

Basics of Web Design

Tuesday 16th February 2010

Liam Heaphy, Manchester Architecture Research Centre

Lecture format

1. **Week 1 – 16th Feb:** Web design basics (all)
2. **Week 2 – 23rd Feb:** Web design advanced tutorial class for webmasters (webmasters, but all welcome)
3. **Week 3 – 2nd Mar:** London Olympic Stadium case study lecture (all)

Web Design 16th Feb - Tutorial Structure

1. Web design methodologies (10 min)
2. Web design languages (20 min)
3. Creating your own simple website II (30 min)
4. Testing your website (10 min)
5. Questions

Website design and design software

- Web pages have special requirements
 - Functional
 - Fast-loading
 - Simplest use of code
 - Optimised media
 - Usable on multiple browsers (Firefox, IE, Chrome, Opera, etc)
 - Easy to read

Introducing web languages

- Web page languages are simple, modular, and varied
- Languages have evolved through shared use, like natural languages
- Overall coordination through W3C (World Wide Web Consortium)

Introducing web languages

W3 – Internet Standards, Origins



- HTML (www.w3.org), CERN, and Tim Berners-Lee
 - *Conseil européen pour la recherche nucléaire* (European Organization for Nuclear Research)
- Table, frames (and all that stuff you need not worry about)

WEB LANGUAGES

- **Client-Side**

- HTML
- JavaScript
- CSS

- **Server-Side**

- PHP
- CGI
- XML

Working with HTML

- Tags
 - <html>
 - <header>
 - <body>
 - <div>
 -

Example

```
<html>
<body>

<h1>My First Heading</h1>

<p>My first paragraph.</p>

</body>
</html>
```

Try it Yourself

What is HTML?

HTML is a language for describing web pages.

- HTML stands for **H**yper **T**ext **M**arkup **L**anguage
- HTML is not a programming language, it is a **markup language**
- A markup language is a set of **markup tags**
- HTML uses **markup tags** to describe web pages

Sources: http://www.w3schools.com/html/html_intro.asp

Working with HTML

- All tags must be opened `<html>` and closed `</html>`

```
<html>
```

```
<head>
```

```
<title>My First Website</title>
```

```
</head>
```

```
<body>
```

```
<h1>A Tale of an Extraordinary Website</h1>
```

```
<p>Stuff what I wrote</p>
```

```
</body>
```

```
</html>
```

Using Cascading Style Sheets (CSS)

- Separating content from presentation
- External style sheets (.css)
- Why cascading? Order of priority
 1. Browser default
 2. External style sheet
 3. Internal style sheet (inside the <head> tag)
 4. Inline style (inside a HTML element)

Source: http://www.w3schools.com/css/css_intro.asp

CSS Code

CSS uses the following format - **selector {property: value}**
– E.g. **body {color: black}**

```
.studentimagecasestudies {  
  text-align: center;  
  width: 600px;  
  height: auto;  
  position: relative;  
  left: 40px;  
  border: 1px solid #000;  
}
```

```
.studentimagevertical{  
  text-align: center;  
  width: 400px;  
  height: auto;  
  position: relative;  
  left: 40px;  
}
```

CSS Functionality

- IDs and Classes – **What is the difference?**
 - Classes can be used as many times as needed within a document
 - IDs can only be applied once within a document
- CLASSES
 - Define different styles for the same type of HTML element

```
p.right {text-align: right}
```

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

CSS Functionality

Using IDs as well as classes

- You can also define styles for HTML elements with the id selector.
- ID selector is defined as a #.

The style rule below will match the element that has an id attribute with a value of "green":

```
#green {color: green}
```

The style rule below will match the p element that has an id with a value of "para1":

```
p#para1  
{  
text-align: center;  
color: red  
}
```

How to Insert a Style Sheet

- **External Style Sheet**

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

- **Internal Style Sheet**

```
<head>
<style type="text/css">
hr {color: sienna}
p {margin-left: 20px}
body {background-image: url("images/back40.gif")}
</style>
</head>
```

- **Inline Styles**

```
<p style="color: sienna; margin-left: 20px">
This is a paragraph
</p>
```



Putting CSS and HTML together

```
<html>
<head>

<style type="text/css">
<!--
h1 { color: green; font-size: 37px; font-family: impact }
p { text-indent: 1cm; background: yellow; font-family: courier }
-->
</style>

<title>My First Website</title>
</head>
<body>
<h1>A Tale of an Extraordinary Website</h1>
<p>Stuff what I wrote</p>
</body>
</html>
```

A Tale of an Extraordinary Website

Stuff what I wrote

Creating a simple website I

- Storyboarding
- Using Dreamweaver
- Testing and Evaluation

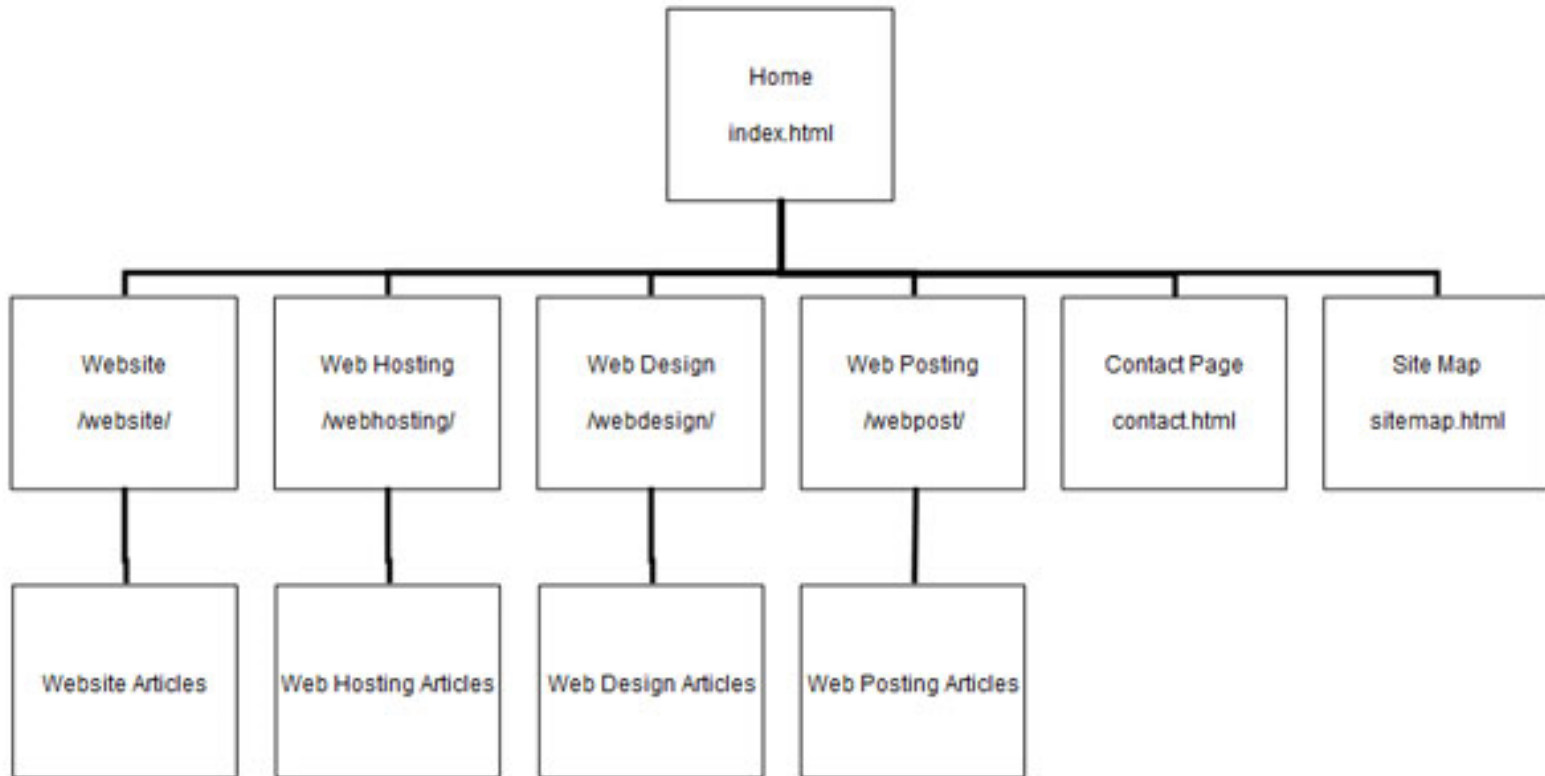
Storyboarding

- GO LOOK AT sites and take note of the ones you like
- SKETCH out a schematic plan of the website
- DRAW on paper a number of interface designs for each unique web page within the plan



Source: <http://www.webhostdesignpost.com/website/websitestoryboarding.html>

Storyboarding



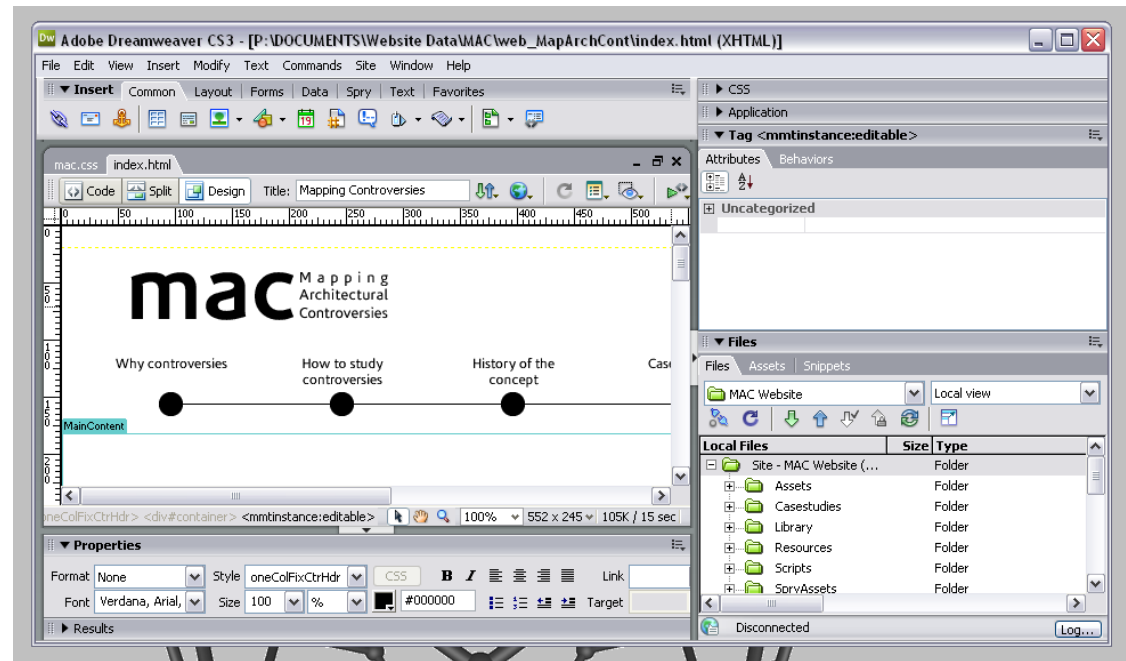
Using WYSIWYG Tools - Dreamweaver

- Complexity
- Scalability
- Functionality

Versions:

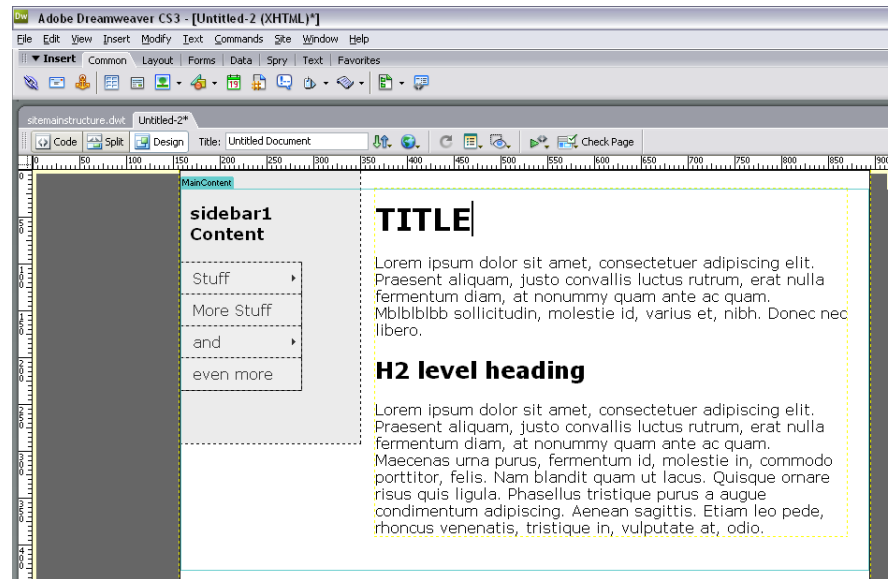
Latest: CS4

UoM: DW 8



Developing the layout

- BUILD a website to match scheme
- SAVE as **template**
- DEFINE editable regions
- RECREATE pages from template



Using DIV tags

- Divides your website into relevant, manageable sections
- Allows you to position content (text, images, video)
- Works in combination with CSS

```
<div class= "personaldetails" >  
  <h3>Who am I?</h3>  
  <p>This is a paragraph.</p>  
</div>
```

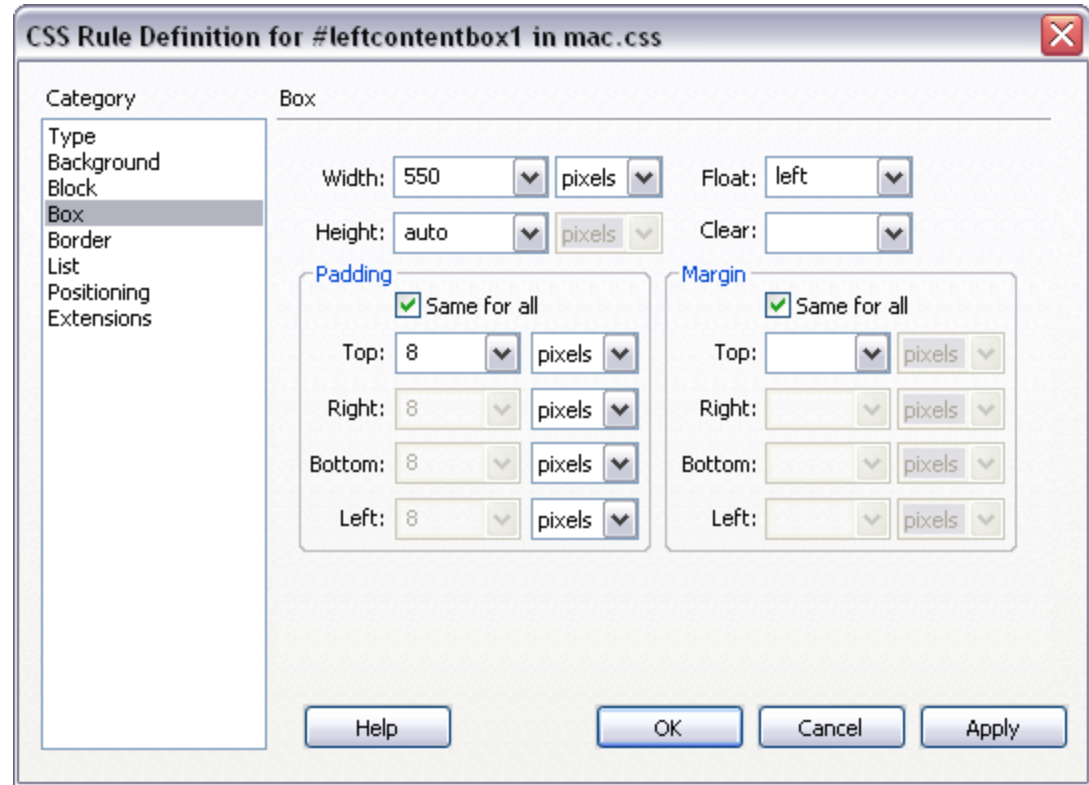
Positioning DIV Tags with CSS

- Positioning
 - Static (default)
 - Relative (to its static position in the document's flow)
 - Absolute (with respect to parent element)
 - Fixed (with respect to top of window)

Source: <http://css.maxdesign.com.au/floatutorial/>

Editing other CSS properties

- Colour
- Margin
- Width
- Padding
- Border



Adding images and video

- Images

- Frame within DIV tags
- Float tags
- Add borders or backgrounds or other effects

- Video

- Encode as .FLV files (web quality med/high)
- Use Dreamweaver to encode within site

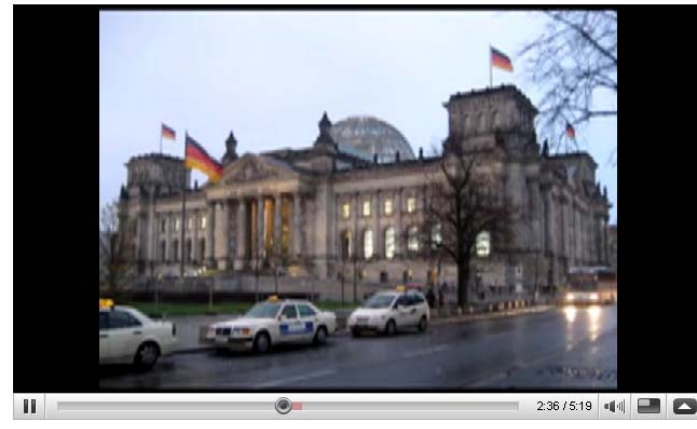
North Lancashire contains Architecture
March 16th, 2009



Pages
[Home](#)
[Contact us](#)
[Read books by us](#)
[Study Architecture](#)
Subscribe

Categories
[Adolf Loos \(2\)](#)
[Aldo Rossi \(2\)](#)
[Alvaro Siza \(2\)](#)
[Andrea Palladio \(12\)](#)
[Architecture Hacks \(34\)](#)
[Aventinus \(105\)](#)
[Buildings at Risk \(17\)](#)
[Carlo Scarpa \(3\)](#)
[Churches \(26\)](#)

Paul Wallot / Norman Foster: Reichstag, Berlin

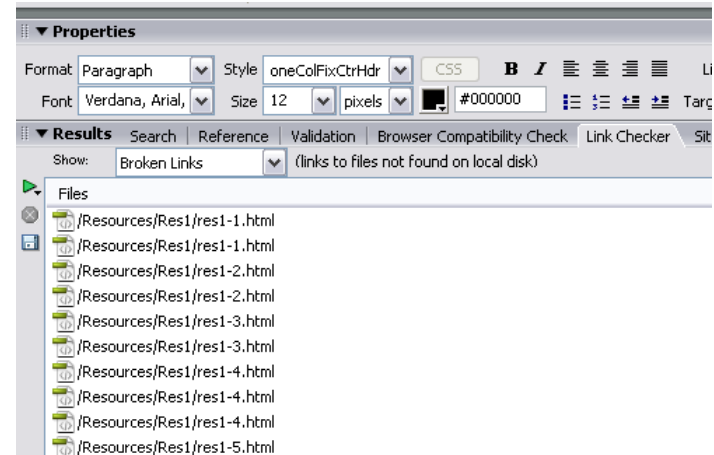


Make your own test site

- DESIGN on paper a website with ~ 5 separate pages.
- Create a webpage with a HEADER, FOOTER, and MENU (either top or sidebar)
- Save as a TEMPLATE
- RECREATE pages from site using the template you have made
- Give each webpage a relevant name, and edit the MENU to link to each page
- Add new content (text, images, video (.FLV, req. encoding))
- PREVIEW with browsers to check layout
- TEST for broken links, bad code, etc

Testing your website

- Constantly check with browser preview
- Use a variety of browsers
- Things to look out for:
 - Mispositioned elements
 - Unviewable video
 - Unclosed tags
 - Broken links
- Use Dreamweaver tools



CMS systems

- Wordpress
 - Create a site on their website or make your own webserver. Simple, not very flexible.
- Drupal
 - Download to your own webserver. Powerful, flexible, and complex
- Joomla
 - Use your own webserver. Flexible, not too complex

Where do we go from here?

- Learn the basics..
 - ...and then start experimenting,
 - but do SAVE and BACKUP your work!

- ‘ Internet ’ fixes everything
 - Search for tutorials
 - Consult books on DW
 - Look on technical forums for specific issues

More resources on platform
<http://www.msa.ac.uk/mac>



Further help

See **Resources** section on

<http://www.mappingcontroversies.co.uk>

(= www.msa.ac.uk/mac)

(or click on MAC icon from MSA website)



Contact Details:

liam.heaphy@postgrad.manchester.ac.uk