Welcome

Web Authoring: HTML - Advanced Topics & Photo Optimisation (Level 3)

Richard Hey & Barny Baggs
Objectives

- To understand the need for photo optimisation
- To develop practical skills focused on photo optimisation
- To develop an understanding of more advanced HTML topics

Pre-requisites (Basic HTML & CSS)

Duration, Breaks & Refreshments

Delivery Style

Course Material
Course Information

- Please sign the attendance sheet
- Please fill in the online feedback at the end of the course (shortcut on the Desktop)
- Let us know if you need assistance; please ask questions
- Related Courses:
  - Java Script for Beginners (Level 3)
Importance of rendering times

• Ideally, webpages should seem to appear instantly.

• In webpage terms, the human brain considers anything that takes 100ms or less to be instantaneous. Conversely, anything that takes longer than 1s is deemed to be slow.
Importance of rendering times

• The rendering time for web pages is dominated by the graphics and images on the page.

• Incorrectly configured images and graphics can therefore render your website almost unusable to those on slow internet connections or mobile devices.

• We need to balance acceptable image quality whilst maximising transfer and rendering efficiency.
# Importance of Rendering Times

<table>
<thead>
<tr>
<th></th>
<th>56 kbps</th>
<th>128 kbps</th>
<th>512 kbps</th>
<th>2 Mbps</th>
<th>20 Mbps</th>
<th>100 Mbps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small web page (100 KB)</td>
<td>14s</td>
<td>6.3s</td>
<td>1.6s</td>
<td>0.4s</td>
<td>&lt; 0.1s</td>
<td>&lt; 0.1s</td>
</tr>
<tr>
<td>Complicated website (250 KB)</td>
<td>36s</td>
<td>16s</td>
<td>3.9s</td>
<td>1.0s</td>
<td>0.1s</td>
<td>&lt; 0.1s</td>
</tr>
<tr>
<td>Image from 5M pixel camera (1.5 MB)</td>
<td>3m 39s</td>
<td>1m 36s</td>
<td>24s</td>
<td>6.0s</td>
<td>0.6s</td>
<td>0.1s</td>
</tr>
<tr>
<td>MP3 music file, good quality (5 MB)</td>
<td>12m 11s</td>
<td>5m 20s</td>
<td>1m 20s</td>
<td>20s</td>
<td>2.0s</td>
<td>0.4s</td>
</tr>
<tr>
<td>Complete CD (650 MB)</td>
<td>26h 25m</td>
<td>11h 33m</td>
<td>2h 53m</td>
<td>43m 20s</td>
<td>4m 20s</td>
<td>52s</td>
</tr>
<tr>
<td>Movie (1.5 GB)</td>
<td>62h 25m</td>
<td>27h 18m</td>
<td>6h 50m</td>
<td>1h 42m</td>
<td>10m 14s</td>
<td>2m 3s</td>
</tr>
</tbody>
</table>
Images are made of Pixels

- Images are made up of pixels (tiny dots).
- Each pixel having colour and brightness attributes.
Size, Resolution, Colour Depth and Quality

- Size = pixels wide by pixels tall
- Resolution = pixels per inch
- Colour Depth = bits per pixel
- Quality = amount of compression
Size

Image Size = pixels wide by pixels tall (Mega Pixels)
Resolution

Resolution = pixels per inch (also dots per inch)

Higher dpi

“Print size” is a determined by image size and dpi

Lower dpi
Colour Depth

Colour Depth = bits per pixel

The number of distinct colours that can be represented by a pixel depends on the number of bits per pixel (bpp).

A 1 bpp image uses 1-bit for each pixel, so each pixel can be either on or off.

Each additional bit doubles the number of colours available:

1 bpp, $2^1 = 2$ colours (monochrome)
2 bpp, $2^2 = 4$ colours
3 bpp, $2^3 = 8$ colours
Colour Depth

Colour Depth = bits per pixel

As the number of bits increases, the number of possible colours becomes impractically large for a colour map. So in higher colour depths, the colour value typically directly encodes relative brightness's of red, green, and blue to specify a colour in the RGB colour model.

8 bpp, $2^8 = 256$ colours

16 bpp, $2^{16} = 65,536$ colours ("Highcolour")

24 bpp, $2^{24} = 16,777,216$ colours ("Truecolour")
Colour Depth

256 Colours

128 Colours
Colour Depth

8 Colours

4 Colours
Quality (amount of compression)

100% (no) Compression
63kB

20% Compression
22kB
Quality (amount of compression)

50% Compression

15kB

Image “artifacts” visible

90% Compression

9kB

8 pixel blocks visible
Web graphic formats

• There is no one perfect image format for all graphic / image website requirements

• GIF, JPEG and PNG are all common formats with different characteristics
GIF Format

- GIF - Graphics Interchange Format
- Lossless compression
- 8 bit Colour
- Transparency
- Interlacing
- Animation – CPU waster (avoid use)
JPEG Format

- JPEG (or JPG) - Joint Picture Expert Group
- Lossy compression
- 24 bit – True Colour
- **Progressive or Baseline**
PNG Format

• PNG – Portable Network Graphics
• Lossless
• Transparency
• Best for screenshots / text
• Interlacing
Anti-Aliased
Aliased – undesirable side effects
Dithering

- Colour reduction from 24 bits to 8
- Uses limited colours to create illusion of more colours
- Avoid as hinders compression
Images in HTML

- Use native image size where possible
- Use HEIGHT and WIDTH
- Avoid text in images (let browser generate text)
Optimisation Guidelines

- Save a copy of your original
- Minimise dimension and maximise crop
- Choose minimum colours from a non-dithering palette
- Avoid anti-aliasing
- Reduce colours by colour mapping into Web palette
- Reduce resolution to 72dpi
- Use optimisation plug-in or program or web utility
Search Engine Optimisation

- Search engine optimisation (SEO) is the process of affecting the visibility of a website or a web page in a search engine's "natural" (i.e. un-paid) search results.

- Relevant results are key and determined by a web pages:
  - Content
  - Performance
  - Authority
  - User Experience
SEO Checklist

- Create unique, accurate page titles
- Make use of the "description" meta tag uniquely on each page
- Use meaningful URLs
- Good structure & easy navigation
- Use meaningful anchor text
- Use “alt” attribute with all images
- Use headings (H1 – H6) sparingly but appropriately
- Correct use of robots.txt (not covered in this course)
HTML Tables (Recap)

- Use `<table>` to define a table
- Use `<tr>` to define a table row
- Use `<td>` to define a table data
- Use `<th>` to define a table heading
- Use `<caption>` to define a table caption
Advanced Tables - CSS

- Use **border** to define a border
- Use **border-collapse** to collapse cell borders
- Use **padding** to add padding to cells
- Use **text-align** to align cell text
- Use **border-spacing** to set the spacing between cells

<table>
<thead>
<tr>
<th>My Caption</th>
</tr>
</thead>
<tbody>
<tr>
<td>**</td>
</tr>
<tr>
<td>**</td>
</tr>
<tr>
<td>data</td>
</tr>
<tr>
<td>data</td>
</tr>
<tr>
<td>data</td>
</tr>
<tr>
<td>data</td>
</tr>
</tbody>
</table>
Advanced Tables – Useful Attributes

- **colspan** to make a cell span many columns
- **rowspan** to make a cell span many rows
- **id** to uniquely define one table

```
<table>
<thead>
<tr>
<th>My Caption</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>Heading 1</td>
</tr>
<tr>
<td>Long column</td>
</tr>
<tr>
<td>Long row top</td>
</tr>
<tr>
<td>Data</td>
</tr>
<tr>
<td>Data</td>
</tr>
<tr>
<td>Long row bottom</td>
</tr>
</tbody>
</table>
```
HTML Forms

- HTML forms are typically used to pass data to a server (e.g. contact or feedback form)
- The `<form>` tag is used to create form
- Forms can contain input elements like text fields, checkboxes, radio-buttons and submit buttons
HTML Forms – The Basics

• `<form>` defines an HTML form for user input

• `<input>` defines an input control

• The attribute “type” defines the type of input. Examples include:
  • type=“text”
  • type=“radio”
  • Type=“checkbox”
  • type=“submit”
<form action="demo_form.asp">

First name: <input type="text" name="FirstName" value="Mickey"><br>
Last name: <input type="text" name="LastName" value="Mouse"><br>
<input type="submit" value="Submit">

</form>
The HTML5 `<canvas>` element is used to draw graphics, on the fly, via scripting (usually JavaScript).

The `<canvas>` element is only a container for graphics. You must use a script to actually draw the graphics.

Canvas has several methods for drawing paths, boxes, circles, text, and adding images.

More on this in the “Java Script for Beginners (Level 3)” course.

Example: [http://andrew-hoyer.com/experiments/cloth/](http://andrew-hoyer.com/experiments/cloth/)
Embedding video, audio

• Always aim to “stream” audio and video using a third party service

• Follow instructions provided by the chosen streaming service

• Avoid simply making files available for download

• Cambridge Streaming Media Service (SMS)

• http://www.sms.cam.ac.uk/
Media queries for responsive design

• Using a @media query, you can write different CSS code for different media types.

• This lets you create a different layout for different media types such as a screen or a printer

• It also lets you create a different layout for different devices, used when making web pages with responsive design.

• You can also have different layout when a user resizes the browser window up or down to a certain width, or height.

• http://people.ds.cam.ac.uk/z350/responsive.html
The rest of this session will be dedicated to a self-paced practical which will be dedicated to practical experimentation using a number of different tools and programs

All instructions are provided in the practical course notes but should you get stuck, find something unclear or have any question about the practical just ask.