LaTeX: A Document Preparation System

An Introduction

Dr. Fred Barnes

S113, School of Computing (ext. 4278) `frmb@kent.ac.uk`

What It Is

- A generic **typesetting** system.
  - often written LaTeX (LaTeX).
  - pronounced **lay-tek** (the ‘X’ is from the Greek letter χ – *chi*).
- Uses TeX as its formatting engine.
  - invented by Donald Knuth, first released in 1978.
  - most recent revision January 2014.
- A structured way of **formatting** documents.
  - particularly suited to **large things**: books, PhD theses, MSc dissertations, project reports, technical manuals, ...
  - the small: letters, reports, handouts, invoices, cover-sheets, ...
  - academic articles and anything with **maths** or science.
- Familiar examples:
  - these slides; the handouts for the practical sessions later; the “Early Computing at Kent” poster in the Cornwallis Foyer.

What It Is

- A generic **typesetting** system.
  - often written LaTeX (LaTeX).
  - pronounced **lay-tek** (the ‘X’ is from the Greek letter χ – *chi*).
- Uses TeX as its formatting engine.
  - invented by Donald Knuth, first released in 1978.
  - most recent revision January 2014.
- A structured way of **formatting** documents.
  - particularly suited to **large things**: books, PhD theses, MSc dissertations, project reports, technical manuals, ...
  - the small: letters, reports, handouts, invoices, cover-sheets, ...
  - academic articles and anything with **maths** or science.
- Familiar examples:
  - these slides; the handouts for the practical sessions later; the “Early Computing at Kent” poster in the Cornwallis Foyer.

What It Is

- A generic **typesetting** system.
  - often written LaTeX (LaTeX).
  - pronounced **lay-tek** (the 'X' is from the Greek letter χ – *chi*).
- Uses TeX as its formatting engine.
  - invented by Donald Knuth, first released in 1978.
  - most recent revision January 2014.
- A structured way of **formatting** documents.
  - particularly suited to **large things**: books, PhD theses, MSc dissertations, project reports, technical manuals, ...
  - the small: letters, reports, handouts, invoices, cover-sheets, ...
  - academic articles and anything with **maths** or science.
- Familiar examples:
  - these slides; the handouts for the practical sessions later; the "Early Computing at Kent" poster in the Cornwallis Foyer.

What It Is

- A generic **typesetting** system.
  - often written LaTeX (LaTeX).
  - pronounced **lay-tek** (the 'X' is from the Greek letter χ – *chi*).
- Uses TeX as its formatting engine.
  - invented by Donald Knuth, first released in 1978.
  - most recent revision January 2014.
- A structured way of **formatting** documents.
  - particularly suited to **large things**: books, PhD theses, MSc dissertations, project reports, technical manuals, ...
  - the small: letters, reports, handouts, invoices, cover-sheets, ...
  - academic articles and anything with **maths** or science.
- Familiar examples:
  - these slides; the handouts for the practical sessions later; the "Early Computing at Kent" poster in the Cornwallis Foyer.

What It Is Not

- LaTeX is **not** a word-processor.
- Nor it is straightforward much of the time.
  - errors can be particularly hard to unpick sometimes.

Why Should I Use It?

- If you care about producing documents of a **professional** standard.
  - where "professional document" is something like a solicitor's letter, newspaper or magazine page.
  - and not the badly laid out flyer or office memo in Comic Sans.
    - there is a place for Comic Sans; most are not it. http://inappropriatecomicsans.tumblr.com/
- If you want to write things that have accented characters.
  - lēgibus pārendum est.
  - À l'œuvre, on connaît l'artisan.
- Or mathematics:

$$X = \sqrt{42}, \qquad Y = \sum_{i=0}^{n-1} \mathbb{E}_i \times \left(\int_0^\infty \mu(s) \; ds \right)$$

Why Should I Use It?

- If you care about producing documents of a **professional** standard.
  - where “professional document” is something like a solicitor’s letter, newspaper or magazine page.
  - and not the badly laid out flyer or office memo in Comic Sans.
    - there is a place for Comic Sans; most are not it. http://inappropriatecomicsans.tumblr.com/
- If you want to write things that have accented characters.
  - lēgibus pārendum est.
  - À l’œuvre, on connaît l’artisan.
- Or mathematics:

$$X = \sqrt{42}, \qquad Y = \sum_{i=0}^{n-1} \mathbb{E}_i \times \left(\int_0^\infty \mu(s)\ ds \right)$$

Why Should I Use It?

- If you care about producing documents of a **professional** standard.
  - where “professional document” is something like a solicitor’s letter, newspaper or magazine page.
  - and not the badly laid out flyer or office memo in Comic Sans.
    - there is a place for Comic Sans; most are not it. http://inappropriatecomicsans.tumblr.com/
- If you want to write things that have accented characters.
  - lēgibus pārendum est.
  - À l’œuvre, on connaît l’artisan.
- Or mathematics:

$$X = \sqrt{42}, \qquad Y = \sum_{i=0}^{n-1} \mathbb{E}_i \times \left(\int_0^\infty \mu(s)\ ds \right)$$

One View ...

- One way to see TEX is as a **programming language**.
  - 'programs' written in this language are **transformed** (compiled) into pages of output (DVI, PostScript and/or PDF).
- LATEX is a set of **macros** (library 'functions') for TEX.
  - vastly simplifies the creation of standard documents (article, book, letter, slides, ...).
  - TEX is more concerned with "layout of things on a page".

One View ...

- One way to see TEX is as a **programming language**.
  - 'programs' written in this language are **transformed** (compiled) into pages of output (DVI, PostScript and/or PDF).
- LATEX is a set of **macros** (library 'functions') for TEX.
  - vastly simplifies the creation of standard documents (article, book, letter, slides, ...).
  - TEX is more concerned with "layout of things on a page".

The Software

- Several predominant **distributions** of LaTeX.
  - TeX Live: http://www.tug.org/texlive/ (my preferred)
  - MacTeX: http://tug.org/mactex/
  - MiKTeX: http://miktex.org/
- Not a vast difference between them now.
  - variations in the individual **packages**, fonts, etc.
  - available packages number in the **thousands**.
- Typically an assortment of **command-line** tools.
  - some that invoke TeX and LaTeX in different ways to produce specific types of output (e.g. "musixtex" for typesetting musical scores).
  - we'll mostly be using **pdflatex** (it's moderately simple!).

The Software

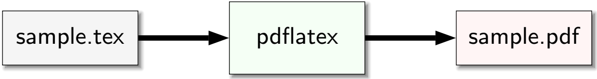
- Several predominant **distributions** of LaTeX.
  - TeX Live: http://www.tug.org/texlive/ (my preferred)
  - MacTeX: http://tug.org/mactex/
  - MiKTeX: http://miktex.org/
- Not a vast difference between them now.
  - variations in the individual **packages**, fonts, etc.
  - available packages number in the **thousands**.
- Typically an assortment of **command-line** tools.
  - some that invoke TeX and LaTeX in different ways to produce specific types of output (e.g. "musixtex" for typesetting musical scores).
  - we'll mostly be using **pdflatex** (it's moderately simple!).

The Software

- Several predominant **distributions** of LaTeX.
  - TeX Live: http://www.tug.org/texlive/ (my preferred)
  - MacTeX: http://tug.org/mactex/
  - MiKTeX: http://miktex.org/
- Not a vast difference between them now.
  - variations in the individual **packages**, fonts, etc.
  - available packages number in the **thousands**.
- Typically an assortment of **command-line** tools.
  - some that invoke TeX and LaTeX in different ways to produce specific types of output (e.g. "musixtex" for typesetting musical scores).
  - we'll mostly be using **pdflatex** (it's moderately simple!).

How it Works

- The simple view:



sample.tex

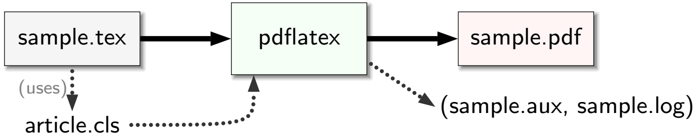
pdflatex

sample.pdf

- What happens in reality is significantly more complex.
  - but we don't need to concern ourselves with that (until it goes wrong..).

How it Works

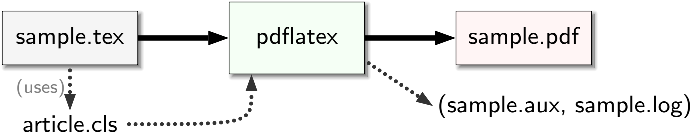
- The simple view:



- What happens in reality is significantly more complex.
    - but we don't need to concern ourselves with that (until it goes wrong..).

How it Works

- The simple view:



- What happens in reality is significantly more complex.
  - but we don't need to concern ourselves with that (until it goes wrong..).

A Simple Example

simple.tex:

```
% comments start with a percent sign and continue to EOL
% comment symbol at EOL is a continuation

\documentclass[12pt]{article}

\begin{document}

Welcome to a very simple \LaTeX~document!

\end{document}
\endinput
```

```
bash$ pdflatex simple
... stuff
```

→ simple.pdf.

LaTeX Document Structure

- First thing is a `\documentclass` directive.
  - determines the overall **type** of document and particular options.
- Then the **preamble**.
  - if other packages are required, or general other setting-up.
- The start of the **document** proper:

  `\begin{document}`

  followed by all of its **content**.
- And the end of the document:

  `\end{document}`

  and end of the input:

  `\endinput`

Aside: you can put whetever after '`\endinput`'; TEX won't try to read it.

LaTeX Document Structure

- First thing is a `\documentclass` directive.
  - determines the overall **type** of document and particular options.
- Then the **preamble**.
  - if other packages are required, or general other setting-up.
- The start of the **document** proper:

  `\begin{document}`

  followed by all of its **content**.
- And the end of the document:

  `\end{document}`

  and end of the input:

  `\endinput`

Aside: you can put whetever after '`\endinput`'; TeX won't try to read it.

LaTeX Document Structure

- First thing is a `\documentclass` directive.
  - determines the overall **type** of document and particular options.
- Then the **preamble**.
  - if other packages are required, or general other setting-up.
- The start of the **document** proper:

  `\begin{document}`

  followed by all of its **content**.
- And the end of the document:

  `\end{document}`

  and end of the input:

  `\endinput`

Aside: you can put whetever after '`\endinput`'; TEX won't try to read it.

LaTeX Document Structure

- First thing is a `\documentclass` directive.
    - determines the overall **type** of document and particular options.
- Then the **preamble**.
    - if other packages are required, or general other setting-up.
- The start of the **document** proper:

  ```
  \begin{document}
  ```

  followed by all of its **content**.
- And the end of the document:

  ```
  \end{document}
  ```

  and end of the input:

  ```
  \endinput
  ```

Aside: you can put whetever after ‘\endinput’; TEX won’t try to read it.

LaTeX Document Structure

- First thing is a `\documentclass` directive.
  - determines the overall **type** of document and particular options.
- Then the **preamble**.
  - if other packages are required, or general other setting-up.
- The start of the **document** proper:

  ```
  \begin{document}
  ```

  followed by all of its **content**.
- And the end of the document:

  ```
  \end{document}
  ```

  and end of the input:

  ```
  \endinput
  ```

Aside: you can put whetever after '`\endinput`'; TeX won't try to read it.

LaTeX Document Structure

- First thing is a `\documentclass` directive.
  - determines the overall **type** of document and particular options.
- Then the **preamble**.
  - if other packages are required, or general other setting-up.
- The start of the **document** proper:

  ```
  \begin{document}
  ```

  followed by all of its **content**.
- And the end of the document:

  ```
  \end{document}
  ```

  and end of the input:

  ```
  \endinput
  ```

Aside: you can put whetever after ‘`\endinput`’; TEX won’t try to read it.

On Structure and Style

- Document **structure**: how something is arranged in chapters, sections, sub-sections, paragraphs, numbered-lists, etc.
- Document **style**: what it looks like, e.g. **bold**, *italics*, cyan.
- In a **markup** language (HTML), structure is the main concern.
  - used not to be the case, but for the vastness of the web now, a good idea to manage style separately (CSS).
- In LaTeX, structure and style have mostly equal footing.
  - style can be separated (and sometimes is).
  - because we're programming in something that is **Turing complete** (the TeX engine) can really do it however we like.
- Downside: authors over the years have done it how they liked.
  - the results of which are both **clever** and **powerful**, but not necessarily consistent, obvious or pretty.

On Structure and Style

- Document **structure**: how something is arranged in chapters, sections, sub-sections, paragraphs, numbered-lists, etc.
- Document **style**: what it looks like, e.g. **bold**, *italics*, cyan.
- In a **markup** language (HTML), structure is the main concern.
  - used not to be the case, but for the vastness of the web now, a good idea to manage style separately (CSS).
- In LaTeX, structure and style have mostly equal footing.
  - style can be separated (and sometimes is).
  - because we're programming in something that is **Turing complete** (the TeX engine) can really do it however we like.
- Downside: authors over the years have done it how they liked.
  - the results of which are both **clever** and **powerful**, but not necessarily consistent, obvious or pretty.

On Structure and Style

- Document **structure**: how something is arranged in chapters, sections, sub-sections, paragraphs, numbered-lists, etc.
- Document **style**: what it looks like, e.g. **bold**, *italics*, cyan.
- In a **markup** language (HTML), structure is the main concern.
  - used not to be the case, but for the vastness of the web now, a good idea to manage style separately (CSS).
- In LaTeX, structure and style have mostly equal footing.
  - style can be separated (and sometimes is).
  - because we're programming in something that is **Turing complete** (the TeX engine) can really do it however we like.
- Downside: authors over the years have done it how they liked.
  - the results of which are both **clever** and **powerful**, but not necessarily consistent, obvious or pretty.

On Structure and Style

- Document **structure**: how something is arranged in chapters, sections, sub-sections, paragraphs, numbered-lists, etc.
- Document **style**: what it looks like, e.g. **bold**, *italics*, cyan.
- In a **markup** language (HTML), structure is the main concern.
  - used not to be the case, but for the vastness of the web now, a good idea to manage style separately (CSS).
- In LaTeX, structure and style have mostly equal footing.
  - style can be separated (and sometimes is).
  - because we're programming in something that is **Turing complete** (the TeX engine) can really do it however we like.
- Downside: authors over the years have done it how they liked.
  - the results of which are both **clever** and **powerful**, but not necessarily consistent, obvious or pretty.

Commands

- Commands in LaTeX (and also TeX) start with a **black-slash**.
    - **arguments** to commands *normally* occur immediately after, surrounded by **curly-braces**.
    - also common to specify **options** in **square-brackets** between the command and its arguments.
- Not limited to this.
    - the free-form nature of the TeX **macro language** means commands can collect arguments in any number of ways.
    - you will doubtless encounter some **very strange** looking things.

```
\def\nrres{\mathbin{\rlap{\raise.05ex\hbox{$-$}}{\rres}}}
```

Commands

- Commands in LaTeX (and also TeX) start with a **black-slash**.
  - **arguments** to commands *normally* occur immediately after, surrounded by **curly-braces**.
  - also common to specify **options** in **square-brackets** between the command and its arguments.
- Not limited to this.
  - the free-form nature of the TeX **macro language** means commands can collect arguments in any number of ways.
  - you will doubtless encounter some **very strange** looking things.

```
\def\nrres{\mathbin{\rlap{\raise.05ex\hbox{$-$}}{\rres}}}
```

Special Characters

- Some characters have a special meaning to LaTeX.
  - foremost, **back-slash**: used to start commands.
  - the curly braces { and }, used to **group** things (usually arguments to commands).
  - the dollar sign $, used to go in and out of **maths-mode**.
  - the ampersand &, used to handle **alignment** in various things.
  - the hash #, used to refer to arguments inside commands.
  - the tilde ˜, that is a **non-breaking space**.
- This means we cannot just **write** these if we want them output.
  - need to be **escaped** in some way.

write:	to get:
`\textbackslash`	\
`\$`	$
`\#`	#
`\textasciitilde`	˜

write:	to get:
`\{`	{
`\}`	}
`\&`	&

Special Characters

- Some characters have a special meaning to LaTeX.
  - foremost, **back-slash**: used to start commands.
  - the curly braces { and }, used to **group** things (usually arguments to commands).
  - the dollar sign $, used to go in and out of **maths-mode**.
  - the ampersand &, used to handle **alignment** in various things.
  - the hash #, used to refer to arguments inside commands.
  - the tilde ˜, that is a **non-breaking space**.
- This means we cannot just **write** these if we want them output.
  - need to be **escaped** in some way.

write:	to get:
`\textbackslash`	\
`\$`	$
`\#`	#
`\textasciitilde`	˜

write:	to get:
`\{`	{
`\}`	}
`\&`	&

Command Types

- High-level structuring:
  - commands such as `\documentclass`, `\section`, `\subsection`.
  - typically take some arguments.
- Environments (grouping):
  - things that start `\begin{env}` and finish `\end{env}` with arbitrary content inbetween (some restrictions).
  - may take various options.
  - may change the way input text is processed within.
- Styling:
  - generally small commands that change how something is displayed.
  - `\textbf{some bold text}` → **some bold text**.
  - `{\bf some bold text}` → **some bold text**.
- Low-level things:
  - commands that define new commands, setup if-then-else structures or even loops.

Command Types

- High-level structuring:
  - commands such as `\documentclass`, `\section`, `\subsection`.
  - typically take some arguments.
- Environments (grouping):
  - things that start `\begin{env}` and finish `\end{env}` with arbitrary content inbetween (some restrictions).
  - may take various options.
  - may change the way input text is processed within.
- Styling:
  - generally small commands that change how something is displayed.
  - `\textbf{some bold text}` → **some bold text**.
  - `{\bf some bold text}` → **some bold text**.
- Low-level things:
  - commands that define new commands, setup if-then-else structures or even loops.

Command Types

- High-level structuring:
  - commands such as `\documentclass`, `\section`, `\subsection`.
  - typically take some arguments.
- Environments (grouping):
  - things that start `\begin{env}` and finish `\end{env}` with arbitrary content inbetween (some restrictions).
  - may take various options.
  - may change the way input text is processed within.
- Styling:
  - generally small commands that change how something is displayed.
  - `\textbf{some bold text}` → **some bold text**.
  - `{\bf some bold text}` → **some bold text**.
- Low-level things:
  - commands that define new commands, setup if-then-else structures or even loops.

Command Types

- High-level structuring:
  - commands such as `\documentclass`, `\section`, `\subsection`.
  - typically take some arguments.
- Environments (grouping):
  - things that start `\begin{env}` and finish `\end{env}` with arbitrary content inbetween (some restrictions).
  - may take various options.
  - may change the way input text is processed within.
- Styling:
  - generally small commands that change how something is displayed.
  - `\textbf{some bold text}` → **some bold text**.
  - `{\bf some bold text}` → **some bold text**.
- Low-level things:
  - commands that define new commands, setup if-then-else structures or even loops.

Commands and Groups

- From the previous slide, two ways of producing bold text.
  - `\textbf{some bold text}` → **some bold text**.
  - `{\bf some bold text}` → **some bold text**.
- The first is a **command**, `\textbf` with one argument.
  - the text to show in bold.
- The second is a **group**.
  - the first command, `\bf`, changes the active formatting style to bold.
  - the effect of this lasts until the end of the group.

Commands and Groups

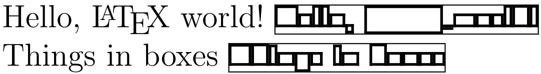
- From the previous slide, two ways of producing bold text.
  - `\textbf{some bold text}` → **some bold text**.
  - `{\bf some bold text}` → **some bold text**.
- The first is a **command**, `\textbf` with one argument.
  - the text to show in bold.
- The second is a **group**.
  - the first command, `\bf`, changes the active formatting style to bold.
  - the effect of this lasts until the end of the group.

Commands and Groups

- From the previous slide, two ways of producing bold text.
  - `\textbf{some bold text}` → **some bold text**.
  - `{\bf some bold text}` → **some bold text**.
- The first is a **command**, `\textbf` with one argument.
  - the text to show in bold.
- The second is a **group**.
  - the first command, `\bf`, changes the active formatting style to bold.
  - the effect of this lasts until the end of the group.

Boxes

- At some level, what TEX does is related to sticking **boxes** together.
  - with fixed **glue** or elastic glue.



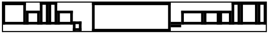
Hello, LATEX world!

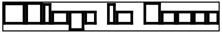
Things in boxes

- Individual characters glue together (fixed) to form words.
  - that also become boxes themselves (e.g. the LATEX macro above).
  - words glue together (elastic) to form paragraphs (also boxes).
  - and these get packed onto the page.
- TEX is aware of **page layout** and is able to:
  - break words at the ends of lines (**hyphenation**).
  - break paragraphs over page boundaries.
- Why do we care?
  - when LATEX doesn't quite do what we want, boxes may be a contributing factor — **warnings** and **errors**.
  - can do some crafty things by manipulating boxes.

Boxes

- At some level, what TEX does is related to sticking **boxes** together.
  - with fixed **glue** or elastic glue.

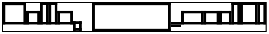
Hello, LATEX world! 

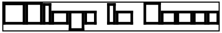
Things in boxes 

- Individual characters glue together (fixed) to form words.
  - that also become boxes themselves (e.g. the LATEX macro above).
  - words glue together (elastic) to form paragraphs (also boxes).
  - and these get packed onto the page.
- TEX is aware of **page layout** and is able to:
  - break words at the ends of lines (**hyphenation**).
  - break paragraphs over page boundaries.
- Why do we care?
  - when LATEX doesn't quite do what we want, boxes may be a contributing factor — **warnings** and **errors**.
  - can do some crafty things by manipulating boxes.

Boxes

- At some level, what TEX does is related to sticking **boxes** together.
  - with fixed **glue** or elastic glue.

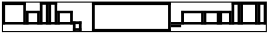
Hello, LATEX world! 

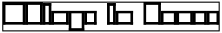
Things in boxes 

- Individual characters glue together (fixed) to form words.
  - that also become boxes themselves (e.g. the LATEX macro above).
  - words glue together (elastic) to form paragraphs (also boxes).
  - and these get packed onto the page.
- TEX is aware of **page layout** and is able to:
  - break words at the ends of lines (**hyphenation**).
  - break paragraphs over page boundaries.
- Why do we care?
  - when LATEX doesn't quite do what we want, boxes may be a contributing factor — **warnings** and **errors**.
  - can do some crafty things by manipulating boxes.

Boxes

- At some level, what TEX does is related to sticking **boxes** together.
  - with fixed **glue** or elastic glue.

Hello, LATEX world! 

Things in boxes 

- Individual characters glue together (fixed) to form words.
  - that also become boxes themselves (e.g. the LATEX macro above).
  - words glue together (elastic) to form paragraphs (also boxes).
  - and these get packed onto the page.
- TEX is aware of **page layout** and is able to:
  - break words at the ends of lines (**hyphenation**).
  - break paragraphs over page boundaries.
- Why do we care?
  - when LATEX doesn't quite do what we want, boxes may be a contributing factor — **warnings** and **errors**.
  - can do some crafty things by manipulating boxes.

Weights and Measures

- A lot of markup in LaTeX is related to **where** and **how big**.
  - to help, TeX supports a whole raft of length related **units**.
- Generally written as a **number** followed by two unit characters:

  `0.5in, 5pt, 4em, 42mm`

  - '`pt`': 1 point = $\frac{1}{72.27}$in = 0.351mm.
  - '`in`': 1 inch = 25.4mm = 72.27pt = 6.022pc.
  - '`mm`': 1 millimeter = 2.845pt.
  - '`ex`': height of a small 'x' in the current font.
  - '`em`': width of capital 'M' in the current font.
  - '`mu`': 1 math-unit = $\frac{1}{18}$em (for positioning in math mode).

Weights and Measures

- A lot of markup in LaTeX is related to **where** and **how big**.
  - to help, TeX supports a whole raft of length related **units**.
- Generally written as a **number** followed by two unit characters:

  `0.5in, 5pt, 4em, 42mm`

  - ‘`pt`’: 1 point $= \frac{1}{72.27}$in $= 0.351$mm.
  - ‘`in`’: 1 inch $= 25.4$mm $= 72.27$pt $= 6.022$pc.
  - ‘`mm`’: 1 millimeter $= 2.845$pt.
  - ‘`ex`’: height of a small ‘x’ in the current font.
  - ‘`em`’: width of capital ‘M’ in the current font.
  - ‘`mu`’: 1 math-unit $= \frac{1}{18}$em (for positioning in math mode).

Weights and Measures

- A lot of markup in LaTeX is related to **where** and **how big**.
    - to help, TeX supports a whole raft of length related **units**.
- Generally written as a **number** followed by two unit characters:

  `0.5in, 5pt, 4em, 42mm`

    - ‘`pt`’: 1 point $= \frac{1}{72.27}$in $= 0.351$mm.
    - ‘`in`’: 1 inch $= 25.4$mm $= 72.27$pt $= 6.022$pc.
    - ‘`mm`’: 1 millimeter $= 2.845$pt.
    - ‘`ex`’: height of a small ‘x’ in the current font.
    - ‘`em`’: width of capital ‘M’ in the current font.
    - ‘`mu`’: 1 math-unit $= \frac{1}{18}$em (for positioning in math mode).

Page Layout

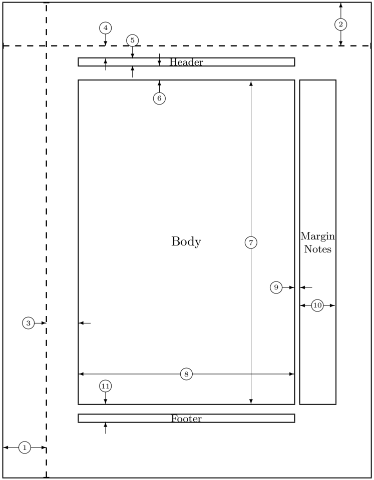
A typical A4 page using the ‘article’ document-class:

- diagram produced using the **layout** package and ‘\layout’ command.

layout.tex

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

\begin{document}

\layout

\end{document}
\endinput
```

→ layout.pdf



```
1   one inch + \hoffset          2   one inch + \voffset
3   \oddsidemargin = 54pt        4   \topmargin = 21pt
5   \headheight = 12pt           6   \headsep = 25pt
7   \textheight = 541pt          8   \textwidth = 360pt
9   \marginparsep = 10pt         10  \marginparwidth = 59pt
11  \footskip = 30pt                 \marginparpush = 5pt (not shown)
    \hoffset = 0pt                   \voffset = 0pt
    \paperwidth = 614pt              \paperheight = 794pt
```

Manipulating The Page Layout

- The standard LaTeX article leaves a lot of whitespace around a page.
  - typical when preparing an **article** for an academic journal, less so for general writing (e.g. your coursework).
- Easy to manipulate these in the document's **preamble**.

```
\addtolength{\oddsidemargin}{-1.0in}
\addtolength{\evensidemargin}{-1.0in}
\addtolength{\textwidth}{1.8in}
\addtolength{\textheight}{1.8in}
\addtolength{\topmargin}{-0.8in}
```

  - Note: **odd** and **even** refer to double-sided pages.
- `layout2.tex` → `layout2.pdf`
- And someone else already did this for us!

```
\usepackage{fullpage}
```

Writing a Document

- Enough with the background, time to write something!

```
\documentclass[a4paper,12pt]{article}
\usepackage{times}       % nicer font

\pagestyle{empty}

\begin{document}

\title{My Assessment}
\author{J.~Random User}
\date{\today}

\maketitle

For this assessment we were asked to find out about how
widgets are made inside a fictitious software system,
which was hard since such a thing doesn't really exist.
Anyhow, this is what I discovered:
```

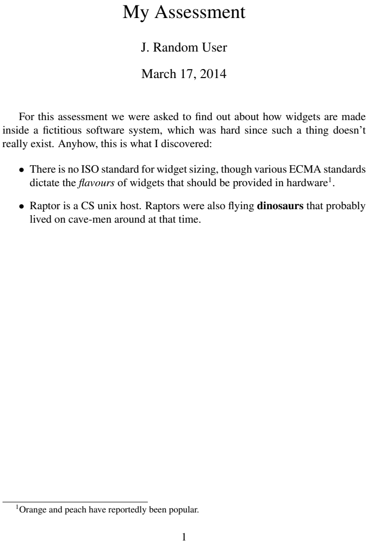
Writing a Document

```
\begin{itemize}
\item   There is no ISO standard for widget sizing,
        though various ECMA standards dictate the
        {\em flavours} of widgets that should be
        provided in hardware\footnote{Orange and
        peach have reportedly been popular.}.
\item   Raptor is a CS unix host.  Raptors were also
        flying {\bf dinosaurs} that probably lived on
        cave-men around at that time.
\end{itemize}

\end{document}
\endinput
```

Writing a Document

- use of the **times** package to get a particular font.
- commands to set **title**, etc. and `\maketitle` to typeset it.
- the `\today` command to get today's date.
- bulleted lists with the `itemize` environment; new items introduced with `\item`.
- emphasised text with '`{\em ...}`'.
- bold text with '`{\em ...}`'.
- footnotes with '`\footnote{...}`'.

`doc1.tex` → `doc1.pdf`



My Assessment

J. Random User

March 17, 2014

For this assessment we were asked to find out about how widgets are made inside a fictitious software system, which was hard since such a thing doesn't really exist. Anyhow, this is what I discovered:

- There is no ISO standard for widget sizing, though various ECMA standards dictate the *flavours* of widgets that should be provided in hardware[1].
- Raptor is a CS unix host. Raptors were also flying **dinosaurs** that probably lived on cave-men around at that time.

[1] Orange and peach have reportedly been popular.

1

Document Sectioning

- Most document types support a range of **sectioning** commands.
  - for article, mainly '`section`', '`subsection`' and '`subsubsection`'.
  - for book and report, '`chapter`' and '`part`' as well.
- These commands typically just appear throughout the document.
  - breaking up the content as appropriate.
- They can have complex **side-effects**:
  - e.g. inserting entries into a *table-of-contents*.

Document Sectioning

- Most document types support a range of **sectioning** commands.
  - for article, mainly ‘`section`’, ‘`subsection`’ and ‘`subsubsection`’.
  - for book and report, ‘`chapter`’ and ‘`part`’ as well.
- These commands typically just appear throughout the document.
  - breaking up the content as appropriate.
- They can have complex **side-effects**:
  - e.g. inserting entries into a *table-of-contents*.

Document Sectioning

- Most document types support a range of **sectioning** commands.
  - for article, mainly '`section`', '`subsection`' and '`subsubsection`'.
  - for book and report, '`chapter`' and '`part`' as well.
- These commands typically just appear throughout the document.
  - breaking up the content as appropriate.
- They can have complex **side-effects**:
  - e.g. inserting entries into a *table-of-contents*.

Document Sectioning

```latex
\documentclass[a4paper,12pt]{article}
\usepackage{times}

\pagestyle{empty}

\begin{document}

\title{My Assessment 2}
\author{J.~Random User\\{\small\tt jru2@kent.ac.uk}}
\date{\today}

\maketitle

\section{Introduction}
For this assessment we were asked to find out about how
widgets are made inside a fictitious software system,
which was hard since such a thing doesn't really exist.
Section~\ref{sec:results} describes what I found.
```

Document Sectioning

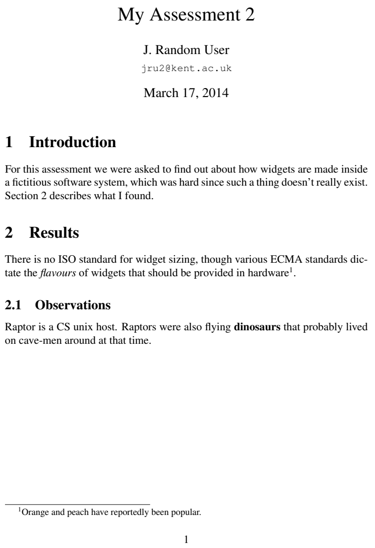
```
\section{Results}\label{sec:results}
There is no ISO standard for widget sizing,
though various ECMA standards dictate the
{\em flavours} of widgets that should be
provided in hardware\footnote{Orange and peach
have reportedly been popular.}.

\subsection{Observations}
Raptor is a CS unix host.  Raptors were also
flying {\bf dinosaurs} that probably lived on
cave-men around at that time.

\end{document}
\endinput
```

Document Sectioning

- the `\author` command takes a more complex argument:
  - `\\` forces a line break.
  - `\small` sets font size.
  - `\tt` sets typewriter font.
- various sectioning commands.
- labels placed with `\label{...}`.
- referenced with `\ref{...}`.
  - or page with `\pageref{..}`.

With this example, need to run pdflatex **twice** to resolve the cross-references.

doc2.tex → doc2.pdf



My Assessment 2

J. Random User
jru2@kent.ac.uk

March 17, 2014

1 Introduction

For this assessment we were asked to find out about how widgets are made inside a fictitious software system, which was hard since such a thing doesn't really exist. Section 2 describes what I found.

2 Results

There is no ISO standard for widget sizing, though various ECMA standards dictate the *flavours* of widgets that should be provided in hardware[1].

2.1 Observations

Raptor is a CS unix host. Raptors were also flying **dinosaurs** that probably lived on cave-men around at that time.

[1] Orange and peach have reportedly been popular.

1

Documentation, Help and Support

- Documentation for the vast range of LaTeX packages varies.
  - many come with neat, although sometimes lengthy, PDF and HTML.
  - often generated using something like **docbook**.
- Whatever question you have, chances are it's been answered already (somewhere out there).
  - Google is your friend: start the query with "TeX" or "LaTeX" to get relevant hits.
- For any serious LaTeX use, "The LaTeX Companion" [1].
  - numerous on-line introductions, guides, references, etc.
  - `http://tobi.oetiker.ch/lshort/lshort.pdf`

Documentation, Help and Support

- Documentation for the vast range of LaTeX packages varies.
  - many come with neat, although sometimes lengthy, PDF and HTML.
  - often generated using something like **docbook**.
- Whatever question you have, chances are it's been answered already (somewhere out there).
  - Google is your friend: start the query with "TeX" or "LaTeX" to get relevant hits.
- For any serious LaTeX use, "The LaTeX Companion" [1].
  - numerous on-line introductions, guides, references, etc.
  - `http://tobi.oetiker.ch/lshort/lshort.pdf`

Documentation, Help and Support

- Documentation for the vast range of LaTeX packages varies.
  - many come with neat, although sometimes lengthy, PDF and HTML.
  - often generated using something like **docbook**.
- Whatever question you have, chances are it's been answered already (somewhere out there).
  - Google is your friend: start the query with "TeX" or "LaTeX" to get relevant hits.
- For any serious LaTeX use, "The LaTeX Companion" [1].
  - numerous on-line introductions, guides, references, etc.
  - `http://tobi.oetiker.ch/lshort/lshort.pdf`

Dealing With Errors

- Before long, you will probably run into an **error**.
  - or more commonly, a **warning**.
- Amongst the messages generated in the output will be some reference to **where** the problem lies.
  - usually a line number in your source file.
  - common mistakes include typos in use of brackets (e.g. wrong ones), not escaping special characters, unbalanced environments.

The Second Lecture

- This is the second lecture `:-)`.
- The earlier lecture briefly covered:
  - introduction to LaTeX and its syntax (**commands**, **groups**, etc.).
  - document **structure** and some **style**.
  - **layout** of the page (margins, etc.).
  - TeX as a box-gluing machine.

Typesetting Maths

- One of the **motivations** for using LaTeX.
  - and one of the reasons Don Knuth invented TeX.
- Maths is integral to the point of TeX having a **maths-mode**.
  - though given its US origins, usually written/said **math-mode**.
- In the flow of text:
  - can switch in and out using '$' (dollar) or '\(' and '\)'.
  - looks much nicer in roman (serif) fonts than in these slides!
- Dedicated **environments**:
  - several environments are automatically math-mode.
  - common ones include '`equation`', '`align`', '`alignat`' and '`multline`'.
  - most have a *starred* version (e.g. '`equation*`') that supresses numbering.

Typesetting Maths

- One of the **motivations** for using LaTeX.
  - and one of the reasons Don Knuth invented TeX.
- Maths is integral to the point of TeX having a **maths-mode**.
  - though given its US origins, usually written/said **math-mode**.
- In the flow of text:
  - can switch in and out using ‘$’ (dollar) or ‘\(’ and ‘\)’.
  - looks much nicer in roman (serif) fonts than in these slides!
- Dedicated **environments**:
  - several environments are automatically math-mode.
  - common ones include ‘`equation`’, ‘`align`’, ‘`alignat`’ and ‘`multline`’.
  - most have a *starred* version (e.g. ‘`equation*`’) that supresses numbering.

Typesetting Maths

- One of the **motivations** for using LaTeX.
  - and one of the reasons Don Knuth invented TeX.
- Maths is integral to the point of TeX having a **maths-mode**.
  - though given its US origins, usually written/said **math-mode**.
- In the flow of text:
  - can switch in and out using '$' (dollar) or '\(' and '\)'.
  - looks much nicer in roman (serif) fonts than in these slides!
- Dedicated **environments**:
  - several environments are automatically math-mode.
  - common ones include '`equation`', '`align`', '`alignat`' and '`multline`'.
  - most have a *starred* version (e.g. '`equation*`') that supresses numbering.

Typesetting Maths

- One of the **motivations** for using LaTeX.
  - and one of the reasons Don Knuth invented TeX.
- Maths is integral to the point of TeX having a **maths-mode**.
  - though given its US origins, usually written/said **math-mode**.
- In the flow of text:
  - can switch in and out using ‘$’ (dollar) or ‘\(’ and ‘\)’.
  - looks much nicer in roman (serif) fonts than in these slides!
- Dedicated **environments**:
  - several environments are automatically math-mode.
  - common ones include ‘`equation`’, ‘`align`’, ‘`alignat`’ and ‘`multline`’.
  - most have a *starred* version (e.g. ‘`equation*`’) that supresses numbering.

Typesetting Maths

math1.tex

→ math1.pdf

1. The point (x, y) as a function of θ is defined as:

$$\mathbb{P}(x, y) = (R \times \sin(\theta), R \times \cos(\theta)) \tag{1}$$

2. The odd looking 'P', $\mathbb{P}$ is from the *blackboard-bold* set.

$$\mathbb{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$$

```
\usepackage{amsmath}        % extra AMS packages in
\usepackage{amsfonts}       % the preamble

\begin{enumerate}
\item The point $(x,y)$ as a function of $\theta$
      is defined as:

      \begin{equation}
          \mathbb{P}(x,y) = (R \times \mathrm{sin}(\theta),
              R \times \mathrm{cos}(\theta))
      \end{equation}
\end{enumerate}
```

Inline math here: $\mathbb{P}(x, y) = (R \times \sin(\theta), R \times \cos(\theta))$.

Typesetting Maths

- The package **amsmath** defines an assorted set of macros for common mathematical typesetting.
  - the package **amsfonts** makes available the *blackboard-bold* font.
  - AMS $=$ American Mathematical Society.
- Ordinary characters appear **italicized** in math-mode.
  - more noticeable in serif (Roman) fonts than sans-serif ones.
- The **equation** environment is used here (`\begin{equation}`).
  - single formula, split over two lines in the input.
- Greek letters available with `\alpha` (α), `\beta` (β), ...
  - multiply symbol with `\times`.
  - ordinary text (in a Roman font) with `\mathrm{...}`.

Typesetting Maths

- The package **amsmath** defines an assorted set of macros for common mathematical typesetting.
  - the package **amsfonts** makes available the *blackboard-bold* font.
  - AMS = American Mathematical Society.
- Ordinary characters appear **italicized** in math-mode.
  - more noticeable in serif (Roman) fonts than sans-serif ones.
- The **equation** environment is used here (`\begin{equation}`).
  - single formula, split over two lines in the input.
- Greek letters available with `\alpha` (α), `\beta` (β), ...
  - multiply symbol with `\times`.
  - ordinary text (in a Roman font) with `\mathrm{...}`.

Typesetting Maths

- The package **amsmath** defines an assorted set of macros for common mathematical typesetting.
  - the package **amsfonts** makes available the *blackboard-bold* font.
  - AMS $=$ American Mathematical Society.
- Ordinary characters appear **italicized** in math-mode.
  - more noticeable in serif (Roman) fonts than sans-serif ones.
- The **equation** environment is used here (`\begin{equation}`).
  - single formula, split over two lines in the input.
- Greek letters available with `\alpha` (α), `\beta` (β), ...
  - multiply symbol with `\times`.
  - ordinary text (in a Roman font) with `\mathrm{...}`.

Typesetting Maths

- The package **amsmath** defines an assorted set of macros for common mathematical typesetting.
  - the package **amsfonts** makes available the *blackboard-bold* font.
  - AMS = American Mathematical Society.
- Ordinary characters appear **italicized** in math-mode.
  - more noticeable in serif (Roman) fonts than sans-serif ones.
- The **equation** environment is used here (`\begin{equation}`).
  - single formula, split over two lines in the input.
- Greek letters available with `\alpha` (α), `\beta` (β), ...
  - multiply symbol with `\times`.
  - ordinary text (in a Roman font) with `\mathrm{...}`.

Making It Nice

- At the moment we have:

$$\mathbb{P}(x, y) = (R \times \sin(\theta), R \times \cos(\theta))$$

- Could use some **extra space**:

- And maybe some larger brackets on the outside:

`math1-d.tex` → `math1-d.pdf`

Making It Nice

- At the moment we have:

$$\mathbb{P}(x,y) = (R \times \sin(\theta), R \times \cos(\theta))$$

- Could use some **extra space**:

```
\mathbb{P}\,(x,y) = (R \times \mathrm{sin}(\theta),
        \: R \times \mathrm{cos}(\theta))
```

$$\mathbb{P}\,(x,y) = (R \times \sin(\theta),\: R \times \cos(\theta))$$

- And maybe some larger brackets on the outside:

`math1-d.tex` → `math1-d.pdf`

Making It Nice

- At the moment we have:

$$\mathbb{P}(x,y) = (R \times \sin(\theta), R \times \cos(\theta))$$

- Could use some **extra space**:

```
\mathbb{P}\,(x,y) = (R \times \mathrm{sin}(\theta),
       \: R \times \mathrm{cos}(\theta))
```

$$\mathbb{P}\,(x,y) = (R \times \mathrm{sin}(\theta),\: R \times \mathrm{cos}(\theta))$$

- And maybe some larger brackets on the outside:

```
\mathbb{P}\,(x,y) = \bigl(R \times \mathrm{sin}(\theta),
       \: R \times \mathrm{cos}(\theta)\bigr)
```

$$\mathbb{P}\,(x,y) = \bigl(R \times \mathrm{sin}(\theta),\: R \times \mathrm{cos}(\theta)\bigr)$$

math1-d.tex → math1-d.pdf

Superscripts and Subscripts

- Have many uses when writing maths.
    - when writing about **code** (arrays).
    - typesetting above and below things like *sum* or *integrate* ($\int$).
    - subscripts typeset with the **underscore**, superscripts with **hat** (caret).

```
\begin{equation}
    \mathit{SumOf}(x) = \sum_{i=0}^{N-1} x_{i},\qquad
    \bar{x} = \frac{\sum_{i=0}^{N-1} x_{i}}{N}
\end{equation}
```

$$\mathit{SumOf}(x) = \sum_{i=0}^{N-1} x_{i},\qquad \bar{x} = \frac{\sum_{i=0}^{N-1} x_{i}}{N} \tag{1}$$

math2.tex → math2.pdf

More Maths Macros

- Extra spacing:
  - `\quad` gives 1 em of spacing; `\qquad` twice this.
  - not used here, but `\!` is a small **negative** space.
- Text typesetting: `\mathit{...}` for italicized text.
  - `\text{...}` can be used for general (non-math-mode) text.
- Symbols:
  - `\sum` for the “big sum”, $\sum$, `\prod` for product, $\prod$.
  - that take an optional **subscript** and **superscript**:

`\sum_{i=0}^{N}` $\Longrightarrow$ $\displaystyle\sum_{i=0}^{N}$

- Fractions with: `\frac{top}{bot}`, e.g. $\frac{x+1}{x-1}$
  - drops general font size for top (numerator) and bottom (denominator) — why ‘sum’ shows differently.

`x_{i+1} = \frac{x_i + 1}{x_i - 1}` $\Longrightarrow$ $\displaystyle x_{i+1} = \frac{x_i + 1}{x_i - 1}$

More Maths Macros

- Extra spacing:
  - `\quad` gives 1 em of spacing; `\qquad` twice this.
  - not used here, but `\!` is a small **negative** space.
- Text typesetting: `\mathit{...}` for italicized text.
  - `\text{...}` can be used for general (non-math-mode) text.
- Symbols:
  - `\sum` for the "big sum", $\sum$, `\prod` for product, $\prod$.
  - that take an optional **subscript** and **superscript**:

$$\texttt{\textbackslash sum\_\{i=0\}\^{}\{N\}} \implies \sum_{i=0}^{N}$$

- Fractions with: `\frac{top}{bot}`, e.g. $\frac{x+1}{x-1}$
  - drops general font size for top (numerator) and bottom (denominator) — why 'sum' shows differently.

$$\texttt{x\_\{i+1\} = \textbackslash frac\{x\_i + 1\}\{x\_i - 1\}} \implies x_{i+1} = \frac{x_i + 1}{x_i - 1}$$

More Maths Macros

- Extra spacing:
  - `\quad` gives 1 em of spacing; `\qquad` twice this.
  - not used here, but `\!` is a small **negative** space.
- Text typesetting: `\mathit{...}` for italicized text.
  - `\text{...}` can be used for general (non-math-mode) text.
- Symbols:
  - `\sum` for the “big sum”, $\sum$, `\prod` for product, $\prod$.
  - that take an optional **subscript** and **superscript**:

$$\texttt{\textbackslash sum\_\{i=0\}\^{}\{N\}} \quad \Longrightarrow \quad \sum_{i=0}^{N}$$

- Fractions with: `\frac{top}{bot}`, e.g. $\frac{x+1}{x-1}$
  - drops general font size for top (numerator) and bottom (denominator) — why ‘sum’ shows differently.

$$\texttt{x\_\{i+1\} = \textbackslash frac\{x\_i + 1\}\{x\_i - 1\}} \quad \Longrightarrow \quad x_{i+1} = \frac{x_i + 1}{x_i - 1}$$

More Maths Macros

- Extra spacing:
  - `\quad` gives 1 em of spacing; `\qquad` twice this.
  - not used here, but `\!` is a small **negative** space.
- Text typesetting: `\mathit{...}` for italicized text.
  - `\text{...}` can be used for general (non-math-mode) text.
- Symbols:
  - `\sum` for the "big sum", $\sum$, `\prod` for product, $\prod$.
  - that take an optional **subscript** and **superscript**:

`\sum_{i=0}^{N}` $\Longrightarrow$ $$\sum_{i=0}^{N}$$

- Fractions with: `\frac{top}{bot}`, e.g. $\frac{x+1}{x-1}$
  - drops general font size for top (numerator) and bottom (denominator) — why 'sum' shows differently.

`x_{i+1} = \frac{x_i + 1}{x_i - 1}` $\Longrightarrow$ $$x_{i+1} = \frac{x_i + 1}{x_i - 1}$$

Typesetting Sets and Related

math3.tex

→ math3.pdf

$$\begin{align*} \mathit{Evens} &= \bigl\{x\,|\,(\exists n \in \mathbb{N}) \wedge (x = 2n)\bigr\}\\ P &= (A \cap B) \cup (A \cap C) = A \cap (B \cup C) \end{align*}$$

```
\begin{align*}
       \mathit{Evens} &= \bigl\{x\,|\,(\exists n \in
              \mathbb{N}) \wedge (x = 2n)\bigr\}\\
       P &= (A \cap B) \cup (A \cap C) = A \cap (B \cup C)
\end{align*}
```

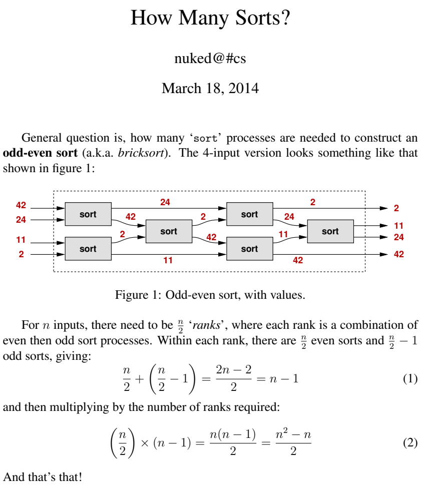
- The **align** environment uses the ampersand (&) to align.
  - with the sans-serif fonts here:

$$\begin{align*} \mathit{Evens} &= \bigl\{x\,|\,(\exists n \in \mathbb{N}) \wedge (x = 2n)\bigr\}\\ P &= (A \cap B) \cup (A \cap C) = A \cap (B \cup C) \end{align*}$$

  - the commands `\cap`, `\cup` and `\wedge` typeset the symbols shown.
  - also: `\setminus` ($\setminus$), `\vee` ($\vee$), `\neq` ($\neq$).
- Note use of `\bigl\{` and `\bigl\}` for larger curly brackets.

Captioned Figures

```
... front-matter

... something like that shown
in figure~\ref{fig:sort}:

\begin{figure}[hbt]
  \centering
  \includegraphics[scale=.6]%
    {oddevensort.pdf}
  \caption{Odd-even sort,
    with values.}
  \label{fig:sort}
\end{figure}

For $n$ inputs, there ...
\begin{equation}
  ...
\end{equation}
```



How Many Sorts?

nuked@#cs

March 18, 2014

General question is, how many ‘sort’ processes are needed to construct an **odd-even sort** (a.k.a. *bricksort*). The 4-input version looks something like that shown in figure 1:

Figure 1: Odd-even sort, with values.

For n inputs, there need to be $\frac{n}{2}$ ‘*ranks*’, where each rank is a combination of even then odd sort processes. Within each rank, there are $\frac{n}{2}$ even sorts and $\frac{n}{2} - 1$ odd sorts, giving:

$$\frac{n}{2} + \left(\frac{n}{2} - 1\right) = \frac{2n-2}{2} = n - 1 \tag{1}$$

and then multiplying by the number of ranks required:

$$\left(\frac{n}{2}\right) \times (n-1) = \frac{n(n-1)}{2} = \frac{n^2 - n}{2} \tag{2}$$

And that’s that!

math4.tex → math4.pdf

A Few More Maths

math5.tex

→ math5.pdf

Beta-reduction in the lambda-calculus is defined by substitution:

$$(\lambda v.E)\ x \quad\longrightarrow_{\beta}\quad E[x/v] \tag{1}$$

$$(\lambda vw.E)\ x \quad\longrightarrow_{\beta}\quad \lambda w.(E[x/v]) \tag{2}$$

where x may be any lambda term (variable, function or application). E.g.:

$$(\lambda xy.(y\,(x\,x\,y)))\:(\lambda xy.(y\,(x\,x\,y))) \\ \longrightarrow_{\beta} \lambda y.(y\,((\lambda xy.(y\,(x\,x\,y)))\:(\lambda xy.(y\,(x\,x\,y)))\, y)) \tag{3}$$

```
\begin{align}
  (\lambda v.E)\ x &\quad\longrightarrow_{\beta}
        \quad E[x/v]\\
  (\lambda vw.E)\ x &\quad\longrightarrow_{\beta}
        \quad \lambda w.(E[x/v])
\end{align}
\begin{multline}
  (\lambda xy.(y\,(x\,x\,y)))\:(\lambda
    xy.(y\,(x\,x\,y)))\\
  \longrightarrow_{\beta} \lambda y.(y\,
    ((\lambda xy.(y\,(x\,x\,y)))\:(\lambda
    xy.(y\,(x\,x\,y)))\, y))
\end{multline}
```

Simulating Typed Text

- A variety of **environments** (some from packages) for this:
  - LaTeX's default 'verbatim' environment:

```
\begin{verbatim}
This is some text that will come out
exactly as is, including specials ^_^.
\end{verbatim}
```

```
This is some text that will come out
exactly as is, including specials ^_^.
```

- The 'alltt' environment (and package) is a bit more featured.
  - **backslash** and **braces** keep their existing meaning.
  - makes it possible to tweak the style: emphasized, blue, etc.

Formatted Code Listings

- We're used to syntax highlighting in our code **editors**.
  - and now we can do it in **printed** stuff too!
- The **listings** package.
  - highly configurable way of typesetting code listings.
  - what is used in these slides.
- Not without its problems though.
  - the 'lstlisting' environment **does not** co-exist with Beamer's step-by-step slides.
  - crude solution: generate the required listings as stand-alone PDFs and include as images.

See: http://en.wikibooks.org/wiki/LaTeX/Source_Code_Listings

Formatted Code Listings

- We're used to syntax highlighting in our code **editors**.
  - and now we can do it in **printed** stuff too!
- The **listings** package.
  - highly configurable way of typesetting code listings.
  - what is used in these slides.
- Not without its problems though.
  - the '`lstlisting`' environment **does not** co-exist with Beamer's step-by-step slides.
  - crude solution: generate the required listings as stand-alone PDFs and include as images.

See: `http://en.wikibooks.org/wiki/LaTeX/Source_Code_Listings`

Formatted Code Listings

- We're used to syntax highlighting in our code **editors**.
  - and now we can do it in **printed** stuff too!
- The **listings** package.
  - highly configurable way of typesetting code listings.
  - what is used in these slides.
- Not without its problems though.
  - the '`lstlisting`' environment **does not** co-exist with Beamer's step-by-step slides.
  - crude solution: generate the required listings as stand-alone PDFs and include as images.

See: `http://en.wikibooks.org/wiki/LaTeX/Source_Code_Listings`

Using BibTeX

- BibTeX is **not** LaTeX, but a mostly separate program.
  - used in conjunction with LaTeX to add references to a document.
- A separate **bibliography** file is kept.
  - contains details of things that can be referenced (e.g. article, book, web-site, private communication).
  - e.g. `test.bib` (some stuff I scooped off ACM's digital library).
- In the document:
  - use `\cite{...}` to insert a citation.
  - the command `\bibliographystyle{...}` selects a particular style, e.g. '`unsrt`' (unsorted — in document order).
  - bibliography style files (`.bst`) are essentially stack machine programs that format entries from a `.bib` file.
  - the command `\bibliography{test}` to select and use '`test.bib`'.

Using BibTeX

- BibTeX is **not** LaTeX, but a mostly separate program.
  - used in conjunction with LaTeX to add references to a document.
- A separate **bibliography** file is kept.
  - contains details of things that can be referenced (e.g. article, book, web-site, private communication).
  - e.g. `test.bib` (some stuff I scooped off ACM's digital library).
- In the document:
  - use `\cite{...}` to insert a citation.
  - the command `\bibliographystyle{...}` selects a particular style, e.g. '`unsrt`' (unsorted — in document order).
  - bibliography style files (`.bst`) are essentially stack machine programs that format entries from a `.bib` file.
  - the command `\bibliography{test}` to select and use '`test.bib`'.

Using BibTeX

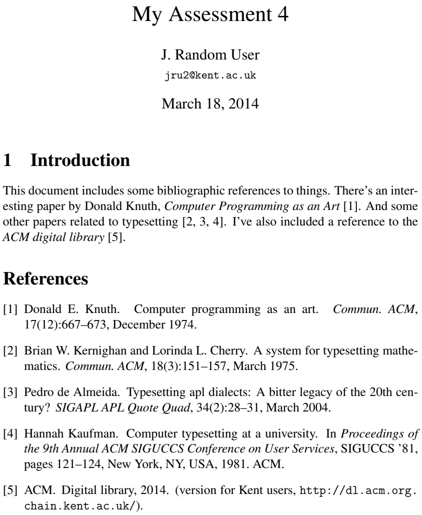
- BibTeX is **not** LaTeX, but a mostly separate program.
  - used in conjunction with LaTeX to add references to a document.
- A separate **bibliography** file is kept.
  - contains details of things that can be referenced (e.g. article, book, web-site, private communication).
  - e.g. `test.bib` (some stuff I scooped off ACM's digital library).
- In the document:
  - use `\cite{...}` to insert a citation.
  - the command `\bibliographystyle{...}` selects a particular style, e.g. '`unsrt`' (unsorted — in document order).
  - bibliography style files (`.bst`) are essentially stack machine programs that format entries from a `.bib` file.
  - the command `\bibliography{test}` to select and use '`test.bib`'.

Using BibTeX

```
\usepackage{natbib}
\usepackage{hyperref}

...  I've also included a
reference to the {\em ACM
digital library}~%
\cite{DL:2014}.

\bibliographystyle{unsrt}
\bibliography{test}
```

doc4.tex → doc4.pdf

(using the **hyperref** package means that the generated PDF will contain links where appropriate)



My Assessment 4

J. Random User
jru2@kent.ac.uk

March 18, 2014

1 Introduction

This document includes some bibliographic references to things. There's an interesting paper by Donald Knuth, *Computer Programming as an Art* [1]. And some other papers related to typesetting [2, 3, 4]. I've also included a reference to the *ACM digital library* [5].

References

[1] Donald E. Knuth. Computer programming as an art. *Commun. ACM*, 17(12):667–673, December 1974.

[2] Brian W. Kernighan and Lorinda L. Cherry. A system for typesetting mathematics. *Commun. ACM*, 18(3):151–157, March 1975.

[3] Pedro de Almeida. Typesetting apl dialects: A bitter legacy of the 20th century? *SIGAPL APL Quote Quad*, 34(2):28–31, March 2004.

[4] Hannah Kaufman. Computer typesetting at a university. In *Proceedings of the 9th Annual ACM SIGUCCS Conference on User Services*, SIGUCCS '81, pages 121–124, New York, NY, USA, 1981. ACM.

[5] ACM. Digital library, 2014. (version for Kent users, http://dl.acm.org.chain.kent.ac.uk/).

Using BibTeX — In Practice

- A minor inconvenience sometimes, but may need to run `pdflatex` or equivalent **several** times:
  1. `pdflatex`: collects citations and other relevant information in the '`.aux`' file.
  2. `bibtex`: uses the `.aux` information to extract and format particular entries using the appropriate style. This results in a `.bbl` file that contains the formatted (in LaTeX) entries.
  3. `pdflatex`: pick up the formatted entries and drop them into the document, assigning numbers/names.
  4. `pdflatex`: pick up the final numbers/names in citation commands.

Tabular Material

- The ‘`tabular`’ environment — and all its horridness.

Box Tricks

- Boxing commands, e.g. ‘`\raisebox`’, ‘`\mbox`’, parbox, etc.

References

Frank Mittelbach and Michel Goossens.
The LaTeX Companion.
Pearson Education, Inc., 2 edition, 2004.
ISBN: 0-201-36299-6.