

# HTML and LaTeX Workshop

---

MATT CHARNLEY

BECKY COULSON

# HTML Outline

---

Writing html code

Uploading html to the internet

Changing formatting: CSS

# How to Write HTML

---

Can be written in any plain text editor.

Notepad is my go-to, but a lot of other things will work as well.

Save documents as .html files.

```
participants - Notepad
File Edit Format View Help
<!DOCTYPE HTML >
<html>
<meta name="viewport" content="width=device-width, user-scalable=no">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>DIMACS REU 2014 Participants</title>
<link rel="stylesheet" type="text/css" href="../style.css">
<link rel="shortcut icon" href="http://dimacs.rutgers.edu/favicon.ico" type="image/x-icon">
<style type="text/css">
img {border-width:1px; border:solid; height:100px; }
img.nat { height:auto; border:none;}
td {text-align:center; border:1px solid black; padding:5px; }
table { border:2px solid black; }
br.small {line-height: 5px; }
div.container { max-width:1100px; }
</style>
<script type="text/javascript">
window.onload = function ()
{
var allimg = document.getElementsByTagName("img");
for (i=0; i<=allimg.length; i++)
{
allimg[i].onerror = function() { this.src = "pics/smiley.jpg" };
allimg[i].src = allimg[i].src ;
};
};
</script>
</head>
<body>
<div class="container">

```

# Starting a new HTML File

---

Grab code from somewhere else.

- Templates are available.
- Just download from an already running site.
- The internet.

Or, you can start one from scratch.

# Starting a file from scratch – What's Important?

---

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
...
```

```
</head>
```

```
<body>
```

```
...
```

```
</body>
```

```
</html>
```

Tell the computer this is html.

Starting and ending tags.

<x> starts an x object.

</x> ends it.

These need to be in order, like brackets in any type of programming.

In the ... is where you write your code.

# What goes in the head?

---

`<title> </title>` - Title of your webpage. This is what shows up on the border of the browser window when someone visits your page.

`<link />` - Used for importing CSS documents (later)

`<meta>` - Used when copying things from the internet.

# Important Tags to Know

---

All of these need the beginning and end thing.

- `<h1>`, `<h2>`, ..., `<h6>` - Headers. Large font that you can format later if needed.
- `<p>` - Paragraph. Standard text section.
- `<b>` - Bold
- `<i>` - Italics
- `<a>` - Hyperlink (more details later)
- `<div>` - Division. Chunking up the document into different blocks.
- `<ul>`, `<ol>` - Unordered and Ordered list respectively.

Single Tag expressions – do not need beginning and end.

- `<br>` - Line break.
- `<hr>` - Horizontal Line across the page.
- `<img />` - Image
- `<li>` - List Item, nested inside `<ul>` or `<ol>`

# Tables

---

<code>&lt;table&gt;</code>	Begin the Table
<code>&lt;tr&gt;</code>	Table row
<code>&lt;th&gt; &lt;/th&gt;</code>	Table head(er)
<code>&lt;td&gt; &lt;/td&gt;</code>	Table data
<code>&lt;/tr&gt;</code>	End row
...	
<code>&lt;/table&gt;</code>	

# Attributes

---

Specifying more things about your objects/tags.

- `<a href= " " >Text</a>` - Specify the destination of the hyperlink.
- `` - Specifies the source of the image file.
- `<table border = "1px" >` - 1 pixel wide border on each cell of the table.
- `<div class = " " >` - Naming pieces of your document for applying CSS formatting.

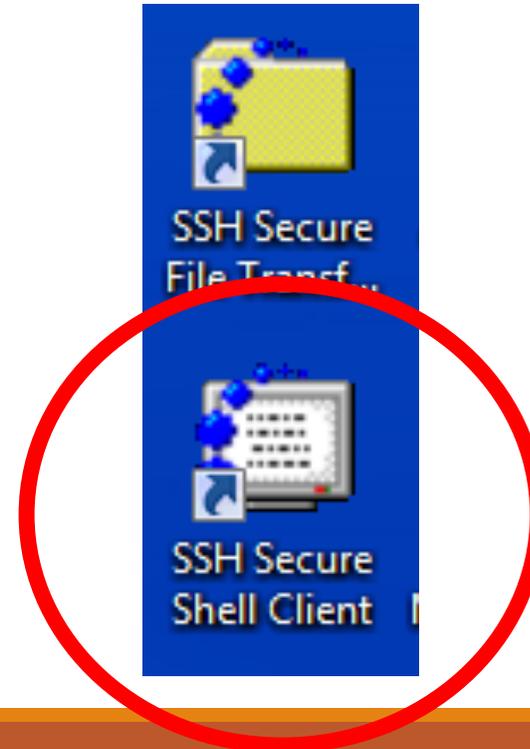
# Putting HTML on the Internet

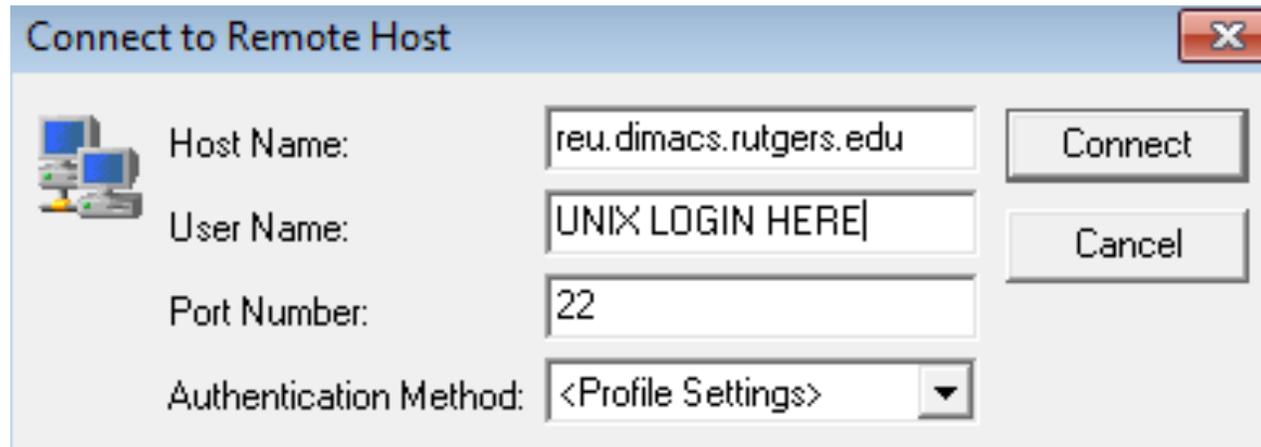
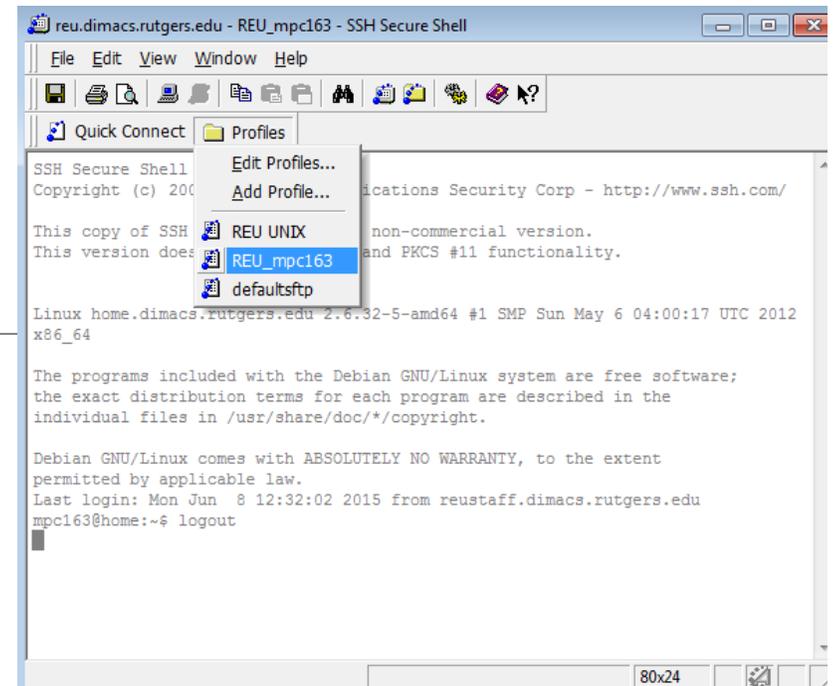
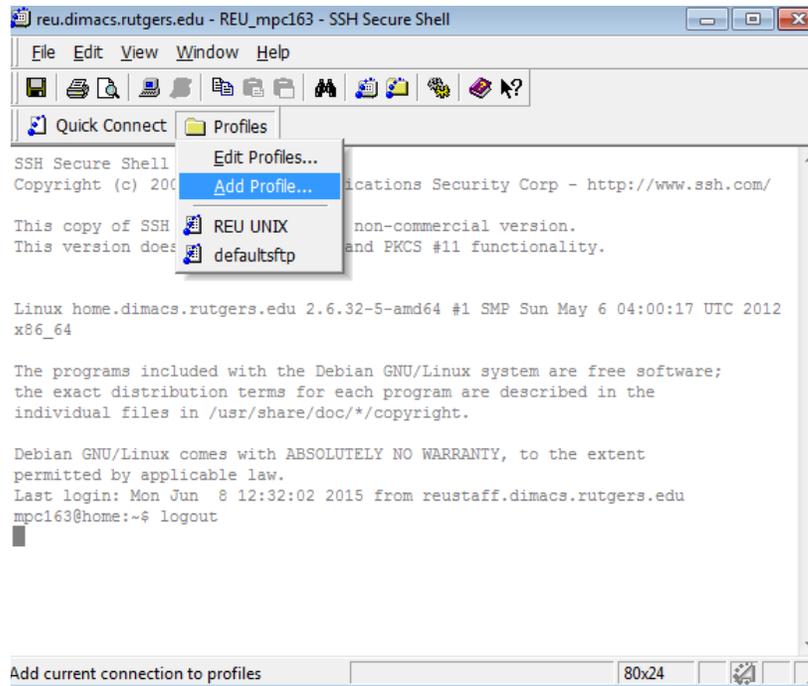
---

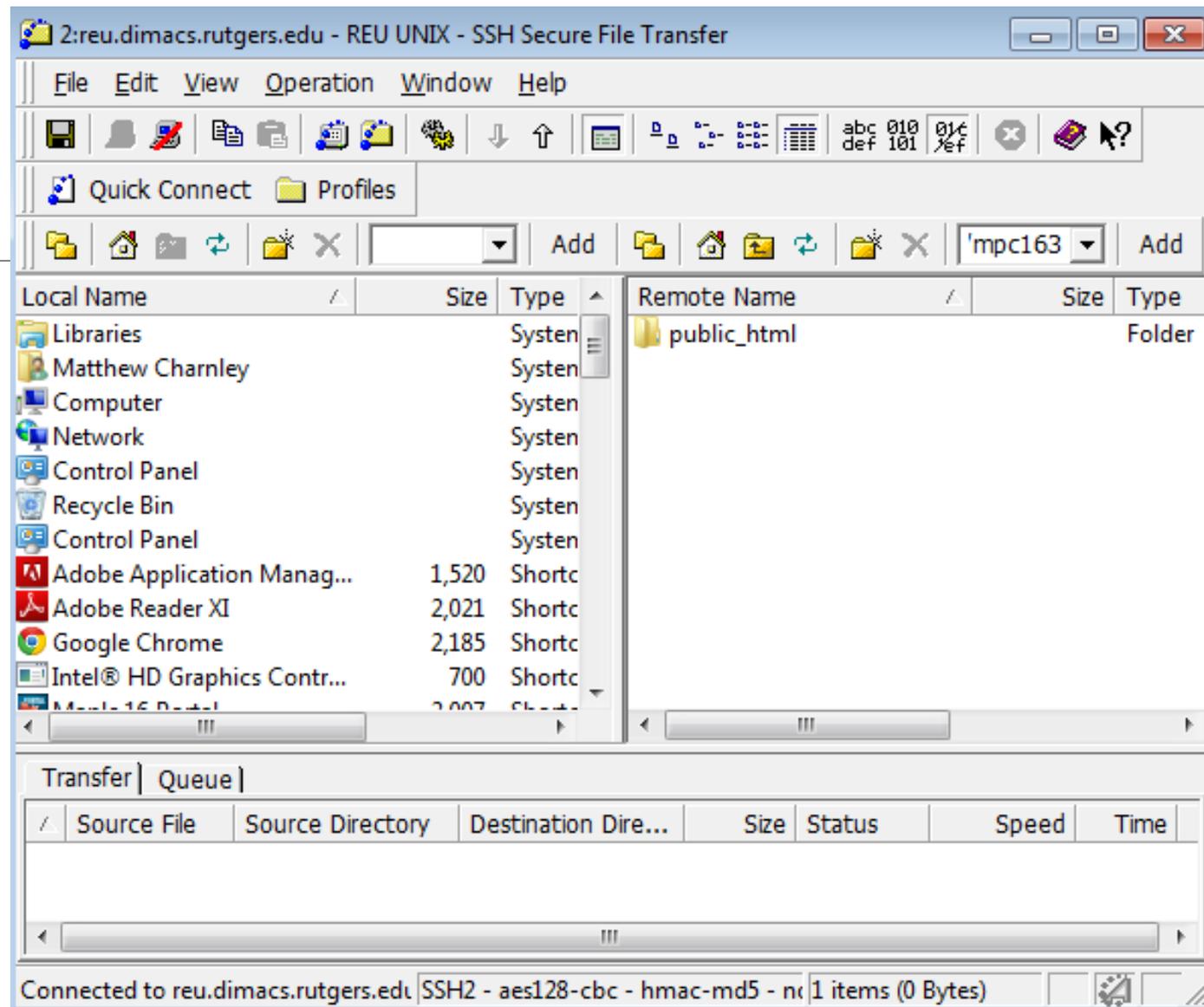
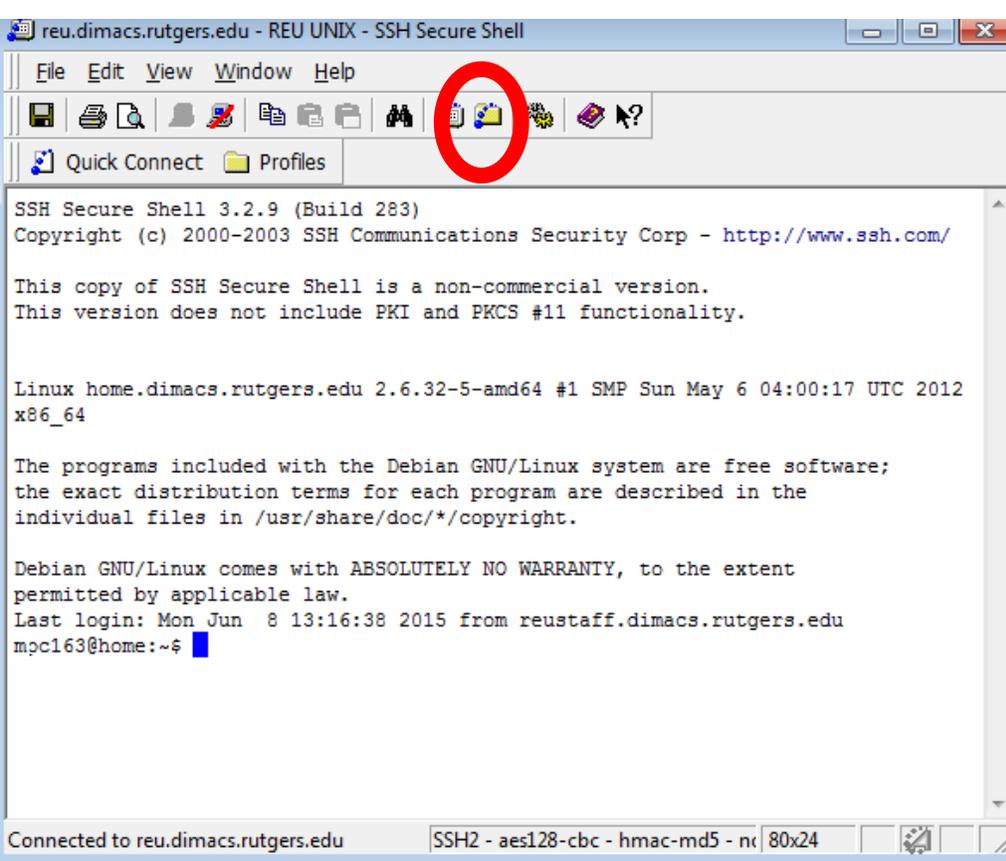
Now that we have this HTML file, we need to put it on the internet.

Luckily, the desktop computers in your offices already have the ability to do this.

- Login to the SSH Terminal
- Open the file moving application.
- Move HTML into the public\_html folder
  - Main Page is titled index.html
- Set permissions







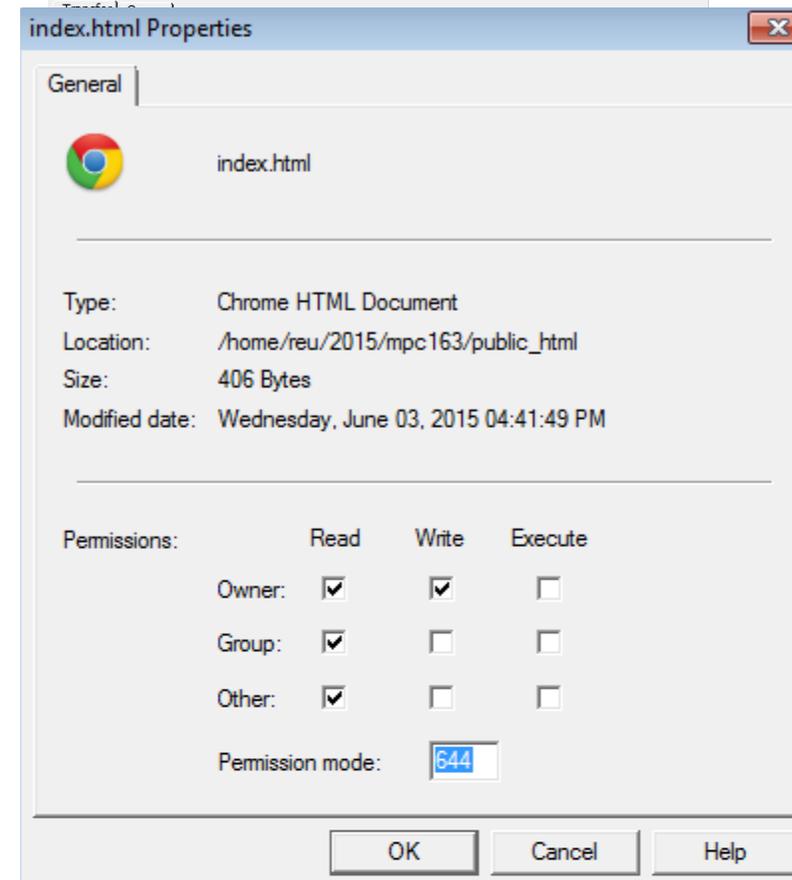
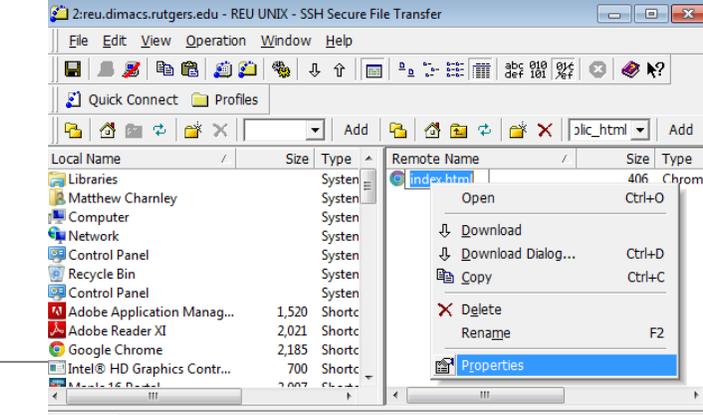
# Permissions

This has thrown me off before. Even if something is in that folder, the internet won't be able to access it unless the permissions are set properly.

You need to be able to RWX (read, write, execute)

Everyone else need to be able to Read the file.

If you have subfolder in your public\_html folder (including the folder itself) those have to have RX permissions for everyone, so that they can get inside the folder.



# CSS – Cascading Style Sheets

---

Making things look pretty (and not just the plain black and white)

Allows you to add color or formatting to your entire page, or just pieces of it, using the div names you made earlier.

Color, position, size, functionality.

```
<link type="text/css" rel="stylesheet" href="styles.css" />
```

# Sample CSS code lines

---

```
h1{  
    text-align:center;  
    text-height: 50px;  
}
```

```
body  
{  
    background-color: #dddddd;  
}
```

```
p.contact{  
    text-align:center;  
}
```

# An Interactive Introduction to $\text{\LaTeX}$

## Part 1: The Basics

Dr John D. Lees-Miller

February 26, 2015



University of  
BRISTOL

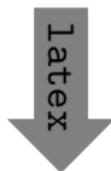
# Why L<sup>A</sup>T<sub>E</sub>X?

- ▶ It makes beautiful documents
  - ▶ Especially mathematics
- ▶ It was created by scientists, for scientists
  - ▶ A large and active community
- ▶ It is powerful — you can extend it
  - ▶ Packages for papers, presentations, spreadsheets, . . .

## How does it work?

- ▶ You write your document in plain text with `commands` that describe its structure and meaning.
- ▶ The latex program processes your text and commands to produce a beautifully formatted document.

```
The rain in Spain falls \emph{mainly} on the plain.
```



```
The rain in Spain falls mainly on the plain.
```

## More examples of commands and their output...

```
\begin{itemize}
\item Tea
\item Milk
\item Biscuits
\end{itemize}
```

- ▶ Tea
- ▶ Milk
- ▶ Biscuits

```
\begin{figure}
\includegraphics{chick}
\end{figure}
```



```
\begin{equation}
\alpha + \beta + 1
\end{equation}
```

$$\alpha + \beta + 1 \quad (1)$$

# Attitude adjustment

- ▶ Use commands to describe ‘what it is’, not ‘how it looks’.
- ▶ Focus on your content.
- ▶ Let  $\text{\LaTeX}$  do its job.

## Getting started

- ▶ A minimal  $\text{\LaTeX}$  document:

```
\documentclass{article}
\begin{document}
Hello World! % your content goes here...
\end{document}
```

- ▶ Commands start with a *backslash*  $\backslash$ .
- ▶ Every document starts with a `\documentclass` command.
- ▶ The *argument* in curly braces  $\{ \}$  tells  $\text{\LaTeX}$  what kind of document we are creating: an `article`.
- ▶ A percent sign  $\%$  starts a *comment* —  $\text{\LaTeX}$  will ignore the rest of the line.

# Getting started with **Overleaf**

- ▶ Overleaf is a website for writing documents in  $\text{\LaTeX}$ .
- ▶ It 'compiles' your  $\text{\LaTeX}$  automatically to show you the results.

Click here to open the example document in **Overleaf**

For best results, please use Google Chrome or a recent FireFox.

- ▶ As we go through the following slides, try out the examples by typing them into the example document on Overleaf.
- ▶ **No really, you should try them out as we go!**

# Typesetting Text

- ▶ Type your text between `\begin{document}` and `\end{document}`.
- ▶ For the most part, you can just type your text normally.

Words are separated by one or more spaces.

Paragraphs are separated by one or more blank lines.

Words are separated by one or more spaces.

Paragraphs are separated by one or more blank lines.

- ▶ Space in the source file is collapsed in the output.

The rain in Spain  
falls mainly on the plain.

The rain in Spain falls  
mainly on the plain.

# Typesetting Text: Caveats

- ▶ Quotation marks are a bit tricky: use a backtick ( ` ) on the left and an apostrophe ( ’ ) on the right.

Single quotes: ‘text’.

Double quotes: “text”.

Single quotes: 'text'.

Double quotes: "text".

- ▶ Some common characters have special meanings in L<sup>A</sup>T<sub>E</sub>X:

% percent sign  
# hash (pound / sharp) sign  
& ampersand  
\$ dollar sign

- ▶ If you just type these, you'll get an error. If you want one to appear in the output, you have to *escape* it by preceding it with a backslash.

`\$%\&\#!`

`$%&#!`

# Handling Errors

- ▶  $\text{\LaTeX}$  can get confused when it is trying to compile your document. If it does, it stops with an error, which you must fix before it will produce any output.
- ▶ For example, if you misspell `\emph` as `\meph`,  $\text{\LaTeX}$  will stop with an “undefined control sequence” error, because “meph” is not one of the commands it knows.

## Advice on Errors

1. Don't panic! Errors happen.
2. Fix them as soon as they arise — if what you just typed caused an error, you can start your debugging there.
3. If there are multiple errors, start with the first one — the cause may even be above it.

# Typesetting Exercise 1

Typeset this in  $\text{\LaTeX}$ : <sup>1</sup>

In March 2006, Congress raised that ceiling an additional \$0.79 trillion to \$8.97 trillion, which is approximately 68% of GDP. As of October 4, 2008, the “Emergency Economic Stabilization Act of 2008” raised the current debt ceiling to \$11.3 trillion.

Click to open this exercise in **Overleaf**

- ▶ Hint: watch out for characters with special meanings!
- ▶ Once you've tried, [click here to see my solution](#).

---

<sup>1</sup>[http://en.wikipedia.org/wiki/Economy\\_of\\_the\\_United\\_States](http://en.wikipedia.org/wiki/Economy_of_the_United_States)

# Typesetting Mathematics: Dollar Signs

- ▶ Why are dollar signs  $\$$  special? We use them to mark mathematics in text.

*% not so good:*

Let `a` and `b` be distinct positive integers, and let `c = a - b + 1`.

*% much better:*

Let `$a$` and `$b$` be distinct positive integers, and let `$c = a - b + 1$`.

Let  $a$  and  $b$  be distinct positive integers, and let  $c = a - b + 1$ .

Let  $a$  and  $b$  be distinct positive integers, and let  $c = a - b + 1$ .

- ▶ Always use dollar signs in pairs — one to begin the mathematics, and one to end it.
- ▶  $\LaTeX$  handles spacing automatically; it ignores your spaces.

Let `$y=mx+b$` be `\ldots`

Let `$y = m x + b$` be `\ldots`

Let  $y = mx + b$  be ...

Let  $y = mx + b$  be ...

# Typesetting Mathematics: Notation

- ▶ Use caret `^` for superscripts and underscore `_` for subscripts.

```
$y = c_2 x^2 + c_1 x + c_0$
```

$$y = c_2x^2 + c_1x + c_0$$

- ▶ Use curly braces `{}` `}` to group superscripts and subscripts.

```
$F_n = F_{n-1} + F_{n-2}$ % oops!
```

$$F_n = F_n - 1 + F_n - 2$$

```
$F_n = F_{n-1} + F_{n-2}$ % ok!
```

$$F_n = F_{n-1} + F_{n-2}$$

- ▶ There are commands for Greek letters and common notation.

```
$\mu = A e^{Q/RT}$
```

$$\mu = Ae^{Q/RT}$$

```
$\Omega = \sum_{k=1}^n \omega_k$
```

$$\Omega = \sum_{k=1}^n \omega_k$$

# Typesetting Mathematics: Displayed Equations

- ▶ If it's big and scary, *display* it on its own line using `\begin{equation}` and `\end{equation}`.

The roots of a quadratic equation are given by

```
\begin{equation}
x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}
\end{equation}
```

where `$a$`, `$b$` and `$c$` are `\ldots`

The roots of a quadratic equation are given by

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad (2)$$

where  $a$ ,  $b$  and  $c$  are ...

Caution:  $\LaTeX$  mostly ignores your spaces in mathematics, but it can't handle blank lines in equations — don't put blank lines in your mathematics.

## Interlude: Environments

- ▶ `equation` is an *environment* — a context.
- ▶ A command can produce different output in different contexts.

We can write

```
$ \Omega = \sum_{k=1}^n \omega_k $
```

in text, or we can write

```
\begin{equation}
```

```
  \Omega = \sum_{k=1}^n \omega_k
```

```
\end{equation}
```

to display it.

We can write  $\Omega = \sum_{k=1}^n \omega_k$   
in text, or we can write

$$\Omega = \sum_{k=1}^n \omega_k \quad (3)$$

to display it.

- ▶ Note how the  $\Sigma$  is bigger in the `equation` environment, and how the subscripts and superscripts change position, even though we used the same commands.

In fact, we could have written `$. . . $` as `\begin{math} . . . \end{math}`.

## Interlude: Environments

- ▶ The `\begin` and `\end` commands are used to create many different environments.
- ▶ The `itemize` and `enumerate` environments generate lists.

```
\begin{itemize} % for bullet points  
\item Biscuits  
\item Tea  
\end{itemize}
```

- ▶ Biscuits
- ▶ Tea

```
\begin{enumerate} % for numbers  
\item Biscuits  
\item Tea  
\end{enumerate}
```

1. Biscuits
2. Tea

## Interlude: Packages

- ▶ All of the commands and environments we've used so far are built into L<sup>A</sup>T<sub>E</sub>X.
- ▶ *Packages* are libraries of extra commands and environments. There are thousands of freely available packages.
- ▶ We have to load each of the packages we want to use with a `\usepackage` command in the *preamble*.
- ▶ Example: `amsmath` from the American Mathematical Society.

```
\documentclass{article}
\usepackage{amsmath} % preamble
\begin{document}
% now we can use commands from amsmath here...
\end{document}
```

# Typesetting Mathematics: Examples with `amsmath`

- ▶ Use `equation*` (“equation-star”) for unnumbered equations.

```
\begin{equation*}
  \Omega = \sum_{k=1}^n \omega_k
\end{equation*}
```

$$\Omega = \sum_{k=1}^n \omega_k$$

- ▶  $\text{\LaTeX}$  treats adjacent letters as variables multiplied together, which is not always what you want. `amsmath` defines commands for many common mathematical operators.

```
\begin{equation*} % bad!
  min_{x,y} (1-x)^2 + 100(y-x^2)^2
\end{equation*}
\begin{equation*} % good!
  \min_{x,y} \{(1-x)^2 + 100(y-x^2)^2\}
\end{equation*}
```

$$\min_{x,y} (1-x)^2 + 100(y-x^2)^2$$

$$\min_{x,y} (1-x)^2 + 100(y-x^2)^2$$

- ▶ You can use `\operatorname` for others.

```
\begin{equation*}
  \beta_i =
  \frac{\operatorname{Cov}(R_i, R_m)}
  {\operatorname{Var}(R_m)}
\end{equation*}
```

$$\beta_i = \frac{\operatorname{Cov}(R_i, R_m)}{\operatorname{Var}(R_m)}$$

## Typesetting Mathematics: Examples with `amsmath`

- ▶ Align a sequence of equations at the equals sign

$$\begin{aligned}(x + 1)^3 &= (x + 1)(x + 1)(x + 1) \\ &= (x + 1)(x^2 + 2x + 1) \\ &= x^3 + 3x^2 + 3x + 1\end{aligned}$$

with the `align*` environment.

```
\begin{align*}
(x+1)^3 &=& (x+1)(x+1)(x+1) & \\
&=& (x+1)(x^2 + 2x + 1) & \\
&=& x^3 + 3x^2 + 3x + 1 & \\
\end{align*}
```

- ▶ An ampersand `&` separates the left column (before the `=`) from the right column (after the `=`).
- ▶ A double backslash `\``\` starts a new line.

## Typesetting Exercise 2

Typeset this in  $\LaTeX$ :

Let  $X_1, X_2, \dots, X_n$  be a sequence of independent and identically distributed random variables with  $E[X_i] = \mu$  and  $\text{Var}[X_i] = \sigma^2 < \infty$ , and let

$$S_n = \frac{1}{n} \sum_i^n X_i$$

denote their mean. Then as  $n$  approaches infinity, the random variables  $\sqrt{n}(S_n - \mu)$  converge in distribution to a normal  $N(0, \sigma^2)$ .

Click to open this exercise in **Overleaf**

- ▶ Hint: the command for  $\infty$  is `\infty`.
- ▶ Once you've tried, [click here to see my solution](#).

## End of Part 1

- ▶ Congrats! You've already learned how to ...
  - ▶ Typeset text in  $\text{\LaTeX}$ .
  - ▶ Use lots of different commands.
  - ▶ Handle errors when they arise.
  - ▶ Typeset some beautiful mathematics.
  - ▶ Use several different environments.
  - ▶ Load packages.
- ▶ That's amazing!
- ▶ In Part 2, we'll see how to use  $\text{\LaTeX}$  to write structured documents with sections, cross references, figures, tables and bibliographies. See you then!

# An Interactive Introduction to L<sup>A</sup>T<sub>E</sub>X

## Part 2: Structured Documents & More

Dr John D. Lees-Miller

February 26, 2015



University of  
BRISTOL

# Outline

## Structured Documents

- Title and Abstract

- Sections

- Labels and Cross-References

- Exercise

## Figures and Tables

- Graphics

- Floats

- Tables

## Bibliographies

- bib $\text{T}_\text{E}\text{X}$

- Exercise

## What's Next?

- More Neat Things

- More Neat Packages

- Installing  $\text{L}_\text{A}\text{T}_\text{E}\text{X}$

- Online Resources

# Structured Documents

- ▶ In Part 1, we learned about commands and environments for typesetting text and mathematics.
- ▶ Now, we'll learn about commands and environments for structuring documents.
- ▶ You can try out the new commands in Overleaf:

Click here to open the example document in **Overleaf**

For best results, please use Google Chrome or a recent FireFox.

- ▶ Let's get started!

# Title and Abstract

- ▶ Tell  $\LaTeX$  the `\title` and `\author` names in the preamble.
- ▶ Then use `\maketitle` in the document to actually create the title.
- ▶ Use the abstract environment to make an abstract.

```
\documentclass{article}

\title{The Title}

\author{A. Author}

\date{\today}

\begin{document}
\maketitle

\begin{abstract}
Abstract goes here...
\end{abstract}

\end{document}
```

The Title

A. Author

February 26, 2015

**Abstract**

Abstract goes here...

# Sections

- ▶ Just use `\section` and `\subsection`.
- ▶ Can you guess what `\section*` and `\subsection*` do?

```
\documentclass{article}
\begin{document}

\section{Introduction}

The problem of \ldots

\section{Method}

We investigate \ldots

\subsection{Sample Preparation}

\subsection{Data Collection}

\section{Results}

\section{Conclusion}

\end{document}
```

## 1 Introduction

The problem of ...

## 2 Method

We investigate ...

### 2.1 Sample Preparation

### 2.2 Data Collection

## 3 Results

## 4 Conclusion

# Labels and Cross-References

- ▶ Use `\label` and `\ref` for automatic numbering.
- ▶ The `amsmath` package provides `\eqref` for referencing equations.

```
\documentclass{article}
\usepackage{amsmath} % for \eqref
\begin{document}

\section{Introduction}
\label{sec:intro}

In Section \ref{sec:method}, we \ldots

\section{Method}
\label{sec:method}

\begin{equation}
\label{eq:euler}
e^{i\pi} + 1 = 0
\end{equation}

By \eqref{eq:euler}, we have \ldots

\end{document}
```

## 1 Introduction

In Section 2, we ...

## 2 Method

By (1), we have ...  $e^{i\pi} + 1 = 0$  (1)

# Structured Documents Exercise

Typeset this short paper in  $\text{\LaTeX}$ : <sup>1</sup>

Click to open the paper

Make your paper look like this one. Use `\ref` and `\eqref` to avoid explicitly writing section and equation numbers into the text.

Click to open this exercise in **Overleaf**

- ▶ Once you've tried, [click here to see my solution](#).

---

<sup>1</sup>From <http://pdos.csail.mit.edu/scigen/>, a random paper generator.

# Outline

## Structured Documents

Title and Abstract

Sections

Labels and Cross-References

Exercise

## Figures and Tables

Graphics

Floats

Tables

## Bibliographies

bibTeX

Exercise

## What's Next?

More Neat Things

More Neat Packages

Installing L<sup>A</sup>T<sub>E</sub>X

Online Resources

# Graphics

- ▶ Requires the `graphicx` package, which provides the `\includegraphics` command.
- ▶ Supported graphics formats include JPEG, PNG and PDF (usually).

```
\includegraphics[  
  width=0.5\textwidth]{big_chick}
```

```
\includegraphics[  
  width=0.3\textwidth,  
  angle=270]{big_chick}
```



Image from [http://www.andy-roberts.net/writing/latex/importing\\_images](http://www.andy-roberts.net/writing/latex/importing_images)

## Interlude: Optional Arguments

- ▶ We use square brackets `[ ]` for optional arguments, instead of braces `{ }`.
- ▶ `\includegraphics` accepts optional arguments that allow you to transform the image when it is included. For example, `width=0.3\textwidth` makes the image take up 30% of the width of the surrounding text (`\textwidth`).
- ▶ `\documentclass` accepts optional arguments, too. Example:

```
\documentclass[12pt,twocolumn]{article}
```

makes the text bigger (12pt) and puts it into two columns.

- ▶ Where do you find out about these? See the slides at the end of this presentation for links to more information.

# Floats

- ▶ Allow  $\LaTeX$  to decide where the figure will go (it can “float”).
- ▶ You can also give the figure a caption, which can be referenced with `\ref`.

```
\documentclass{article}
\usepackage{graphicx}
\begin{document}

Figure \ref{fig:chick} shows \ldots

\begin{figure}
\centering
\includegraphics[%
width=0.5\textwidth]{big_chick}
\caption{\label{fig:chick}Aww\ldots.}
\end{figure}

\end{document}
```



Figure 1: Aww....

Figure 1 shows ...

# Tables

- ▶ Tables in  $\text{\LaTeX}$  take some getting used to.
- ▶ Use the `tabular` environment from the `tabularx` package.
- ▶ The argument specifies column alignment — left, right, right.

```
\begin{tabular}{lrr}  
Item & Qty & Unit \ $ \\  
Widget & 1 & 199.99 \\  
Gadget & 2 & 399.99 \\  
Cable & 3 & 19.99 \\  
\end{tabular}
```

Item	Qty	Unit \$
Widget	1	199.99
Gadget	2	399.99
Cable	3	19.99

- ▶ It also specifies vertical lines; use `\hline` for horizontal lines.

```
\begin{tabular}{|l|r|r|} \hline  
Item & Qty & Unit \ $ \\  
Widget & 1 & 199.99 \\  
Gadget & 2 & 399.99 \\  
Cable & 3 & 19.99 \\  
\end{tabular}
```

Item	Qty	Unit \$
Widget	1	199.99
Gadget	2	399.99
Cable	3	19.99

- ▶ Use an ampersand `&` to separate columns and a double backslash `\\` to start a new row (like in the `align*` environment that we saw in part 1).

# Outline

## Structured Documents

Title and Abstract

Sections

Labels and Cross-References

Exercise

## Figures and Tables

Graphics

Floats

Tables

## Bibliographies

`bibTeX`

Exercise

## What's Next?

More Neat Things

More Neat Packages

Installing `LATEX`

Online Resources

- ▶ Put your references in a .bib file in 'bibtex' database format:

```
@Article{Jacobson1999Towards,  
  author = {Van Jacobson},  
  title = {Towards the Analysis of Massive Multiplayer Online  
          Role-Playing Games},  
  journal = {Journal of Ubiquitous Information},  
  Month = jun,  
  Year = 1999,  
  Volume = 6,  
  Pages = {75--83}}  
  
@InProceedings{Brooks1997Methodology,  
  author = {Fredrick P. Brooks and John Kubiawicz and  
          Christos Papadimitriou},  
  title = {A Methodology for the Study of the  
          Location-Identity Split},  
  booktitle = {Proceedings of OOPSLA},  
  Month = jun,  
  Year = 1997}
```

- ▶ Most reference managers can export to bibtex format.

## bibT<sub>E</sub>X 2

- ▶ Each entry in the .bib file has a *key* that you can use to reference it in the document. For example, Jacobson1999Towards is the key for this article:

```
@Article{Jacobson1999Towards,  
  author = {Van Jacobson},  
  ...  
}
```

- ▶ It's a good idea to use a key based on the name, year and title.
- ▶ L<sup>A</sup>T<sub>E</sub>X can automatically format your in-text citations and generate a list of references; it knows most standard styles, and you can design your own.

## bibT<sub>E</sub>X 3

- ▶ Use the natbib package<sup>2</sup> with `\citet` and `\citep`.
- ▶ Reference `\bibliography` at the end, and specify a `\bibliographystyle`.

```
\documentclass{article}
\usepackage{natbib}
\begin{document}

\citet{Brooks1997Methodology}
show that \ldots. Clearly,
all odd numbers are prime
\citep{Jacobson1999Towards}.

\bibliography{bib-example}
% if 'bib-example' is the name of
% your bib file

\bibliographystyle{plainnat}
% try changing to abbrunat

\end{document}
```

Brooks et al. [1997] show that .... Clearly, all odd numbers are prime [Jacobson, 1999].

### References

Frederick P. Brooks, John Kubiawicz, and Christos Papadimitriou. A methodology for the study of the location-identity split. In *Proceedings of OOPSL* June 1997.

Van Jacobson. Towards the analysis of massive multiplayer online role-play games. *Journal of Ubiquitous Information*, 6:75-83, June 1999.

<sup>2</sup>There is a new package with more features named biblatex but most of the articles templates still use natbib.

## Exercise: Putting it All Together

Add an image and a bibliography to the paper from the previous exercise.

1. Download these example files to your computer.

[Click to download example image](#)

[Click to download example bib file](#)

2. Upload them to Overleaf (use the project menu).

# Outline

## Structured Documents

Title and Abstract

Sections

Labels and Cross-References

Exercise

## Figures and Tables

Graphics

Floats

Tables

## Bibliographies

bibTeX

Exercise

## What's Next?

More Neat Things

More Neat Packages

Installing L<sup>A</sup>T<sub>E</sub>X

Online Resources

## More Neat Things

- ▶ Add the `\tableofcontents` command to generate a table of contents from the `\section` commands.

- ▶ Change the `\documentclass` to

```
\documentclass{scrartcl}
```

or

```
\documentclass[12pt]{IEEEtran}
```

- ▶ Define your own command for a complicated equation:

```
\newcommand{\rperf}{%  
  \rho_{\text{perf}}}  
$$  
\rperf = {\bf c}'{\bf X} + \varepsilon  
$$
```

$$\rho_{\text{perf}} = \mathbf{c}'\mathbf{X} + \varepsilon$$

## More Neat Packages

- ▶ `beamer`: for presentations (like this one!)
- ▶ `todonotes`: comments and TODO management
- ▶ `tikz`: make amazing graphics
- ▶ `pgfplots`: create graphs in  $\text{\LaTeX}$
- ▶ `listings`: source code printer for  $\text{\LaTeX}$
- ▶ `spreadtab`: create spreadsheets in  $\text{\LaTeX}$
- ▶ `gchords`, `guitar`: guitar chords and tabulature
- ▶ `cwpuzzle`: crossword puzzles

See <https://www.overleaf.com/latex/examples> and <http://texample.net> for examples of (most of) these packages.

# Installing L<sup>A</sup>T<sub>E</sub>X

- ▶ To run L<sup>A</sup>T<sub>E</sub>X on your own computer, you'll want to use a L<sup>A</sup>T<sub>E</sub>X *distribution*. A distribution includes a latex program and (typically) several thousand packages.
  - ▶ On Windows: MikT<sub>E</sub>X or T<sub>E</sub>XLive
  - ▶ On Linux: T<sub>E</sub>XLive
  - ▶ On Mac: MacT<sub>E</sub>X
- ▶ You'll also want a text editor with L<sup>A</sup>T<sub>E</sub>X support. See [http://en.wikipedia.org/wiki/Comparison\\_of\\_TeX\\_editors](http://en.wikipedia.org/wiki/Comparison_of_TeX_editors) for a list of (many) options.
- ▶ You'll also have to know more about how latex and its related tools work — see the resources on the next slide.

## Online Resources

- ▶ The  $\LaTeX$  Wikibook — excellent tutorials and reference material.
- ▶  $\TeX$  Stack Exchange — ask questions and get excellent answers incredibly quickly
- ▶  $\LaTeX$  Community — a large online forum
- ▶ Comprehensive  $\TeX$  Archive Network (CTAN) — over four thousand packages plus documentation
- ▶ Google will usually get you to one of the above.

Thanks, and happy T<sub>E</sub>Xing!