CM0133 Internet Computing

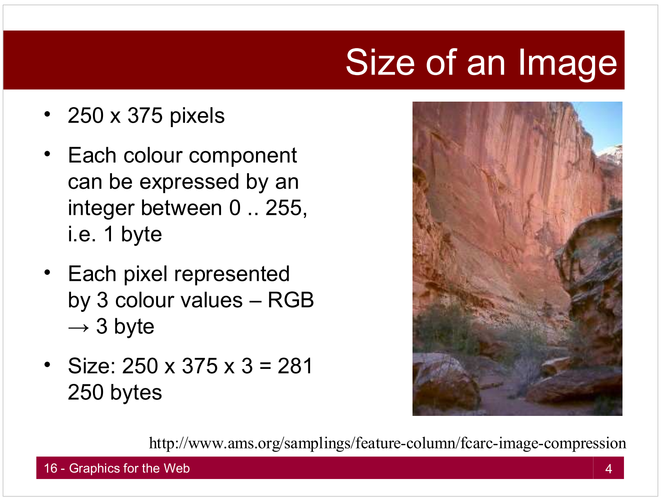
16. Graphics for the Web

Objectives

- Learn about the various possibilities to use and create graphics for and on web pages
  - Vector graphics (Flash, SVG)
  - Raster Graphics (JPEG, GIF, PNG)
- Appreciate different graphic formats and ways to create them
- Learn how to animate graphics

Issues with Images on the Web

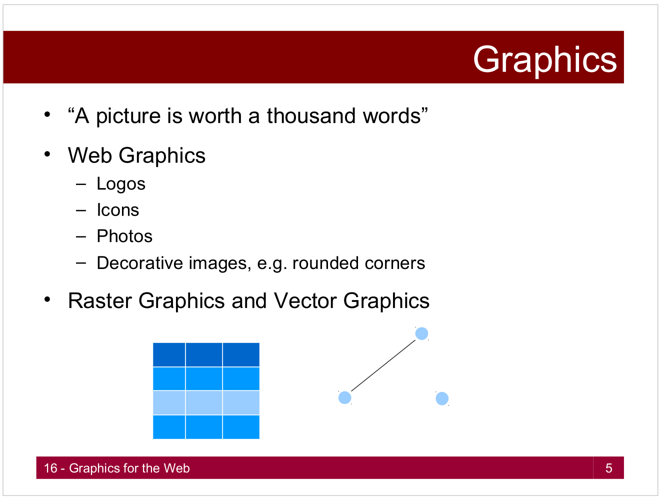
- Image sizes
  - Sizes need to be kept at a minimum
  - Sydney Olympics took 230 000 000 page views in 16 days
  - If a page is 500kb bigger than necessary
  - 115 Terra Bytes extra traffic!!
- Image Quality
  - Different image types are good for different applications
  - We need to choose carefully to balance image quality vs size
  - Does it need to scale?



Size of an Image

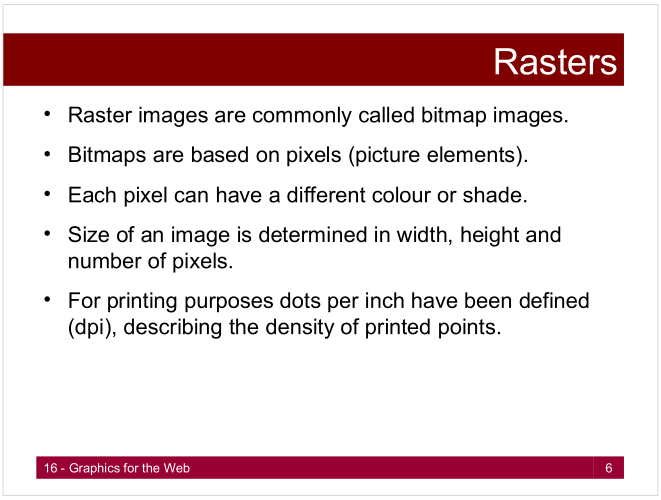
- 250 x 375 pixels
- Each colour component can be expressed by an integer between 0 .. 255, i.e. 1 byte
- Each pixel represented by 3 colour values – RGB → 3 byte
- Size: 250 x 375 x 3 = 281 250 bytes

http://www.ams.org/samplings/feature-column/fcarc-image-compression



Graphics

- “A picture is worth a thousand words”
- Web Graphics
  - Logos
  - Icons
  - Photos
  - Decorative images, e.g. rounded corners
- Raster Graphics and Vector Graphics

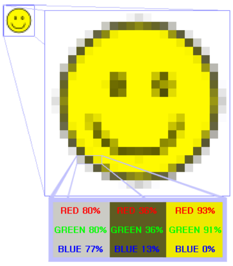


Rasters

- Raster images are commonly called bitmap images.
- Bitmaps are based on pixels (picture elements).
- Each pixel can have a different colour or shade.
- Size of an image is determined in width, height and number of pixels.
- For printing purposes dots per inch have been defined (dpi), describing the density of printed points.

Bitmap Images

- A data structure representing a grid of pixels (picture element)
- Each pixel will have colour information
- This can be b&w, greyscale, rgb, rgba etc



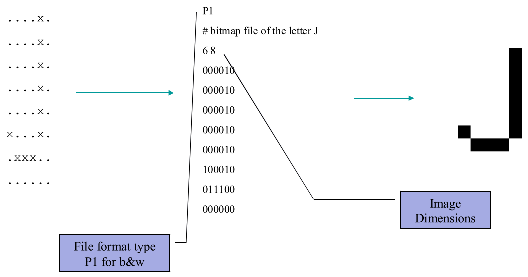
RED 80%	RED 36%	RED 93%
GREEN 80%	GREEN 36%	GREEN 91%
BLUE 77%	BLUE 13%	BLUE 0%

Taken from wikipedia

Bitmap Image Types

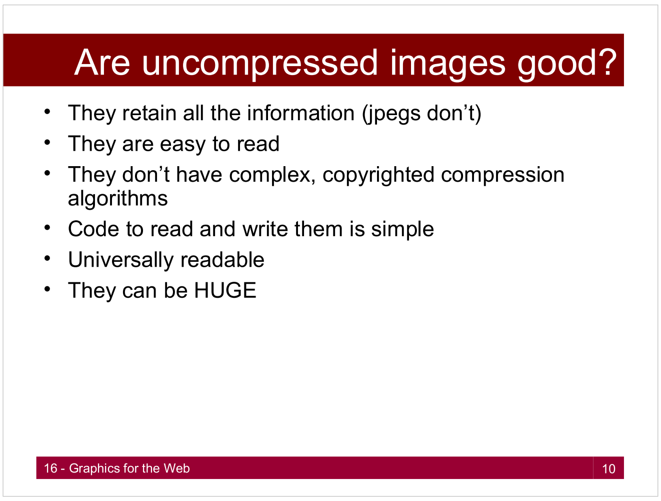
- Bitmap Images come in two variants
  - Uncompressed
    - Windows Bitmap (BMP), Portable Bitmap (PBM)…..
  - Compressed
    - Graphics Interchange Format (GIF), Joint Photographics Experts Group (JPEG), Tagged Image File Format(TIF), Portable Networks Graphic (PNG) ……

Uncompressed Bitmap Images



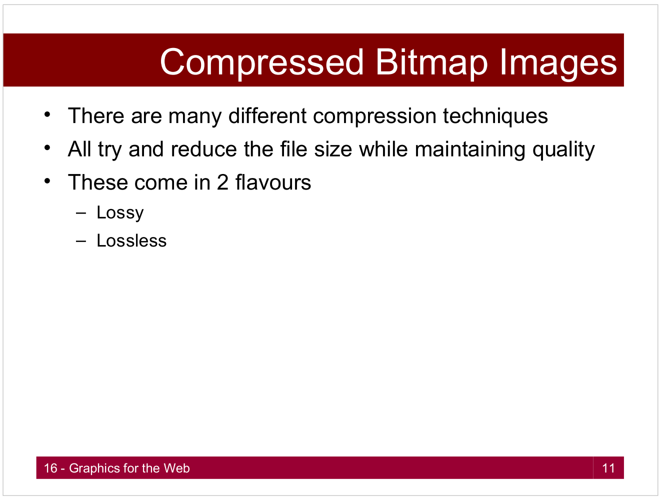
```
....x.
....x.
....x.
....x.
....x.
x...x.
.xxx..
......
```

```
P1
# bitmap file of the letter J
6 8
000010
000010
000010
000010
000010
100010
011100
000000
```



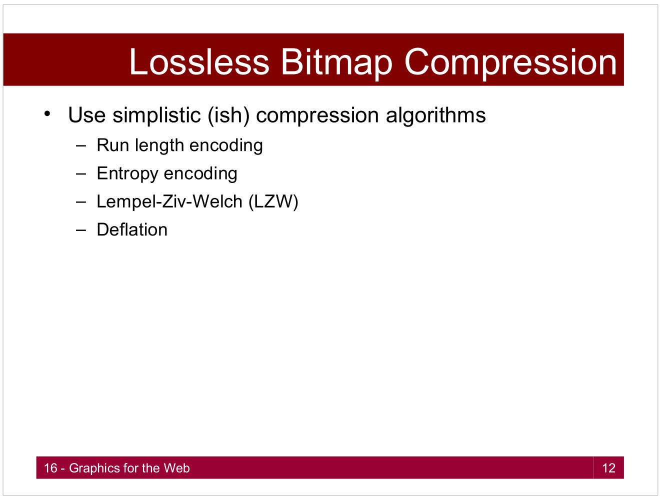
Are uncompressed images good?

- They retain all the information (jpegs don’t)
- They are easy to read
- They don’t have complex, copyrighted compression algorithms
- Code to read and write them is simple
- Universally readable
- They can be HUGE



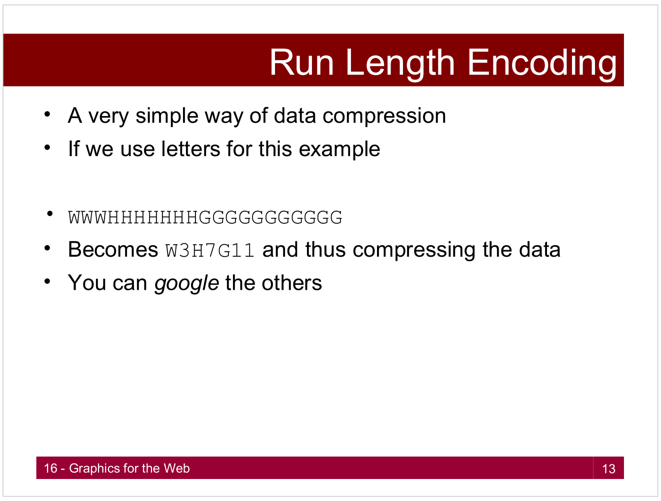
Compressed Bitmap Images

- There are many different compression techniques
- All try and reduce the file size while maintaining quality
- These come in 2 flavours
  - Lossy
  - Lossless



Lossless Bitmap Compression

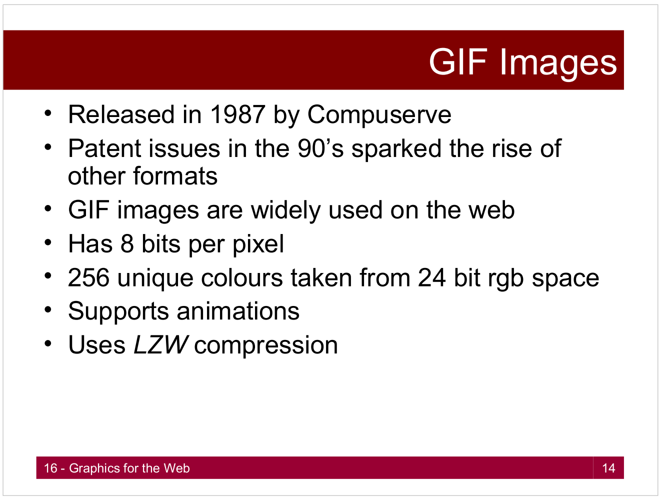
- Use simplistic (ish) compression algorithms
  - Run length encoding
  - Entropy encoding
  - Lempel-Ziv-Welch (LZW)
  - Deflation



Run Length Encoding

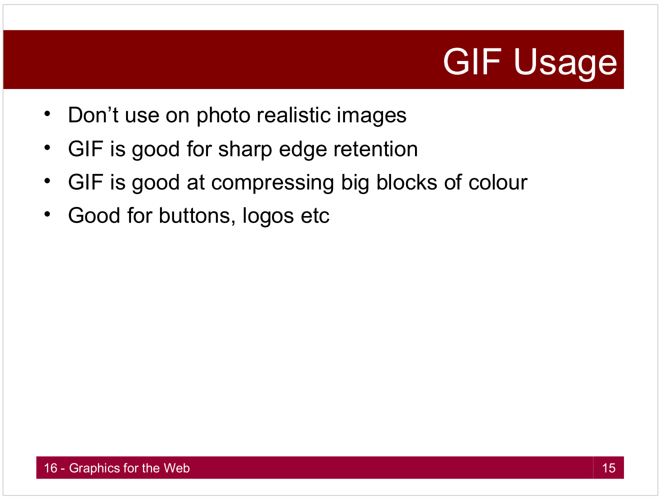
- A very simple way of data compression
- If we use letters for this example

- `WWWHHHHHHHGGGGGGGGGGG`
- Becomes `W3H7G11` and thus compressing the data
- You can *google* the others



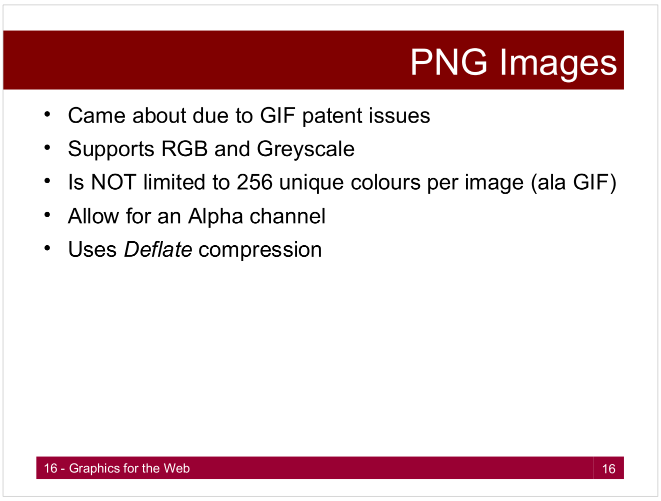
GIF Images

- Released in 1987 by Compuserve
- Patent issues in the 90’s sparked the rise of other formats
- GIF images are widely used on the web
- Has 8 bits per pixel
- 256 unique colours taken from 24 bit rgb space
- Supports animations
- Uses *LZW* compression



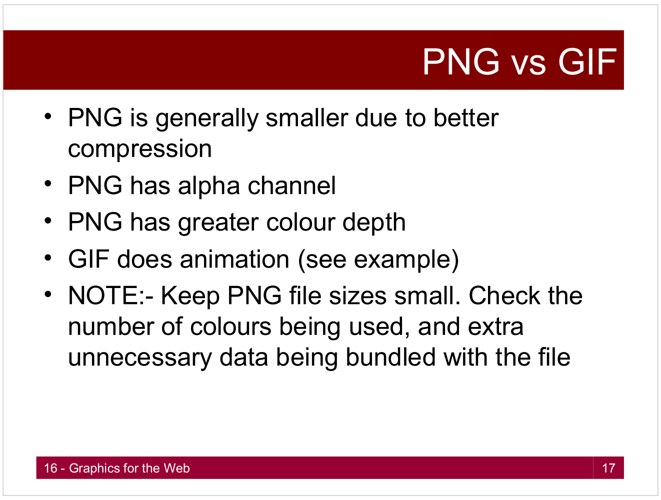
GIF Usage

- Don’t use on photo realistic images
- GIF is good for sharp edge retention
- GIF is good at compressing big blocks of colour
- Good for buttons, logos etc



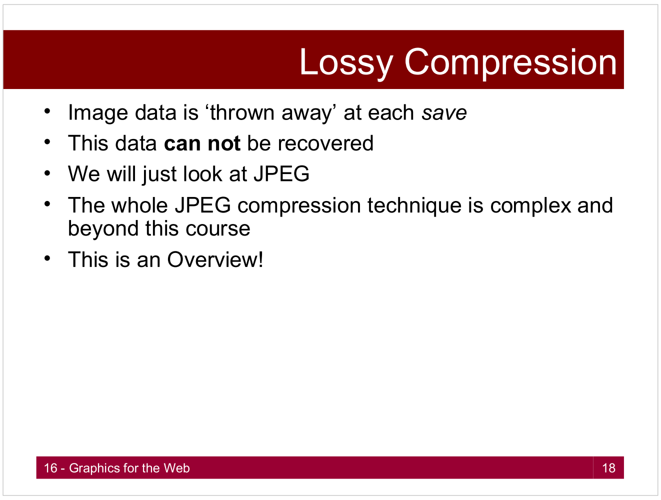
PNG Images

- Came about due to GIF patent issues
- Supports RGB and Greyscale
- Is NOT limited to 256 unique colours per image (ala GIF)
- Allow for an Alpha channel
- Uses *Deflate* compression



PNG vs GIF

- PNG is generally smaller due to better compression
- PNG has alpha channel
- PNG has greater colour depth
- GIF does animation (see example)
- NOTE:- Keep PNG file sizes small. Check the number of colours being used, and extra unnecessary data being bundled with the file



Lossy Compression

- Image data is ‘thrown away’ at each *save*
- This data **can not** be recovered
- We will just look at JPEG
- The whole JPEG compression technique is complex and beyond this course
- This is an Overview!

JPEG Bitmap Images

- Designed for storing photographic images (or photo style)
- Poor for lettering and sharp edges (like cartoons)
- Has a sister, MPEG, for motion pictures

JPEG Compression

- The human visual system has flaws
- JPEG tries to throw away information that the human visual system wont notice
- We find it hard to notice small colour changes
- We do notice small brightness changes
- It most definitely isn’t perfect and can be mislead
- Usefully, the amount of compression can be controlled

How small?

- JPEG achieves very good compression

- Take a full colour image 24bits per pixel
- Lossless will give around 2:1 compression
- JPEG will give 10/20:1 with no visible effects
- It can give 30/50:1 with small effects

JPEG effects

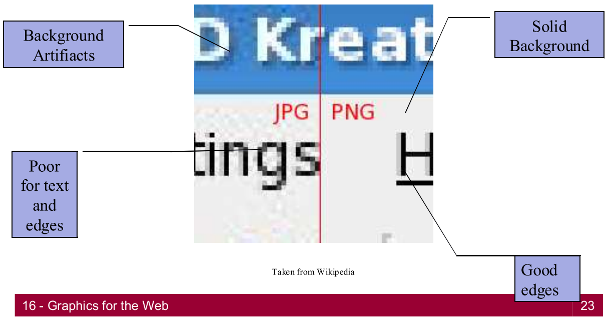
1mb

500kb

100k

JPEG Effects close-up

- Click to add an outline



Background Artifacts

Solid Background

Poor for text and edges

Good edges

Taken from Wikipedia

JPEG Algorithm. Step 1

- Image is divided into blocks of pixels, e.g. 8 x 8. Each block is processed without reference to the others.
- A color space conversion from RGB to YCbCr is carried out.
- YCbCr
  - Y = Luma (brightness)
  - C = Chroma (colour), b … blue, r … red
- Now we aim to reduce the resolution in chroma channels
  - Eye is insensitive to this information
  - Humans can see considerably more fine detail in the brightness of an image (the Y component) than in the color of an image (the Cb and Cr components). Using this knowledge, encoders can be designed to compress images more efficiently.

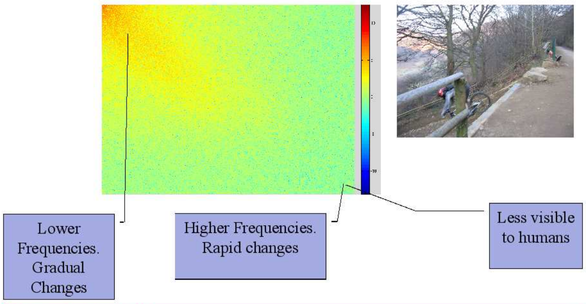
JPEG Algorithm. Step 2

- Perform a Discrete Cosine Transform (DCT)
- This allows us to see the different *frequencies* in the image
- Detail is at high frequencies and simple structures at low frequencies
- We can detect gradual change well but not so much with the detail
- Reduce the resolution of the high frequencies, by replacing individual pixel values by e.g. average frequency values.

http://en.wikipedia.org/wiki/JPEG#JPEG_compression

DCT Image

- Click to add an outline



Lower Frequencies. Gradual Changes

Higher Frequencies. Rapid changes

Less visible to humans

JPEG Algorithm. Step 2. Cont

- Think of these frequencies in an audio signal
- A graphic equaliser/spectrum splits the signal into different frequencies bass/treble
- We can then remove/amplify parts of the signal
- We can SEE this using audio software
- JPEG does the same thing with images

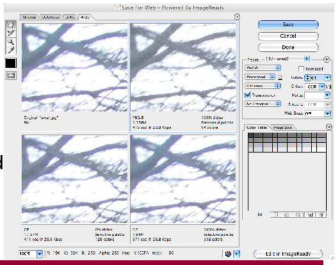
JPEG Algorithm. Step 3

- Some more compression happens with the DCT information.
- This is lossless and called **Huffman Coding**
- This sees how often something occurs and represents that with a smaller bit in a binary tree (higher up the tree)

Saving Images

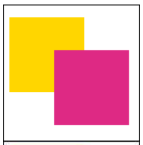
- In Photoshop make use of the '*Save for web*' option
- Change the colour depth, image type, and quality.
- Look at the file sizes and quality of image

- Similar functionality in GIMP.
- Possibility to **animate** gif images by storing them as several frames and defining a delay between them → DEMO

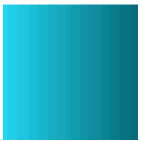


JPEG vs GIF example

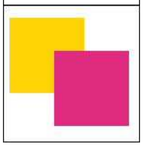
- Click to add an outline



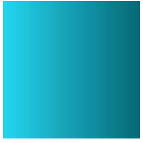
818b GIF



3723b GIF



3729b JPEG



1816b JPEG

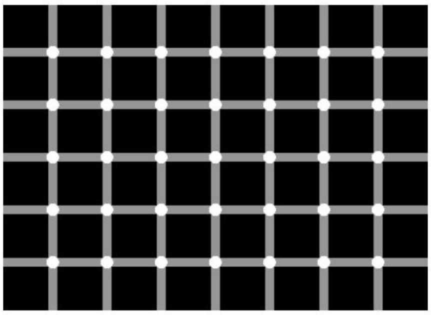
Image file formats

- GIF – Graphics Interchange Format (.gif)
  - 256 colours adapted to image
  - compressed (not good for photos)
  - options for transparency and animation (GIF89A)
- JPEG – Joint Photographic Experts Group (.jpg or .jpeg)
  - sophisticated compression
  - image quality can be chosen (good for photos)
- PNG – Portable Network Graphics (.png)
  - non-proprietary GIF (with better colour quality)
- X-Bitmap (.xbm)
  - black and white (transparent)
- X-Pixelmap (.xpm)
  - 8 bits per pixel (colour)

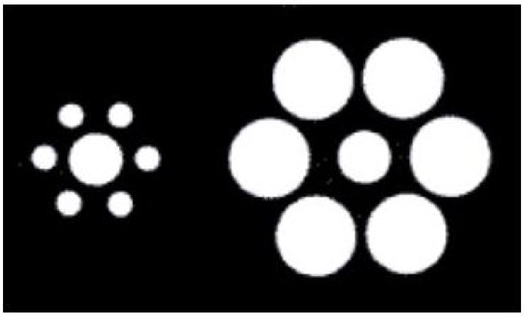
Use of images

- Images can help to illustrate concepts and may make site attractive. However, they also waste screen space and can take too long to download.
- Ensure that any images serve to increase the information content of the page. Provide thumbnail versions of large images.
- Background images can divert attention from the information content of the web page
- Be careful when choosing colour combinations for the text and background
  - dark on light usually better than light on dark

How many black dots do you see?



Is the left center circle bigger?



Vector Images

- These do not supply information for each pixel (as per bitmaps)
- These use mathematical equations to describe geometric primitives
  - Points
  - Lines
  - Curves
  - Polygons
  - These are then rendered and rasterized for display

Why use Vectors?

- Vectors have may advantages over bitmaps
- Their size generally will be smaller
  - To draw a line you just need
    - Line Equation
    - Start point
    - Length
    - Line type and colour

Why use Vectors?

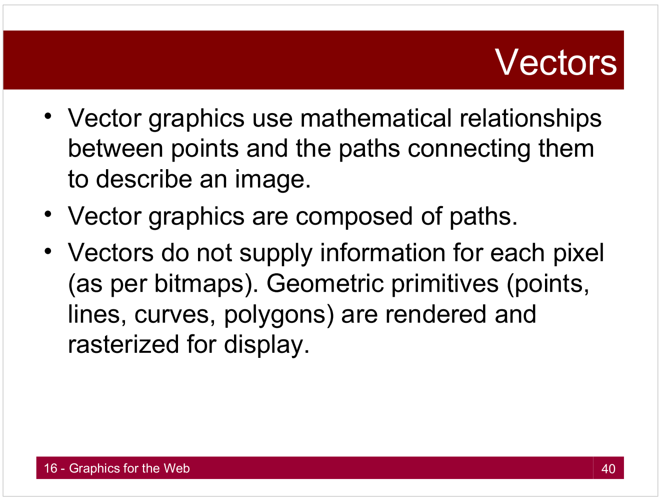
- Images can be zoomed without loosing quality
- Items can be scaled, coloured and even animated
- Fonts can be vectors as well (TrueType)
- We can see this in Acrobat Reader with PDFs
- Edges remain sharp

Vectors and the WWW

- These points make it obvious that Vector graphics is good for WWW use
- Animations and menu design are well suited to this
- Due to modern technology they can become very dynamic/interactive as well
- This is not possible with bitmap style pages

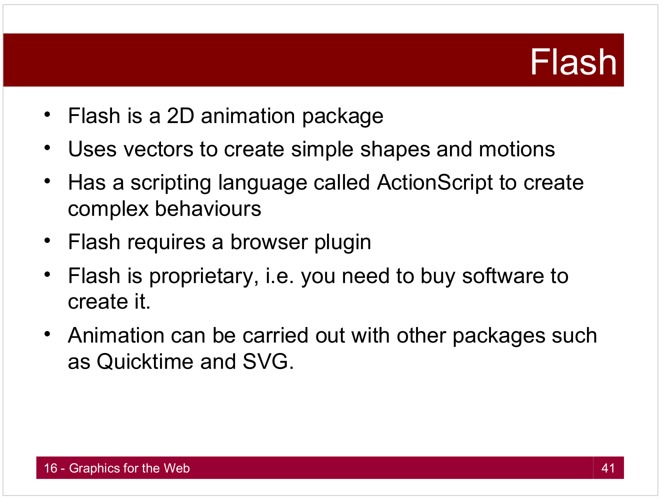
Vectorisation

- Bitmap images can be converted to vectors
- Often needs manual interaction
- Can end up with many many shapes, increasing the file size
- We will see this in **Flash and SVG**.



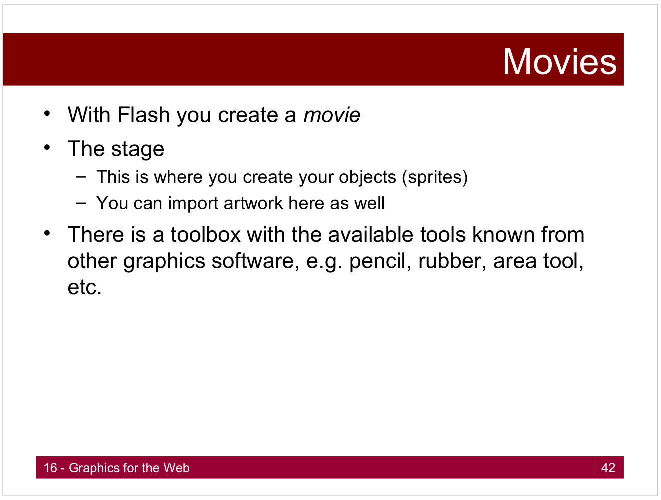
Vectors

- Vector graphics use mathematical relationships between points and the paths connecting them to describe an image.
- Vector graphics are composed of paths.
- Vectors do not supply information for each pixel (as per bitmaps). Geometric primitives (points, lines, curves, polygons) are rendered and rasterized for display.



Flash

- Flash is a 2D animation package
- Uses vectors to create simple shapes and motions
- Has a scripting language called ActionScript to create complex behaviours
- Flash requires a browser plugin
- Flash is proprietary, i.e. you need to buy software to create it.
- Animation can be carried out with other packages such as Quicktime and SVG.



Movies

- With Flash you create a *movie*
- The stage
  - This is where you create your objects (sprites)
  - You can import artwork here as well
- There is a toolbox with the available tools known from other graphics software, e.g. pencil, rubber, area tool, etc.

Animation with Flash Software

- Define a **timeline** that is split into **frames**.
  - The timeline controls how the *movie* acts over time.
  - Frames are single snapshots of the animation.
- **Layers** allow to organize and group objects
- **Libraries** gather create objects.
  - You create instances of a library objects
  - Instances can be edited, rotated, skewed, set transparency, etc.

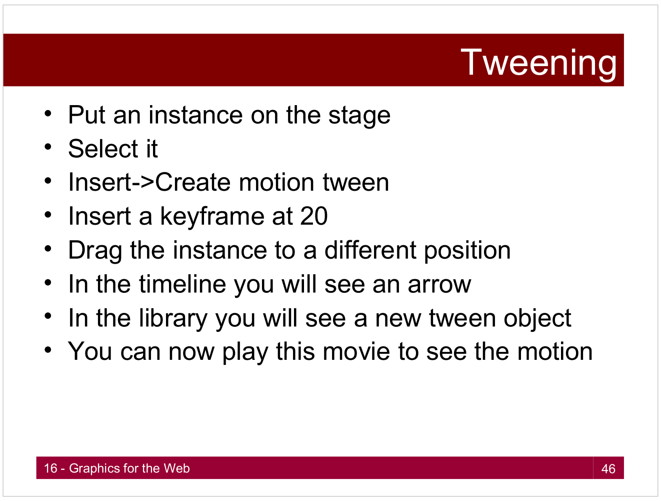
http://www.youtube.com/watch?v=xI38fbBu9VA&feature=related

Animation

- In flash moving an instance along a path is called tweening
- You set the start and end points
- Size, rotation, colour etc can all be tweened
- Flash will interpolate the other frames
- You can also do your own frame by frame animations

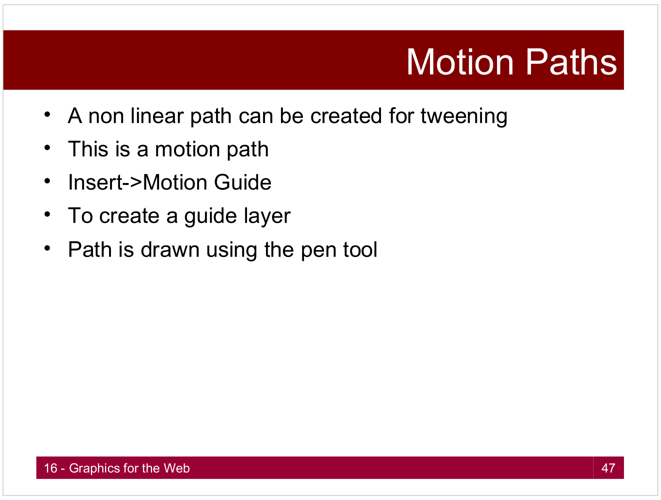
Key Frames

- Key frames are set at important moments
- It is where something changes
- With tweening the parts between key frames are filled in by flash
- Start and end points are key frames
- They are shown as little circles in the timeline



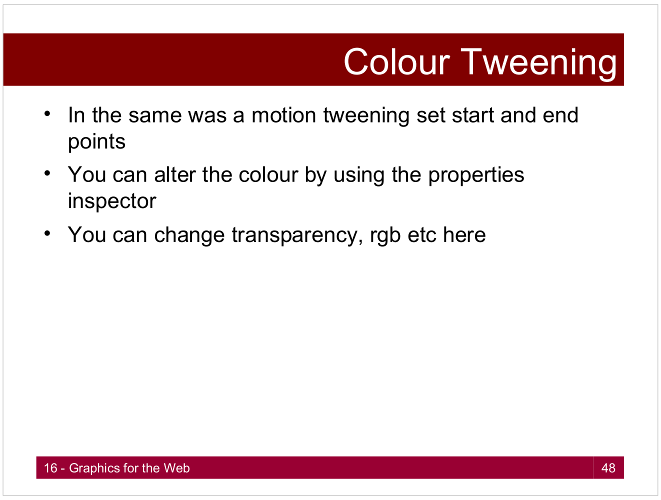
Tweening

- Put an instance on the stage
- Select it
- Insert->Create motion tween
- Insert a keyframe at 20
- Drag the instance to a different position
- In the timeline you will see an arrow
- In the library you will see a new tween object
- You can now play this movie to see the motion



Motion Paths

- A non linear path can be created for tweening
- This is a motion path
- Insert->Motion Guide
- To create a guide layer
- Path is drawn using the pen tool



Colour Tweening

- In the same was a motion tweening set start and end points
- You can alter the colour by using the properties inspector
- You can change transparency, rgb etc here

Shape Tweening

- You can also tween between shapes
- So if you start with a square
- End with a circle
- Flash can fill in the transform between the 2 so the shape morphs

Buttons

- Flash can create buttons for navigation etc
- You have to create the Up, Down and Over states
- Also specify a hit area
- Also some interaction can be applied using actionscript or by adding actions

ActionScript

- To do anything complex in Flash you will need ActionScript
- This scripting language opens up Flash
- ActionScript can be attached to objects, instances and movie frames
- Use Window->Actions to use actionscript

ActionScript II

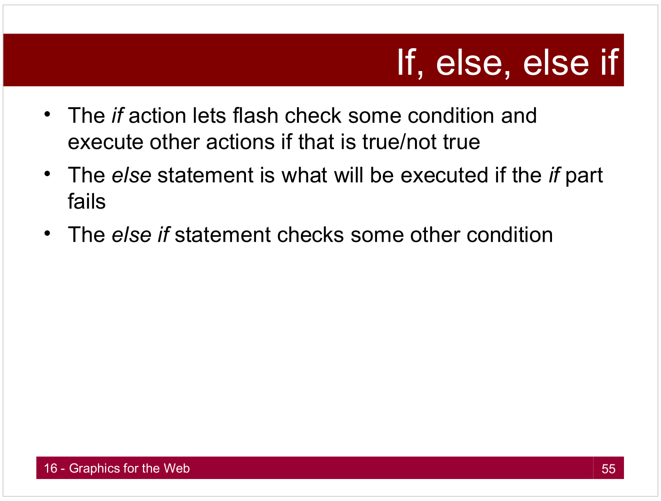
- For each actionscript action there is a reference
- In the actionscript window click the reference button
- This will describe the action
- Use *Check Syntax* to look for errors in your code

Variables

- Local
  - Available in their own block of code
- Global
  - Available to any timeline if you do not use a target path
- Timeline
  - Available to any timeline if you do use a target path
  - Target path is the timeline

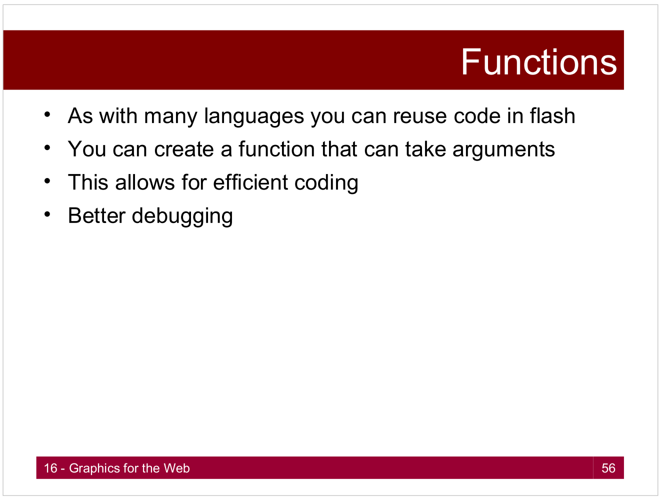
Displaying Information

- Use dynamic text boxes
- These have variable names
- Use actionscript to control what is shown in the text box and when



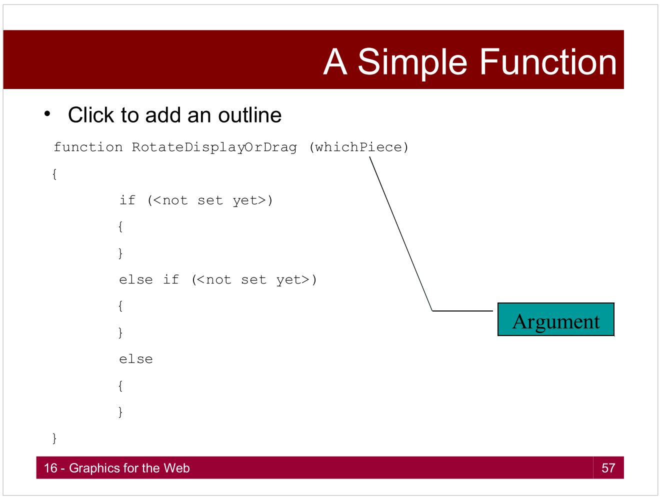
If, else, else if

- The *if* action lets flash check some condition and execute other actions if that is true/not true
- The *else* statement is what will be executed if the *if* part fails
- The *else if* statement checks some other condition



Functions

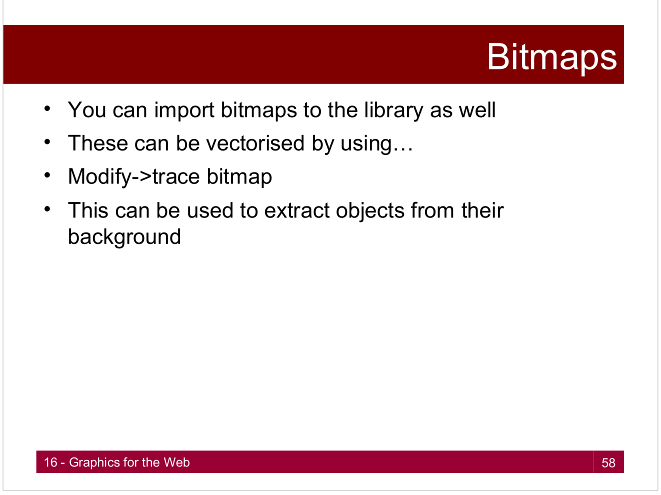
- As with many languages you can reuse code in flash
- You can create a function that can take arguments
- This allows for efficient coding
- Better debugging



A Simple Function

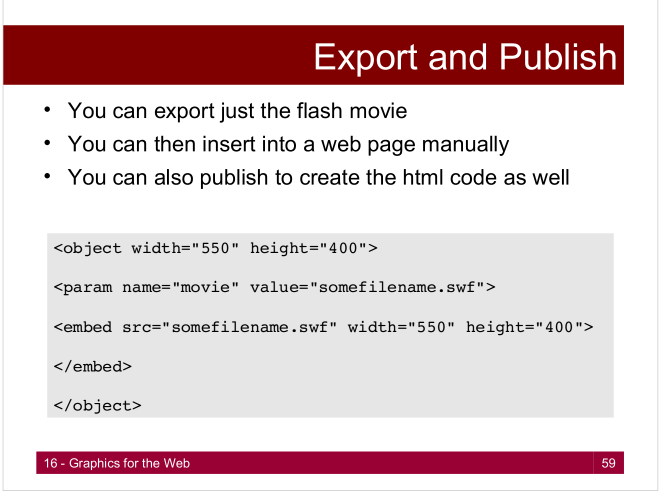
- Click to add an outline

```
function RotateDisplayOrDrag (whichPiece)
{
        if (<not set yet>)
        {
        }
        else if (<not set yet>)
        {
        }
        else
        {
        }
}
```



Bitmaps

- You can import bitmaps to the library as well
- These can be vectorised by using…
- Modify->trace bitmap
- This can be used to extract objects from their background



Export and Publish

- You can export just the flash movie
- You can then insert into a web page manually
- You can also publish to create the html code as well

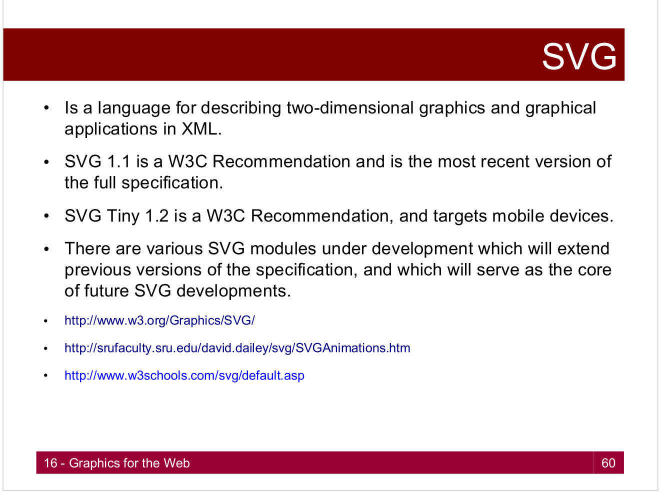
```
<object width="550" height="400">

<param name="movie" value="somefilename.swf">

<embed src="somefilename.swf" width="550" height="400">

</embed>

</object>
```



SVG

- Is a language for describing two-dimensional graphics and graphical applications in XML.
- SVG 1.1 is a W3C Recommendation and is the most recent version of the full specification.
- SVG Tiny 1.2 is a W3C Recommendation, and targets mobile devices.
- There are various SVG modules under development which will extend previous versions of the specification, and which will serve as the core of future SVG developments.
- http://www.w3.org/Graphics/SVG/
- http://srufaculty.sru.edu/david.dailey/svg/SVGAnimations.htm
- http://www.w3schools.com/svg/default.asp

Resources

- http://www.ams.org/samplings/feature-column/fcarc-image-compression
- http://www.w3.org/Graphics/SVG/
- http://srufaculty.sru.edu/david.dailey/svg/SVGAnimations.htm
- www.flashkit.com
- http://www.youtube.com/watch?v=xl38fbBu9VA&feature=related

Software

- Vector
  - PDF files
    - Portable Documents
  - Adobe Illustrator
    - Image manipulation
  - Inkscape
  - Macromedia Flash
    - Interactive web
- Raster
  - Adobe Photoshop
  - GIMP

Vector Based Sites

- Digital Science
- Martin Hawkins
- 2Advanced
- Air Atlantis
- Powerbright
- Courseavenue
- Dream Studio