LaTeX Tutorial
Home
Introduction
The folder path to images
Changing the image size and rotating the picture
Positioning
Captions
Labels and cross-references
Generating high-res and low-res images
Reference guide
Units of LaTeX and length
Types of Images in LaTeX
Vector format or Bit-map format?

Inserting Images In LaTeX

> Images are an integral part of documents. They make the reader understand the content and relate to it. LaTeX provides various options to handle the images and present them in a way you want. In the coming topics, we will explain how to include images, compress, enlarge, and to add a reference for them within the document.

Mangler du en vikar?
Fleksibelt vikarbureau med dag-til-dag booking - Få hurtig hjælp her!
Power People Danmark ApS
Åbn

Table of Contents

1. Introduction
2. The folder path to images
3. Changing the image size and rotating the picture
4. Positioning
5. Captions
6. Labels and cross-references
7. Generating high-res and low-res images
8. Reference guide
9. Units of LaTeX and length
10. Types of Images in LaTeX
11. Vector format or Bit-map format?

Introduction

Below is an example on how to import a picture.

```
\documentclass{article}
\usepackage{graphicx}
\graphicspath{ {./images/} }

\begin{document}
Sun rises in the east and sets in the west.

\includegraphics{sun}

There's a picture of Sun above
\end{document}
```

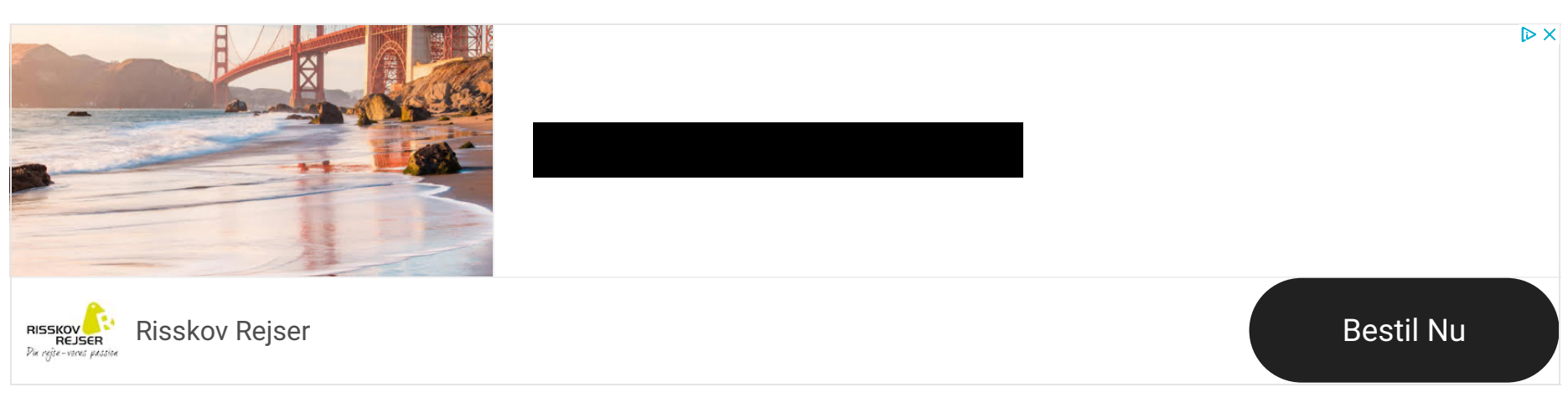
We use the **graphicx** package to manage images in LaTeX. To include that, please write the following command in the preamble: `\usepackage{graphicx}`

The place where the images are kept is denoted by the command `\graphicspath{ {./images/} }` . The folder "images" is under the directory of the main document.

The `\includegraphics{sun}` command includes the image named "sun" in the document. Here, we have not added any extension or the image format to the name. Also, the name of the file should not contain white spaces or multiple dots.

Please note that the file extension can be included, but it is usually recommended to omit it. This way, LaTeX will search all the supported format for that file within the folder.

The folder path to images



[redacted]
Risskov Rejser
Bestil Nu

It is a good practice to keep the images in a separate dedicated folder for a particular project. The command `\graphicspath{ {pictures/} }` tells LATEX that "pictures" is the folder to look for the images. The compiler will look for the folder in the current working directory.

This method might create confusion when .tex files within folders are included in the main .tex file. In this case, the compiler may end up looking for the folder of pictures at the wrong place. Therefore, it is always recommended to **specify the graphics path to be relative to the main .tex file**, denoting the main .tex file directory as ./ , for instance

```
%Path relative to the main .tex file
\graphicspath{ {./pictures/} }
```

as in the introduction.

The path can be **absolute** also meaning, the path has the exact location of the file on your system. For example,

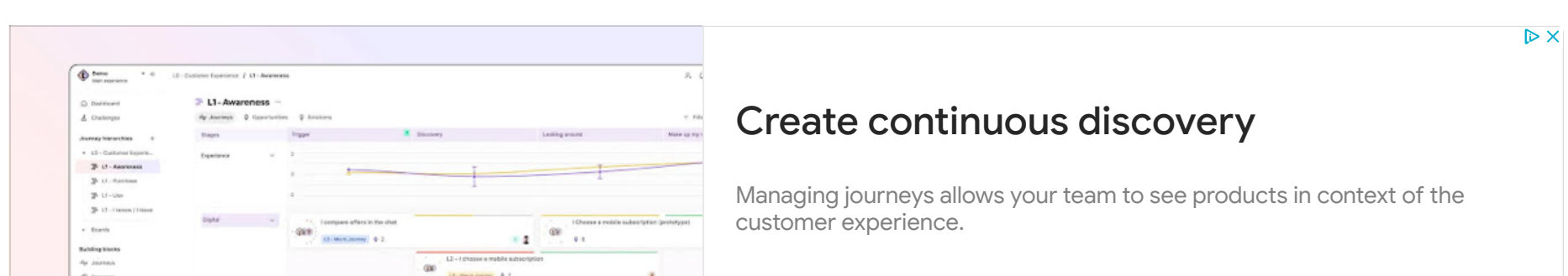
```
%Path in Windows format:
\graphicspath{ {c:/user/pictures/} }

%Path in Unix-like (Linux, Mac OS) format
\graphicspath{ {/home/user/pictures/} }
```

Notice that this command requires a trailing slash / and that the path is in between double braces.

Multiple paths can be set in case the images are saved in more than one folder. For instance, if there are two folders named pictures1 and pictures2, you can use the command:

```
\graphicspath{ {./pictures1/}{./pictures2/} }
```

Changing the image size and rotating the picture



Create continuous discovery
Managing journeys allows your team to see products in context of the customer experience.
TheyDo Journey Management
Open

We can set the specifications of the image like, height, length etc by using the below mentioned format:

```
\begin{document}

Here, we are learning to change the specifications of the image.

\includegraphics[scale=2.5]{moon}
```

The command `\includegraphics[scale=2.5]{moon}` includes the image "moon" in the document. Here, the parameter **scale** will make the image as 2.5 times of its original image.

Latex Figure Scale Width and height

Images can also be scaled to a particular height and width.

```
\begin{document}

Here, we are learning to change the width and height of the image.

\includegraphics[width=2cm, height=3cm]{moon}
```

Here, the parameters width and height set the width and height of the image respectively to 2cm and 3cm respectively. Other units of measurement can also be used. If you pass the value only for width, the height will be adjusted to keep the aspect ratio intact.

The length units can be set relative to some elements of the document. For example, if you want to set the width of the image as the width of the text, you can do it like this:

```
\begin{document}

Honesty is the best policy.

\includegraphics[width=\textwidth]{vegetables}
```

Other LaTeX lenth parameters which can be used instead of \textwidth are: \textheight, \columnsep, \paperheight, \linewidth, etc.

There is another feature of rotating a picture. Following is an instance of the same.

```
\begin{document}

Now, we will see how the image is getting rotated in LaTeX.

\includegraphics[scale=2.2, angle=55]{box}
```

The picture is rotated 55 degrees counter-clockwise with the parameter angle=55. To rotate the picture clockwise, use a negative number.

Positioning

In this section, we will learn how to position the image in the document. A new environment will be introduced to implement this.

```
In the below example, the image will be positioned
right below this sentence.

\begin{figure}[h]
\includegraphics[width=9cm]{sun}
\end{figure}
```

The **figure** environment displays the pictures as floating elements within the document. By including the picture inside the figure environment, you don't have to worry about it's placement. LATeX will help position it in a way that it fits into the flow of the document.

We can also pass an additional parameter to determine the figure positioning. In the example above, begin{figure}[h], the parameter 'h' inside the brackets set the position of the figure to **here**. Below is a table listing the possible positioning values.

Parameter	Position
!	Override internal parameters LaTeX uses for determining "good" float positions.
b	Position at the bottom of the page.
h	Place the float here, i.e., approximately at the same point it occurs in the source text (however, not exactly at the spot)
H	Places the float at precisely the location in the LATEX code. Requires the float package, though may cause problems occasionally. This is somewhat equivalent to h!.
p	Put on a special page for floats only.
t	Position at the top of the page.

Now, let's look at another example. In this, we will see the picture at the top of the document, despite being declared below the text.

```
Now, here we see a different thing. Let's explore.

\begin{figure}[t]
\includegraphics[width=9cm]{Nature}
\centering
\end{figure}
```

The command `\centering` will place the picture in the centre. By default, the alignment is left.

There is also a feature to wrap the text around a figure. If the document has multiple small pictures, this makes it look better.

```
\begin{wrapfigure}{r}{0.35\textwidth} %this figure is placed at the right
    \centering
    \includegraphics[width=0.35\textwidth]{nature}
\end{wrapfigure}

Now, here we will see different images wrapped around the text. We have added this content here to make you understand the wrapping better. Hope this helps!

\begin{wrapfigure}{l}{0.35\textwidth}
    \centering
    \includegraphics[width=0.35\textwidth]{trees}
\end{wrapfigure}

You will see how the images are getting displayed. We have added two pictures here, one of the nature and the other of tress. Let's see how this gets displayed in the document.
Simple repetition of the above text for understanding wrap feature. You will see how the images are getting displayed. We have added two pictures here, one of the nature and the other of tress. Let's see how this gets displayed in the document.
```

To make the commands work, we have to import the **wrapfig** package. For this, add this line to the preamble: `\usepackage{wrapfig}`.

The **wrapfigure** environment is defined by the following commands: `\begin{wrapfigure}{l}{0.35\textwidth} \end{wrapfigure}` . This environment has two additional parameters. Below is an explanation of the above code:

- `{l}` : This parameter defines the alignment of the figure. For left and right, use l and r respectively. Also, for book or other similar format, use o for the outer edge and i for the inner edge of the page.
- `{0.35\textwidth}` : This denotes the width of the figure box. It is not the width of the image itself, that has to be set in the **includegraphics** command.

Captioning, labelling and referencing

Captions act as a brief description of the images and labels denote further references for the image. This helps to make the document more interactive.

Captions

Here is one example:

```
\begin{figure}[h]
\caption{Example of a plot ($\sin (x), \cos(x), x$)}
\centering
\includegraphics[width=0.5\textwidth]{spiralpic}
\end{figure}
```

Simply write `\caption{Some caption}` and at the underlined position, write your caption for the image. The caption is placed wherever you write the command. If the caption is written above the includegraphics command then, the caption will be on top of it, and if it is written below then, the caption will also be set below the figure.

The sidecap package helps to accomplish this.

```
\documentclass{article}
\usepackage[leftcaption]{sidecap}

\usepackage{graphicx} %package to manage images
\graphicspath{ {pictures/} }

\begin{SCfigure}[0.6][h]
\caption{This is a caption}
\includegraphics[width=0.7\textwidth]{galaxy}
\end{SCfigure}
```

`\usepackage[leftcaption]{sidecap}` : This command will import the sidecap package. It takes an additional parameter: **leftcaption**. With this parameter, the caption is placed at the right of the picture. Similarly, you can also use **rightcaption**. Parameters outercaption and innercaption are also available in book like documents.

`\begin{SCfigure}[0.6][h] \end{SCfigure}`: This command defines an environment similar to figure. There are two parameters here. The first is the width of the caption relative to the size of the image and the second one works exactly same as in the figure environment.

Labels and cross-references

References can be added for figures within the text. This can be done by simply adding a label to the figure or SCfigure environment, and then use that label to refer to the picture.

```
\begin{figure}[h]
    \centering
    \includegraphics[width=0.35\textwidth]{sun}
    \caption{Sun rays}
    \label{fig:sun1}
\end{figure}

Now, we can see in the figure \ref{fig:sun1}, the
function grows near 0. Also, in the page \pageref{fig:sun1}
is the same example.
```

There are three commands that generate cross-references in this example.

- `\label{fig:sun1}`: This command sets a label for this figure. We have also used a prefix here "fig".
- `\ref{fig:sun1}` : This command inserts the number assigned to the figure. It gets automatically generated and will be updated if we insert some other figure before the referenced one.
- `\pageref{fig:sun1}` : This command prints the page number where the referenced image appears.

To reference a figure, `\caption` is mandatory.

We can also generate a list of figures in LaTeX by using a simple command `\listoffigures`

But this command will work on figures with captions, since it uses the caption in the table. The above example will list the images in this article.

Important Note: While using cross-references, the LaTeX project should be compiled twice, otherwise the references, the page references and the table of figures won't work.

Generating high-res and low-res images

So far, we have followed the practice of omitting file extension with the name of the image while using \includegraphics. Though it is useful, we can still include file extension. In the case when the file extension is omitted, LaTeX will search for any supported image format in that directory, and will search for various extensions in the default order (which can be modified later).

This is useful while switching between production and development environments. In a development environment when the book/report/article that you are writing is still in progress, it is good to use images with low-resolution (typically in .png format) so that they can be quickly complied for the preview. In the production environment when we are almost ready with final version of the book/report/article , it is better to include the images with high-resolution.

This can be implemented by:

- Not including the file extension in the \includegraphics command
- Writing the desired extension in the preamble

Suppose, we have two versions of one image, for example, chart1.pdf (high resolution) and chart1.png (low resolution) then, we can write the following line in the preamble to use the mow resolution i,e., .png version while developing the report.

```
\DeclareGraphicsExtensions{.png,.pdf}
```

This command ensures that whenever two files are there with the same name but different extension (as in the example above, chart1.pdf and chart1.png) then, .png – low resolution version will be used first, and when it is not there, .pdf – high resolution version will be used.

Once the report is complete, to use the high-resolution .pdf version, we can change the line in the preamble to:

```
\DeclareGraphicsExtensions{.pdf,.png}
```

If we don't have a low resolution image, we can request LaTeX for the same while compiling the document. To do that, include the following line in the preamble.

```
\usepackage{graphicx}

\usepackage{epstopdf}
\epstopdfDeclareGraphicsR/ule{.pdf}{png}{.png}{convert #1 \OutputFile}
\DeclareGraphicsExtensions{.png,.pdf}
```

If chart1.pdf exists but chart1.png doesn't then, chart1-pdf-converted-to.png will be created and loaded in its place. The command convert #1 does the conversion and additional parameters may be passed between convert and #1. For example - convert -density 100 #1.

There are some important things to have in mind though:

- To make the automatic conversion work, call pdflatex with the --shell-escape option.
- For the final production version, comment out the `\epstopdfDeclareGraphicsRule`, so that only high-resolution images get loaded. We might also have to change the order of precedence.

Reference guide

Units of LaTeX and length

Abbreviation	Definition
\columnsep	distance between columns
\columnwidth	width of the column
\linewidth	width of the line in the current environment
\paperheight	height of the page
\paperwidth	width of the page
\textheight	height of the text
\textwidth	width of the text
\unitleght	units of length in the picture environment.
cm	a centimeter
em	the width of an m in the current font
ex	the height of an x in the current font
in	an inch
mm	a millimeter
pt	A point, is the default length unit. About 0.3515mm

Types of Images in LaTeX

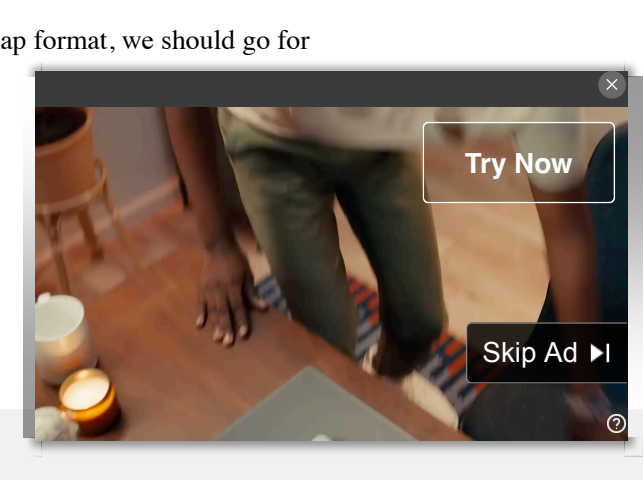
- **Latex** : When compiling with latex, only EPS images can be used, which is a vector format.
- **Pdflatex** While compiling using "pdflatex" to produce a PDF, a number of image formats can be used.
  - **JPG**: Best choice to insert photoss
  - **PNG**: Best choice to insert diagrams (if a vector version could not be generated) and screenshots
  - **PDF**: We have mostly seen PDF for documents but images can also have this extension
  - **EPS**: EPS images can be included using the epstopdf package (Simply install the package, there is no need to use \usepackage{} in our document)

Vector format or Bit-map format?

Images can have either vector format or bit-map format. We don't have to worry much about it, but in case, we do know the format the image is in, we can use this information to choose an appropriate image format to include in our LaTeX document. If we have an image in the vector format, we should go for PDF or EPS extension and if we have the image in bit-map format, we should go for JPG or PNG. This is because storing bit-map pictures in PDF or EPS takes a lot of disk space.

All the best!

Back to Top



Try Now
Skip Ad