

BEAMER-REVEAL

Walter Daems (`walter.daems@uantwerpen.be`)

Release 2026-01-19 — v1.03

1 Introduction

BEAMER [1] is a very powerful and convenient document class to create presentations and slides. However, integrating multimedia in it, is still a bit of a faff. On one side you have the integrated multimedia facility of BEAMER [1] and on the other side the `movie15` [2] and `media9` [3] packages. But unless you use the stock acrobat reader on Microsoft Windows, these things barely work, if at all.

Next to this \LaTeX ecosystem for slides, you have the `reveal.js` framework, that allows easy integration of multimedia content. Why not combine both worlds? With that I mean:

- make your slides in BEAMER, using all the nice packages that come with \LaTeX ,
- include any browser-compatible multimedia content you'd like, even generate some \LaTeX animations,
- convert that presentation to the `reveal.js` framework.

That's exactly what this package does.

2 Rationale

Let's talk about the elephant in the room: why not work in the `reveal` framework directly? I see two reasons:

- It avoids a new learning curve, allowing you to continue to build on your BEAMER expertise.
- It avoids that you have to convert your thousands of slides into a new format.

For me, those are all the reasons I need.

3 Synopsis

Figure 1 shows the overall flow. You start by making slides (frames) the usual way using the `beamer` class. Your source file (`slides.tex`) uses the package `beamer-reveal.sty` and references the multimedia files of your choice. These multimedia files (e.g. videos) can be super-imposed on any slide you like. Your favorite \LaTeX -compiler typesets your slides to PDF and produces an auxiliary `.rv1`-file containing extra information for the conversion script. The conversion script `beamer-reveal.pl` fuses the PDF and the auxiliary file into a `reveal.js` presentation, using the templates provided by the BEAMER-REVEAL package. In fact, this is done by converting your slides to JPG format and using them as the background on the REVEAL-slides. The multimedia content appears as HTML5 elements on that background. You then use your favorite browser as your viewer instead of the good old PDF reader.

As a nice bonus, the BEAMER-REVEAL package allows you to generate \LaTeX animations. It's not on the level of Manim [4], but for shorter animations inlined in your deck of slides it is more than functional!

Finally, you can also include material in iframes. Iframes are HTML constructs that allow you to incorporate multiple multimedia blocks in HTML. As an example, this allows incorporating asymptote material that has been generated for displaying as HTML using WebGL. Note that in many cases, your browser

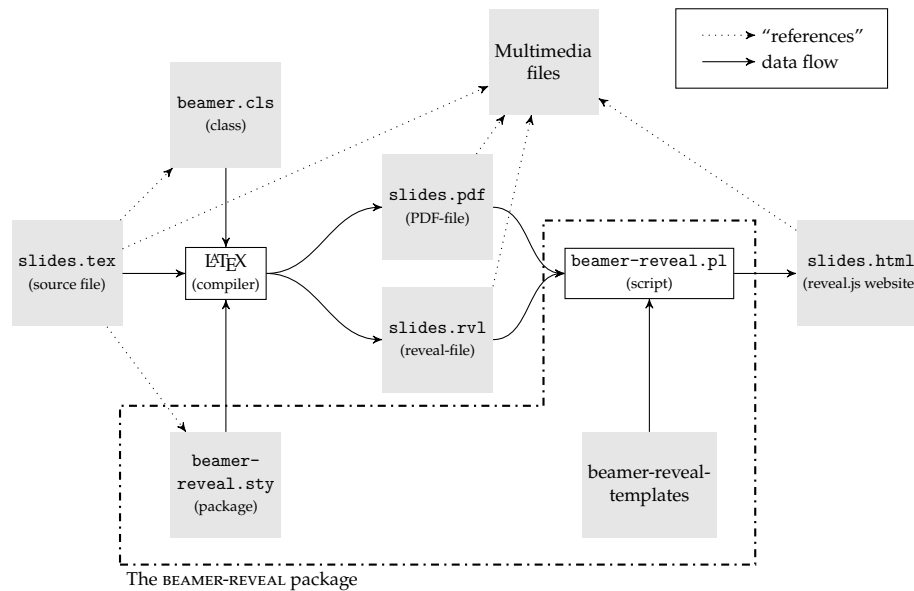


Figure 1: Workflow imposed by the BEAMER-REVEAL package

will prohibit these iframes to load external web content. To solve this, you can run a local webserver and access your presentation through it. This is explained in Section 7.

4 Quirks

Combining BEAMER and REVEAL posed one major challenge: how to make sure that the HTML5 elements appear exactly where you want them to be, i.e. perfectly aligned with your L^AT_EX content as it has been converted to PDF. The core of the problem is twofold:

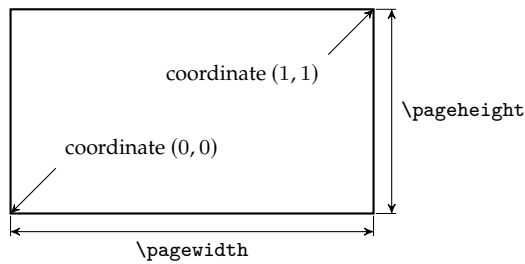
- In BEAMER the aspectratio of the slides is determined by the class option `aspectratio`. Your PDF viewer uses letterboxing (black bars on the side) if the aspectratio of your presentation does not correspond to the aspectratio of your screen. To the other hand, REVEAL puts your slides fullscreen without letterboxing, and therefore the aspectratio is determined by the screen resolution.
- Given the vector-nature of L^AT_EX and PDF, resolution is not a parameter you normally care about (everything is vector graphics anyway), while given the pixel-based nature of traditional multimedia files, resolution is an issue that you need to think about carefully.

There is only one good solution: ask the users to (1) set the correct `aspectratio` using the beamer class options and (2) specify the width or height of their displaying screen using a BEAMER-REVEAL class option.

The latter ensures that (a) the background content and the multimedia files that are generated based on the L^AT_EX source, are generated with sufficient resolution, and that (b) the file sizes stay within acceptable limits.

Actually, there is a one more problem, and that is that the canvas size and aspectratio of the display area in your browser is dependent on whether you are viewing full screen or not. Therefore, an extra requirement arises: (3) when presenting a slide-deck only you must use your browser in full-screen mode. Otherwise, the alignment is off.

In order not to have to care about resolutions or the actual `\pagewidth` or `\pageheight` of your presentation, I've chosen to impose another constraint: (4) the user must specify locations on the screen (where to put the multimedia content) as relative fractions (x, y) where $(0, 0)$ corresponds to the bottom left of the screen and $(1, 1)$ corresponds to the top right of the screen.



However, if you have to specify width and heights of the multimedia boxes that you want to superimpose on your slides, you don't want to specify width and height as relative numbers (of the slide width and height). In that scenario, displaying a 16:9 video on a 16:10 slide with a width equal to a quarter of the slide width, would require you to specify: `width=0.25,height=0.225`, which is weird. In addition, if you'd have to present at a venue where the projector has a different aspectratio from the one you anticipated, you would be forced to recalculate all the widths and heights of every video or image. You don't want that. Presenting brings sufficient stress without that extra worry.

Therefore, I imposed a fifth constraint: (5) the user must always specify either width or height of the box (as a fraction of the respective slide width or height) together with the aspectratio. Given width and aspectratio the BEAMER-REVEAL package can correctly determine its relative height; likewise height and aspectratio can be converted to the correct relative width. Therefore, it is illegal to specify both width and height at the same time. As the package knows the aspectratio of the screen (set correctly by the user) no recalculations need to be done. In the example above, that would mean specifying: `width=0.25,aspectratio=16/9`, which is very logical.

When confronted with the situation where you need to present on an old 4:3 projector, instead of the 16:10 you are used to, this allows you to just change the beamer aspectratio to 43, recompile, rerun `beamer-reveal.pl` and you are good to go on stage.

So summarized, these are the five rules to go by:

1. Set the correct aspectratio as a beamer class option.
2. Specify the X or Y-resolution of your displaying screen as a package option to the BEAMER-REVEAL package.
3. Always put your browser in full-screen-mode when presenting.
4. Specify positions relative to the slide width and height, (0,0) being bottom left and (1,1) being top right.
5. Specify box sizes of the HTML5 content always as the width or length in combination with the aspectratio.

5 Portability

These class files should be ready to use with all common modern \LaTeX compilers that produce PDF (XeLaTeX, Lua \LaTeX , ...) from the major \TeX -distributions (TeX \TeX , TeXLive, MikTeX). If you experience problems with one of these, please inform the author.

The script `beamer-reveal.pl` is a Perl script, that works on all major platforms (UNIX, Linux, BSD, Debian, MS-Windows, MAC-OS, ...). It makes use of the Poppler library and its `pdftoppm` tool, which is also available for those platforms. In case you want to use \LaTeX animations, it also uses your very own favorite \LaTeX -compiler, `pdfcrop` (which is part of the major TeX-distributions) and `ffmpeg` a well-known video conversion tool.

If these tools are available on your platform, then all should be well.

6 Installing the beamer-reveal.pl script

6.1 On Linux-like operating systems

Open a terminal with a shell. Below '\$' represents your shell prompt. If needed, update your package list first. Then install the required tools:

- on a Debian-like OS:

```
$ sudo apt install -y perl cpanminus poppler-utils \
  texlive-extra-utils ffmpeg
```
- on a Redhat-like OS:

```
$ sudo dnf install -y epel-release
$ sudo dnf install -y https://download1.rpmfusion.org/free/el/rpmfusion-free-release-
$(rpm -E %rhel).noarch.rpm
$ sudo dnf install -y perl perl-App-cpanminus poppler-utils \
  texlive-pdfcrop ffmpeg
```
- on openSUSE:

```
$ sudo zypper install -y perl perl-App-cpanminus \
  poppler-tools texlive-pdfcrop ffmpeg
```
- on an Arch-like OS:

```
$ sudo pacman -Syu --needed perl cpanminus poppler \
  texlive-binextra ffmpeg
```
- on Alpine:

```
$ sudo apk add perl cpan-app-cpanminus poppler-utils \
  texlive-extra-utils ffmpeg
```
- on macOS:

```
$ brew install perl cpanminus poppler texlive ffmpeg
```

6.2 On MS windows operating systems

Open a powershell. Below '\$' represents the powershell prompt. Install the required tools as follows:

```
$ winget install StrawberryPerl.StrawberryPerl
$ winget install -e --id oschwartz10612.Poppler
$ winget install MiKTeX.MiKTeX
$ winget install Gyan.FFMpeg
$ winget install --id Python.Python.3 --source winget
```

You probably need to answer 'Yes' quite some times. If you're asked to restart, you only need that before moving over to section 6.4 'Checking your setup'. You probably already have MiKTeX installed. In that case you can skip that. However, make sure the 'pdfcrop' program (part of MiKTeX) is installed and available on your path.

6.3 Install BEAMER-REVEAL

Open a shell on Linux/macOS, or a powershell prompt on MS-Windows. Run the following command at the prompt:

```
$ cpanm BeamerReveal
```

This will take a while as this needs to install a number of extra Perl libraries that have been used in BEAMER-REVEAL. You can use the same command to update the script if a new release has been made.

The current release number of the Perl package is: 20260119.1636

You will see that number appear during the installation. Any version more recent version than this will work. New releases will fix bugs that are reported. If there is a breaking change in the Perl package, a new CTAN update will be released as well.

6.4 Checking your setup

Open a shell on Linux/macOS, or a command line or powershell prompt on MS-Windows. Then see whether you can invoke the help information for the tools.

```
$ cpanm -h
$ pdftoppm -h
$ pdfcrop --help
$ ffmpeg -h
```

To test BEAMER-REVEAL on MS-Windows, run:

```
$ beamer-reveal -h
```

On any other platform, run:

```
$ beamer-reveal.pl -h
```

If they all display correct help info, you're good to go.

7 Using the beamer-reveal.pl script

If you prepared your BEAMER-presentation according to Section 11, then converting it into a reveal.js HTML presentation, is as simple as running the following command. Make sure the current directory of the shell is the directory your L^AT_EX-source file and your compiled PDF-file is in. For all commands below, stay in that same working directory!

If your document is called `jobname.tex`, then just run:

```
$ lualatex jobname.tex
$ lualatex jobname.tex
```

On MS-Windows, run:

```
$ beamer-reveal jobname
```

On any other platform, run:

```
$ beamer-reveal.pl jobname
```

Conversion is very fast. Of course, if you need to render some L^AT_EX animations, it may take a little more than a jiffy. Especially if you are working on MS-Windows, because there the generation is fully single-threaded.

Some side notes:

- you need to run `beamer-reveal.pl` from the same working directory as your L^AT_EX-compiler; the script needs to be able to access your source files from the same viewpoint as your compiler.
- `beamer-reveal.pl` also has some convenient command-line options to set the aux directory (that contains your `.rvl` file), the pdf directory that is the output directory of your L^AT_EX-run and the output directory (in which it will place the resulting reveal site). This allows easy integration with some build frameworks such as `latexmk`. Checkout the help message to learn about it.

```
$ beamer-reveal.pl -h
```

Next, you can load your document in your browser, e.g.:

```
$ firefox jobname.html
```

If your presentation contains `iframe` content, you need to start a local webserver. You don't need a network connection for that. This is how:

```
$ python -m http.server
```

Then, you can access it through: <http://localhost:8000>.

8 Demo

If you want to see and try out the result of the example that is embedded in this documentation, check:

- a 16-by-9 version on: <https://www.digmanwaves.net/beamer-reveal/169>
- a 16-by-10 version on: <https://www.digmanwaves.net/beamer-reveal/1610>

9 Outreach

Is there any feature you are missing? Some problems you encounter? Some inconsistencies in the interface or the documentation? Some additional features that Reveal supports, but are not in BEAMER-REVEAL? Let me know by dropping me an e-mail.

If you think BEAMER-REVEAL makes no sense, let me know why you think so. I'm keen to learn.

On the other hand, if you like BEAMER-REVEAL and are using it, just send me a kind word. It keeps me going way better than wine or pizza.

10 Thanks

Thanks to Paul Levrie for proofreading this documentation and testing the package.

11 Usage

11.1 Package options

The following package options are available:

<code>width</code>	the width (in pixels) of the screen you will display the presentation on
<code>height</code>	the height (in pixels) of the screen you will display the presentation on

Only specify one of the two options. The other dimension will be deduced from the `aspectratio` option that passed onto the beamer class.

Higher values will give higher resolution of the slides (in the background), but also larger file sizes. A safety factor is already used when converting the slides to jpg-format, so taking the true height or width is recommended. Only when you are bothered with jpg-artifacts in the final result, you should consider increasing the width- or height-value.

11.2 An enhanced frame environment

`frame (env.)` The frame environment is defined by `BEAMER`. However, it is equipped with four new environment options by the `BEAMER-REVEAL` package:

<code>titleslide</code>	specifies that this slide is a title slide. It will be on the top level of the reveal menu. This menu can be invoked by pressing 'm' in your presentation (or clicking on the pan-cake icon on the bottom left of your presentation).
<code>sectionslide</code>	specifies that this slide is a section slide. It will be on the second level of the reveal menu.
<code>subsectionslide</code>	specifies that this slide is a subsection slide. It will be on the third level of the reveal menu.
<code>transition</code>	one of none, zoom, fade, concave or convex; these correspond to the available reveal.js transitions. Not recommended for use. Why? Animation without purpose is bad practice.

Slides that are not titleslides, sectionslides or subsectionslides are ordinary slides. They will appear on the lowest level in the reveal menu.

11.3 The macros to use inside the frame environment

To understand the operation of the macros, it is important to realize that the slide content of your beamer-generated PDF will be put as a background image onto the reveal.js slides. The extra material such as videos, images and audio will be put as an overlay on top of that background. The macros allow you to define what material to put in overlay and where and how big it should appear. All macros take options. Some of the options need a value, some not. Options that take no value are marked with a dagger-symbol in superscript (§).

`\video` This macro will create a video box on the current slide. Of course a video box can cover the entire slide if desired.

Its syntax is:

```
\video<overlay-spec> [options] at (x,y) { filename }
```

The arguments are:

`overlay-spec` a standard beamer overlay specification that allows you to determine on which overlays the video is to appear; this argument will end up in a traditional `\only<>{}` clause that beamer provides.

`options` the following keys are available. In general they require a value: key=value.

Size options:

<code>width</code>	the width of the video box (a fraction relative to the width of the slide)
--------------------	--

height the height of the video box (a fraction relative to the height of the slide)

aspectratio the aspectratio of the video box

Remember that you never specify both width and height, only one of those two in combination with the aspectratio.

Placement options:

anchor value is one of center, north, west, south, east, north east, north west, south east, south west; this specifies where the anchor of the video is positioned; the anchor will be positioned at (x,y)

above[†] synonym to anchor=south

below[†] synonym to anchor=north

left[†] synonym to anchor=west

right[†] synonym to anchor=east

above left[†] synonym to anchor=south west

above right[†] synonym to anchor=south east

below left[†] synonym to anchor=north west

below right[†] synonym to anchor=north east

Appearance options:

fit the way the video should occupy the box: fill, cover or fit

background the color of the background of the box

draw[†] generates an outline around the box that allows you to inspect where the video will end up in your PDF-file

autoplay[†] causes the video to start playing as soon as it appears on the slide

controls[†] causes player controls to appear below your video

muted[†] silences the audio of the player

x, y the x- and y-coordinate of the box, specified as a fraction of the slide width and slide height (dimensionless), fixing the anchor of the box, w.r.t. the bottom left of your slide.

filename a filename or URL that leads to the video file (e.g. an mp4 file). Any file playable by your browser will work.

`\audio` This macro will create an audio block on the current slide. This block is rather an abstract concept, unless you activate the controls of the player. Indeed, the audio block will be invisible unless you specify the option to display the controls of the player.

Its syntax is

```
\audio<overlay-spec> [options] at (x,y) { filename }
```

The arguments are:

overlay-spec a standard beamer overlay specification that allows you to determine on which overlays the audio is to appear; this argument will end up in a traditional `\only<>{}` clause that beamer provides.

options the following keys are available. In general they require a value: key=value.

Size options:

width the width of the audio box (a fraction relative to the width of the slide)

height the height of the audio box (a fraction relative to the height of the slide)

aspectratio the aspectratio of the audio box

Remember that you never specify both width and height, only one of those two in combination with the aspectratio.

Placement options:

anchor	value is one of center, north, west, south, east, north east, north west, south east, south west; this specifies where the anchor of the audio box is positioned; the anchor will be positioned at (x,y)
above [†]	synonym to anchor=south
below [†]	synonym to anchor=north
left [†]	synonym to anchor=west
right [†]	synonym to anchor=east
above left [†]	synonym to anchor=south west
above right [†]	synonym to anchor=south east
below left [†]	synonym to anchor=north west
below right [†]	synonym to anchor=north east

Appearance options:

fit	the way the audio block should occupy the box: fill, cover or fit
background	the color of the background of the box
draw [†]	generates an outline around the box that allows you to inspect where the audio box will end up in your PDF-file
autoplay [†]	causes the audio to start playing as soon as it appears on the slide
controls [†]	causes player controls to appear
muted [†]	silences the audio of the player

x, y the *x*- and *y*-coordinate of the box, specified as a fraction of the slide width and slide height (dimensionless), fixing the anchor of the box, w.r.t. the bottom left of your slide.

filename a filename or URL that leads to the audio file (e.g. an mp3 or ogg vorbis file). Any file playable by your browser will work.

\iframe This macro will create an iframe box on the current slide. Of course an iframe box can cover the entire slide if desired.

Its syntax is:

```
\iframe<overlay-spec> [options] at (x,y) { filename }
```

The arguments are:

overlay-spec a standard beamer overlay specification that allows you to determine on which overlays the iframe is to appear; this argument will end up in a traditional `\only<>{}` clause that beamer provides.

options the following keys are available. In general they require a value: key=value.

Size options:

width	the width of the iframe box (a fraction relative to the width of the slide)
height	the height of the iframe box (a fraction relative to the height of the slide)
aspectratio	the aspectratio of the iframe box

Remember that you never specify both width and height, only one of those two in combination with the aspectratio.

Placement options:

anchor	value is one of center, north, west, south, east, north east, north west, south east, south west; this specifies where the anchor of the video is positioned; the anchor will be positioned at (x,y)
above	synonym to anchor=south; ¹
below	synonym to anchor=north; ¹

¹this key allows no value

left [†]	synonym to anchor=east; ¹
right [†]	synonym to anchor=west; ¹
above left [†]	synonym to anchor=south east; ¹
above right [†]	synonym to anchor=south west; ¹
below left [†]	synonym to anchor=north east; ¹
below right [†]	synonym to anchor=north west; ¹

Appearance options:

fit	the way the iframe should occupy the box: fill, cover or fit
background	the color of the background of the box
draw	generates an outline around the box that allows you to inspect where the iframe will end up in your PDF-file; ¹

x, y the *x*- and *y*-coordinate of the box, specified as a fraction of the slide width and slide height (dimensionless), fixing the anchor of the box, w.r.t. the bottom left of your slide.

filename a filename or URL that leads to the iframe content (e.g. an HTML file generated by asymptote). Note that for the content to work, you might need to serve your presentation through a local html server:

```
$ python -m http.server
```

`\image` This macro will create an image box on the current slide. Of course an image box can cover the entire slide if desired.

Note that using the `\includegraphics` command of the `graphicx` package is still preferred to include the standard image formats, such as PDF, PNG, TIFF and JPG. However, the `image` command also allows to include (animated) GIFs and SVG files.

Its syntax is:

```
\image<overlay-spec>[options] at (x,y) { filename }
```

The arguments are:

overlay-spec a standard beamer overlay specification that allows you to determine on which overlays the image is to appear; this argument will end up in a traditional `\only<>{}` clause that beamer provides.

options the following keys are available. In general they require a value: `key=value`.

Size options:

width	the width of the image box (a fraction relative to the width of the slide)
height	the height of the image box (a fraction relative to the height of the slide)
aspectratio	the aspectratio of the image box
Remember that you never specify both width and height, only one of those two in combination with the <code>aspectratio</code> .	

Placement options:

anchor	value is one of center, north, west, south, east, north east, north west, south east, south west; this specifies where the anchor of the image is positioned; the anchor will be positioned at (<i>x, y</i>)
above [†]	synonym to anchor=south; ²
below [†]	synonym to anchor=north; ²
left [†]	synonym to anchor=east; ²
right [†]	synonym to anchor=west; ²
above left [†]	synonym to anchor=south east; ²
above right [†]	synonym to anchor=south west; ²

²this key allows no value

<code>below left</code> [†]	synonym to <code>anchor=north east</code> ; ²
<code>below right</code> [†]	synonym to <code>anchor=north west</code> ; ²

Appearance options:

<code>fit</code>	the way the image should occupy the box: <code>fill</code> , <code>cover</code> or <code>fit</code>
<code>background</code>	the color of the background of the box
<code>draw</code>	generates an outline around the box that allows you to inspect where the image will end up in your PDF-file; ²

`x, y` the x - and y -coordinate of the box, specified as a fraction of the slide width and slide height (dimensionless), fixing the anchor of the box, w.r.t. the bottom left of your slide.

`filename` a filename or URL of the image file (e.g. a GIF file).

`\animation` This macro will create an animation box on the current slide. Different from the other boxes, this box will determine its own width and height, based on the dimensions of the LaTeX content embedded in it. In fact, it is illegal to specify `aspectratio`, `width` or `height`.

The animation is generated as follows: the content of macro will be written by the `beamer-reveal.pl` script to a separate LaTeX file using the `standalone` class. The preamble that it uses will be the part of the preamble in your beamer source file, in between the loading of the `beamer-reveal` package and the line containing `\begin{document}`. The animation block will be embedded in a loop that will be executed `duration · framerate` times, providing a macro `\progress` that contains a fraction that goes up from 0 (first iteration) to 1 last iteration. The PDF-file that is generated with this standalone file, is converted to an mp4-file that is included as a video on your slide.

The tools used for this are the LaTeX-compiler you used for your beamer sourcefile, `pdfcrop` [5], `pdftoppm` [6] and `ffmpeg` [7].

This animation generation takes advantage of the multicore nature of your computer, by simple, but smart parallelization (on non-MS-Windows operating systems).

Its syntax is:

```
\animation<overlay-spec> [options] at (x,y) { animation-LaTeX-content }
```

The arguments are:

`overlay-spec` a standard beamer overlay specification that allows you to determine on which overlays the animation is to appear; this argument will end up in a traditional `\only<>{}` clause that beamer provides.

`options` the following keys are available. In general they require a value: `key=value`.

Size options: no size options; size is determined from the LaTeX code itself!

Generation options:

<code>framerate</code>	number of frames per second that the animation should contain.
<code>duration</code>	duration (in seconds) of the animation
<code>pdfprogress</code>	value that the progress macro will take on when a picture of the animation frame is made in your PDF-file.

The number of frames that will be generated for the animation is:

$$\text{number-of-frames} = \text{framerate} \cdot \text{duration} \quad (1)$$

Your PDF file will contain a single shot of the animation (as if it were a preview shot). For this frame, the value of the `\progress` macro will be set to `pdfprogress`.

Placement options:

<code>anchor</code>	value is one of <code>center</code> , <code>north</code> , <code>west</code> , <code>south</code> , <code>east</code> , <code>north east</code> , <code>north west</code> , <code>south east</code> , <code>south west</code> ; this specifies where the anchor of the animation is positioned; the anchor will be positioned at (x, y)
<code>above</code> [†]	synonym to <code>anchor=south</code>

<code>below[†]</code>	synonym to <code>anchor=north</code>
<code>left[†]</code>	synonym to <code>anchor=east</code>
<code>right[†]</code>	synonym to <code>anchor=west</code>
<code>above left[†]</code>	synonym to <code>anchor=south east</code>
<code>above right[†]</code>	synonym to <code>anchor=south west</code>
<code>below left[†]</code>	synonym to <code>anchor=north east</code>
<code>below right[†]</code>	synonym to <code>anchor=north west</code>

Appearance options:

<code>background</code>	the color of the background of the box
<code>draw[†]</code>	generates an outline around the box that allows you to inspect where the video will end up in your PDF-file
<code>autoplay[†]</code>	causes the video to start playing as soon as it appears on the slide
<code>controls[†]</code>	causes player controls to appear below your video
<code>muted[†]</code>	silences the audio of the player
<code>x, y</code>	the x - and y -coordinate of the box, specified as a fraction of the slide width and slide height (dimensionless), fixing the anchor of the box, w.r.t. the bottom left of your slide.
<code>animation-LaTeX-content</code>	the LaTeX code that generates every frame based on the <code>\progress</code> 'time variable'.

12 Example

12.1 Using the example

An example will allow you to get an idea of the convenience of the package. The following steps will allow you to observe it in your browser:

```
$ lualatex beamer-reveal-example.tex
$ beamer-reveal.pl beamer-reveal-example
$ python -m http.server
```

Then open in your browser: localhost:8000 and enjoy!

12.2 The source code of the example

```
<*example>
\documentclass[11pt,aspectratio=169,t]{beamer}
\setbeamertemplate{navigation symbols}{}

\usepackage[width=1920]{beamer-reveal}
\usepackage{tikz}
\usepackage{siunitx}

\newcommand\eu{\mathrm{e}}
\newcommand\ju{\mathrm{j}}

\title{Test slide deck}
\subtitle{\textsc{beamer-reveal}}
\author{Walter Daems}

\begin{document}

\begin{frame}[titleslide]
  \titlepage
\end{frame}

\AtBeginSection{
  \begin{frame}[sectionslide]{Overview}
    \tableofcontents[currentsection]
  \end{frame}
}
\AtBeginSubsection{
  \begin{frame}[subsectionslide]{Overview}
    \tableofcontents[currentsection,currentsubsection]
  \end{frame}
}

\section{Introduction}
\subsection{Slide making}

\begin{frame}
  {Good news}
  {}
  You can keep on making your slides the way you are used to!
  \begin{itemize}
    \item all the nice \LaTeX{} stuff at your fingertips
    \item no temptation to use too much unnecessary animation
  \end{itemize}
  \bigskip

  Indeed, there are no tools that can typeset equations like the tools form the \TeX-ecosystem:
  \begin{equation}
    \eu^{-\ju\pi}+1=0
  \end{equation}

```

```

\end{equation}
\end{frame}

\begin{frame}[transition=concave]
{Pr\^et-\`a-porter}
{A dummy slide}
\vfll
Showing off the 'concave' slide transition animation. Not recommended!
\vfll
\end{frame}

\begin{frame}[transition=convex]
{Très chique}
{A dummy slide}
\vfll
Showing off the 'convex' slide transition animation. Not recommended!
\vfll
\end{frame}

\subsection{Pimping your slides}

\begin{frame}
{And even more good news}
{\ldots almost seems to good to be true\ldots}
\small
However, now you can pimp your slides like never before. You can incorporate:
\begin{itemize}
\item videos and audio fragments
\item animated GIFs and LaTeX animations
\item iframe content
\end{itemize}
without being tied to Acrobat reader.
In addition, there are some extra features
\begin{itemize}
\item press '?' for keyboard help, amongst which you will find:
\item press 'm' to open the slide menu on the left
\item press 'o' to get an overview of the slides
\item press 's' to start a speaker view
\item press 'g' to go to a specific slide by typing its slide number
\end{itemize}
The pancake menu on the bottom left also opens the menu.
\end{frame}

\begin{frame}[transition=zoom]
{A dymmy slide}
{number three}
\vfll
Showing off the 'zoom slide transition animation. Not recommended!
\vfll
\end{frame}

\section{In detail}

\subsection{Candy for the eye}

\begin{frame}
{Placing videos}
{}
\only<1>{On this first slide there is nothing to see. On the next animation frame, a video will appear.}
\only<2>{Here it is!}
\video<2>[above,draw,autoplay,height=0.7,aspectratio=16/9,
background=yellow,fit=contain]
at (0.5,0.1) {Media/beamer-reveal-testvideo.mp4}
\end{frame}

```

```

\begin{frame}
  {Placing images (possibly animated)}
  {}
  \begin{columns}
    \column[T]{0.45\textwidth}
    Below you will find a png (for which you don't need reveal, BTW).
    \vspace*{1cm}

    But you can exploit the transparency!
    \vspace*{2.5cm}

    And on the top right you will find a swinging pendulum (an animated GIF).
    \column[T]{0.45\textwidth}
  \end{columns}
  \image[width=0.33,aspectratio=1,fit=contain]
  at (0.7,0.6) {Media/beamer-reveal-AnimatedPendulum.gif}
  \image[width=0.25,aspectratio=1,fit=contain]
  at (0.2,0.6) {Media/beamer-reveal-WiresTp.png}
\end{frame}

\begin{frame}
  {Placing iframe material (possibly animated)}
  {e.g. generated with asymptote}

  Click and drag on the iframe below. You can manipulate it! Use your mouse
  scroll-wheel to zoom in or out.
  \iframe[draw,anchor=north,height=0.6,aspectratio=16/9,
    fit=cover]
  at (0.5,0.7) {Media/beamer-reveal-PCB.html}
\end{frame}

\subsection{Resonance for the ear}

\begin{frame}
  {Adding audio to your slides}
  {}
  On the very bottom right, there is an audio block
  that automatically starts playing.
  \audio[draw,autoplay,controls,width=0.1,aspectratio=16/9,
    background=blue,fit=cover]
  at (0.85,0.1) {Media/beamer-reveal-AudioSample.ogg}
\end{frame}

\subsection{Make (video) animations with LaTeX}

\begin{frame}[fragile]
  {Making animations with LaTeX (using TikZ as example)}
  {It is easier than ever before}
  \small The animation content is exported to a standalone \LaTeX-document that creates a
  loop over it, for a duration of \texttt{duration} seconds at
  \texttt{framerate} frames per second providing a \texttt{\textbackslash{}progress}
  variable that goes gradually from 0 to 1 in \texttt{duration} $\times$
  \texttt{framerate} frames. The beamer-reveal.pl script transforms it to mp4 exploiting
  your full potential of your multi-core hardware.

  \animation[framerate=25,duration=7.5,pdfprogress=0.1,autoplay,loop] at (0.5,0.35) {
    \begin{tikzpicture}[font=\footnotesize,transform shape,scale=0.75]
      \pgfmathsetmacro{angle}{\progress*540}%
      \clip (-2,-5.25) rectangle (8,2);
      \node[below left,inner sep=1pt] at (0,0) {\tiny 0};
      \node[below left,inner sep=1pt] at (2.5,0) {\tiny 0};
      \node[above right,inner sep=1pt] at (0,-2) {\tiny 0};

      \begin{scope}[every node/.style={right}]
        \node[thick,draw,rectangle] at (2.5,-2)

```

```

\large  $x(t) = A \cdot e^{j\omega t}$ ;
\node at (3.5,-3)
{\large  $e^{j\alpha} = \cos\alpha + j\sin\alpha$ };
\node at (2.5,-4)
{\large  $x(t) = \underbrace{A \cos \omega t}_{\text{\textcolor{orange}{real}}} + \underbrace{j A \sin \omega t}_{\text{\textcolor{olive}{imaginary}}}$ };
\end{scope}
\draw[>,thick] (3,-2.4) -- (3,-3.4);
\draw[blue,thick] (0,0) circle (1);

\draw[>] (-1.25,0) -- (1.25,0) node[below] {Re};
\draw[>] (0,-1.25) -- (0,1.25) node[left] {Im};

% circle
\draw[olive,very thick] (0,0) -- (0,{sin(\angle)});
\draw[orange,very thick] (0,0) -- ({cos(\angle)},0);
\draw[blue,thick,>] (0,0) -- node[left,font=\tiny] {A} +(\angle:1);
\draw[>] (0.4,0) arc (0:\angle:0.4);
\node at (0.5*\angle:0.7) {\scriptsize  $\omega \tilde{t}$ };

% right graph
\draw[very thick,olive] ({2.5+\angle/180},0) -- +(0,{sin(\angle)});
\draw[densely dotted] ({min(0,cos(\angle))},{sin(\angle)})
-- ({2.5+\angle/180},{sin(\angle)});
\draw[thick] ({2.5+\angle/180},0) +(0,1pt) -- +(0,-1pt) node[below] { $\tilde{t}$ };

% bottom graph
\draw[very thick,orange] (0,{2-\angle/180}) -- +({cos(\angle)},0);
\draw[densely dotted] ({cos(\angle)},{max(0,sin(\angle))})
-- ({cos(\angle)},{2-\angle/180});
\draw[thick] (0,{2-\angle/180}) +(1pt,0) -- +(-1pt,0) node[left] { $\tilde{t}$ };

% right graph
\foreach \y/\l in {-1/-A,1/A} {
\draw[gray,densely dotted] (2.5,\y) -- (6.25,\y);
\draw (2.5,\y) +(1pt,0) -- +(-1pt,0) node[left] { $\l$ };
}
\draw[>] (2.0,0) -- (6.5,0) node[below] { $t$ };
\draw[>] (2.5,-1.25) -- (2.5,1.25) node[left] { $\text{Im}(x(t))$ };
\draw[olive,thick,domain=-0.25:3.5,samples=30,smooth] plot
({x+2.5},{sin(pi*x r)});

% bottom graph
\foreach \y/\l in {-1/-A,1/A} {
\draw[gray,densely dotted] (\y,-2) -- (\y,-4.5);
\draw (2.5,\y) +(1pt,0) -- +(-1pt,0) node[left] { $\l$ };
}
\draw[>] (-1.25,-2) -- (1.25,-2) node[above] { $\text{Re}(x(t))$ };
\draw[>] (0,-1.5) -- (0,-5) node[left] { $t$ };

\draw[orange,thick,domain=-0.25:2.6,samples=30,smooth] plot
({cos(pi*x r)},{2-x});
\end{tikzpicture}
}
\end{frame}

\end{document}
</example>

```


13 Implementation

13.1 The preamble of the package

```
1 <*reveal>
2 \ExplSyntaxOn
3 \RequirePackage{l3keys2e}
4 </reveal>
```

13.2 Error/warning messages

```
5 <*reveal>
6 \msg_new:nnn{ beamer-reveal } { inconsistent-dimensions } {
7   aspect-ratio-of-beamer~(#1)~and-reveal~(#2:#3)~are-not-consistent.\\
8   You-must-specify-consistent-values-for-width/height-and-aspectratio~
9   otherwise-your-reveal-items~(videos/images/animations)-will-not-appear~
10  on-the-right-locations-on-your-reveal-slidedeck.
11 }
12 \msg_new:nnn{ beamer-reveal } { missing-aspectratio } {
13   missing-aspect-ratio.\\
14   You-need-to-specify-at-least-an-aspect-ratio-for-a-beamer-reveal-item~
15   you-want-to-put-on-the-reveal-slide.
16 }
17 \msg_new:nnn{ beamer-reveal } { missing-width-or-height } {
18   missing-width-or-height.\\
19   You-need-to-specify-at-least-a-width-or-a-height-for-a-beamer-reveal-item~
20   you-want-to-put-on-the-reveal-slide.
21 }
22 \msg_new:nnn{ beamer-reveal } { overconstrained-box } {
23   overconstrained-box.\\
24   You-cannot-both-specify-the-width-and-the-height-of-a-beamer-reveal-item~
25   you-want-to-put-on-the-slide.
26   Specify-width-and-aspectratio-or-height-and-aspectratio.
27 }
28 \msg_new:nnn{ beamer-reveal } { dynamic-option-for-staticcontent } {
29   dynamic-option-given~(autoplay,~controls,~loop,~muted)~for-static~
30   content~(\image,~\iframe).\\
31   These-options-make-no-sense-for-the~\image~command.~Remove-them.
32 }
33 \msg_new:nnn{ beamer-reveal } { animation-option-for-nonanimation } {
34   duration-and-framerate-are-options-that-can-only-be-given-for-the~
35   \animation~command.\\
36   Remove-them-from-the~\video,~\audio,~\image~and~\iframe~commands.
37 }
38
39 </reveal>
```

13.3 Some pdf_latex backwards compatibility

```
40 <*reveal>
41 \@ifundefined{pagewidth}{%
42   \let\pagewidth\pdfpagewidth
43   \let\pageheight\pdfpageheight
44 }{}
45 </reveal>
```

13.4 Package options

First some global variables to store the global width and height of the presentation, that can be specified as package options:

```
46 <*reveal>
47 \tl_new:N \g_@@_beameraspectratio_tl
48 \tl_set:Nn \g_@@_beameraspectratio_tl {43}
```

```

49\fp_new:N \g_@@_canvaswidth_fp
50\fp_set:Nn \g_@@_canvaswidth_fp { \dim_to_fp:n {\paperwidth} }
51\fp_new:N \g_@@_canvasheight_fp
52\fp_set:Nn \g_@@_canvasheight_fp { \dim_to_fp:n {\paperheight} }
53\fp_new:N \g_@@_canvasaspectratio_fp
54\fp_set:Nn \g_@@_canvasaspectratio_fp { \g_@@_canvaswidth_fp / \g_@@_canvasheight_fp }
55\int_new:N \g_@@_canvaswidth_int
56\int_set:Nn \g_@@_canvaswidth_int {0}
57\int_new:N \g_@@_canvasheight_int
58\int_set:Nn \g_@@_canvasheight_int {0}
59\keys_define:nn { beamerreveal } {
60  width .int_set:N          = \g_@@_canvaswidth_int,
61  width .value_required:n   = true,
62  height .int_set:N         = \g_@@_canvasheight_int,
63  height .value_required:n  = true,
64}
65\ProcessKeyOptions[beamerreveal]
66\message{ PW~: \fp_use:N \g_@@_canvaswidth_fp }
67\message{ PH~: \fp_use:N \g_@@_canvasheight_fp }
68\int_compare:nNnTF { \g_@@_canvaswidth_int } = {0}
69  {
70    \int_compare:nNnTF { \g_@@_canvasheight_int } = {0}
71    {
72      % we assume 4x3 on an HD screen
73      \int_set:Nn \g_@@_canvaswidth_int { 1920 }
74      \int_set:Nn \g_@@_canvasheight_int
75        { \fp_eval:n { round( \g_@@_canvaswidth_int / \g_@@_canvasaspectratio_fp ) } }
76    }
77    {
78      % we assume 4x3
79      \int_set:Nn \g_@@_canvaswidth_int
80        { \fp_eval:n { round( \g_@@_canvasheight_int * \g_@@_canvasaspectratio_fp ) } }
81    }
82  }
83  {
84    \int_compare:nNnTF { \g_@@_canvasheight_int } = {0}
85    {
86      % we assume 4x3 on an HD screen
87      \int_set:Nn \g_@@_canvasheight_int{ \fp_eval:n
88        { round( \g_@@_canvaswidth_int / \g_@@_canvasaspectratio_fp ) } }
89    }
90    {
91      % both are set, we're good to go
92      \fp_new:N \l_@@_canvasaspectratioresidue_fp
93      \fp_set:Nn \l_@@_canvasaspectratioresidue_fp
94        { \fp_abs:n { \g_@@_canvaswidth_int / \g_@@_canvasheight_int - \g_@@_canvasaspectratio_fp } }
95      \fp_compare:nNnTF { \l_@@_canvasaspectratioresidue_fp } > {0.001}
96      {
97        \msg_warning:nneee { beamer-reveal } { inconsistent-dimensions }
98        { \tl_use:N \g_@@_beameraspectratio_tl }
99        { \int_use:N \g_@@_canvaswidth_int }
100       { \int_use:N \g_@@_canvasheight_int }
101      }{}
102    }
103  }
104</reveal>

```

13.5 Extra options for the frame environment of BEAMER

```

105<*reveal>
106\bool_new:N \g_@@_titlepage_bool
107\define@key{beamerframe}{titleslide}[true]{%
108  \ExplSyntaxOn
109  \bool_gset_true:N \g_@@_titlepage_bool

```

```

110 \ExplSyntaxOff
111 }
112 \bool_new:N \g_@@_sectionslide_bool
113 \define@key{beamerframe}{sectionslide}[true]{%
114 \ExplSyntaxOn
115 \bool_gset_true:N \g_@@_sectionslide_bool
116 \ExplSyntaxOff
117 }
118 \bool_new:N \g_@@_subsectionslide_bool
119 \define@key{beamerframe}{subsectionslide}[true]{%
120 \ExplSyntaxOn
121 \bool_gset_true:N \g_@@_subsectionslide_bool
122 \ExplSyntaxOff
123 }
124 \tl_new:N \g_@@_transition_tl
125 \define@key{beamerframe}{transition}[none]{%
126 \ExplSyntaxOn
127 \tl_gset:Nn \g_@@_transition_tl { #1 }
128 \ExplSyntaxOff
129 }
130 </reveal>

```

13.6 File writing

File handles and auxiliary functions to write data to the .rvl file.

```

131 <*reveal>
132 \iow_new:N \g_@@_rvlfile
133 \iow_open:Nn \g_@@_rvlfile {\jobname.rvl}
134 </reveal>

```

\writecomment_@@:n

```

135 <*reveal>
136 \cs_new:Npn \writecomment_@@:n #1
137 { \iow_now:Ne \g_@@_rvlfile {\c_percent_str\c_percent_str\c_space_tl #1} }
138 </reveal>

```

\writecontrol_@@:n

```

139 <*reveal>
140 \cs_new:Npn \writecontrol_@@:nn #1 #2
141 { \iow_now:Ne \g_@@_rvlfile {@ @#1:~#2} }

```

\writeliteral_@@:n

```

142 \cs_new:Npn \writeliteral_@@:n #1
143 { \iow_now:Nx \g_@@_rvlfile {#1} }

```

\riteraw_@@:n

```

144 \cs_new:Npn \riteraw_@@:n #1
145 { \iow_now:Nn \g_@@_rvlfile { #1 } }

```

Now initialize our .rvl file.

```

146 \writecomment_@@:n {Beamer-reveal driver file}
147 \writecontrol_@@:nn {Presentation} {}

```

```

148\tl_new:N \l_@@_my_compiler_tl
149\tl_set:Nn \l_@@_my_compiler_tl { unknown }
150\sys_if_engine_pdftex:T { \tl_set:Nn \l_@@_my_compiler_tl { pdflatex } }
151\sys_if_engine_xetex:T { \tl_set:Nn \l_@@_my_compiler_tl { xelatex } }
152\sys_if_engine luatex:T { \tl_set:Nn \l_@@_my_compiler_tl { luatex } }
153\writeliteral_@@:n {-parameters:
154  compiler={\tl_use:N \l_@@_my_compiler_tl },
155  canvaswidth={\int_use:N \g_@@_canvaswidth_int},
156  canvasheight={\int_use:N \g_@@_canvasheight_int}
157}
158</reveal>

```

13.7 Frame generation

```

159<*reveal>
160\AddToHook{ env / frame / begin} {
161  \bool_gset_false:N \g_@@_titlepage_bool
162  \bool_gset_false:N \g_@@_sectionslide_bool
163  \bool_gset_false:N \g_@@_subsectionslide_bool
164  \tl_gset:Nn \g_@@_transition_tl {none}
165}
166\AddToHook{ env / beamer@frameslide / before} {
167  \writecontrol_@@:nn {BeamerFrame} {}
168}
169\AddToHook{ env / beamer@frameslide / after}
170{
171  \writeliteral_@@:n
172    {-parameters:rawpage={\insertpagenumber},
173     truepage={\insertframenumbers},
174     overlay={\insertoverlaynumber},
175     transition={\tl_use:N \g_@@_transition_tl},
176     \bool_if:NTF \g_@@_sectionslide_bool
177       {
178         title={\secname},toc={section}
179       }
180       {
181         \bool_if:NTF \g_@@_subsectionslide_bool
182           {
183             title={\subsecname},toc={subsection}
184           }
185           {
186             \bool_if:NTF \g_@@_titlepage_bool
187               {
188                 title={\inserttitle},toc={titlepage}
189               }
190               {
191                 \sys_if_engine_pdftex:TF % avoid pdflatex screwing up old accents
192                   {title={\unexpanded\expandafter{\beamer@frametitle}}}
193                   {title={\beamer@frametitle}}
194               }
195             }
196         }
197      }
198}
199</reveal>

```

13.8 Common keys for the macros

```

200<*reveal>
201\fp_new:N \l_@@_mediawidth_fp
202\fp_new:N \l_@@_mediaheight_fp
203\fp_new:N \l_@@_mediaframerate_fp
204\fp_new:N \l_@@_mediaduration_fp
205\fp_new:N \l_@@_mediapdfprogress_fp

```

```

206\tl_new:N \l_@@_mediafit_tl
207\tl_new:N \l_@@_mediabackground_tl
208\fp_new:N \l_@@_xposdelta_fp
209\fp_new:N \l_@@_yposdelta_fp
210\bool_new:N \_@@_mediaautoplay_bool
211\bool_new:N \_@@_medialoop_bool
212\bool_new:N \_@@_mediaboxdraw_bool
213\bool_new:N \_@@_mediamuted_bool
214\bool_new:N \_@@_mediacontrols_bool
215\tl_new:N \l_@@_mediaanchor_tl
216
217\msg_new:nnn { beamerreveal / media } { unknown-key }
218{ Unknown-key~'#1'~for~media~(video,~animated,~...~)~command. }
219\msg_new:nnn { beamerreveal / media } { illegal-keys-video }
220{ Illegal-key(s)~'#1'~for~a~\video. }
221\msg_new:nnn { beamerreveal / media } { illegal-keys-image }
222{ Illegal-key(s)~'#1'~for~an~\image. }
223\msg_new:nnn { beamerreveal / media } { illegal-keys-iframe }
224{ Illegal-key(s)~'#1'~for~an~\iframe. }
225\msg_new:nnn { beamerreveal / media } { illegal-keys-animation }
226{ Illegal-key(s)~'#1'~for~an~\animation. }
227\keys_define:nn { beamerreveal / media } {
228  width .fp_set:N          = \l_@@_mediawidth_fp,
229  width .value_required:n  = true,
230  width .initial:n         = 0,
231  width .groups:n          = { size },
232  height .fp_set:N         = \l_@@_mediaheight_fp,
233  height .value_required:n = true,
234  height .initial:n        = 0,
235  height .groups:n         = { size },
236  aspectratio .fp_set:N    = \l_@@_mediaaspectratio_fp,
237  aspectratio .value_required:n = true,
238  aspectratio .groups:n    = { size },
239  fit .tl_set:N            = \l_@@_mediafit_tl,
240  fit .value_required:n    = true,
241  fit .initial:n           = fill,
242  fit .groups:n            = { fit },
243  background .tl_set:N     = \l_@@_mediabackground_tl,
244  background .value_required:n = true,
245  background .initial:n    = white,
246  background .groups:n     = { draw },
247  draw .bool_set:N         = \l_@@_mediaboxdraw_bool,
248  draw .groups:n           = { draw },
249  autoplay .bool_set:N     = \l_@@_mediaautoplay_bool,
250  autoplay .initial:n      = false,
251  autoplay .groups:n       = { dynamic },
252  loop .bool_set:N         = \l_@@_medialoop_bool,
253  loop .initial:n          = false,
254  loop .groups:n           = { dynamic },
255  controls .bool_set:N     = \l_@@_mediacontrols_bool,
256  controls .initial:n     = false,
257  controls .groups:n      = { dynamic },
258  muted .bool_set:N        = \l_@@_mediamuted_bool,
259  muted .initial:n         = false,
260  muted .groups:n          = { dynamic },
261  duration .fp_set:N       = \l_@@_mediaduration_fp,
262  duration .initial:n      = 0,
263  duration .value_required:n = true,
264  duration .groups:n       = { animation },
265  pdfprogress .fp_set:N    = \l_@@_mediapdfprogress_fp,
266  pdfprogress .initial:n   = 0,
267  pdfprogress .value_required:n = true,
268  pdfprogress .groups:n    = { animation },
269  framerate .fp_set:N      = \l_@@_mediaframerate_fp,
270  framerate .initial:n     = 0,

```

```

271 framerate .value_required:n = true,
272 framerate .groups:n         = { animation },
273 anchor .choice:,
274 anchor / center .code:n      = { \fp_set:Nn \l_@@_xposdelta_fp { -0.5 }
275                                \fp_set:Nn \l_@@_yposdelta_fp { -0.5 }
276                                \tl_set:Nn \l_@@_mediaanchor_tl {#1} },
277 anchor / west .code:n        = { \fp_set:Nn \l_@@_xposdelta_fp { 0 }
278                                \fp_set:Nn \l_@@_yposdelta_fp { -0.5 }
279                                \tl_set:Nn \l_@@_mediaanchor_tl {#1} },
280 anchor / north~west .code:n  = { \fp_set:Nn \l_@@_xposdelta_fp { 0 }
281                                \fp_set:Nn \l_@@_yposdelta_fp { 0 }
282                                \tl_set:Nn \l_@@_mediaanchor_tl {#1} },
283 anchor / north .code:n       = { \fp_set:Nn \l_@@_xposdelta_fp { -0.5 }
284                                \fp_set:Nn \l_@@_yposdelta_fp { 0 }
285                                \tl_set:Nn \l_@@_mediaanchor_tl {#1} },
286 anchor / north~east .code:n  = { \fp_set:Nn \l_@@_xposdelta_fp { -1 }
287                                \fp_set:Nn \l_@@_yposdelta_fp { 0 }
288                                \tl_set:Nn \l_@@_mediaanchor_tl {#1} },
289 anchor / east .code:n        = { \fp_set:Nn \l_@@_xposdelta_fp { -1 }
290                                \fp_set:Nn \l_@@_yposdelta_fp { -0.5 }
291                                \tl_set:Nn \l_@@_mediaanchor_tl {#1} },
292 anchor / south~east .code:n  = { \fp_set:Nn \l_@@_xposdelta_fp { -1 }
293                                \fp_set:Nn \l_@@_yposdelta_fp { -1 }
294                                \tl_set:Nn \l_@@_mediaanchor_tl {#1} },
295 anchor / south .code:n       = { \fp_set:Nn \l_@@_xposdelta_fp { -0.5 }
296                                \fp_set:Nn \l_@@_yposdelta_fp { -1 }
297                                \tl_set:Nn \l_@@_mediaanchor_tl {#1} },
298 anchor / south~west .code:n  = { \fp_set:Nn \l_@@_xposdelta_fp { 0 }
299                                \fp_set:Nn \l_@@_yposdelta_fp { -1 }
300                                \tl_set:Nn \l_@@_mediaanchor_tl {#1} },
301 anchor .value_required:n     = true,
302 anchor .initial:n            = center,
303 anchor .groups:n             = { position },
304 above .meta:n                = { anchor = south },
305 above .groups:n              = { position },
306 below .code:n                = { anchor = north },
307 below .groups:n              = { position },
308 left .code:n                 = { anchor = east },
309 left .groups:n               = { position },
310 right .code:n                = { anchor = west },
311 right .groups:n              = { position },
312 above~left .code:n           = { anchor = south east },
313 above~left .groups:n         = { position },
314 above~right .code:n          = { anchor = south west },
315 above~right .groups:n        = { position },
316 below~left .code:n           = { anchor = north east },
317 below~left .groups:n         = { position },
318 below~right .code:n          = { anchor = north west },
319 below~right .groups:n        = { position },
320 unknown .code:n =
321 {
322   \msg_error:nne { beamerreveal / media } { unknown-key } {\l_keys_key_str}
323 },
324 }
325 </reveal>

```

13.9 Fiddling with relative widths/heights

As mentioned in the section ‘quirks’ the package only allows specifying box dimensions as height and an aspect ratio, or width and an aspect ratio. BEAMER-REVEAL will recalculate these into fractional values relative to the slide width and height.

In order not to get lost in the calculations, let’s find the equations we need to implement. Let’s call the absolute width and height W and L and the relative width and height w and h . Let W_P and H_P be the absolute page width

and height. Obviously:

$$w = \frac{W}{W_p} \qquad h = \frac{H}{H_p} \quad (2)$$

Now let's denote the aspectratio of an object as A and the aspectratio of the page as A_p .

$$A = \frac{W}{H} \qquad A_p = \frac{W_p}{H_p} \quad (3)$$

This allows to calculate w from h and vice versa:

$$w = \frac{W}{W_p} = \frac{AH}{A_p H_p} = \frac{A}{A_p} h \qquad h = \frac{H}{H_p} = \frac{W/A}{W_p/A_p} = \frac{A_p}{A} w \quad (4)$$

The following macro calculates verifies if at least the aspectratio A has been given otherwise a fatal error is generated. It then calculates w from h or h from w unless both w and h are missing, or unless both have been specified.

`\process_a_w_h_@@:NNN`

```

326 <*reveal>
327 \cs_new:Npn \process_a_w_h_@@:NNN #1#2#3 {
328   \fp_compare:nNnTF { #1 } = {0} {
329     \msg_fatal:nn { beamer-reveal } { missing-aspectratio }
330   } {
331     \fp_compare:nNnTF { #2 } = {0} {
332       \fp_compare:nNnTF { #3 } = {0} {
333         \msg_fatal:nn { beamer-reveal } { missing-width-or-height }
334       } {
335         % calculating w
336         \fp_gset:Nn \l_tmpa_fp { round( #1 / \g_@@_canvasaspectratio_fp * #3, 6 ) }
337         \fp_gset:Nn \l_tmpb_fp { #3 }
338       }
339     } {
340       \fp_compare:nNnTF { #3 } = {0} {
341         } {
342         \msg_fatal:nn { beamer-reveal } { overconstrained-box }
343       } {
344         % calculating h
345         \fp_gset:Nn \l_tmpa_fp { #2 }
346         \fp_gset:Nn \l_tmpb_fp { round( \g_@@_canvasaspectratio_fp / #1 * #2, 6 ) }
347       }
348     }
349   }
350   \fp_set_eq:NN #2 \l_tmpa_fp
351   \fp_set_eq:NN #3 \l_tmpb_fp
352 }
353 </reveal>

```

13.10 Auxiliary functions

`\extractlleft`

```

354 <*reveal>
355 \newdimen\xlleft
356 \newdimen\ylleft
357 \newcommand*\extractlleft[1]{\path (#1);\pgfgetlastxy{\xlleft}{\ylleft};}
358 </reveal>

```

`\extractupright`

```

359 <*reveal>
360 \newdimen\xupright
361 \newdimen\yupright
362 \newcommand*\extractupright[1]{\path (#1);\pgfgetlastxy{\xupright}{\yupright};}
363 </reveal>

```

Finally, some native l3exp fp variables to hold x and y position.

```

364 <*reveal>
365 \fp_new:N \l_@@_xpos_fp
366 \fp_new:N \l_@@_ypos_fp
367 </reveal>

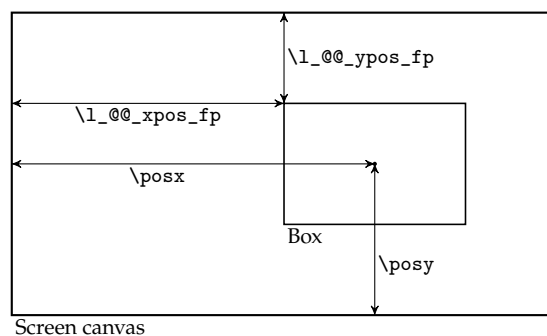
```

13.11 Macros

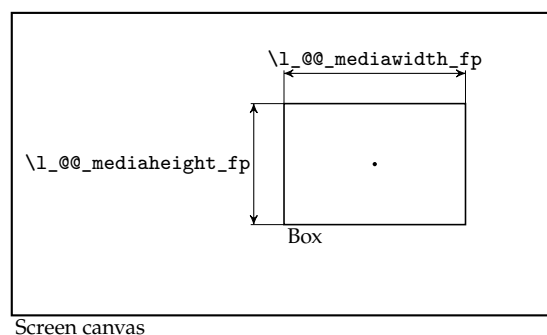
13.11.1 Main macros

The macro's fiddle with positions and sizes of boxes. Therefore it helps to have the following pictures in mind when reading the code.

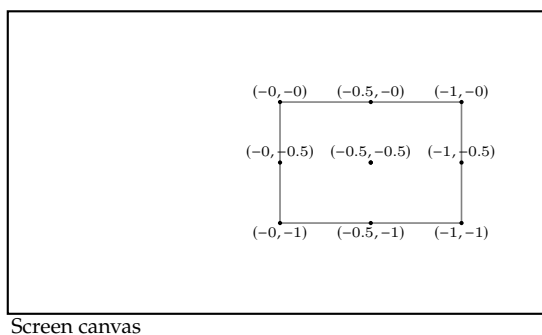
The parameters `\posx` and `\posy` are the relative *BEAMER* coordinates of the box. The parameters `\l_@@_xpos_fp` and `\l_@@_ypos_fp` are the HTML top and left distances of the top left corner of the box. All parameters are relative to the screen canvas dimensions (and therefore have values in between 0 and 1).



The width and height of the box have dedicated parameters.



The anchor of a box can be any of the main 8 wind directions (north, east, south, west, and the ones in between those). This anchor location in combination with the width and height of the box, allow for a correction on the HTML coordinates. The correction coefficients (relative to width and height) are:



\video

```

368 <*reveal>
369 \tl_new:N \l_@@_keyoverflow_tl
370 \NewDocumentCommand\video{D<>{1-} 0{} u{at} r() m}{
371   \only<#1> {
372     % begin group to keep the key setting local
373     \group_begin:
374     \keys_set_exclude_groups:nnnN { beamerreveal / media } { animation } { #2 } \l_@@_keyoverflow_tl
375     \tl_if_empty:NTF \l_@@_keyoverflow_tl {} {
376       \msg_fatal:nne { beamerreveal / media } { illegal-keys-video } { \l_@@_keyoverflow_tl }
377     }
378     \dynamicContent_@@:nnn {#4} {#5} {video}
379     \group_end:
380   }
381 }
382 </reveal>

```

\audio

```

383 <*reveal>
384 \NewDocumentCommand\audio{D<>{1-} 0{} u{at} r() m}{
385   \only<#1> {
386     % begin group to keep the key setting local
387     \group_begin:
388     \keys_set_exclude_groups:nnnN { beamerreveal / media } { animation } { #2 } \l_@@_keyoverflow_tl
389     \tl_if_empty:NTF \l_@@_keyoverflow_tl {} {
390       \msg_fatal:nne { beamerreveal / media } { illegal-keys-audio } { \l_@@_keyoverflow_tl }
391     }
392     \dynamicContent_@@:nnn {#4} {#5} {audio}
393     \group_end:
394   }
395 }
396 </reveal>

```

\iframe

```

397 <*reveal>
398 \NewDocumentCommand\iframe{D<>{1-} 0{} u{at} r() m}{
399   \only<#1> {
400     % begin group to keep the key setting local
401     \group_begin:
402     \keys_set_exclude_groups:nnnN { beamerreveal / media } { dynamic animation } { #2 } \l_@@_keyoverflow_tl
403     \tl_if_empty:NTF \l_@@_keyoverflow_tl {} {
404       \msg_fatal:nne { beamerreveal / media } { illegal-keys-iframe } { \l_@@_keyoverflow_tl }
405     }
406     \dynamicContent_@@:nnn {#4} {#5} {iframe}
407     \group_end:

```

```

408 }
409 }
410 </reveal>

```

`\image`

```

411 <*reveal>
412 \NewDocumentCommand\image{D<>{1-} 0{} u{at} r() m}{
413   \only<#1> {
414     % begin group to keep the key setting local
415     \group_begin:
416     \keys_set_exclude_groups:nnnN { beamerreveal / media } { dynamic } { #2 } \l_@@_keyoverflow_tl
417     \tl_if_empty:NTF \l_@@_keyoverflow_tl {} {
418       \msg_fatal:nne { beamerreveal / media } { illegal-keys-image } { \l_@@_keyoverflow_tl }
419     }
420     \staticContent_@@:nnn {#4} {#5} {image}
421     \group_end:
422   }
423 }
424 </reveal>

```

`\animation`

```

425 <*reveal>
426 \NewDocumentCommand\animation{D<>{1-} 0{} u{at} r() +m}{
427   \only<#1> {
428     % begin group to keep the key setting local
429     \group_begin:
430     \keys_set_exclude_groups:nnnN { beamerreveal / media } { size, fit } { #2 } \l_@@_keyoverflow_tl
431     \tl_if_empty:NTF \l_@@_keyoverflow_tl {} {
432       \msg_fatal:nne { beamerreveal / media } { illegal-keys-animation } { \l_@@_keyoverflow_tl }
433     }
434     \fixedContent_@@:nnn {#4} {#5} {animation}
435     \group_end:
436   }
437 }
438 </reveal>

```

13.11.2 Auxiliary macros

`\dynamicContent_@@:nnn`

```

439 <*reveal>
440 \cs_new:Npn \dynamicContent_@@:nnn #1#2#3 {
441   % convert combination width/aspectratio to height, or
442   % height/aspectratio to width
443   \process_a_w_h_@@:NNN \l_@@_mediaaspectratio_fp \l_@@_mediawidth_fp \l_@@_mediaheight_fp
444
445   % extract relative position from bottom left of page (0,0) to top right (1,1)
446   \seq_set_split:Nnn \l_tmpa_seq { , } { #1 }
447   \pgfmathsetmacro{\posx}{\seq_item:Nn \l_tmpa_seq {1}}
448   \pgfmathsetmacro{\posy}{\seq_item:Nn \l_tmpa_seq {2}}
449   % convert to html coordinates from top left corner, and correct for anchor location
450   \fp_set:Nn \l_@@_xpos_fp { \posx + \l_@@_xposdelta_fp * \l_@@_mediawidth_fp }
451   \fp_set:Nn \l_@@_ypos_fp { (1-\posy) + \l_@@_yposdelta_fp * \l_@@_mediaheight_fp }
452
453   \writeliteral_@@:n {
454     -#3:
455     width={\fp_use:N \l_@@_mediawidth_fp},

```

```

456     height={\fp_use:N \l_@@_mediaheight_fp},
457     fit={\tl_use:N \l_@@_mediafit_tl},
458     background={\tl_use:N \l_@@_mediabackground_tl},
459     \bool_if:NTF \l_@@_mediaautoplay_bool {autoplay={},} {}
460     \bool_if:NTF \l_@@_mediacontrols_bool {controls={},} {}
461     \bool_if:NTF \l_@@_medialoop_bool {loop={},} {}
462     \bool_if:NTF \l_@@_mediamuted_bool {muted={},} {}
463     x={\fp_use:N \l_@@_xpos_fp},
464     y={\fp_use:N \l_@@_ypos_fp},
465     file={#2}
466 }
467
468 \bool_if:NTF \l_@@_mediaboxdraw_bool {
469   \begin{tikzpicture}[overlay,remember~picture,font=\tiny]
470     \extractlloleft{$(current~page.south~west)$}
471     \extractupright{$(current~page.north~east)$}
472     \node[
473       anchor = \tl_use:N \l_@@_mediaanchor_tl,
474       minimum~width={\fp_use:N \l_@@_mediawidth_fp * (\xupright - \xlloleft)},
475       minimum~height={\fp_use:N \l_@@_mediaheight_fp * (\yupright - \ylloleft)},
476       draw,
477       fill=\tl_use:N \l_@@_mediabackground_tl,
478     ] at ({\xlloleft*(1-\posx)+\xupright*\posx},{\ylloleft*(1-\posy)+\yupright*\posy})
479       {\textcolor{gray}{#2}};
480   \end{tikzpicture}
481 }{}
482 }
483 </reveal>

```

`\staticContent_@:nnn`

```

484 <*reveal>
485 \cs_new:Npn \staticContent_@:nnn #1#2#3 {
486   % convert combination width/aspectratio to height, or
487   % height/aspectratio to width
488   \process_a_w_h_@@:NNN \l_@@_mediaaspectratio_fp \l_@@_mediawidth_fp \l_@@_mediaheight_fp
489
490   % extract relative position from bottom left of page (0,0) to top right (1,1)
491   \seq_set_split:Nnn \l_tmpa_seq { , } { #1 }
492   \pgfmathsetmacro{\posx}{\seq_item:Nn \l_tmpa_seq {1}}
493   \pgfmathsetmacro{\posy}{\seq_item:Nn \l_tmpa_seq {2}}
494   % convert to html coordinates from top left corner, and correct for anchor location
495   \fp_set:Nn \l_@@_xpos_fp { \posx + \l_@@_xposdelta_fp * \l_@@_mediawidth_fp }
496   \fp_set:Nn \l_@@_ypos_fp { (1-\posy) + \l_@@_yposdelta_fp * \l_@@_mediaheight_fp }
497
498   \writeliteral_@@:n {
499     -#3:
500     width={\fp_use:N \l_@@_mediawidth_fp},
501     height={\fp_use:N \l_@@_mediaheight_fp},
502     fit={\tl_use:N \l_@@_mediafit_tl},
503     background={\tl_use:N \l_@@_mediabackground_tl},
504     \bool_if:NTF \l_@@_mediaautoplay_bool {autoplay={true},} {}
505     x={\fp_use:N \l_@@_xpos_fp},
506     y={\fp_use:N \l_@@_ypos_fp},
507     file={#2}
508   }
509
510   \bool_if:NTF \l_@@_mediaboxdraw_bool {
511     \begin{tikzpicture}[overlay,remember~picture,font=\tiny]
512       \extractlloleft{$(current~page.south~west)$}
513       \extractupright{$(current~page.north~east)$}
514       \node[
515         anchor = \tl_use:N \l_@@_mediaanchor_tl,

```

```

516     minimum~width={\fp_use:N \l_@@_mediawidth_fp * (\xupright - \xlleft)},
517     minimum~height={\fp_use:N \l_@@_mediaheight_fp * (\yupright - \ylleft)},
518     draw,
519     fill=\tl_use:N \l_@@_mediabackground_tl,
520   ] at ({\xlleft*(1-\posx)+\xupright*\posx},{\ylleft*(1-\posy)+\yupright*\posy})
521     {\textcolor{gray}{#2}};
522   \end{tikzpicture}
523 }{}
524 }
525 \</reveal>

```

\fixedContent_@@:nnn

```

526 \<*\reveal>
527 \cs_new:Npn \fixedContent_@@:nnn #1#2#3 {
528   % first create the box with the content and measure it
529   \pgfmathsetmacro\progress{ \fp_use:N \l_@@_mediapdfprogress_fp }%
530   \hbox_set:Nn \l_tmpa_box {#2}
531   \fp_set:Nn \l_tmpa_fp { \dim_eval:n { \box_wd:N \l_tmpa_box } }
532   \fp_set:Nn \l_tmpb_fp { \dim_eval:n { \pagewidth } }
533   \fp_set:Nn \l_@@_mediawidth_fp { \fp_eval:n { \l_tmpa_fp / \l_tmpb_fp } }
534   \fp_set:Nn \l_tmpa_fp { \dim_eval:n { \box_ht:N \l_tmpa_box } }
535   \fp_set:Nn \l_tmpb_fp { \dim_eval:n { \pageheight } }
536   \fp_set:Nn \l_@@_mediaheight_fp { \fp_eval:n { \l_tmpa_fp / \l_tmpb_fp } }
537
538   % extract relative position from bottom left of page (0,0) to top right (1,1)
539   \seq_set_split:Nnn \l_tmpa_seq { , } { #1 }
540   \pgfmathsetmacro{\posx}{\seq_item:Nn \l_tmpa_seq {1}}
541   \pgfmathsetmacro{\posy}{\seq_item:Nn \l_tmpa_seq {2}}
542   % convert to html coordinates from top left corner, and correct for anchor location
543   \fp_set:Nn \l_@@_xpos_fp { \posx + \l_@@_xposdelta_fp * \l_@@_mediawidth_fp }
544   \fp_set:Nn \l_@@_ypos_fp { (1-\posy) + \l_@@_yposdelta_fp * \l_@@_mediaheight_fp }
545
546   \writeliteral_@@:n {
547     -#3:
548     width={ \fp_use:N \l_@@_mediawidth_fp },
549     height={ \fp_use:N \l_@@_mediaheight_fp },
550     framerate={ \fp_use:N \l_@@_mediaframerate_fp },
551     duration={ \fp_use:N \l_@@_mediaduration_fp },
552     x={\fp_use:N \l_@@_xpos_fp},
553     y={\fp_use:N \l_@@_ypos_fp},
554     fit={fit},
555     background={\tl_use:N \l_@@_mediabackground_tl},
556     \bool_if:NTF \l_@@_mediaautoplay_bool {autoplay={},} {}
557     \bool_if:NTF \l_@@_medialoop_bool {loop={},} {}
558     \bool_if:NTF \l_@@_mediacontrols_bool {controls={},} {}
559   }
560
561   \begin{tikzpicture}[overlay,remember~picture,font=\tiny]
562     \extractlleft{$(current~page.south~west)$}
563     \extractupright{$(current~page.north~east)$}
564     \node at ({\xlleft*(1-\posx)+\xupright*\posx},{\ylleft*(1-\posy)+\yupright*\posy})
565       {\box_use:N \l_tmpa_box};
566   \end{tikzpicture}
567
568   \writeraw_@@:n {
569     #2
570   }
571 }
572 \</reveal>

```

13.12 Postamble

```
573 <*reveal>
574 \AtEndDocument{
575   \ExplSyntaxOn
576   \iow_close:N \g_@@_rvlfile
577   \ExplSyntaxOff
578 }
579 </reveal>
```

References

- [1] Till Tantau et al., "beamer - A \LaTeX class for producing presentations and slides", <https://ctan.org/pkg/beamer>, online, accessed in December 2025.
- [2] Alexander Grahn, "movie15 - Multimedia inclusion package", <https://ctan.org/pkg/movie15>, online, accessed in December 2025.
- [3] Alexander Grahn, "media9 - Multimedia inclusion package with Adobe Reader-9/X compatibility", <https://ctan.org/pkg/media9>, online, accessed in December 2025.
- [4] Grant Sanderson, "manim - A community maintained Python library for creating mathematical animations", <https://www.manim.community>, online, accessed in December 2025.
- [5] Heiko Oberdiek, "pdfcrop — Crop PDF graphics", <https://ctan.org/pkg/pdfcrop>, online, accessed in December 2025.
- [6] Poppler developers, "Poppler Utilities (pdftoppm)", <https://poppler.freedesktop.org>, online, accessed in December 2025.
- [7] FFmpeg developers, "ffmpeg tool", <https://ffmpeg.org>, online, accessed in December 2025.

Change History

v0.80	General: Alpha 1 embryonic demo version 1	v1.01	General: Restored pdflatex compatibility (including macro-based accented characters) 1
v0.85	General: Alpha 2 tested by Walter, functional on Linux 1	v1.02	General: Restored pdflatex compatibility (including macro-based accented characters) - needed to correct faulty upload 1
v0.90	General: Beta 1 tested by Paul, not functional on MS-Windows 1	v1.03	General: Introduced new command-line options to accomodate users of latexmk 1
v0.95	General: Beta 2 tested by Paul, functional on MS-Windows 1		
v1.00	General: First appearance on CTAN 1		

Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in *roman* refer to the code lines where the entry is used.

Symbols	J	W
\ifundefined 41	\jobname 133	\writecomment 136, 146
A	P	\writecomment_@@:n 134
\animation 35, 226, <u>424</u> , 426	\pdfpageheight 43	\writecontrol 140, 147, 167
\audio 36, <u>382</u> , 384	\pdfpagewidth 42	\writecontrol_@@:n 138
D	\pgfgetlastxy 357, 362	\writeliteral
\dynamicContent . . 378, 392, 406, 440	\posx 447, 450, 478, 492, 495, 520, 540, 543, 564 142, 153, 171, 453, 498, 546
\dynamicContent_@@:nnn 438	\posy 448, 451, 478, 493, 496, 520, 541, 544, 564	\writeliteral_@@:n 141
E	\process 327, 443, 488	\writeraw 144, 568
\expandafter 192	\process_a_w_h_@@:NNN . . . 325	\writeraw_@@:n 143
\extractlleft <u>353</u> , 357, 470, 512, 562	\progress 529	X
\extractupright <u>358</u> , 362, 471, 513, 563	S	\xlleft 355, 357, 474, 478, 516, 520, 564
F	\staticContent 420, 485	\xupright
\fixedContent 434, 527	\staticContent_@@:nnn 483	. 360, 362, 474, 478, 516, 520, 564
\fixedContent_@@:nnn 525	U	Y
I	\unexpanded 192	\ylleft 356, 357, 475, 478, 517, 520, 564
\iframe 30, 36, 224, <u>396</u> , 398	V	\yupright
\image 30, 31, 36, 222, <u>410</u> , 412	\video 36, 220, <u>367</u> , 370	. 361, 362, 475, 478, 517, 520, 564