# Getting Started

## Introduction

- LaTeX is not a WYSIWYG system. That means, it is **not** "what you see is what you get". Microsoft word *is* a WYSIWYG system.
- With LaTeX , you begin by writing a text file (in WinEdt). Other than spaces, you will only ever use the following characters:

```
a b c d e f g h i j k l m n o p q r s t u v w x y z

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

0 1 2 3 4 5 6 7 8 9

! " # $ \% & ’ ‘ ( ) * + , - . / :

; < = > ? @ [ \ ] ^ _ { | } ~
```

- In this tutorial, we'll be writing an example of a LaTeX *input* or text file using (some of) these characters. The file name you will produce has two parts. The first part is the *base name* or *first name* (Diller 1999, p. 7), the second part is the *file extension*. The file you will create is called `practice.tex` with `practice` being the first name and `.tex` being the file extension.
- Now, open WinEdt, and type the following:

```
\documentclass[11pt]{article}
\title{My Day at the Department}
\author{Arthur Spirling}
\begin{document}
\maketitle

\noindent This is an abridged account of a day I spent
in the department just before Christmas. Early in the morning,
Tu\={g}ba G\"{u}ven\c{c} came by for a chat. She was my math
teacher, so I asked her a few questions: ``Tugba", I said, ``how
do I calculate $\frac{\partial{\pi}}{\partial x}$ for this week's
homework?". She was unimpressed---now seemingly in a rush---and
told to me to check Simon \& Blume (pages 345--352) for an answer.

Later in the day, Prof.\ Duggan dropped by to ask me about my
earlier enquiries.  He said that, in future, I should ask
\textit{him} directly about partial derivatives since then he
could gauge the average level of the class's knowledge and how
`effective' his teaching had been.  ``Right-ho, chief!", I chimed.

\end{document}
```

- note that some words are coming up in red. This is because WinEdt doesn't think they are spelt correctly. To check spelling, right click on them
- note that the symbol you need to produce ` is on the top left of your keyboard, just below the Escape key, and to the left of the `1` key. It shares a key with the `~` (called a tilde).

### I've finished writing, what next?

- Now you need to run LaTeX on the file. To do this, look on the tool bar at the top of the screen for a 'button' with the word LaTeX on it. It is below the word "Accessories" and between a picture of a bear's head, and the word TeX. Click it once.
- Now WinEdt will show you a 'save as' box. It cannot run LaTeX until it knows what to call your file (running LaTeX actually saves a copy of your file in its current form). Call the document `practice`. Note that the "save as type" already says TeX: don't change that bit! Click 'Save'.
- Now, a black box with writing will appear and some lines of words and symbols in white will scroll. When it says `Press any key to continue...`, press any key.
- Now we want to see how what we have written looks. We press the button with the letters `DVI` and a picture of a magnifying glass on it (by the LaTeX button). Either of the buttons (with or without a green arrow) is fine. Depending on which you choose, you may need to press the button again.
- It looks nice, doesn't it? Minimize the `DVI` window, and you should be looking at your `.tex` file again. Alternatively, you can set the size of your WinEdt window so it is on the left of your screen, whilst your DVI window can be sized so that it is on the right (personally, I find this leaves me a bit cramped).

## Understanding LaTeX Commands

- LaTeX has ten special characters: `# $ % & \ ^ _ { } ~`
- the character `%` tells LaTeX that everything after it on that line is a comment. Everything between the character `%` and the next end-of-line character will not appear on the final document produced by LaTeX .
- the characters `{` and `}` are for grouping. They must always be matched. If you put the cursor on one in WinEdt, it will turn yellow if the left brace is matching something on the right, and green if the right brace

is matching something on the left. If the brace doesn't seem to match anything, WinEdt turns the cursor red.

- the backslash, \ begins every LaTeX command. Examples in the document you have produced today include `\noindent` which told LaTeX not to indent the line of the paragraph and `\textit` which told LaTeX to write the words in the braces in italics.
- note that LaTeX is CaSe SeNSiTIvE.
- LaTeX treats any number of spaces as a *single* space. In my WinEdt file, I put five spaces between 'of' and 'spaces' in the previous sentence, but it makes no difference to how it looks.
- we will go over the use of the other special characters in due course: what do you think `$` does?
- if you want to write—in text— the symbols #, %, $, { and }, you need to use a backslash before the symbol. So to get #, I wrote `\#`. This is also true of `&`: not because it is 'special' like the other characters, but because it tells LaTeX something about math if you write it without a backslash (we'll come to that later).

## Environments

- in our practice today, we used the `document` environment. The command `\begin{document}` was used to open the `document` environment and `\end{document}` was used to close it. Actually, the `document` environment is unusual because only one instance of it can occur in any one file (and it is necessary if you want to write anything).
- usually, several instances of an environment can be in a file, or can be nested within other environments. At the moment, I am writing in the `itemize` environment.

## Structure of an Input File

- Generally speaking, you cannot run LaTeX on a document unless you have provided an `\end{ }` command for every environment you have begun. So, when you want to write a document, and you want to check how it looks before you finish, make sure you have `\documentclass{article}`, followed by `\begin{document}`, followed by your text and math, followed by `\end{document}` somewhere down the page (nothing you write after `\end{document}` will be read by LaTeX ).

- go back and look at the start of the document you wrote in WinEdt. It looks like this:

```
\documentclass[11pt]{article}
\title{My Day at the Department}
\author{Arthur Spirling}
\begin{document}
\maketitle
\noindent
```

- the first command here is `\documentclass[11pt]{article}`. This means that LaTeX will use the `article` document class. There are other classes, like `report`, `book`, `letter` and `slides`, but you won't be learning them in this course.

- in the first command in the document we have written, the `11pt` option has been chosen; the default is 10pt and if you leave out the square brackets all together, that is what will be chosen.

- the part of the file that occurs between the `\documentclass` command and the actual start of your document is called the *preamble*. In the case of `practice.tex`, the preamble contains two 'declarations':

```
\title{My Day at the Department}
\author{Arthur Spirling}
```

  The first command tells LaTeX what the title of the article will be, and the second tells it that Arthur Spirling is the author. The actual title (and the date) are produced by typing the `\maketitle` command that occurs within the `document` environment.

- If you don't want a date in your article, use `\date{}` with nothing in the braces. If you recompile your LaTeX at a later date, LaTeX will *automatically* alter the date to the newer one.

- in fact, you don't have to use these preamble commands at all if you don't want to. For example, an alternative way to write the start of the document here is:

```
\documentclass[11pt]{article}
\begin{document}
\begin{center} \LARGE{My Day at the Department}\end{center}
\noindent
```

  which gives:

# My Day at the Department

### Modes

The LaTeX files you will process will be in one of two modes: paragraph or math. Paragraph mode is for ordinary text. To begin a new paragraph, leave a blank line in the input file. If you want a math formula in the text use dollar signs: `$x+y+5=\alpha$` gives you $x + y + 5 = \alpha$. If you want math to be displayed, use `\[` at the start, and `\]` at the end. So, `\[y-2x=z\]` will give you

$$y - 2x = z.$$

We'll be doing a lot more on math mode in the coming tutorials.

## It isn't working- it's broken and I hate it!

- yes it is, no it isn't, no you don't
- people make coding mistakes (I make a lot) and LaTeX sometimes can't understand you. Don't forget it is a language, and languages have rules!

### Common Errors

A common error is to **incorrectly input the name of a command**. For example, go back to your practice file and change `\textit{him}` to `\texit{him}`. If you do this, when you run LaTeX , it will give you a bunch of white writing followed by something like:

```
Document Class: article 2003/02/14 v1.3p Standard LaTeX document
class
(/UR/PSC/star/tex/inputs/latex2e/base/size11.clo))(practice.aux)

! Undefined control sequence

1.14 \texit
?
```

If you type `h` in response to this, LaTeX (may) be able to give you a bit more information about what went wrong. In this case it will tell you the control message is not `def'ed` by which it means 'defined'. It is telling you it doesn't know what that command means. What you do then is as follows:

- take note of the error and the line it occurs on (`1.14` means line 14 above)
- close the LaTeX window
- correct the error

- run LaTeX again
- repeat these steps till all errors are dealt with and the box gives you the `press any key to continue...` prompt

Another very common error is **to miss a closing brace**. So, if you write `\begin{document` without the final brace `}` , LaTeX won't be happy. The error message will have the words

```
runaway argument?
```

somewhere. But, again, it will tell you where the (that) error is on your input file. Make sure that you **close math modes** in the right way! That is, make sure your dollar signs and `\[` and `\]` match. If they don't, LaTeX will let you know.

**Missing a backslash** before special characters that you want it text will upset LaTeX too. For example, try typing `&` in your input file where you have previous written `\&`. Now type `h` in response to the error. Now correct the error as is required before LaTeX is willing to run again.

- try and be patient: in a short period of time, you will make fewer errors and know immediately how to correct those you do make.