1 Raster and Vector Graphics

There are two types of computer graphics, vector graphics and raster graphics. **Raster graphics** list the color of each pixel of a grid. **Vector graphics**, in contrast, are a description of how to draw the image. To do this, they contain lists of commands like "make a red disk with radius 1 cm in the middle of the images". Because pixels are not explicitly mentioned, vector graphics are resolution-independent. In practice, this means you can zoom into vector images indefinitely and they will look the same. Raster images, conversely, will look pixelated after zooming in. In general, vector graphics will look better and have a smaller filesize. However, they cannot be used to display some content which is inherently pixelized, such as digital photographs. Vector graphics should be used whenever possible.

Files with the extensions `.jpg`, `.gif`, `.png` or `.tiff` are raster graphics. Files with the extension `.eps` or `.svg` are vector. Some file types, such as `.pdf` can have both formats simultaneously.

2 Figures

To put images into LaTeX, first add the line `\usepackage{graphicx}` to your preamble. To add an indivudal image to your document, first move the image into the same directory as your `.tex` file. If the image is called `vector.eps`, use the command

Code

```
\usepackage{graphicx}
.
.
.
\includegraphics{vector.eps}
```

Output

The `includegraphics` command takes one argument, the path to the image file relative to your .tex file. This can be used to place images into a TeX file from other directories. For instance, if you a directory called `images` in the same directory as your .tex file, then you can include an image inside the `images` folder with the command `\includegraphics{images/vector.eps}`.

If you want to be a little more fancy, and get effects like captions you can use the `figure` environment.

Code

```
\usepackage{graphicx}
.
.
.
\begin{figure}[h]
\center
\includegraphics{vector.eps}
\caption{An example image}
\label{fig:example_image}
\end{figure}
You can the reference images like this: Image \ref{fig:example_image
    }.
```

Output

Figure 1: An example image

You can the reference images like this: Figure 1.

There is an optional argument for the `figure` environment which controls where the figure is placed on the page. It has three possibilies: `[h]`, for "here" on the page, `[b]` for the bottom of the page or `[t]` for the top of the page.

Positioning images can be quite tricky in LaTeX. Since images are automatically positioned, it is hard to predict where they will appear. If the h,t, or b option doesn't put it where you want, then try messing with the size of the image, or where you placed the `figure` environment in your code. With references, you can direct the reader to a figure which is on a different page than the revelent text — but try not to put it too far away. There is a good rule for doing figure positioning as easily as possible:

Position figures only after the text of your document is complete.

You may want to try the `wrapfig` package to images into text so that the text wraps around the image. This is often useful for small images that don't need a blank horizontal band, or figures that don't need much emphasis.

3 Floats

The figure environment is an example of a type of environment called a `float`. Floating environments are different from other environments in that they are automatically positioned on the page in the typesetted version, usually not in the same location as in the `.tex` file.

The other commonly used float environment is the `table` environment. This is *not* the same as the `tabular` environment. The `tabular` environment, which was introduced in the second workshop, is for making tables. The `table` environment is for positioning them. It uses the same basic syntax as the `figure` environment.

Code

```
\begin{table}
\caption{An example table}
\begin{tabular}[t]{ll}
        \textbf{Animal} & \textbf{Color}\\ \hline
        Elephant & Grey\\
        Lion & Yellow\\
        Blue Whale & Blue
\end{tabular}
\label{tb:example_table}
\end{table}
```

Output

Animal	**Color**
Elephant	Grey
Lion	Yellow
Blue Whale	Blue

Table 1: An example table

4 Counters & Advanced Lists

Counters are LaTeX's internal way of keeping track of things that need to be counted. There are numerous examples of counters that you've already seen. Any time something is automatically numbered, there is a counter for it. A table of some of the most common counters is below. A more complete list can be found at http://en.wikibooks.org/wiki/LaTeX/Counters.

Environment/Command	**Counter Name**
Section Numbering	`section`
Subsection Numbering	`subsection`
Page Numbering	`page`
Enumerated List (top level)	`enumi`
Enumerated List (first nested level)	`enumii`
Enumerated List (second nested level)	`enumiii`
Equation Numbering	`equation`
Figure Numbering	`figure`

So there are lots of counters in LaTeX. Why do they matter if they're always automatic? Sometimes you want to manually change the behaviors of counters. There are two ways to do this.

1. Change the numerical value of a counter.
2. Change the way the counter is displayed. For instance, have the count displayed in roman numerals (I, II, III ...) instead of arabic ones (1, 2, 3...).

The commands for changing and displaying the numerical value of counters are below.

Command	Output
`\theCOUNTER`	Displays the numerical value of the counter. For instance, `\thepage` will display 4. In programming terms, this casts the counter as a string and prints it.
`\value{COUNTER}`	Returns the value of the counter as a number, which cannot be printed, but only used by other counters or in calculations.
`\setcounter{COUNTER}{VALUE}`	This sets the counter to any numerical value. For instance, `\setcounter{page}{30}` will make page numbering jump to 30.

Changing the displayed style of conters is more difficult. To do this, one must rewrite the internal LaTeX command used to display the counter's value. Because these commands are already defined, this must be done using the `\renewcommand` command. This is similar to the `\newcommand` command discussed in the second workshop, except it overwrites already existing commands. The command that must be modified is usually called `\labelCOUNTER`. An example of changing the style of a enumerated list is below.

Code

```
Default enumerated list style.
\begin{enumerate}
        \item The first item.
        \item The second item.
        \item The third item.
\end{enumerate}
Modified list style.
\renewcommand{\labelenumi}{(\Alph{enumi})}
\begin{enumerate}
        \item The first item.
        \item The second item.
        \item The third item.
\end{enumerate}
```

Output

Default enumerated list style.

1. The first item.
2. The second item.
3. The third item.

Modified list style.

(A) The first item.

(B) The second item.

(C) The third item.

There are several useful commands for displaying different numbering styles. Each takes the name of the counter as the argument, and output the value of the counter in a given style. Of course, these can also be used outside the `\renewcommand` in normal text.

Command	Style
`\arabic`	1, 2, 3...
`\alph`	a, b, c...
`\Alph`	A, B, C...
`\roman`	i, ii, iii...
`\Roman`	I, II, III...

5 Miscellaneous

There are lots of other neat formatting tools in LaTeX. The output of virtually every command can be customized, and there are packages to provide any feature you will ever need. Just as an example, let's go through one more command and one more package.

5.1 Footnotes

Footnotes are extremely easy to do in LaTeX.

Code

```
If you have a bunch of text and want to add a footnote, all you need
    to do is this.\footnote{I'm a footnote!}This is another footnote.\
    footnote{Another footnote!}
```

Output

If you have a bunch of text and want to add a footnote, all you need to do is this.[1]This is another footnote.[2]

Of course, footnotes are another example of a counter. As you might guess, they use the `footnote` counter.

5.2 The `hyperref` Package

The `hyperref` package is an extremely useful package for adding clickable links to a TeX document. It's usage is also very simple.

Code

```
If you want to add a link to your document, use the command \texttt{
    href} like this: \href{http://www.brown.edu}{Brown's Website}.
```

Output

If you want to add a link to your document, use the command `href` like this: Brown's Website.

Additionally, the `href` package will automatically make all references clickable. Every time you use `\ref` or `\footnote`, the reference number can be clicked to jump inside the document to that place.

[2]I'm a footnote!
[2]Another footnote!