

# Customization and tables

Math 351

Almost everything in  $\text{\LaTeX}$  can be customized, including  $\text{\LaTeX}$  commands themselves. The standard fonts, margin sizes, and so on, were carefully designed to make technical documents readable, so changing the out-of-the-box behavior is not always wise. Nonetheless, there are times when it may be appropriate to customize.

## 1 Font size and face

To change to a 11 point font, use the [11pt] option in the initial `\documentclass` command in this way:

```
\documentclass[11pt]{article}
```

This document is typed in 11pt font. The font size can also be 10pt (the default option) or 12pt. Another option in the `\documentclass` command is `twocolumn`, which has the effect of producing two columns per page.

The font face can be changed by including certain packages in the preamble of the document. For example, to change both the text and mathematics to the “Times” font, include the command `\usepackage{mathptmx}` before the `\begin{document}`. There are more cutting edge ways to change the font using XeLaTeX (which we will see later in the course), but for now these packages are still used by many.

Most versions of  $\text{\LaTeX}$  have a number of pre-installed fonts, some of which can be selected by calling these packages:

Package	Roman	Math	Sans serif	Typewriter
(none)	CM Roman	CM Roman	CM Sans	CM Typewriter
mathpazo	Palatino	Palatino		
mathptmx	Times	Times		
helvet			Helvetica	
avant			Avant Garde	
courier				Courier
chancery	Chancery			
bookman	Bookman		Avant Garde	Courier
newcent	New Century		Avant Garde	Courier
charter	Charter			
fourier	Utopia	Fourier		
eulervm		Euler		

An empty entry indicates that a package does not have an effect on a given font face. The last two font selections are listed separately because they are not usually found in the basic version of  $\text{\LaTeX}$ , but are given by the “texlive-full” version. Later in the course we will show how to include arbitrary font faces.

To select the roman font, math font, sans serif font, and typewriter font separately, include consecutive `\usepackages` in a correct order. For example, the commands

```
\usepackage[sc]{mathpazo}
\usepackage{charter}
\usepackage[scaled]{helvet}
```

uses the Palatino math font, Charter roman font, Helvetica sans serif font (invoked by `\textsf{..}`), and Computer Modern Typewriter font (as seen in verbatim statements). These are the choices made for this document. The `[sc]` option on `mathpazo` gives a better small caps font, and the `[scaled]` option on `helvet` gives an adjusted version of the Helvetica font.

To set the default font face for the entire document to the sans serif font, say, then place `\renewcommand{\familydefault}{\sfdefault}` in the preamble. If using this command, then roman font can be invoked using `\textrm{..}`.

## 2 Margins and spacing

The margins of the document can be controlled with the `geometry` package. To set the left, top, right, and bottom margins to specific values, place a command such as

```
\usepackage[left=35mm,top=2cm,right=35mm,bottom=2cm]{geometry}
```

in the preamble. The margins used in this document are those values shown above. As another example,

```
\usepackage[landscape, margin=2in]{geometry}
```

changes all margins to be 2 inches and prints in landscape mode.

It usually bad form to manually adjust the vertical or horizontal spacing inside the body of the document when writing an article or book, but it might be appropriate to manually force white space when creating documents such as syllabi, exams, or resumes.

The line break command `\\` has an option to add extra space; to add an extra 1cm, use `\\[1cm]`. Alternatively, to force an extra vertical space of 1cm between two paragraphs, the `\vspace{1cm}` command can be used. Sometimes  $\TeX$  will think adding a vertical space using `\vspace` is a bad idea and won't cooperate; such a vertical space can be demanded with the command `\vspace*{1cm}`.

The `\vfill` command produces a length which can stretch or shrink vertically, pushing the text after the `\vfill` command as far down the page as possible. This command can be used in tandem with the `\newpage` command, which forces a new page to begin. For instance, if writing a mathematics exam, the  $\TeX$  commands

```
\begin{problem} Evaluate  $\int \ln x \, dx$ . \end{problem}
\vfill
\begin{problem} Evaluate  $\int \sin x \, dx$ . \end{problem}
\vfill
\newpage
```

will produce the next page in the document (provided the `amsthm` package is loaded and `\theoremstyle{definition}` and `\newtheorem{problem}{}{}` both appear in the preamble).

1. Evaluate  $\int \ln x \, dx$ .

2. Evaluate  $\int \sin x \, dx$ .

Analogous to the `\vspace`, `\vspace*`, and `\vfill` commands are the `\hspace`, `\hspace*`, and `\hfill` commands, which produce horizontal space, a forced horizontal space, and a rubber horizontal fill.

### 3 Newcommands and style files

L<sup>A</sup>T<sub>E</sub>X commands themselves can be customized. To create a new command, place

$$\backslash\text{newcommand}\{\backslash\text{name}\}\{\text{definition}\}$$

in the preamble. Then, to call the command, use `\name` in the document. The compiler will complain if `\name` is a predefined `TEX` command. The `\newcommand` command has an optional parameter to include input. For instance,

$$\int_{\mathbb{R}} \#1 \, d\#2$$

defines the command `\integral` which takes in 2 inputs, placed where the #1 and #2 appear in the definition. If this command was in the preamble, then calling the command `\integral{\sin x}{x}` would produce  $\int_{\mathbb{R}} \sin x \, dx$ . There can be up to 9 inputs.

For function names in math mode which are not predefined (such as  $\ln x$ ,  $\sin x$ ,  $\arctan x$ ), use the command such as `\DeclareMathOperator{\dimension}{dim}` to define a command `\dimension`. This produces “dim”, a math mode symbol.

To redefine a previously defined L<sup>A</sup>T<sub>E</sub>X command, use the syntax

```
\renewcommand{\old}{\new}
```

For instance, `\renewcommand{\phi}{\varphi}` changes the appearance of  $\varphi$  throughout the document. As another example, this command can be used to change the end-of-proof symbol in the `amsthm` proof environment into creating a black square pushed to the right of the line by including `\renewcommand{\qed}{\hfill \(\ \blacksquare \)}` in the preamble.

Analogous to `\newcommand`, there is a `\newenvironment` command to create custom environments. The syntax to be placed in the preamble is

$$\backslash\text{newenvironment}\{\text{name}\}\{\text{before}\}\{\text{after}\}$$

To call the command, use `\begin{name} .. \end{name}`. Then, commands in `before` are run before `..` and commands in `after` are run after `...`. Just like `\newcommand`, there is an option for up to 9 input variables. For example, the command

```
\newenvironment{tinytext}{tiny text:\begin{tiny}}{\end{tiny}!}
```

can be used to create tiny text: This text really is quite small!

After loading packages, choosing the font, setting margins, and defining new commands and new environments, the preamble can become lengthy. Especially when reusing the same preamble for multiple documents, it may be convenient to store the preamble in a file with the `.sty` extension (the `.sty` stands for “style file”). In the first line of the style file, place the command

`\ProvidesPackage{351package}`

and then call the package just like any other package with the `\usepackage` command in the preamble of the main document.

## 4 Tables

$\text{\LaTeX}$  is not the software of choice to make complicated, detailed spreadsheets, but for simple and short tables there is the tabular environment. The syntax to create a table is

```
\begin{tabular}{column specification}  
.. & .. & .. \\  
.. & .. & .. \\  
\end{tabular}
```

where `column specification` is a string containing these characters:

- `l` for a column of left aligned text,
- `c` for a column of centered text,
- `r` for a column of right aligned text,
- `|` to create a vertical bar, or
- `p{width}` for a column with wraparound text of length `width`.

The syntax for moving to the next column and starting a new line is the same syntax as when using the `\begin{matrix}..\end{matrix}` command in math mode.

To create a horizontal line in the tabular environment, use the command `\hline`. We end with one more example of a tabular command, giving the addition table for the additive group  $\mathbb{Z}_4$ :

$\mathbb{Z}_4$	0	1	2	3
0	0	1	2	3
1	1	2	3	0
2	2	3	0	1
3	3	0	1	2

There are many more examples of tables of varying complexity on pages 45–48 of the text. Later, when we discuss packages, we will introduce the wonderful `booktabs` environment for better looking tables.