Text with CSS and HTML

PGCertInfoTech

Programming with Web Technologies

Auckland
ICT Graduate School

Further HTML Elements: Lists

- Unordered lists (bullet lists)

```
<ul>
  <li>item 1</li>
  <li>item 2</li>
  <li>item 3</li>
</ul>
```

- item 1
- item 2
- item 3

- Ordered lists (numbered lists)

```
<ol>
  <li>item 1</li>
  <li>item 2</li>
  <li>item 3</li>
</ol>
```

1. item 1
2. item 2
3. item 3

- Definition lists

```
<dl>
  <dt>Term being defined</dt>
  <dd>Definition of the term</dd>
  <dt>HTML</dt>
  <dd>HyperText Markup Language</dd>
</dl>
```

Term being defined
    Definition of the term
HTML
    HyperText Markup Language

Tables

```
<!DOCTYPE html> <html>
    <head>
        <meta charset="UTF-8">
        <title>Table</title>
    </head>
    <body>
         <table>
             <caption>An example table</caption>
             <thead>
                 <tr>
                     <th> header 1</th><th> header 2 </th>
                 </tr>
             </thead>
             <tfoot>
                 <tr>
                     <td> footer 1</th><th> footer 2 </td>
                 </tr>
             </tfoot>
             <tbody>
                 <tr>
                     <td> data 1</th><th> data 2 </td>
                 </tr>
             </tbody>
         </table>
     </body>
 </html>
```

An example table

header 1	header 2
data 1	data 2
footer 1	footer 2

The Tables Debate

Tables are very flexible and have been used for page layout in HTML document for many years

- However there is a debate in the web design community about whether they should be used for this purpose

Pro: Tables are useful and well-supported by browsers

Con: Tables are intended for tabular data *not* for layout

The trend in HTML5 is towards the web design point of view:

- HTML5 does not support many tags and attributes that could be used for structuring and layout
- For that they expect you to use CSS

http://www.w3.org/TR/html5/tabular-data.html http://www.w3schools.com/html/html_tables.asp

A Conundrum

How do you write '<' so it appears in a **rendered** HTML page without it being interpreted as the start of an HTML tag?

- For example produce the following in a displayed HTML page:

  The <head> and the <body> elements should be inside the <html> element

- Entered verbatim as HTML, it would be rendered as:

  The and the elements should be inside the element

Solution: HTML entities

- Some characters in HTML are reserved in HTML
  - < and > are reserved for tags
  - " is reserved for enclosing attribute values
  - etc.

- To display a reserved character we use HTML entities
  - To display < use <
  - To display > use >

HTML Entities

Character	HTML entity
<	<
>	>
"	"
'	'
&	&

For a complete list of entities, see:

`http://dev.w3.org/html5/html-author/charref`

CSS Selectors

We have already seen one kind of CSS selector which is the **type selector:**

- A type selector is a type of HTML element (h1)
- A comma separated list of HTML elements (h1, h2, h3)

There are other kinds of selectors:

- attribute
- id
- class
- child
- descendent

CSS attribute selectors

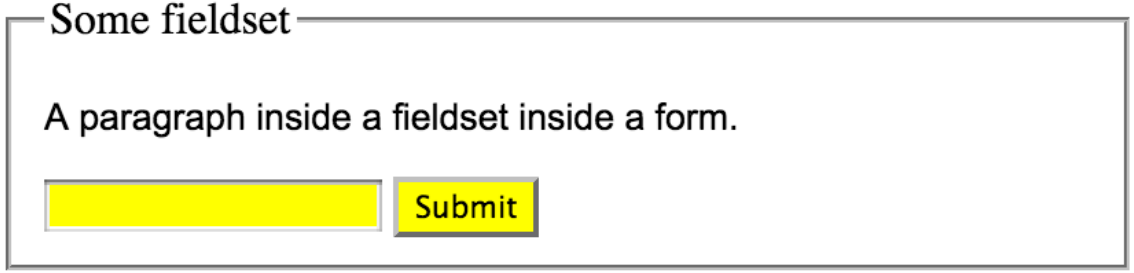
- We know that HTML elements can have attributes, so we can therefore use attributes to select elements in CSS rules
- For example, set the background colour of text boxes in forms
- Text boxes are <input> elements... but so are checkboxes, radio buttons, file selectors, submit buttons etc
- So this rule:

```
input {
    background-color : yellow;
}
```

would change the colour of **all** <input> elements

CSS attribute selectors

```
<form>
    <fieldset>
        <legend>
             Some fieldset
        </legend>
             <p>
               A paragraph inside a fieldset
               inside a form.
             </p>
             <input type="text" />
             <input type="submit" value="Submit" />
    </fieldset>
</form>
```



CSS attribute selectors

- We can use attribute selectors here, which can differentiate between elements on the basis of the attribute value
- Attribute selectors take the form:

  element-type[attribute="value"]

- So to better control what gets changed, we can write:

```
 input[type="text"] {
            background-color : yellow;
}
```

CSS attribute selectors

We can use multiple attributes in the selector if we want to refine our selection:

```
input[type="text"][name="firstName"]
{
    color : yellow;

}
```

CSS id selector

Recall that we can assign a unique identifier to HTML elements using the **id** attribute:

- Id selectors are used to do precise formatting (formatting for a single item)

```
#para1
{
   …
}
```

CSS class selector

- The class attribute specifies one or more class names for an element.
- A class attribute is generally used to make a virtual group of elements called a class. Any style applied to this class will apply to all the elements in this virtual group

```
<h2 class="heading"> </h2>
<thead class="inverted-table-header"></thead>
<img class="bordered-image" src="foo.png" alt="foo">
<input class="required" type="text">
<th class="heading required" > </th>
```

CSS class selectors

Use class selectors like so:

```
.heading {
    color: blue;
}

.required {
   background- color: green;
}
```

- These rules select any elements with the given class attribute value
- This is independent of the type of element that would be selected

CSS class selectors

- What about this situation in a form?

```
<input class="required" type="text">
<textarea class="required" rows="4” cols="40">
</textarea>
```

- It is logical to be able to style text boxes and text-areas on the basis that the user is required to enter values But what if the styles need to be different?

CSS class selectors

This will not discriminate between the element types:

```
.required {
  background- color: green;
}
```

But we can combine type, attribute and class selectors, thus:

```
textarea.required {
  background- color: green;
}
```

CSS selectors

So far we have seen:

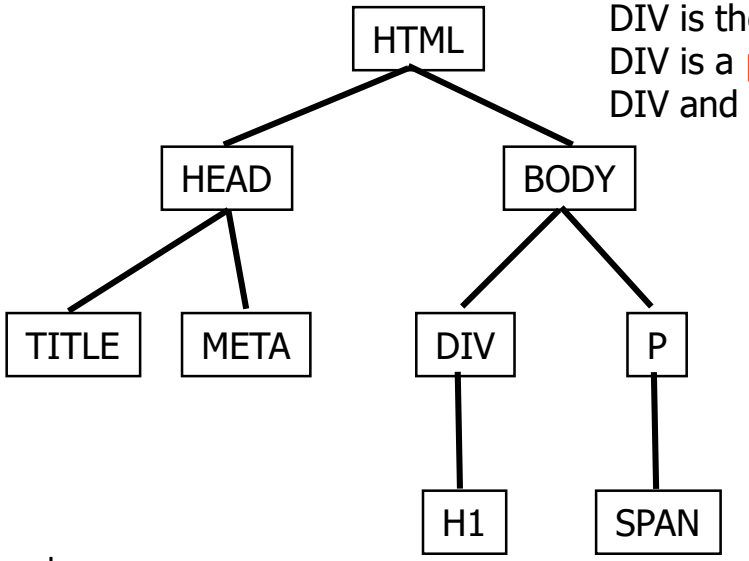
- type
- attribute
- id
- class

Next:

- Child selector
- Descendent selector
- Pseudo classes

HTML Document Tree

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>
      Inline CSS
    </title>
  </head>
  <body>
    <div>
      <h1>
        heading
      </h1>
    </div>
    <p>
      A <span> new </span> paragraph.
    </p>
  </body>
</html>
```



DIV is a descendant of BODY and HTML
DIV is a child of BODY
DIV is the first-child of BODY
DIV is a parent of H1
DIV and P are siblings

CSS child selector

- If there are patterns in the way we want to apply style to our HTML we can use child and descendent selectors to define these patterns

```
<p>
     A paragraph outside a form.
</p>
<form action="" method="GET">
     <p>
          A paragraph inside a form.
     </p>
     <!-- other form content -->
</form>
```

- What if we have this in our HTML and we want paragraphs that occur inside the forms to be styled differently than paragraphs outside the forms

CSS child selector

- We can select **p** elements that are children (at the 1st level inside) of **form** elements
- The CSS rules

```
p {
    font-family : Arial;
    font-size : 10pt;
}

form > p {
     font-family : Times;
     font-size : 12pt;
     color : blue;
}
```

A paragraph outside a form.

A paragraph inside a form styled using a child selector.

- The parent and child element types are separated by a greater than sign, **>**

CSS child selector

- Given those rules, what do you expect to see for the 3 paragraphs in this HTML?

```
<p>
    A paragraph outside a form.
</p>
<form action="" method="GET">
    <p>
        A paragraph inside a form styled
    </p>
    <fieldset>
        <legend>
            Some fieldset
        </legend>
        <p>
            A paragraph inside a fieldset inside a form.
        </p>
    </fieldset>
</form>
```

A paragraph outside a form.

A paragraph inside a form styled using a child selector.

Some fieldset

A paragraph inside a fieldset inside a form.

The child selector **parent > child** only applies to direct descendants

CSS descendent selectors

- What if we want all paragraphs inside a form to be of the same style
- We can use descendent selector for this
- Format
  - **ancestor** whitespace **descendant**

```
form p {
    font-family : Times;
    font-size   : 12pt;
    color       : blue;
}
```

CSS descendent selectors

- make emphasized text () anywhere in a table red

```
table em
{
        color : red;
}
```

```
<table>
    <caption>
        An example HTML table (<em>this is the caption</em>)
    </caption>
    <tbody>
        <tr>
            <td>row 1 col 2</td>
        </tr>
        <tr>
            <td><em>row 2 col 1</em></td>
            <td>row 2 col 2</td>
        </tr>
    </tbody>
</table>
```

An example HTML table (*this is the caption*)

row 1 col 2

row 2 col 1 row 2 col 2

CSS pseudo classes

All of the different kinds of selectors so far identify elements using information given in the HTML document about elements

- type
- id
- attribute names and values
- Class

What about characteristics of elements that aren't represented in the document?

- a link hasn't been visited
- a link has been visited
- the pointer is hovering over an element
- an element has input focus

CSS pseudo classes

- Pseudo classes specify the state of the element to be selected
- For example, to style link anchors in our page when
  - link unvisited **:link**
  - link previous visited **:visited**
  - pointer over the link **:hover**
  - user has clicked link but not released mouse button **:active**

```
selector:pseudo-class {
    property:value;
}
```

```
a:link    { color : blue; }
a:visited { color : red; }
a:hover   { color : green; }
a:active  { color : yellow; }
```

CSS pseudo classes

- Obviously **:link** and **:visited** only make sense for <a> elements
- But we can apply **:hover** to other elements

```
p:hover {
  font-size: 18pt;
}
fieldset:hover {
  background-color : gray;
}
```

- Normally apply the **:focus** pseudo-class to text input elements

```
input[type="text"]:focus {
  border-color : green;
}
```

List of pseudo classes: http://www.w3schools.com/css/css_pseudo_classes.asp