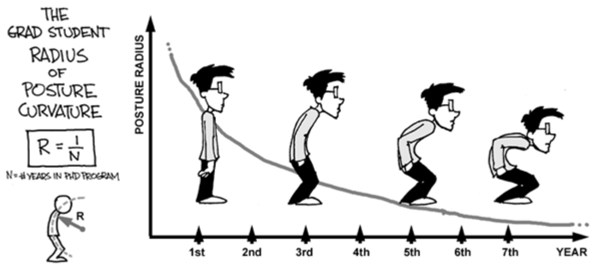
Introduction to LaTeX

adding images, tables

Contents

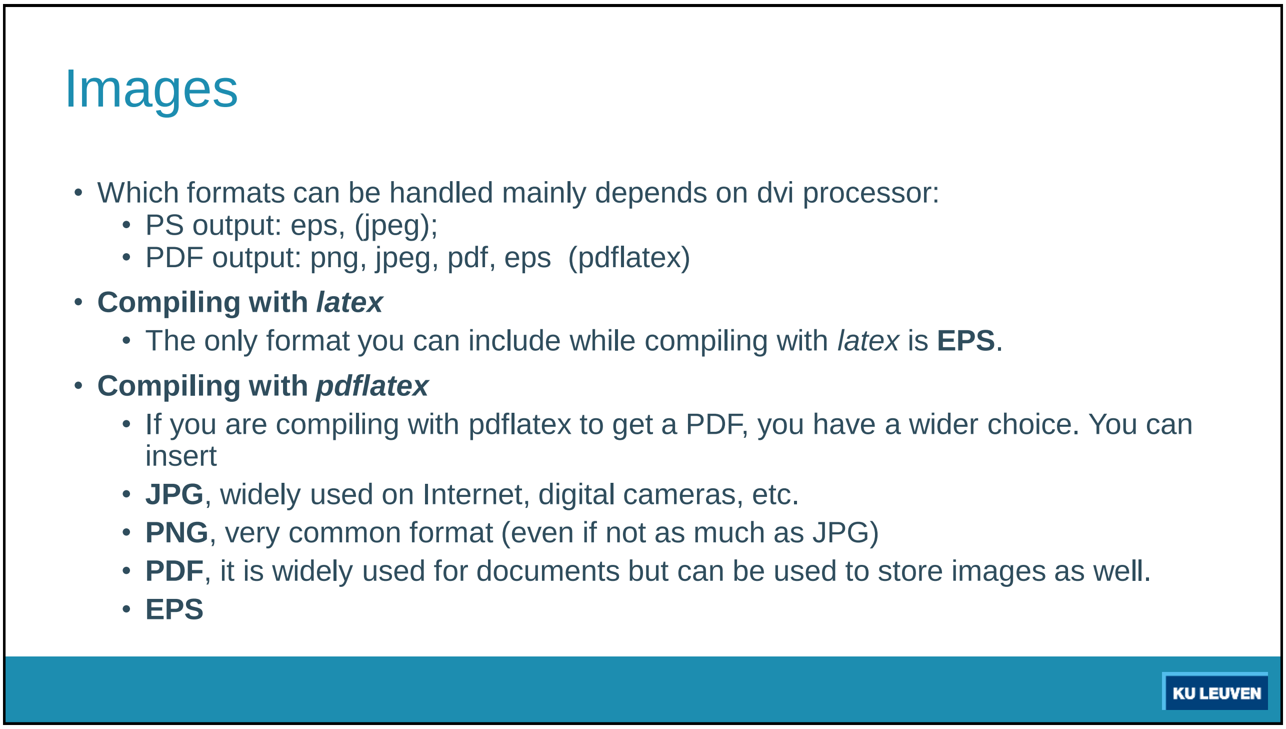
- Images
- Tabular
- Floats


THE GRAD STUDENT RADIUS OF POSTURE CURVATURE
$R = \frac{1}{N}$
N = # YEARS IN PHD PROGRAM
R
POSTURE RADIUS
1st
2nd
3rd
4th
5th
6th
7th
YEAR

Inserting images

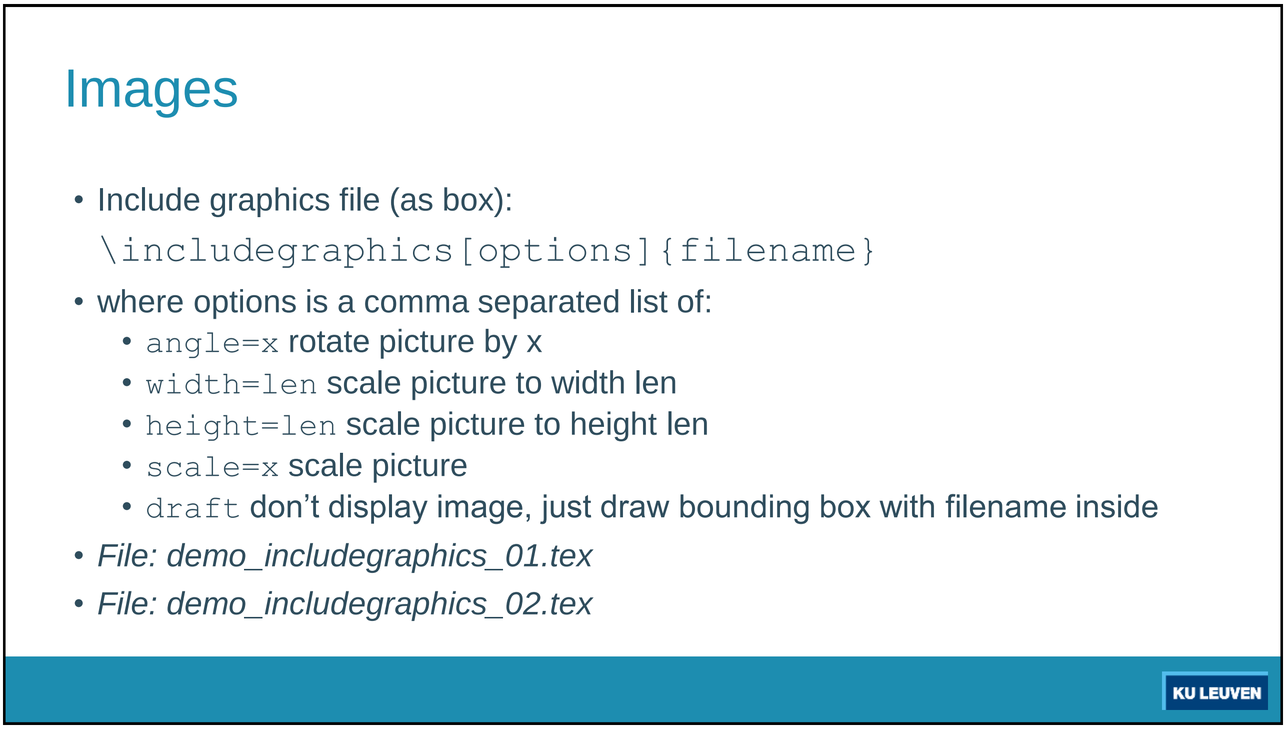
Images

- Ensure that the images are high quality.
- Additional package needed: `graphicx` package provides commands to include images
- Images behave quite similar to characters, just in larger boxes
- https://en.wikibooks.org/wiki/LaTeX/Importing_Graphics
- https://www.overleaf.com/learn/latex/Inserting_Images



Images

- Which formats can be handled mainly depends on dvi processor:
  - PS output: eps, (jpeg);
  - PDF output: png, jpeg, pdf, eps (pdflatex)
- **Compiling with *latex***
  - The only format you can include while compiling with *latex* is **EPS**.
- **Compiling with *pdflatex***
  - If you are compiling with pdflatex to get a PDF, you have a wider choice. You can insert
  - **JPG**, widely used on Internet, digital cameras, etc.
  - **PNG**, very common format (even if not as much as JPG)
  - **PDF**, it is widely used for documents but can be used to store images as well.
  - **EPS**



Images

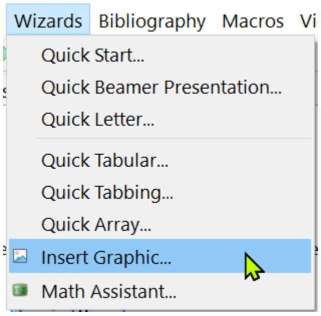
- Include graphics file (as box):

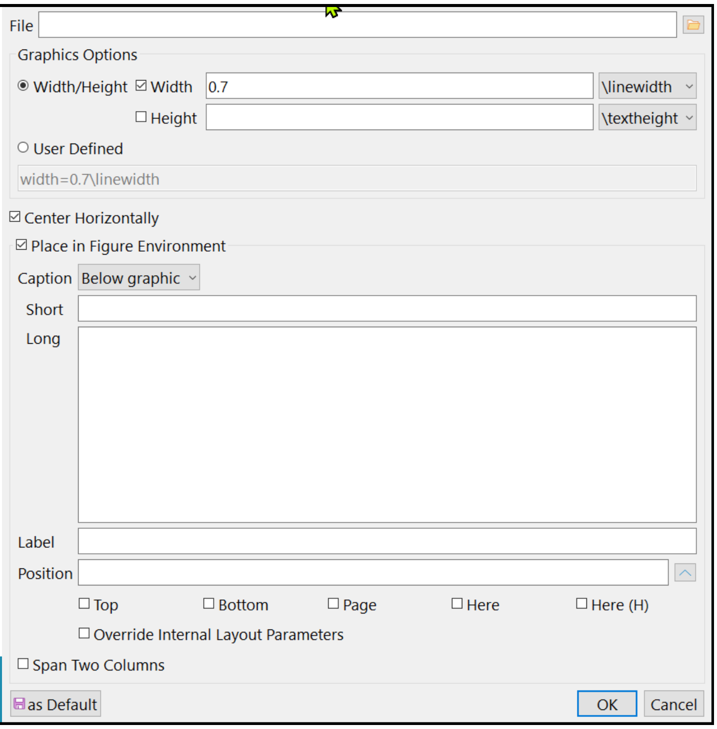
```
\includegraphics[options]{filename}
```

- where options is a comma separated list of:
  - `angle=x` rotate picture by x
  - `width=len` scale picture to width len
  - `height=len` scale picture to height len
  - `scale=x` scale picture
  - `draft` don’t display image, just draw bounding box with filename inside
- *File: demo_includegraphics_01.tex*
- *File: demo_includegraphics_02.tex*

Images

- TeXstudio wizard




File

Graphics Options

● Width/Height ☑ Width 0.7 \linewidth

☐ Height \textheight

○ User Defined

width=0.7\linewidth

☑ Center Horizontally

☑ Place in Figure Environment

Caption Below graphic

Short

Long

Label

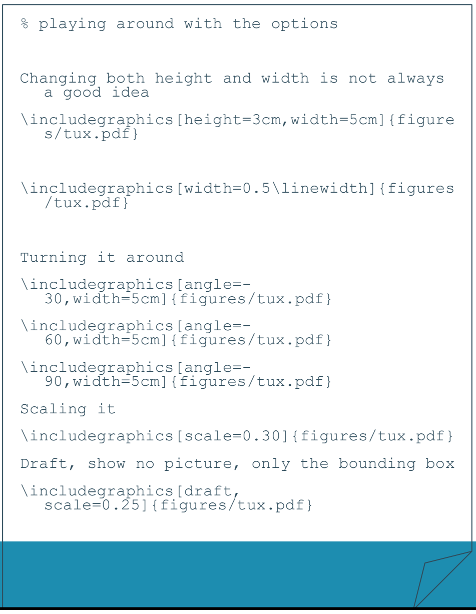
Position

☐ Top ☐ Bottom ☐ Page ☐ Here ☐ Here (H)

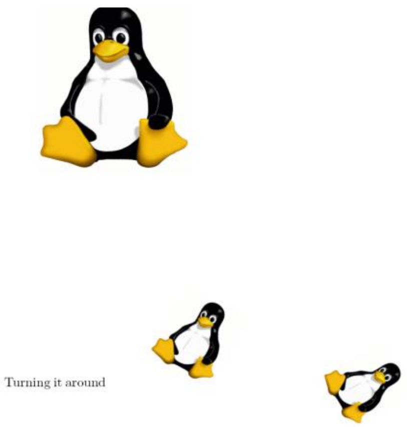
☐ Override Internal Layout Parameters

☐ Span Two Columns

Images

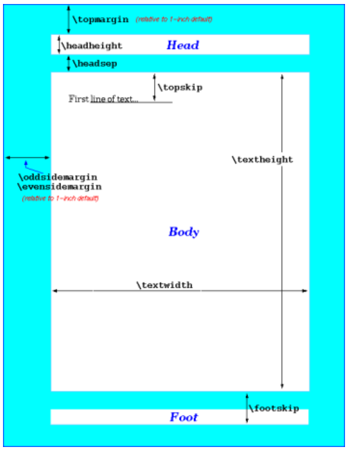


```
% playing around with the options


Changing both height and width is not always
   a good idea

\includegraphics[height=3cm,width=5cm]{figure
   s/tux.pdf}


\includegraphics[width=0.5\linewidth]{figures
   /tux.pdf}


Turning it around

\includegraphics[angle=-
   30,width=5cm]{figures/tux.pdf}

\includegraphics[angle=-
   60,width=5cm]{figures/tux.pdf}

\includegraphics[angle=-
   90,width=5cm]{figures/tux.pdf}

Scaling it

\includegraphics[scale=0.30]{figures/tux.pdf}

Draft, show no picture, only the bounding box

\includegraphics[draft,
   scale=0.25]{figures/tux.pdf}
```



Images

- the most obvious thing to set is the `width` or the `height` of an image.
- Tip: use dimensions relative to the `\textwidth` or `\linewidth` and `\textheight`.
- `\textwidth` is the width of the text block on the physical page
- `\linewidth` is the current width, which might locally be different (ex. documentclass option `twocolumn`).



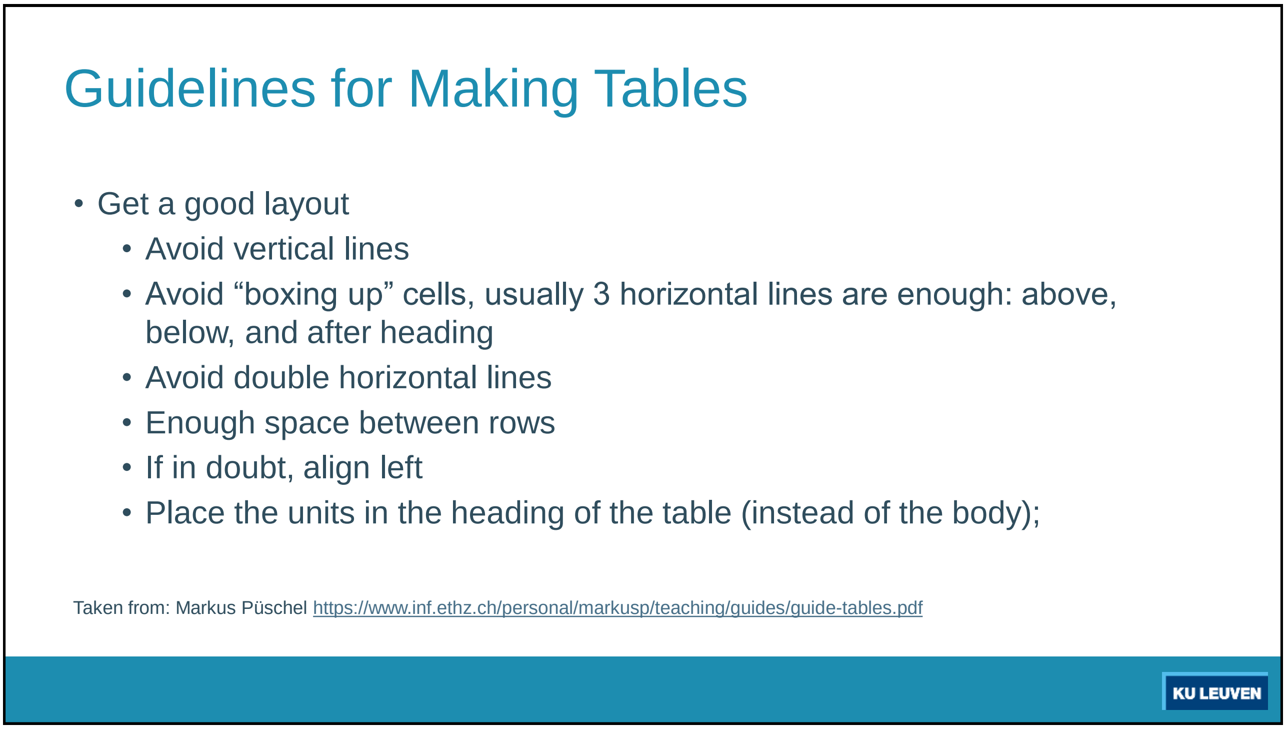
Location of images

- Tell LaTeX where to look for images
- Images can be stored centrally for use in many different documents.
- `\graphicspath` : provide an additional directory path to search for images (relative path)
- `\graphicspath{{/var/lib/images/}} \graphicspath{{./images/}}`
  `\graphicspath{{images_folder/}{other_folder/}{third_folder/}}`
- Or specify absolute path
  `\includegraphics{D:/Cursus_Final/Latex/examples/Basics-3/figures/atomium.jpg}`
- *File: demo_includegraphics_03.tex*

Hands-on

- Use the file demo_includegraphics_01 and change the options, check the result.
- Use google images, to search for some pictures to enclose.

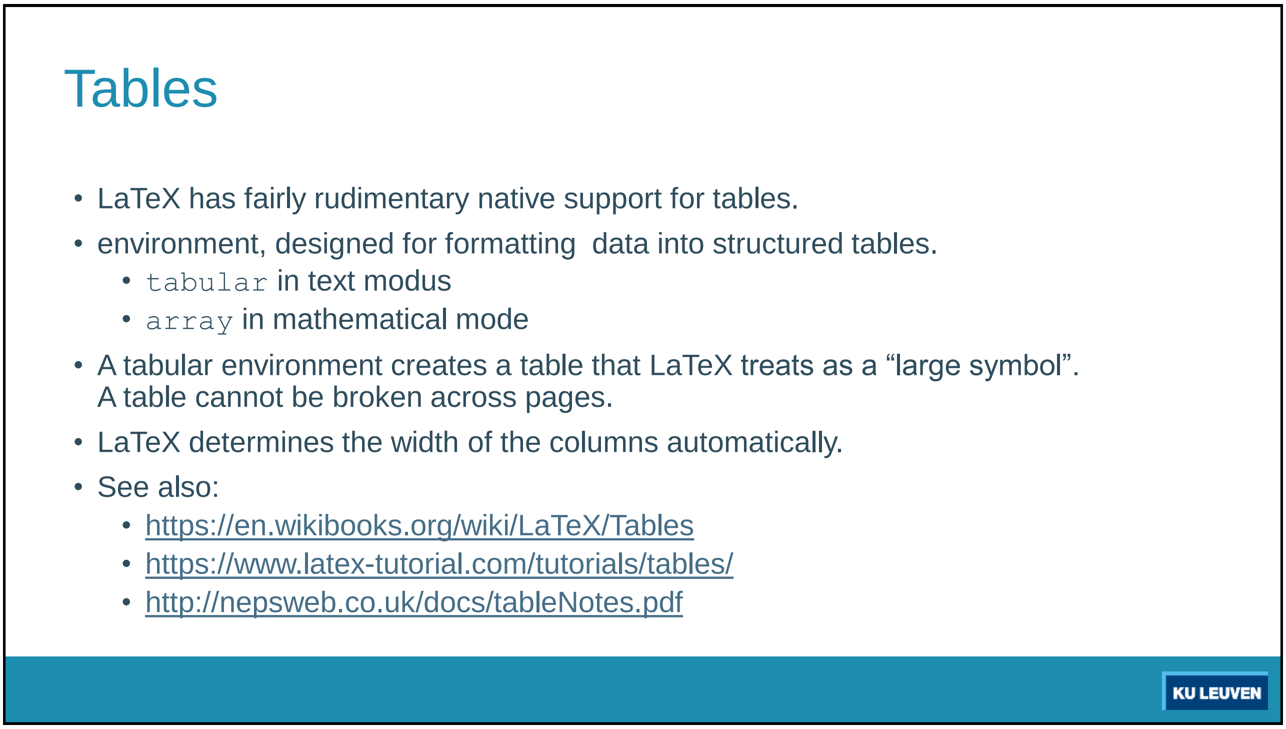
Creating tables



Guidelines for Making Tables

- Get a good layout
  - Avoid vertical lines
  - Avoid “boxing up” cells, usually 3 horizontal lines are enough: above, below, and after heading
  - Avoid double horizontal lines
  - Enough space between rows
  - If in doubt, align left
  - Place the units in the heading of the table (instead of the body);

Taken from: Markus Püschel https://www.inf.ethz.ch/personal/markusp/teaching/guides/guide-tables.pdf



Tables

- LaTeX has fairly rudimentary native support for tables.
- environment, designed for formatting data into structured tables.
  - `tabular` in text modus
  - `array` in mathematical mode
- A tabular environment creates a table that LaTeX treats as a “large symbol”. A table cannot be broken across pages.
- LaTeX determines the width of the columns automatically.
- See also:
  - https://en.wikibooks.org/wiki/LaTeX/Tables
  - https://www.latex-tutorial.com/tutorials/tables/
  - http://nepsweb.co.uk/docs/tableNotes.pdf

Tables

- (standard) Tables are made in LaTeX using the `tabular` environment
- Syntax

```
\begin{tabular}[pos]{cols}
  column 1 entry  &column 2 entry  ...  &column n entry \\
  ...
\end{tabular}
```

- `pos`: Optional. Specifies the table's vertical position. `t` aligns the table so its top row matches the baseline of the surrounding text, and `b` aligns on the bottom row.
- `cols`: Required. Specifies the formatting of columns. It consists of a sequence of specifiers, corresponding to the types of column and intercolumn material. Commonly used: `l, r, c, |`

tabular

- A row of a tabular is separated into columns by & (alignment character)
- A row end is indicated by \\
- `\hline` horizontal line
- Rows may contain less, but not more columns than specified by tabular argument
- Width of a column is determined by the width of the largest cell

`l`	left-justified column
`c`	centered column
`r`	right-justified column
`p{'width'}`	paragraph column with text vertically aligned at the top
`m{'width'}`	paragraph column with text vertically aligned in the middle (requires array package)
`b{'width'}`	paragraph column with text vertically aligned at the bottom (requires array package)
`|`	vertical line
`||`	double vertical line

tabular

- *File: demo_tabular_basic.tex*

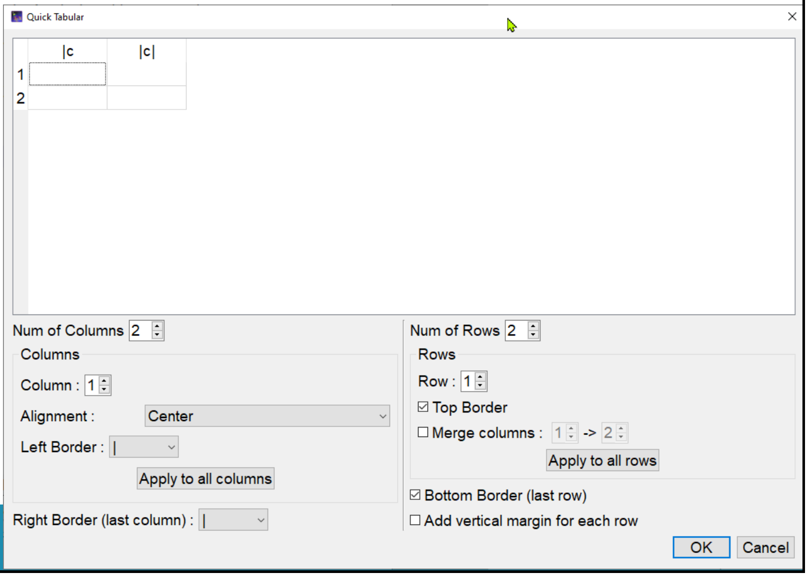
```
\begin{tabular}{l|r}
Track (100 m): & 25 sec \\
Swim (50 m): & 10 min \\
Bike (1 km): & 5 min \\
\end{tabular}
```

Track (100 m):	25 sec
Swim (50 m):	10 min
Bike (1 km):	5 min

- *File: demo_table_02.tex*

Table wizard (TeXstudio)

- Get help from your editor



Quick Tabular

	|c	|c|
1		
2		

Num of Columns 2

Columns

Column : 1

Alignment : Center

Left Border : |

Right Border (last column) : |

Num of Rows 2

Rows

Row : 1

☑ Top Border

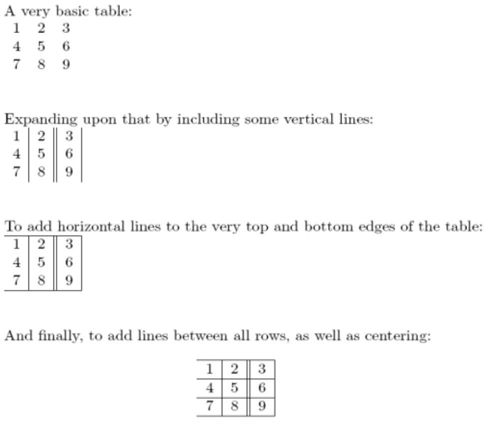
□ Merge columns : 1 -> 2

☑ Bottom Border (last row)

□ Add vertical margin for each row

Hands-on

- Write a file, building these simple tables. (*handson_tabular_01.tex*)



A very basic table:

1	2	3
4	5	6
7	8	9

Expanding upon that by including some vertical lines:

1	2	3
4	5	6
7	8	9

To add horizontal lines to the very top and bottom edges of the table:

1	2	3
4	5	6
7	8	9

And finally, to add lines between all rows, as well as centering:

1	2	3
4	5	6
7	8	9

Combining rows and columns

- Columns can be combined in a bigger cell: `\multicolumn{cols}{pos}{text}`
  - Combines the next cols to single column with alignment pos and contents text
  - Must be at the beginning of a row or directly after &
- To combine rows: package multirow
- *File: demo_tabular_multi.tex*

Horizontal, vertical lines

- Vertical lines are marked by | in column specification
- Horizontal lines are inserted with `\hline`
- A horizontal line from column x to y:

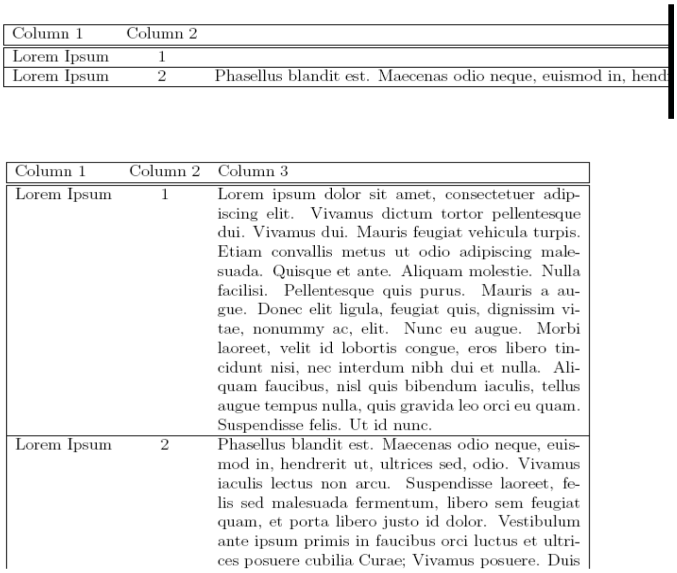
  `\cline{x-y}`
- A vertical line, over the height of a cell

  `\vline`
- *File: demo_tabular_more.tex*

Limiting width tabular

- defines a paragraph column with the specified *width*
  - `p{width}` (top alignment)
  - `m{width}` (center alignment)
  - `b{width}` (bottom alignment)
- *File: demo_tabular_limit.tex*

```
\begin{tabular}{|l c
   p{90mm} |}
\hline
Column 1 & Column 2 &
   Column 3\\
\hline\hline
Lorem Ipsum & 1 &  Lorem
   ipsum dolor sit amet,
   consectetuer adipiscing
   elit. Vivamus dictum
   tortor pellentesque dui.
   Vivamus dui. Mauris
   feugiat vehicula turpis.
   Etiam convallis metus ut
   odio adipiscing
   malesuada. Quisque et
   ante. Aliquam molestie.
```



Column 1	Column 2	
Lorem Ipsum	1	
Lorem Ipsum	2	Phasellus blandit est. Maecenas odio neque, euismod in, hend

Column 1	Column 2	Column 3
Lorem Ipsum	1	Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Vivamus dictum tortor pellentesque dui. Vivamus dui. Mauris feugiat vehicula turpis. Etiam convallis metus ut odio adipiscing malesuada. Quisque et ante. Aliquam molestie. Nulla facilisi. Pellentesque quis purus. Mauris a augue. Donec elit ligula, feugiat quis, dignissim vitae, nonummy ac, elit. Nunc eu augue. Morbi laoreet, velit id lobortis congue, eros libero tincidunt nisi, nec interdum nibh dui et nulla. Aliquam faucibus, nisl quis bibendum iaculis, tellus augue tempus nulla, quis gravida leo orci eu quam. Suspendisse felis. Ut id nunc.
Lorem Ipsum	2	Phasellus blandit est. Maecenas odio neque, euismod in, hendrerit ut, ultrices sed, odio. Vivamus iaculis lectus non arcu. Suspendisse laoreet, felis sed malesuada fermentum, libero sem feugiat quam, et porta libero justo id dolor. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Vivamus posuere. Duis

Package tabularx

- Package `tabularx`
- Introducing the column type X that works like a p-column with automatically calculated width.
- It is possible to use more than one X-column. By default, all of them are typeset to the same width, but it is possible to manually adjust how the available space is divided. Check the documentation.
- *File:demo_tabular_01*

Package booktabs

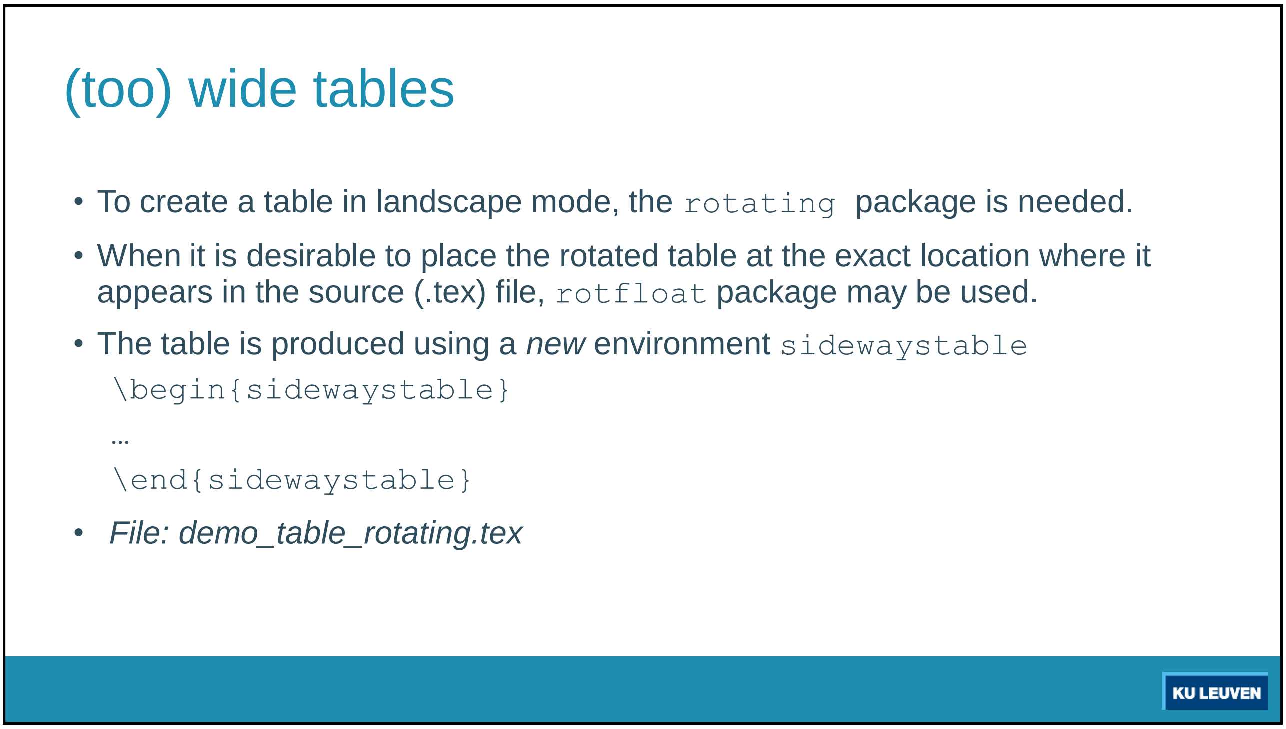
- Package `booktabs`
- Professional tables often appear as a three-line table, i.e., top line, middle line and bottom line. The three lines are usually thicker than other lines that appear in the table.
- Booktabs package provides: `\toprule, \bottomrule, \midrule`
- `\cmidrule:` a line which only spans a few columns
- *File: demo_booktabs_01*
- Web table generators allow export to booktabs
  - https://www.tablesgenerator.com/
  - https://www.latex-tables.com/

KU LEUVEN

(too) Long tables

- package `longtable:` use when a table is longer than one page
- The syntax for `longtable` is similar to `tabular`:
- Tables that can be broken by the standard LaTeX page-breaking algorithm.
- There are four elements to set which define the headings and captions:
  - `\endfirsthead`: Line(s) to appear as head of the table on the first page
  - `\endhead`: Line(s) to appear at top of every page (except first)
  - `\endfoot`: Last line(s) to appear at the bottom of every page (except last)
  - `\endlastfoot`: Last line(s) to appear at the end of the table
- *File: demo_tabular_longtable.tex*

KU LEUVEN

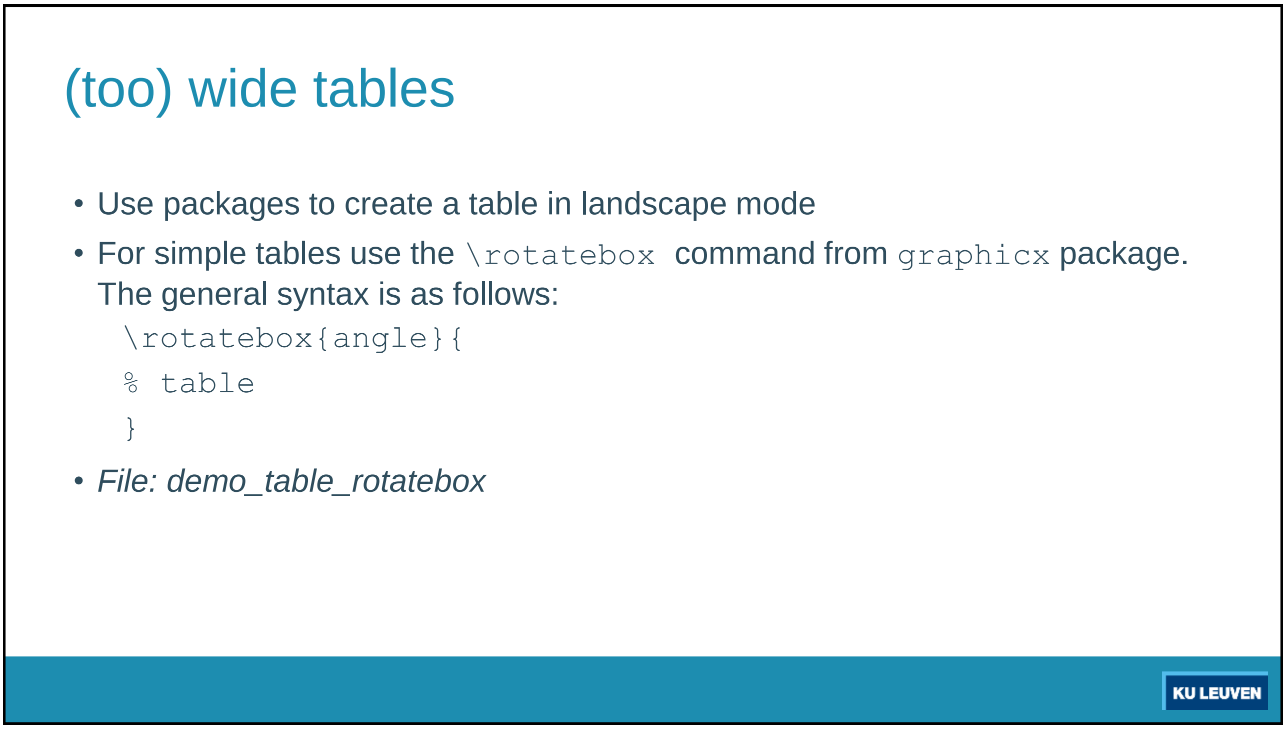


(too) wide tables

- To create a table in landscape mode, the `rotating` package is needed.
- When it is desirable to place the rotated table at the exact location where it appears in the source (.tex) file, `rotfloat` package may be used.
- The table is produced using a *new* environment `sidewaystable`

```
\begin{sidewaystable}
…
\end{sidewaystable}
```

- *File: demo_table_rotating.tex*



(too) wide tables

- Use packages to create a table in landscape mode
- For simple tables use the `\rotatebox` command from `graphicx` package. The general syntax is as follows:

```
\rotatebox{angle}{
% table
}
```

- *File: demo_table_rotatebox*

Color a table

- All elements in a table can be customized to use a specific colour. This functionality is provided by xcolor: `\usepackage[table]{xcolor}`
- `\arrayrulecolor`: Color of the lines.
- `\cellcolor`: Background color of a cell. You can either enter the name directly inside the braces (red, gray, green, etc.) or pass a format parameter inside brackets
- `\rowcolor`: Background color of a row.
- Background color of a column. Tricky, but the easiest way is to define a new column type.
- *File: demo_table_color*

More with tables

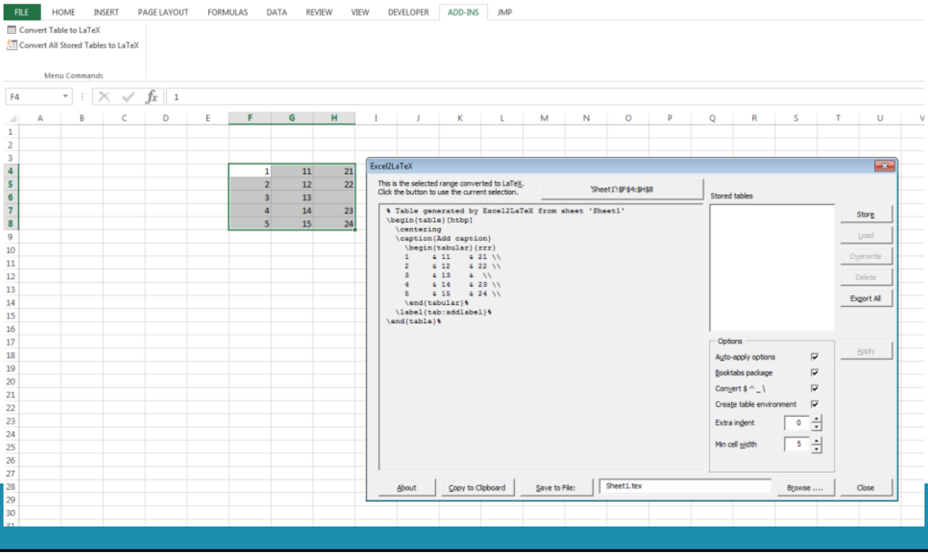
- Package `wrapfig`
  - Allows text to be wrapped around a table
  - *File: demo_wraptable.tex*

Tables from excel

- LaTeX code to generate tables can be cryptic at first.
  - use software to write this code
  - create tables in Excel and export them to LaTeX.
- Excel tables can be converted using:
  - excel2latex (https://www.ctan.org/tex-archive/support/excel2latex/)
    tested: works with Office 2013
- Note:
  - Some features will not be supported
  - Extra editing can be needed

excel2latex

- Office 2013



F4: 1

F	G	H
1	11	21
2	12	22
3	13	
4	14	23
5	15	24

Excel2LaTeX

This is the selected range converted to LaTeX.
Click the button to use the current selection.

'Sheet1'!$F$4:$H$8

```
% Table generated by Excel2LaTeX from sheet 'Sheet1'
\begin{table}[htbp]
  \centering
  \caption{Add caption}
    \begin{tabular}{rrr}
    1     & 11    & 21 \\
    2     & 12    & 22 \\
    3     & 13    &  \\
    4     & 14    & 23 \\
    5     & 15    & 24 \\
    \end{tabular}%
  \label{tab:addlabel}%
\end{table}%
```

Stored tables

Options
- Auto-apply options ☑
- Booktabs package ☑
- Convert $ ^ _ \ ☑
- Create table environment ☑
- Extra indent: 0
- Min cell width: 5

Save to File: Sheet1.tex

Webtool

- Create the table in your browser, copy the code into your LaTeX document
- https://tablesgenerator.com/
- https://www.latex-tables.com/
- https://truben.no/table/
- http://ericwood.org/excel2latex/

Floats

Floats

- LaTeX breaks paragraphs and sentences across pages to avoid partially filled pages.
- Problem: table or image is too large to be placed on the page
  - Pictures and tables, cannot be split;
  - *floated* to convenient places, such as the top of the following page
  - Get a minimum on whitespace
- Floats: Objects, depending on the space available, will be placed where they are invoked or further on in the text.
- Best practice: do not place floats manually, let LaTeX do the job.
- https://en.wikibooks.org/wiki/LaTeX/Floats,_Figures_and_Captions

Floats

- Works for images and tables: `table` and `figure` are two of the environments provided by LaTeX
- Common way of including images, tables:
  - Use the correct environment (`table` or `figure`).
  - Center the content, if necessary.
  - Include the content.
  - Add a caption (caption must proceed label)
  - Add a label for cross-referencing.
- floats may become a major source of frustration, when LaTeX does not put them where you want them to be.
- *File: demo_float_01.tex*

Floats

```
\begin{figure}[placement specifier]
  \centering
  \includegraphics...
  \caption{⟨legend⟩}
  \label{⟨identifier⟩}
\end{figure}
```

```
\begin{table}[placement specifier]
  \centering
  \begin{tabular}...
  \end{tabular}
  \caption{⟨legend⟩}
  \label{⟨identifier⟩}
\end{table}
```

Floats

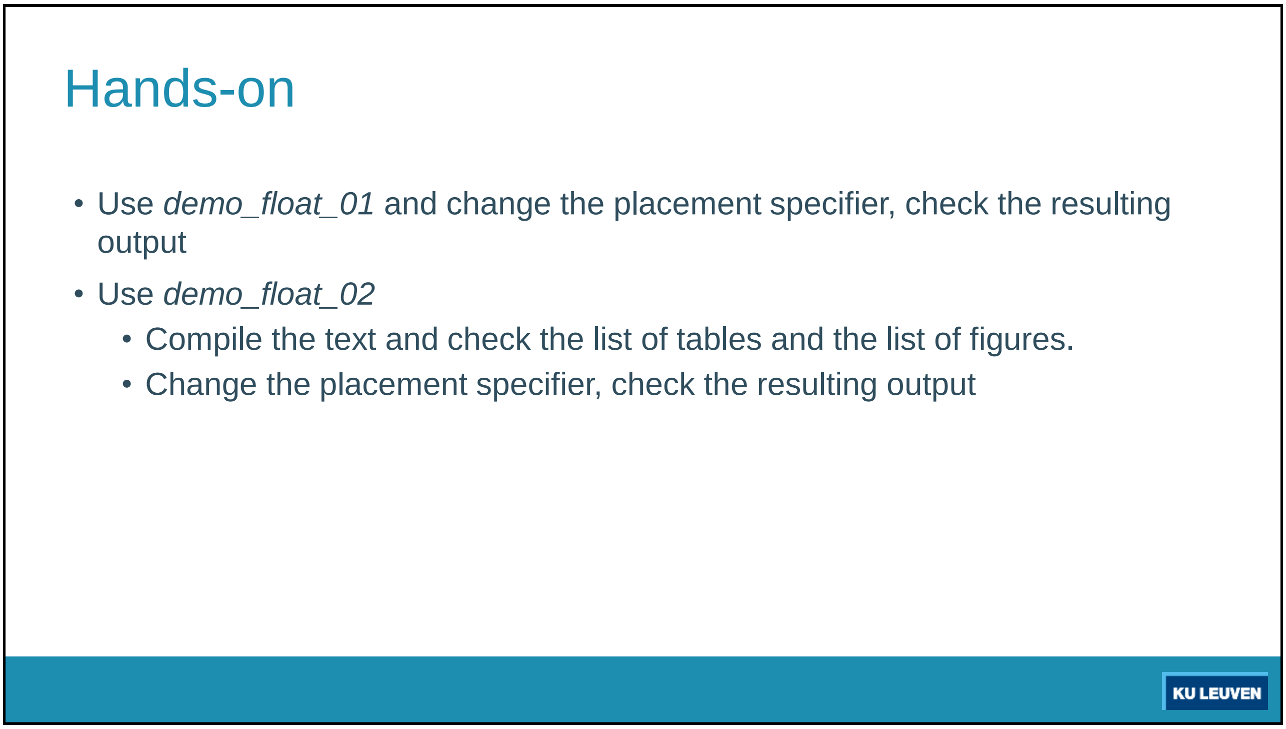
- Any material enclosed in a figure or table environment will be treated as floating matter.
- *placement specifier*. parameter used to indicate the locations to which the float is allowed to be moved.
  - **h (*Here)***: at the very place in the text where it occurred.This is useful mainly for small floats.
  - **t (*Top)***: at the *top* of a page.
  - **b(*Bottom)***: at the *bottom* of a page
  - **p(*Page of floats)***: on a special *page* containing only floats.
  - **! *Force!*** (does not work with p)

Rules

- Floating objects will not appear prior to the page where they are referred on
- If floating objects can not be placed, they will appear at the end of the document.
- `\clearpage` can force the pending objects to be placed
- package `placeins`. This provides the command `\FloatBarrier` which causes all unprocessed floats to be processed at that point, but does not start a new page unless it is necessary.
  To keep floats in the sections in which they were included, use:
  `\usepackage[section]{placeins}`

caption

- It is always good practice to add a caption to any figure or table.
  `\caption[`*`shortform`*`]{text}`
- Place caption on top or at the bottom
- Needs to be in a table or figure environment
- Automatic numbering:
  - Table nr:
  - Figure nr:
- The shortform will be used in the list of figures, list of tables.
- *File: demo_caption_01*



Hands-on

- Use *demo_float_01* and change the placement specifier, check the resulting output
- Use *demo_float_02*
  - Compile the text and check the list of tables and the list of figures.
  - Change the placement specifier, check the resulting output