Overview

- **Part I**: Portraying the Internet as a collection of online information systems
- **Part II**: Design a website using HTML/XHTML & **CSS**
XHTML validation

This page is not Valid XHTML 1.0 Strict!

Below are the results of checking this document for XML well-formedness and validity.

1. **Error**
   Line 4 column 6: document type does not allow element "html" here.

   ```html
   <html>  
   ```

   The element named above was found in a context where it is not allowed. This could mean that you have incorrectly nested elements — such as a "style" element in the "body" section instead of inside "head" — or two elements that overlap (which is not allowed).

   One common cause for this error is the use of XHTML syntax in HTML documents. Due to HTML’s rules of implicitly closed elements, this error can create cascading effects. For instance, using XHTML’s "self-closing" tags for "meta" and "link" in the "head" section of an HTML document may cause the parser to interpret the end of the "head" section and the beginning of the "body" section (where "link" and "meta" are not allowed; hence the reported error).
What is wrong?

```xml
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en"
 lang="en">
 <html>
  <head>
   <title>This bit goes in the titlebar</title>
  </head>
  <body>
   <h1>This is the heading of the page</h1>
   <p><i>This is a paragraph</i></p>
   <p>This is another paragraph</p>
  </body>
 </html>
```

- Two opening `<html>` tags
Some smaller errors

- Remember:
  - Always need opening and closing tags
  - Single tags need to be closed too: `<br/>
  - Nesting tags
    - `<p>`<i>Some italic text</i>`</p>`
    - `<i>`<p>Some italice text</i>`</p>`

- Single tags also need to be closed off:
  - `<br>`
  - `<br/>`
If your document still doesn’t validate

- Maybe not strict XHTML
  - Do you have **font** colours?
- CSS to the rescue: **Cascading Style Sheets**
W3C creates CSS Standard

- World Wide Web Consortium (W3C)
  - Created HTML 4.0 standard (and in turn XHTML)
  - Created CSS in addition to separate style from content
  - All major browsers support Cascading Style Sheets.
XHTML & CSS

- XHTML: only content
- Colour, fonts, style of text etc: CSS document
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd> 

<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>simple css example</title>
  </head>
  <body>
    <h1>This is a heading</h1>
    <p>this is a paragraph</p>
  </body>
</html>
Your first css

This is a heading

this is a paragraph
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd>
<link rel="stylesheet" type="text/css" href="firststyle.css" />
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>simple css example</title>
</head>
<body>
<h1>This is a heading</h1>
<p>this is a simple paragraph</p>
</body>
</html>
Stylesheet link

<link rel="stylesheet" type="text/css" href="firststyle.css" />

- Shows that the document **will use** a stylesheet
  - href (remember the `<a href=...>` link tag?)

- Shows where the stylesheet is and what its called
The css file: firststyle.css

h1 {color: red}
Your first css page

This is a heading

this is a paragraph
Adding to our style sheet

h1 {color: red}

p{
  font-family: "lucida calligraphy", arial, 'sans serif';
}

Adding to our style sheet

This is a heading

this is a paragraph
Changing background colour

h1 {color: red}

p{font-family: "lucida calligraphy", arial, 'sans serif';}

body{
background-color:gray;
}
Changing background colour

This is a heading

this is a paragraph
Changing paragraph alignment

h1 {
    color: red
}

p {
    font-family: "lucida calligraphy", arial, 'sans serif';
    text-align: right
}
Changing paragraph alignment

This is a heading

this is a paragraph
Why CSS?

- **Font** tags (change heading colour & paragraph colour)
- **Align** tags (change paragraph alignment)
- **Body** tag (add bg-color to <body> tag)
An example from CSS Zen Garden: Same content, different stylesheet

http://www.csszengarden.com/
An example from CSS Zen Garden: Same content, different stylesheet

http://www.csszengarden.com/
An example from CSS Zen Garden: Same content, different stylesheet

http://www.csszengarden.com/
An example from CSS Zen Garden: Same content, different stylesheet

http://www.csszengarden.com/
You have got valid XHTML...

- W3C validator checks you XHTML for errors and conformance to standards (STRICT or TRANSITIONAL)

- W3C CSS validator checks your CSS for errors and conformance to standards
http://jigsaw.w3.org/css-validator
CSS Syntax

- CSS syntax consists of 3 parts:
  - **Selector**{**Property**: **Value**}
- You’ve seen this already in our simple example:
  ```
  h1  {color: red}
  p   {
      font-family: "lucida calligraphy", arial, 'sans serif';
      text-align: right
  }
  body {background-color: gray}
  ```
CSS Syntax: grouping styles

- Grouping selectors:
- We can have:
  - h1 {color: red}
  - h2 {color: red}
  - h3 {color: red}
- But we can also say:
  - h1,h2,h3 {color: red}
  - Think again of uniform styles & saving time!
  - Easier to change this one style in css file than change all 3 above
CSS Syntax: The Class Attribute

- The Class attribute:

  ```html
  p{
    font-family: "lucida calligraphy", arial, 'sans serif'
  }

  p.right{text-align: right}
  p.left{text-align: left}
  p.center{text-align: center}
  ```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>simple css example</title>
<link rel="stylesheet" type="text/css" href="firststyle.css" />
</head>
<body>
<h1>This is a heading</h1>
<p class="right">this is a simple paragraph</p>
<p class="center">this is a simple paragraph</p>
<p class="left">this is a simple paragraph</p>
</body>
</html>
CSS Syntax: The Class Attribute
Generic Class Attributes

- Suppose we want to have h1, h2 & p have center classes:
  - h1.center \{text-align: center\}
  - p.center \{text-align: center\}
  - h2.center \{text-align: center\}

- To simplify, we can make one class attribute usable by all these (h1, h2 & p)
  - .center \{text-align: center\}
Generic Class Attributes

h1 {
    color: red
}

p{
    font-family: "lucida calligraphy", arial, 'sans serif';
}

.right{text-align: right}
.left{text-align: left}
.center{text-align: center}

body{background-color:gray;}
Generic Class Attributes

```html
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>simple css example</title>
<link rel="stylesheet" type="text/css" href="firststyle.css"/>
</head>
<body>
<h1 class="center">This is a heading</h1>
<p class="right">this is a simple paragraph</p>
<p class="center">this is a simple paragraph</p>
<p class="left">this is a simple paragraph</p>
</body>
</html>
```
Generic Class Attributes
Using `<span></span>`

- So:
  - What if we don’t want to change the style of a whole paragraph with:
    ```html
    p{
        font-family: "lucida calligraphy", arial, 'sans serif';
    }
    ```
  - But just the ‘simple’ part of ‘this is a simple paragraph’
  - Use a generic class attribute combined with the `<span>` tag:
    ```html
    .nicefont{
        font-family: "lucida calligraphy", arial, 'sans serif';
    }
    ```
Using `<span>`</span>
Using `<span>`</span>
Remember the <div></div> tag?

```xml
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>simple css example</title>
<link rel="stylesheet" type="text/css" href="firststyle.css" />
</head>
<body>
<h1 class="center">This is a heading</h1>
<div class="nicefont">
<p class="right">this is a simple paragraph</p>
<p class="center">this is a simple paragraph</p>
<p class="left">this is a simple paragraph</p>
</div>
</body>
</html>
```
Remember the `<div></div>` tag?

This is a heading

this is a simple paragraph

this is a simple paragraph

this is a simple paragraph
Background Images in CSS

- Some people have already added a background image to their XHTML

  `<body background="leaves.gif"></body>`

- With CSS, we can do more advanced things with our backgrounds
Background Images in CSS

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
Move our background to CSS file

```css
body{
    background-color: gray;
    background-image: url("leaves.gif");
}
```
Move our background to CSS file
Stop the background repeating?

body{
  background-color: gray;
  background-image: url("leaves.gif");
  background-repeat: no-repeat;
}
Stop the background repeating?
Fixing the background

body{
background-color: gray;
background-image: url("leaves.gif");
background-repeat: no-repeat;
background-attachment: fixed
}
Positioning the background

body{
  background-color: gray;
  background-image: url("leaves.gif");
  background-repeat: no-repeat;
  background-attachment: fixed;
  background-position: bottom
}
CSS Syntax: Comments

- In XHTML: Comments look like
  - <!-- a comment in XHTML -->
- CSS comments, similarly, can be used to increase the readability of your CSS code, especially if you make use of shorthand format when writing your CSS rules:
  - /* a comment in css */
Using `<div></div>`

- No visible difference using `<div></div>`
- But remember - structural tags:
  - Don’t alter the presentation of the document
  - Show structure or meaning of document
- In this case:
  - `<div></div>` allows us to split our document into sections
  - CSS defines how `<div></div>` sections are presented
CSS for how `<div>` are displayed

```html
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
  <head>
    <title>Positioning In CSS</title>
    <link href="mystyle.css" rel="stylesheet" type="text/css" />
  </head>
  <body>
    <div class="one">One</div>
    <div class="two">Two</div>
    <div class="three">Three</div>
    <div class="four">Four</div>
  </body>
</html>
```
CSS for how `<div>` are displayed

```css
div{
    position: relative;
    width: 100px;
    height: 100px;
    color: black;
    text-align: center
}

div.one{background-color: red}

div.two{background-color: green}

div.three{background-color: blue;}

div.four{background-color: yellow}
```
CSS for how `<div></div>` are displayed
CSS for how `<div></div>` are displayed

- `<div></div>` define:
  - The size of area covered (using width and height)
  - The position of the text in the tag (centered here)

- So what does the `position: relative` part mean?
Positioning: Relative & Absolute

- **Relative:**
  - Item follows previous item in layout
  - Default positioning

- **Absolute:**
  - Specify exact position of element on page
  - Where is should be displayed in terms of either:
    - The page itself
    - Other things on the page
Absolute positioning

div{
  position: absolute;
  width: 100px;
  height: 100px;
  color: black;
  text-align: center
}

div.one{background-color: red}

div.two{background-color: green}

div.three{background-color: blue;}

div.four{background-color: yellow}
Where did one, two, three go?
Change position slightly

```css
div{
  position: absolute;
  width: 100px;
  height: 100px;
  color: black;
  text-align: center
}

div.one{
  background-color: red;
  left:0px;
  top:0px
}

div.two{
  background-color: green;
  left:25px;
  top:25px
}

div.three{
  background-color: blue;
  left:50px;
  top:50px
}

div.four{
  background-color: yellow;
  left:75px;
  top:75px
}
```
Change position slightly
Does four always have to be on top?

- **Absolute positioning:**
  - Lets us specify by pixel where things go
  - Allows us to overlap areas of our webpage defined by `<div></div>` tags

- **Can we specify order that they overlap?**
  - i.e. can we stack them by:
    - two, four, three, one instead of
    - four, three, two, one?
Z-index

- Think of page in terms of x, y & z coordinates

![Diagram showing x, y, and z coordinates with a red square and a green rectangle. The red square is located at (0, 3) and the green rectangle is behind it.]
How do things stack up?

div{
  position: absolute;
  width: 100px;
  height: 100px;
  color: black;
  text-align: center
}

div.one{background-color: red;
  z-index:0;
  left:0px;
  top:0px}

div.two{background-color: green;
  z-index:3;
  left:25px;
  top:25px}

div.three{background-color: blue;
  z-index:1;
  left:50px;
  top:50px}

div.four{background-color: yellow;
  z-index:2;
  left:75px;
  top:75px}
How do things stack up?