# randomplaceholder user manual

yaxinm

September 2022

Want to add bit of fun to your writing process? Insert a random image into your article by typing `\placeholder`.


Figure 1: A random figure.

## 1 Supported platforms

**Unix** systems and **Overleaf**, compiled with pdfLaTex. *Shell-escape must be enabled.* Shell commands are used to gather the list of images.

The package has been tested on Ubuntu 20.04 (x86_64) and Overleaf.

Sadly, it doesn't work on **Windows** yet.

## 2 Installation

Installation is different for local LaTeX project or Overleaf.

### 2.1 Install package for one project

Use this method for **Overleaf**. Copy the content of the folder `randomplaceholder` into your project, place the `.sty` right next to your `main.tex`.

### 2.2 Install package for all local projects

Copy the folder `randomplaceholder` into your TeX home directory. On linux, using the default, you should copy `randomplaceholder` into the directory `~/texmf/tex/latex/`.

You can find your TeX home with the command `kpsewhich -var-value=TEXMFHOME`.

# 3 How to use

## 3.1 Usepackage

Add this line to the top of your file.

```
\usepackage{randomplaceholder}
```

Now, prepare a folder full of fun images. Make sure that folder contains only images (any format) and the names of both the directory and the files follow the Unix naming conventions. For example, `placeholderimage/` in this repo.

## 3.2 Initialise

Initialise the package by adding the following command at the top of your file.

```
\placeholderinit{$YourImageFolder}
```

`$YourImageFolder` is a valid path to a directory, no space between the {, } and the path is allowed.

By running `\placeholderinit`, a file containing the macros will be created base on the content of `$YourImageFolder`. Once the macro file is present, `\placeholderinit` will not overwrite it even if the content of `$YourImageFolder` has changed or if you recompile. To reinitiate, call `\placeholderclean` before calling `\placeholderinit`.

## 3.3 Add a placeholder

Use `\placeholder` to add a image randomly selected from `$YourImageFolder`, no argument necessary. For more control, see next section.

# 4 Available commands

`\placeholderinit{#1}` Initialise the randomplaceholder by creating a `placeholdermacros.tmp` in the project directory. The command is skipped over if the macro file already exist.

- `#1` path to the folder with the space holder images. Only images should be present in the folder and names should follow the Unix conventions.

`\placeholderclean` Remove `placeholdermacros.tmp` so the package can be reinitialised to reflect changes.

`\placeholder[#1][#2][#3][#4]` Insert a randomly selected image. Use a empty argument `[]` to use default values.

- `#1` Optional, default=htb. Position of the figure, takes floating figure position specifiers.
- `#2` Optional, default=`\textwidth`. Takes a width specifier.
- `#3` Optional, default=None. Text in the figure caption.
- `#4` Optional, default=None. Text in the figure label.

`\includeplaceholder[#1]` Use this in place of `\includegraphics` in the `figure` environment.

`#1` Optional, default=`width=\textwidth`. Take key-value list for command `\includegraphics` from the `graphicx` package.

`\placeholderimage{#1}` You won't need this noramlly, but here it is: returns the file name of the `#1`$^{\text{th}}$ image in the image folder.

# 5 Examples

Here are some codes and their results

```
\placeholder
\placeholder[htb][5cm]
\placeholder[][0.3\textwidth][The third argument is the caption]
\placeholder[][0.3\textwidth][][fig5]
Reference figure 5 with \ref{fig5}

% or use a image in your own \begin{figure}
\begin{figure}
    \centering
    \includeplaceholder[width=5cm]
    \caption{Use 'includeplaceholder' for more control.}
\end{figure}
```

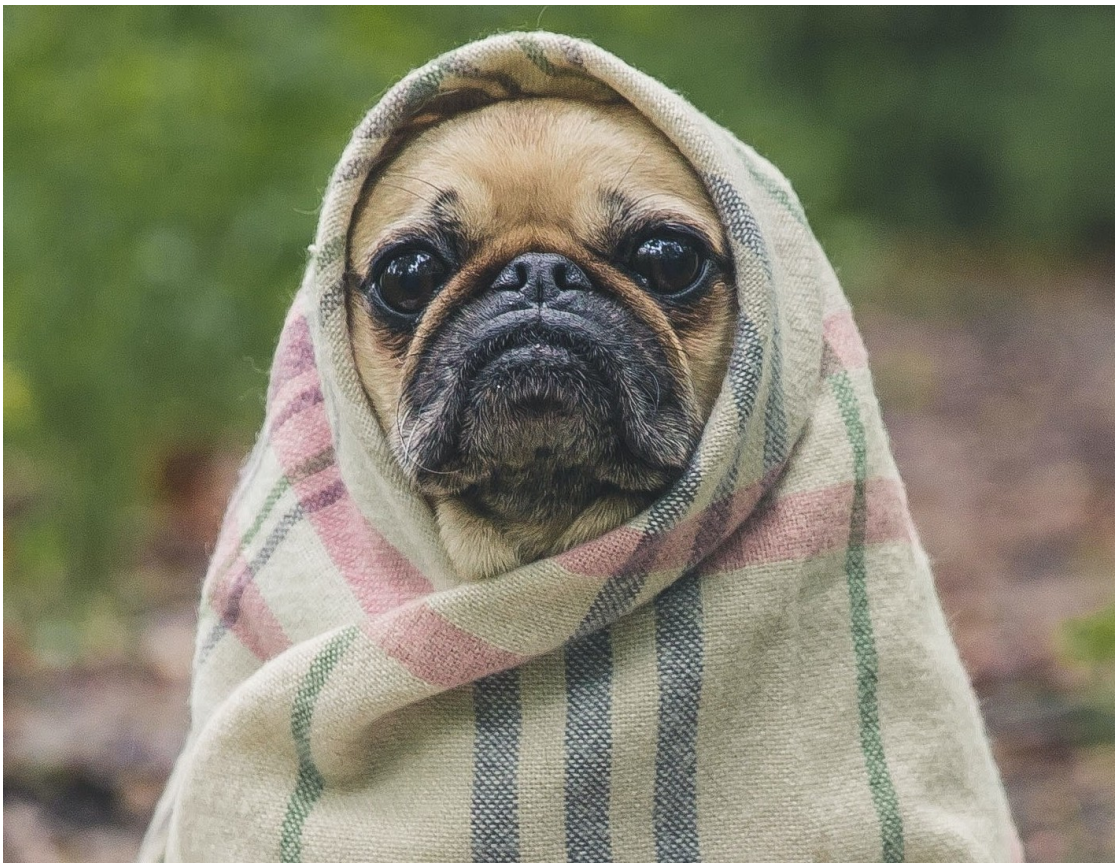

Figure 2:

Reference figure 5 with 5

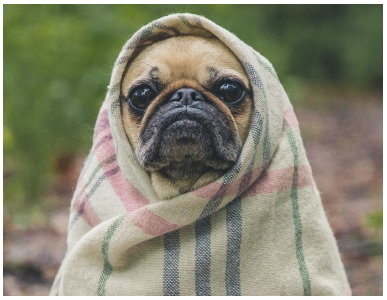

Figure 3:

Figure 4: The third argument is the caption

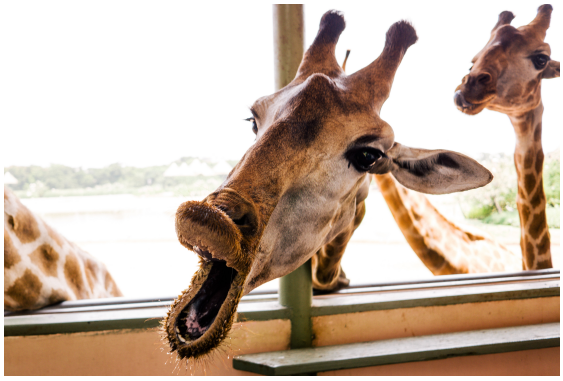

Figure 5:


Figure 6: Use ‘includeplaceholder’ for more control.