although in a different way.}
13.
14.   \end{frame}
15.
16.   \end{document}
```

which produces the output:



3. Bold math in beamer

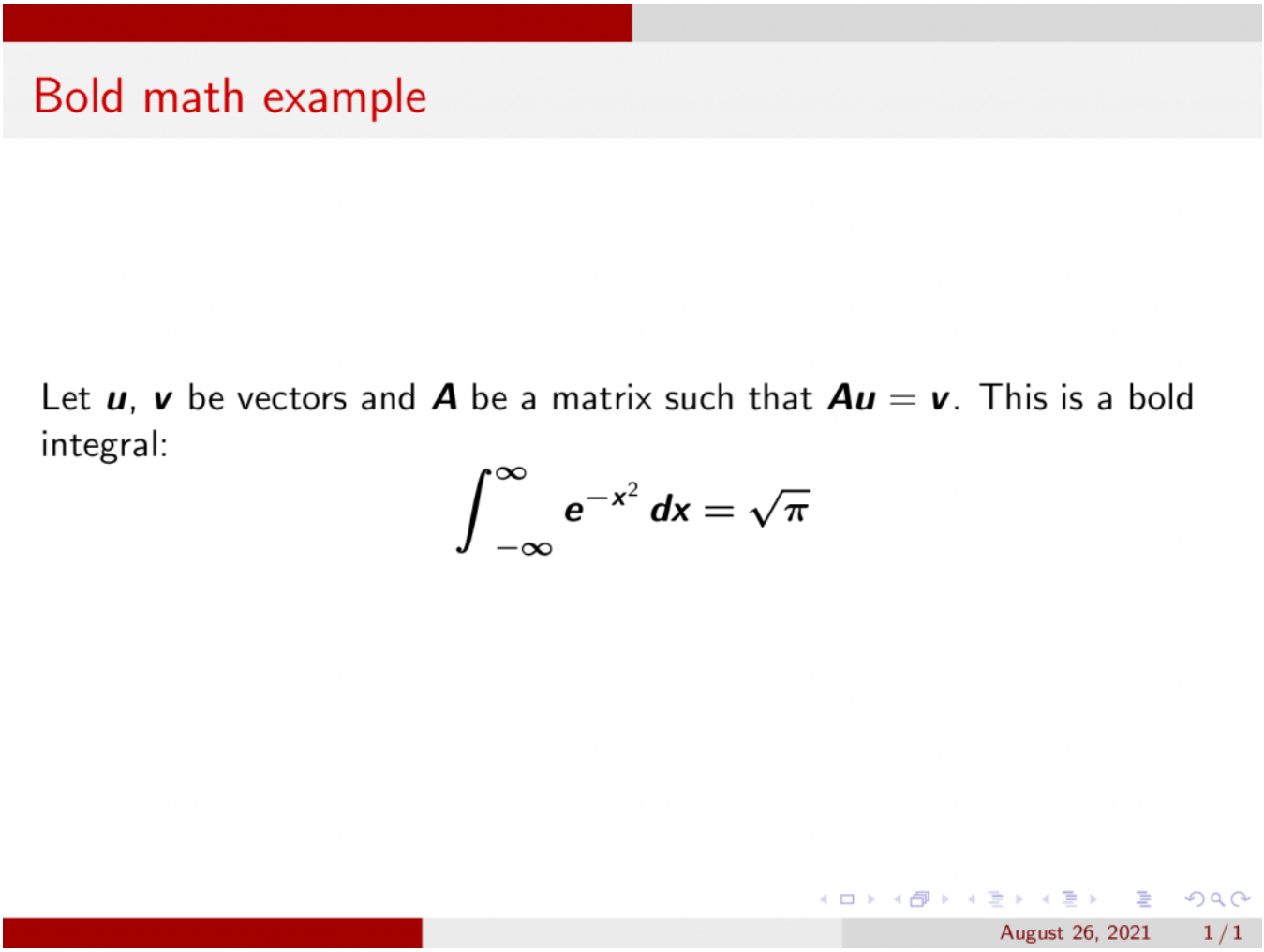
As a side note, maybe it is also worth commenting that sometimes we want to use a bold font inside math mode, for instance to denote vectors or matrices. To do so we cannot use the **`\textbf`** command, but instead we have to load the package bm (which stands for “bold math”) and use the **`\bm`** command.

The following example shows how to do it:

```
% Emphasized text
\documentclass{beamer}

% Theme choice
\usetheme{CambridgeUS}

% Required package
\usepackage{bm}

\begin{document}

\begin{frame}{Bold math example}

Let $\bm{u}$, $\bm{v}$ be vectors and $\bm{A}$ be a
matrix such that $\bm{Au}=\bm{v}$.
This is a bold integral:
\[
\bm{\int_{-\infty}^{\infty} e^{-x^2}\,dx=\sqrt{\pi} }
\]

\end{frame}

\end{document}
```

Copy

and you can see the result of it in the following illustration:



4. Text decorations in Beamer

We already saw how to underline text with the **`\underline`** command that comes with LaTeX. Here we want to go one step further and learn how to do all kinds of "decorations" to our text. To do so, we will be using the ulem package.

This packages mainly changes the way **`\emph`** works, since instead of emphasizing using the regular italics, it emphasizes the text underlining it. It also introduces the command **`\uline`** to underline text. By the way, the new underlining is not like the one provided by **`\underline`**, since the latter will not break at the end of a line, while the former will.

But the ulem package capabilities don't stop here, since it provides another six more different ways to decorate text.

Description	Command
Underline text solid line	\uline{}
Double-Underlined text	\uuline{}
Dashed Underline text	\dashuline{}
Dotted Underline text	\dotuline{}
Wavy-Underlined text	\uwave{}
Strikethrough text	\sout{}
Struck with Hatching text	\xout{}

Text decorations commands in LaTeX

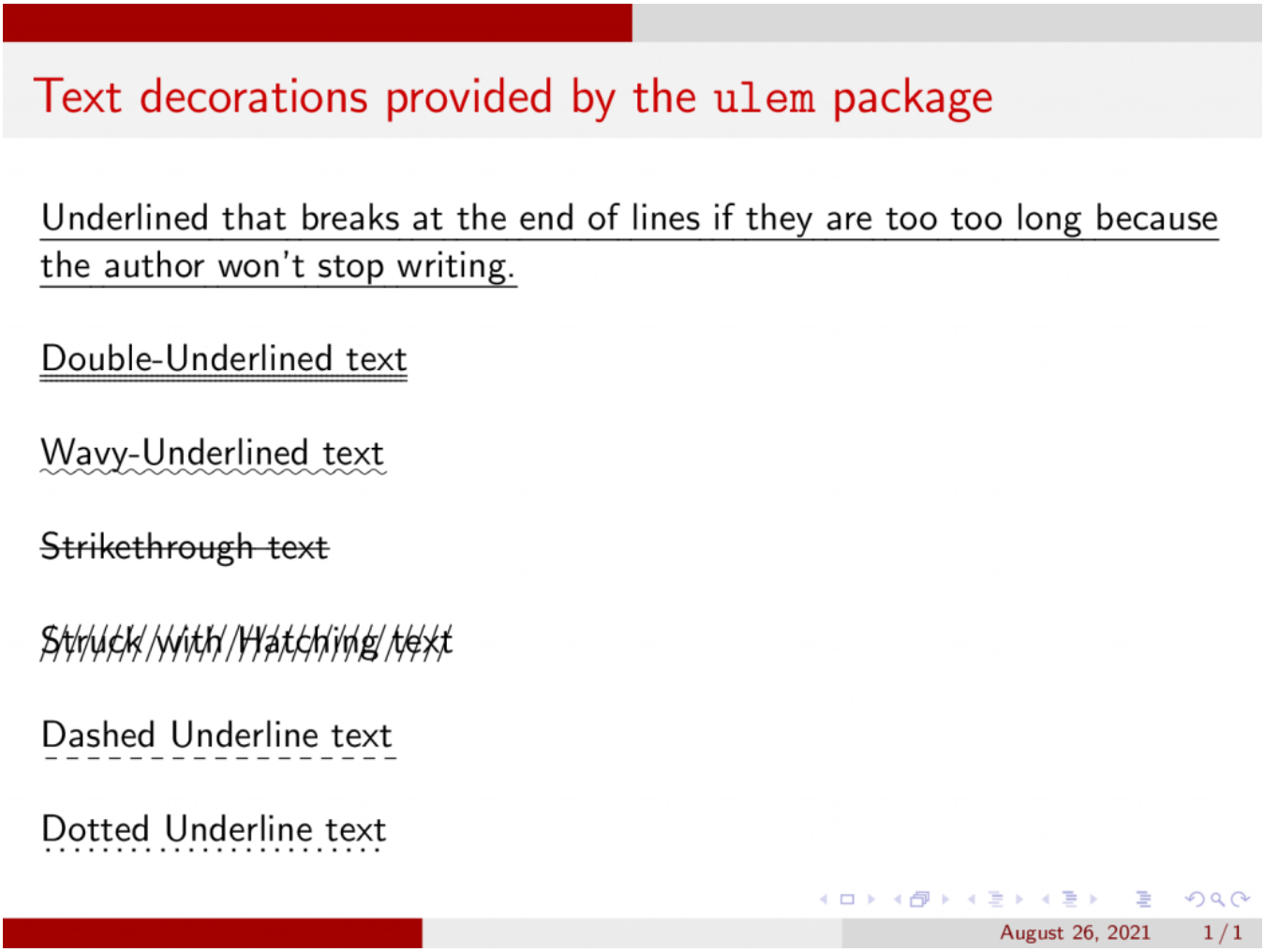
The following example shows how they work:

Copy

```
% Text decorations in Beamer
\documentclass{beamer}

% Theme choice
\usetheme{CambridgeUS}

% Required package
\usepackage{ulem}


\begin{document}

\begin{frame}{Text decorations provided by the
\texttt{ulem} package}


\uline{Underlined that breaks at the end of lines if
they are too too long because the author won't stop
writing.} \\~\\


\uuline{Double-Underlined text} \\~\\

\uwave{Wavy-Underlined text} \\~\\

\sout{Strikethrough text} \\~\\
```

```
27.
28.   \xout{Struck with Hatching text} \\~\\
29.
30.   \dashuline{Dashed Underline text} \\~\\
31.
32.   \dotuline{Dotted Underline text}
33.
34.   \end{frame}
35.
36.
37.   \end{document}
```

Here is the output of this code:



You may remarked the use of //~// at the end of each paragraph. It allows us to create noticeable separate paragraphs in Beamer.

5. Changing the font size in beamer

This part has been extensively discussed in the following posts:

- Beamer Font: Change its Size, Family and style
- How to change the font size on a given frame
- How to use too small font sizes in Beamer?
- Change Math Font style in Beamer

6. Change text color

We are going to see how to change the text color in beamer to improve the appearance of our presentation, and guide the attention of our audience.

The simplest way to use colors in any LaTeX document is the xcolor package.
This package provides a set of commands for color manipulation. The easiest
to use of these is the **`\color`** command, which let's us set, by name, the color
of an environment.

The command accepts most of the usual color names. Here is a minimal working example on how to use it:

Copy

```
% Text color in Beamer
\documentclass{beamer}

% Theme choice
\usetheme{CambridgeUS}


\begin{document}

\begin{frame}{Colors in beamer}

{\color{blue} This is a blue block.}

{\color{red} And this is a red one.}

\begin{itemize}
```

```
17.    \color{green}
18.        \item This is a list.
19.        \item And it is colored!
20.        \end{itemize}
21.    \end{frame}
22.
23.    \end{document}
```

which yields the following:

A basic use of colors in beamer

Here is a list of predefined colros:

The actual list of color names is determined by the driver used. For instance, you can use the dvipsnames as the package option to make the color names for the driver dvips available. For color names, check this interesting tutorial!

6.1 Highlight text

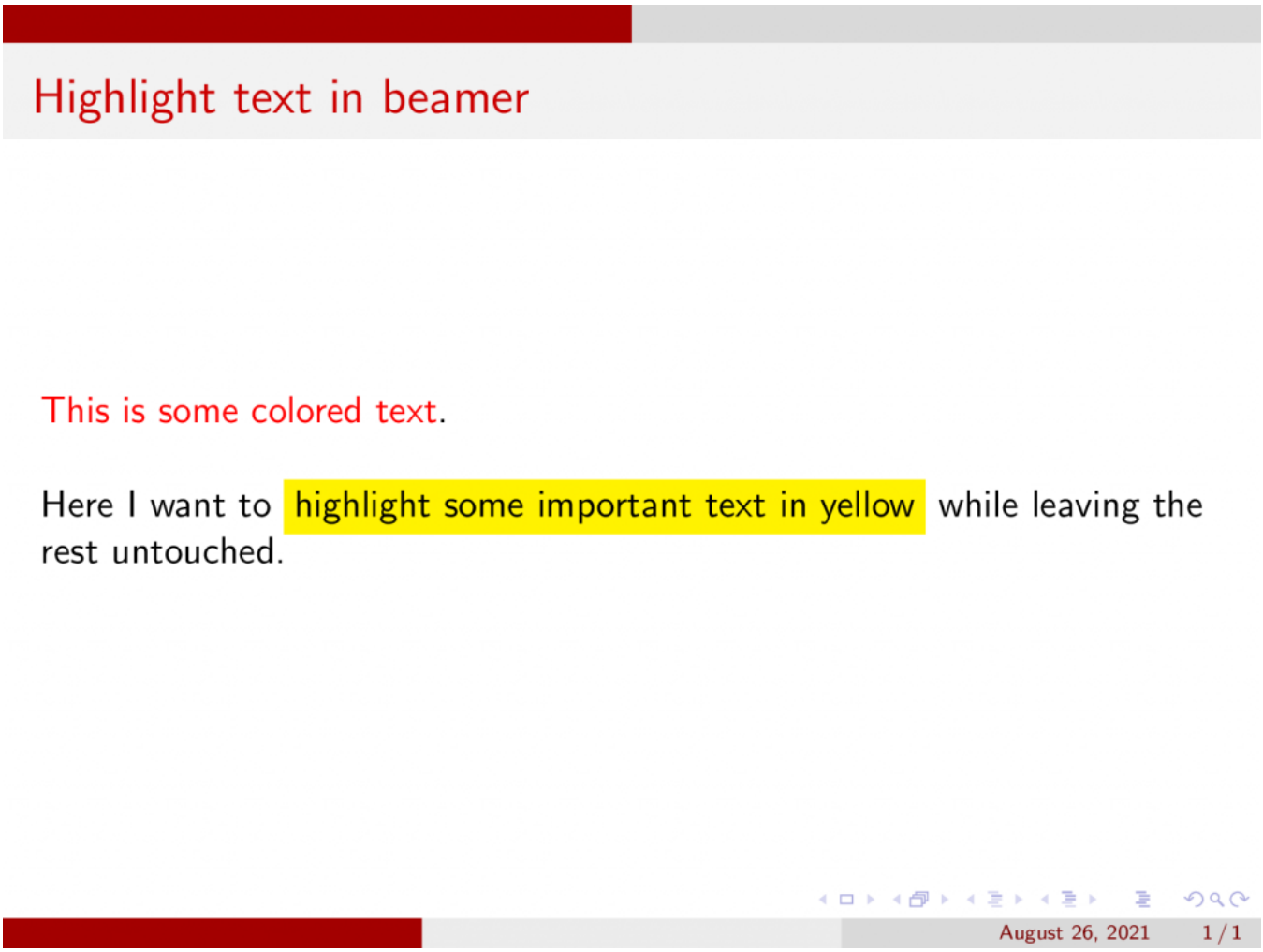
In the following example, we show the usage of the useful commands `\textcolor`, to easily change the color of the text, and `\colorbox` to highlight text:

```
% Highlight Text in Beamer
\documentclass{beamer}

% Theme choice
\usetheme{CambridgeUS}


\begin{document}

\begin{frame}{Highlight text in beamer}

\textcolor{red}{This is some colored text}.\\~\\
```

Copy

```
13.
14.   Here I want to \colorbox{yellow}{highlight some
15.   important text in yellow} while leaving the rest
16.   untouched.
17.
18.   \end{frame}
19.
20.   \end{document}
```

We obtain the following result:



How to easily change the color of the text and highlight it in beamer.

6.1 Define custom colors

As was mentioned before, you can also define your own custom colors, if you find that the ones defined by your driver are not enough. The manner in which the color is defined depends on the color models supported by your driver. In general, the command to define a color is:

```
\definecolor{name}{model}{color definition}
```

where name is the name which will take the color, model is the model used
to define it and color definition is the definition according to the selected model. The most common color models and their definition syntax are the following:

- rgb: Is specified as three comma-separated values between 0 and 1 that represent the amount of red, green and blue (in this order) that the color has.
- RGB: The same as before but with the values going between 0 and 255.
- cmyk: Is specified as four comma-separated values between 0 and 1 that determine the amount of cyan, magenta, yellow and black (in this order) to be added using the additive model in most printers.
- gray: A value in the grey scale between 0 and 1.
- HTML: Consists of 6 hexadecimal digits that represent the color in HTML code.
- wave: This is a fun one that may be useful when writing documents related to the field of Optics. It is a single number between 363 and 814 that specifies the wave length of the color in nanometres.

This is the more general and flexible way to create a new color, but maybe it is
not the easiest.

The most practical is using the command:

```
\colorlet{name}{combination}
```

where name is the name of the new color and combination is a combination of preexisting colors. The percentage of every color in the

combination is written between ! signs. For instance:

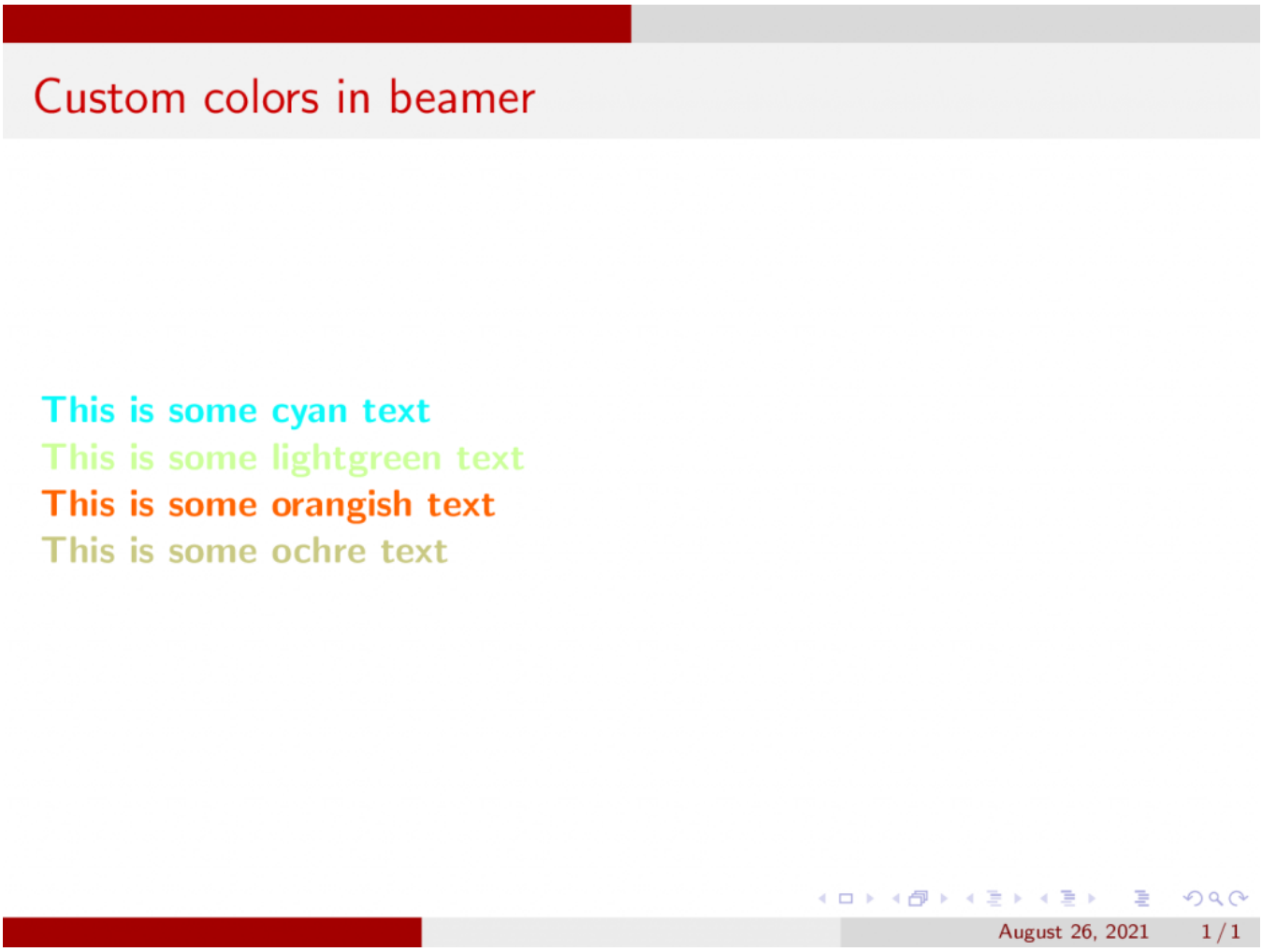
```
\colorlet{ochre}{blue!20!yellow!80!}
```

In the following example, we put these new techniques in use:

Copy

```
% Define colors in Beamer
\documentclass{beamer}

% Theme choice
\usetheme{CambridgeUS}

% Custom colors
\definecolor{cyanish}{RGB}{10,250,250}
\definecolor{lightgreen}{HTML}{CCFF99}
\definecolor{orangish}{wave}{620}
\colorlet{ochre}{blue!30!yellow!70!}

\begin{document}

\begin{frame}{Custom colors in beamer}

\textcolor{cyanish}{\textbf{This is some cyan text}}

\textcolor{lightgreen}{\textbf{This is some lightgreen text}}

\textcolor{orangish}{\textbf{This is some orangish text}}

\textcolor{ochre}{\textbf{This is some ochre text}}

\end{frame}

\end{document}
```

and the result of it is shown below:



7. Text alignment in beamer

In general LaTeX documents, paragraphs are usually fully justified, that is, flush both the left and right margins. If you want to change this justification, LaTeX offers the built-in environments flushleft, flushright and center to produce left justified, right justified and centred paragraphs, respectively.

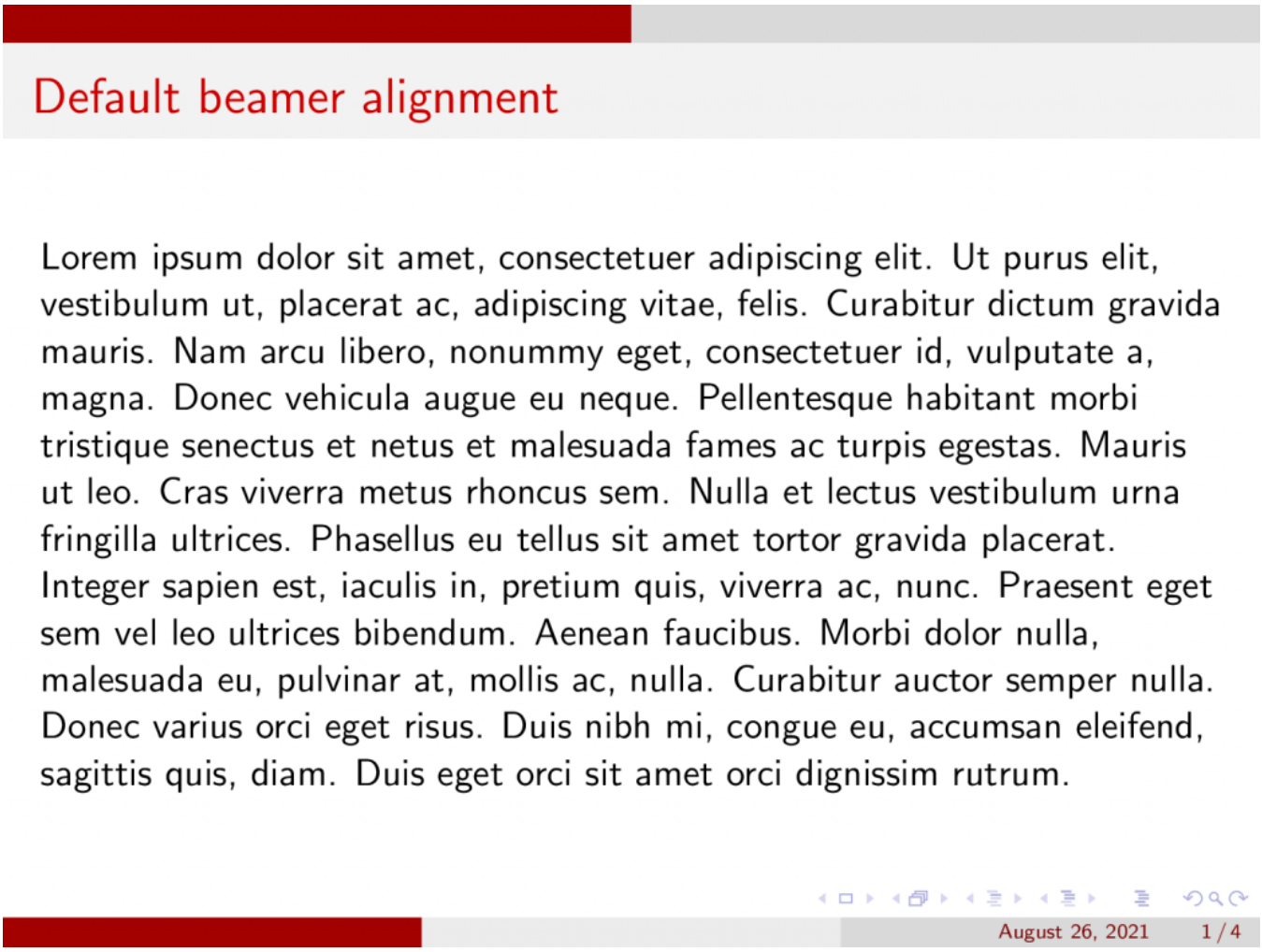
However, there is no built-in environment in LaTeX for fully justified text; and in beamer, by default, the text is left justified. This means that there is no straightforward way of making text fully justified in beamer. This is solved by the ragged2e package, which provides the \justifying command. This command, used inside the frame environment, or any other, will produce justified text inside that environment.

The following example shows how to use the different text alignments:

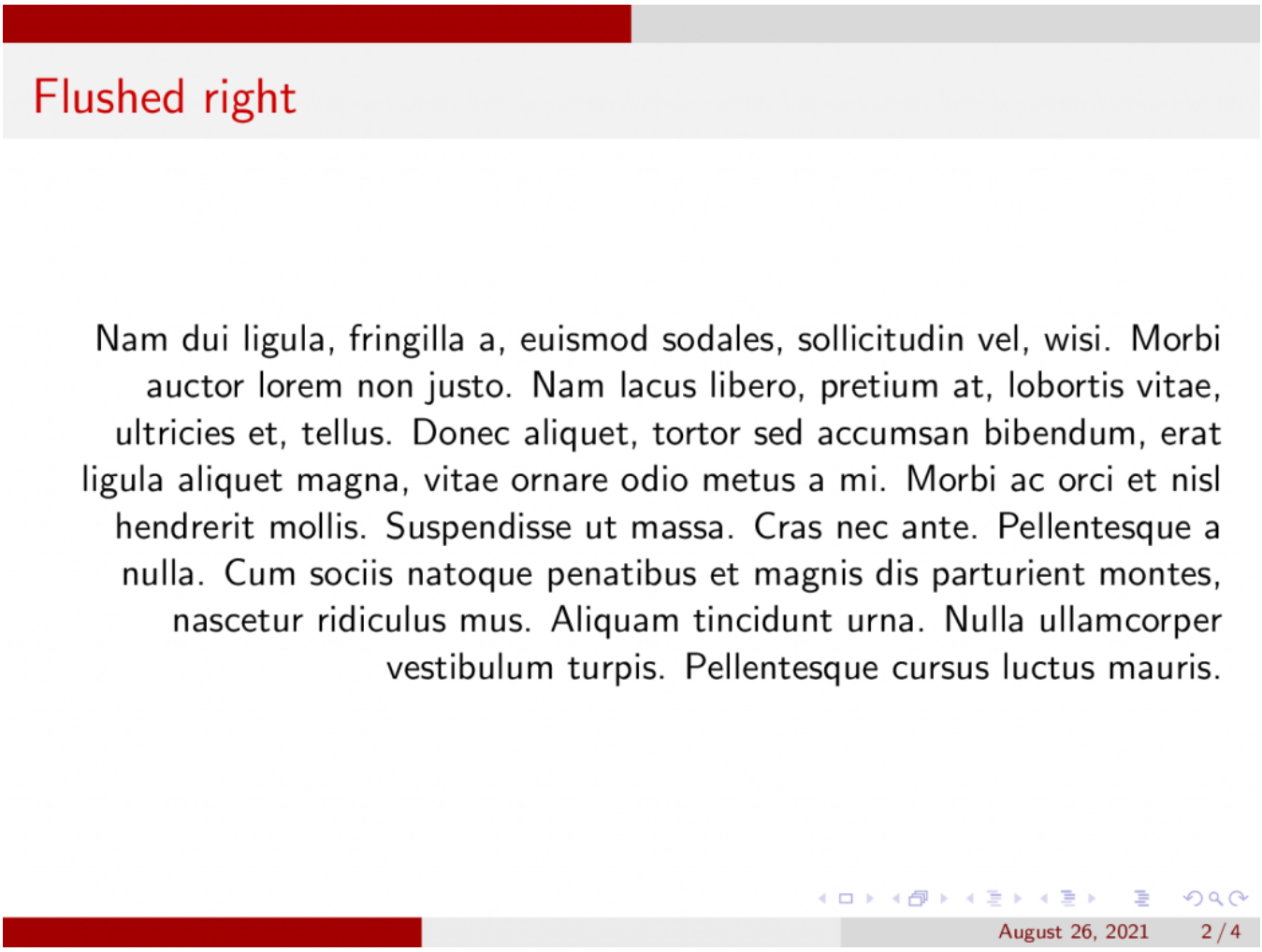
Copy

```
% Text alignment in beamer
\documentclass{beamer}

% Theme choice
\usetheme{CambridgeUS}

% to generate dummy text
\usepackage{lipsum}

% provides the \justifying command
\usepackage{ragged2e}

\begin{document}

% Default alignment
\begin{frame}{Default beamer alignment}
    \lipsum[1]
\end{frame}

% Flushed right alignment
\begin{frame}{Flushed right}

\begin{flushright}
    \lipsum[2]
\end{flushright}

\end{frame}

% Centered alignment
\begin{frame}{Centered}
\begin{center}
    \lipsum[3]
\end{center}
\end{frame}

% Fully justified alignment
\begin{frame}{Fully justified}
    \justifying
    \lipsum[4]
\end{frame}

\end{document}
```

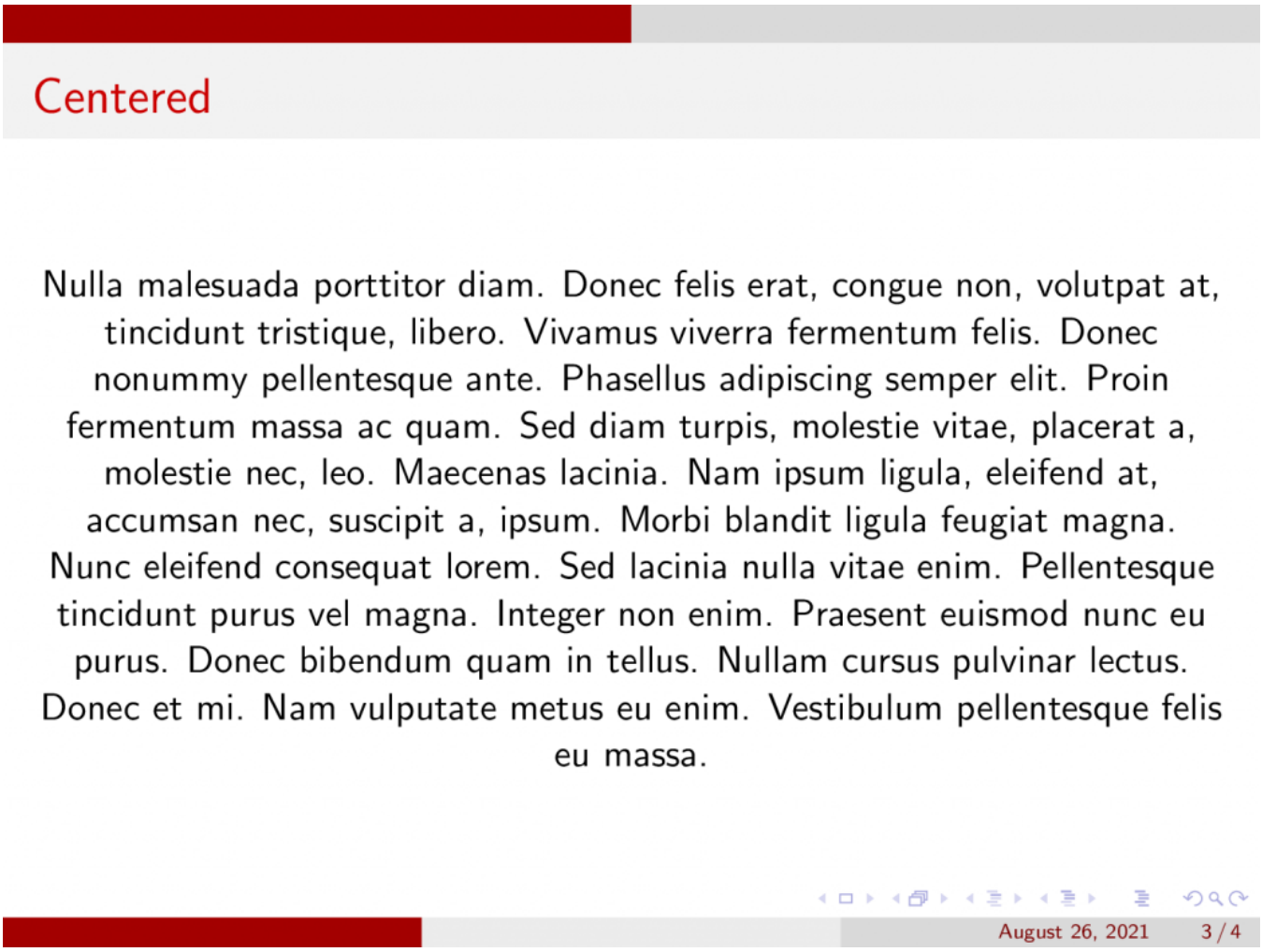
which yields the following:



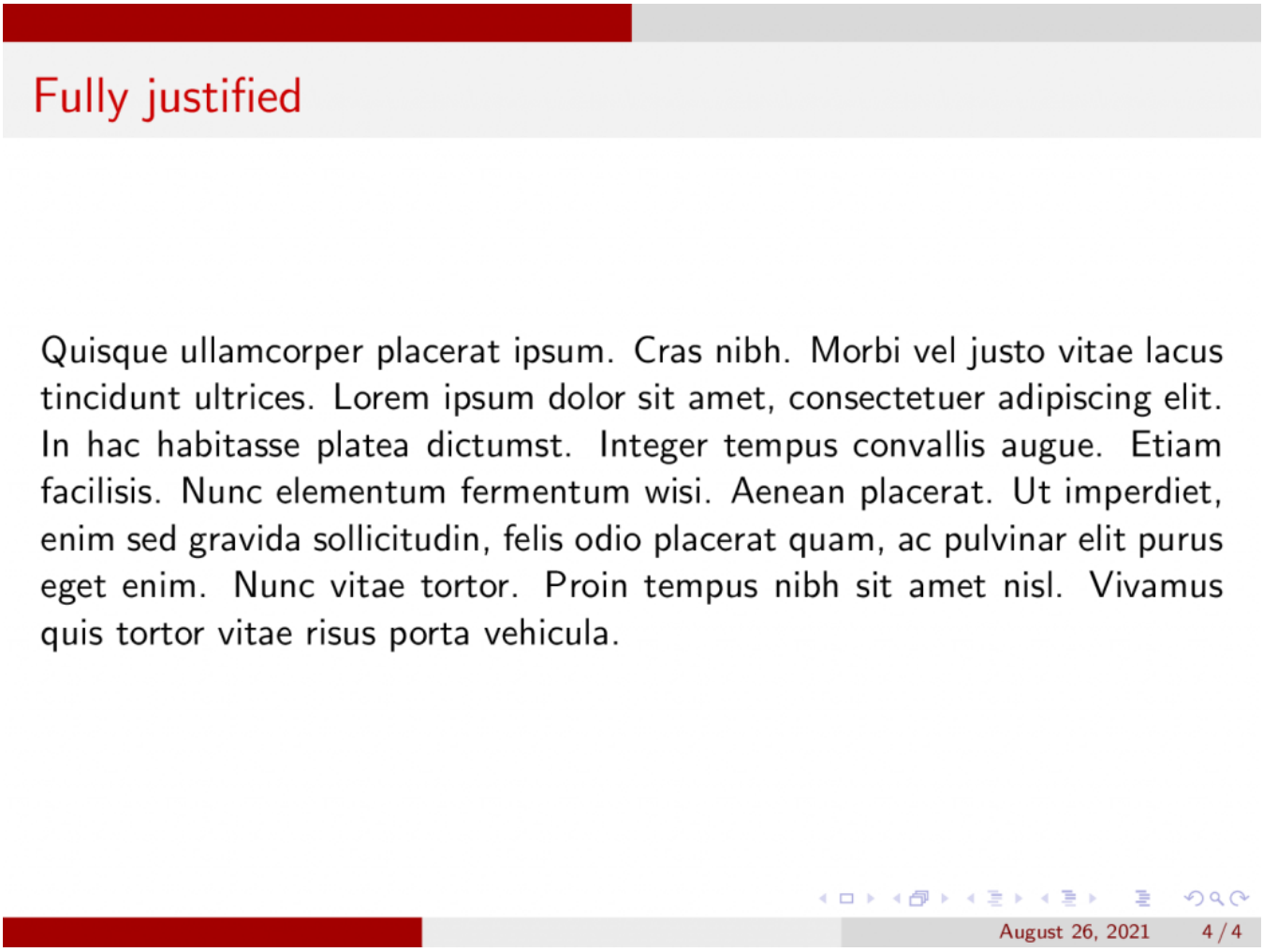
How the default alignment looks in beamer.



How the flushed right alignment looks in beamer



How the centered alignment looks in beamer.



How the fully justified alignment looks in beamer.

8. Line spacing in beamer

If you want to use larger interline spacing in your beamer presentation, you can change its value by using the command:

```
\linespread{factor}
```

in the preamble of your document. The usage of factor is far from intuitive. Because of TeX internal dynamics, **`\linespread{1.3}`** will stand for one and a half spacing and **`\linespread{1.6}`** will stand for double spacing.

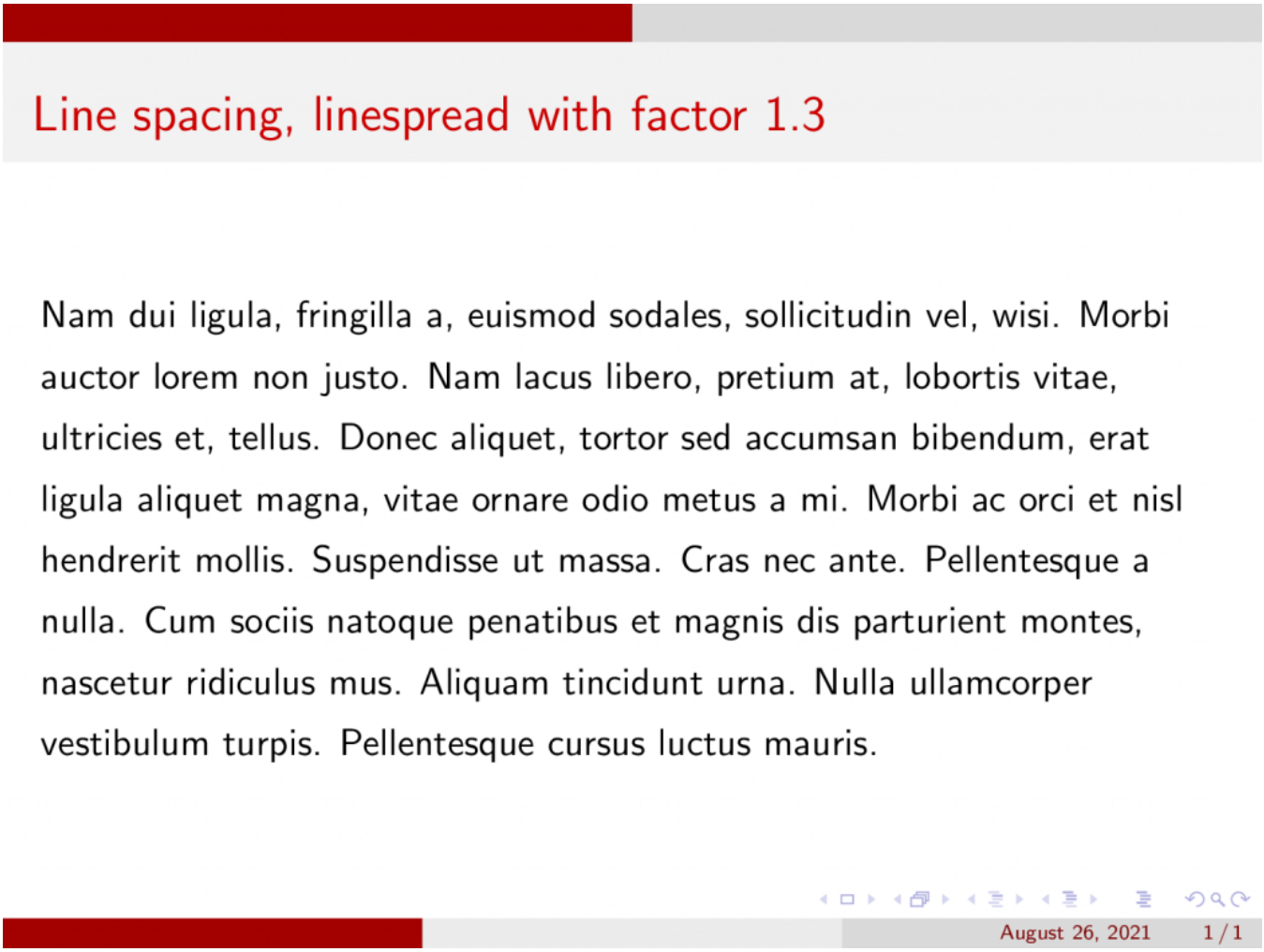
Check the following example:

```
1.   % Change line spacing
```

Copy

```
2.  \documentclass{beamer}
3.
4.  % Theme choice
5.  \usetheme{CambridgeUS}
6.
7.  % to generate dummy text
8.  \usepackage{lipsum}
9.
10. % Change line spacing
11. \linespread{1.3}
12.
13. \begin{document}
14.
15. \begin{frame}{Line spacing, linespread with factor 1.3}
16.     \lipsum[2]
17. \end{frame}
18.
19. \end{document}
```

Compiling this code yields:



How different line spacing options look in beamer.

You can compare it with the default one:

Default Line spacing

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

If you are not satisfied with this strange command, the setspace package allows a more fine-grained control over line spacing.

Next Lesson: 10 Creating Overlays in Beamer

Search

About LaTeX-Beamer.com

LaTeX-Beamer.com is a personal website about creating stylish and modern presentations in LaTeX, through step-by-step lessons.