

CM1102: Introduction to CSS (part 1)

Martin Caminada

Cardiff University

HTML and CSS overview

- *HTML: Hypertext Markup Language*
specifies the structure of the document
(e.g. where do the paragraphs start and end, and what are the section titles)
- *CSS: Cascading Style Sheets*
specifies how to display the HTML document
(e.g. use 11pt font size for paragraphs and 18pt font size for section titles)

*What are the advantages
of separating structure and display?*

CSS functionality

Cascading Style Sheets (CSS) are used to set how web content is displayed. This includes:

- fonts, colours, backgrounds, borders, etc.
- layout of content on a page (position, sizing, etc.)
- Some aspects of interaction with elements (e.g. what happens on mouse-clicks)

CSS Style Rules

- A style rule has:
 - a selector
 - a set of declarations
- The selector specifies which HTML elements the style should be applied to
- The declarations specify how the selected elements should be displayed
- Each declaration consists of:
 - a property
 - a value

```
selector {  
    property: value;  
    property: value;  
    property: value;  
}
```

```
h1 {  
    color: red;  
    background-color: white;  
}  
  
p {  
    color: blue;  
    font-family: sans-serif;  
}
```

Where to put CSS

- inline
in an individual element,
using the style attribute
- in an internal style sheet
inside of a <style> element
(inside of the <head> element)
- in an external style sheet
in a separate file referenced
by the <link> element

```
<p style="color: blue;  
font-family: sans-serif;">
```

```
<style>  
  p {color: blue;  
      font-family: sans-serif;}  
</style>
```

```
p {color: blue;  
font-family: sans-serif;}
```

External Style Sheets

- external style sheet:
CSS is contained in a separate file
- The HTML file should then contain a link:

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

- **rel** specifies the type of link being used
- **type** specifies the MIME (Multipurpose Internet Mail Extensions) / IMT (Internet Media Type) of this file
- **text/css** specifies “cascading style sheets” text
- **href** specifies where to find the file

CSS Selectors

Selectors are used for internal and external style sheets (but not for inline CSS)

Some of the most useful selectors:

- type selector `p`
- class selector `.blah`
- id selector `#blah`
- child selector `p > em`
- descendant selector `p span`
- adjacent sibling `h1+p`
- general sibling `h1~p`

More selectors can be found in your book.

Type Selectors

Type selectors apply to a particular type of element (e.g. h1, p or em).

`<h1>` *Main Attractions* `</h1>`
`<p>` ``*Splott Beach*`` and
 ``*Splott Market*`` are
 the main attractions in Splott
`</p>`

`h1,p` { color: red; }
`em` { color: blue; }

Main Attractions

Splott Beach and *Splott Market* are
the main attractions in Splott

Class Selectors

Class selectors apply to elements whose class attribute has a particular value

```
<ul> <li class="win"> MS Windows </li>
      <li class="unix"> Linux </li>
      <li class="unix"> FreeBSD </li> </ul>
<p class="unix"> Unix comes in many flavours. </p>
```

li.win	{ color: red; }
li.unix	{ color: blue; }
.unix	{ font-style: italic; }

- MS Windows
- Linux
- FreeBSD

Unix comes in many flavours.

ID Selectors

ID selectors apply to the (unique!) element whose ID attribute has a particular value

```
<ul> <li id="win"> MS Windows </li>  
      <li id="linux"> Linux </li>  
      <li id="freebsd"> FreeBSD </li> </ul>
```

```
#linux {color: red; }
```

- MS Windows
- Linux
- FreeBSD

Child Selectors

Child selectors apply to the children (direct sub-elements) of a particular element

```
<ul> <li> Windows </li>
```

```
  <li> <em>Linux</em> </li>
```

```
  <li> FreeBSD </li> </ul>
```

```
<p> <em>Unix</em> comes in many flavours. </p>
```

```
li>em    {color: red; }
```

- Windows
- *Linux*
- FreeBSD

Unix comes in many flavours.

Descendant Selectors

Descendant selectors apply to the descendants (children, children's children, etc) of a particular element

```
<ul> <li> Windows </li>  
      <li> <em>Linux</em> </li>  
      <li> FreeBSD </li> </ul>
```

```
<p> <em>Unix</em> comes in many flavours. </p>
```

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

- Windows
- **Linux**
- FreeBSD

Unix comes in many flavours.

Adjacent Sibling Selectors

Adjacent sibling selectors apply to the first sibling (child of the same parent) of a particular element

<h1> Transport </h1>

<p> Take bus 11 towards Pengam Green. </p>

<p> Alternatively, just walk. </p>

h1+p {color: red; }

Transport

Take bus 11 towards Pengam Green.

Alternatively, just walk.

General Sibling Selectors

General sibling selectors apply to any subsequent sibling of a particular element

<h1> Transport </h1>

<p> Take bus 11 towards Pengam Green. </p>

<p> Alternatively, just walk. </p>

h1~p {color: red; }

Transport

Take bus 11 towards Pengam Green.

Alternatively, just walk.

CSS Properties and Values (1/3)

Fonts:

- **font-family**: <family name> [<generic family>]
- **font-style**: normal|italic|oblique
- **font-weight**: normal|bold|bolder|lighter
- **font-size**: small|medium|large|smaller|larger

Colours and backgrounds for text:

- **color**: <value>
- **background-color**: <value> | transparent
- **background-image**: URL | none

(URL notation: url("selfie.jpg"))

CSS *Font-Family*

`font-family` can be given multiple *alternative* values

```
body {  
font-family: Verdana, Arial, sans-serif;}
```

In the above example:

- The browser first looks for the `Verdana` font
- If this is not on the system, it looks for the `Arial` font
- Last resort: browser uses a generic `sans-serif` font

Generic font classes:

`serif` `sans-serif`

`monospace` *cursive*

`fantasy`

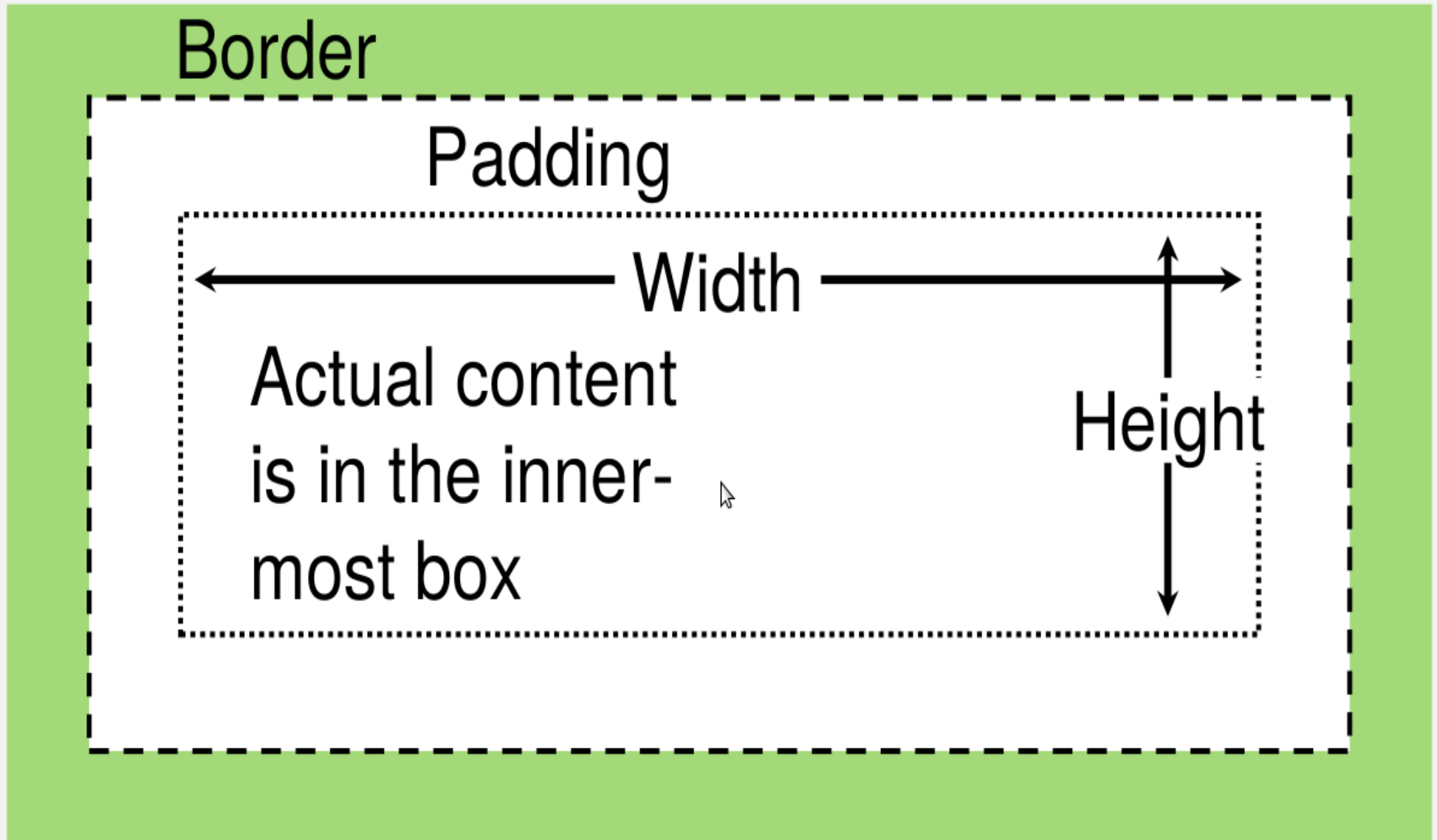
CSS Properties and Values (2/3)

Text:

- **text-decoration:**
none|underline|overline|line-through
- **text-transformation:**
none|capitalize|uppercase|lowercase
- **text-align:**
left|right|center|justify
- **text-indent:**
length|percentage

The CSS Box Model

Margin



CSS Properties and Values (3/3)

- **margin**: width (e.g. 5px or 5%)
margin-top, margin-right, margin-left, margin-bottom
- **padding**: width (e.g. 5px or 5%)
- **border-width**: thin|thick|medium (or width units)
- **border-color**: colour name or hex/rgb values
- **border-style**: none|dotted|dashed|solid|double|groove|ridge|inset|outset

margin, padding, border-width, border-color and border-style
also have top/bottom/left/right variants

(e.g. **border-top-width**, **border-bottom-width**, **border-left-width**, **border-right-width**)

See your book for details.

Size Units for Width

Size can be specified using the following units:

- **px**: pixels (e.g. 5px)
- **em**: related to font size
(so 1em=12pt if the current font size is 12pt)
- **ex**: related to font size
(height of lowercase x)
- **pt**: points (1/72 inch)
- **pc**: pica (12 points)
- **%**: percentage of the entire box width

Shorthand for Box Widths

Shorthand for box widths
of border, margin and padding:

- 1 value: all edges
`margin: 10px;`
- 2 values: top+bottom, left+right
`border-width: 10px 15px;`
- 3 values: top, left+right, bottom
`padding: 4px 8px 10px;`
- 4 values: top, right, bottom, left
(so clockwise from top)
`border: 2px 4px 6px 8px;`

List Styles

- The bullets or counters can be set with
`list-style-type: disc|circle|square|decimal|lower-alpha|upper-alpha|lower-roman|upper-roman`
- To use your own image for the bullet/marker:
`list-style-image: url("mybullet.png");`
- To remove bullets or counters entirely:
`list-style: none;`

CSS Event Pseudo-Classes

CSS can be used to detect mouse events

- Not yet visited
`a:link { color: blue; }`
- Already visited
`a:visited { color: purple; }`
- Mouse goes over element
`a:hover { color: green; }`
- Element is being clicked
`a:active { color: red; }`

*:hover and :active
also work on other elements than links*

More Information on CSS

- “Beginning HTML and CSS”
chapter 7+8 (chapter 9 is optional)
- <http://www.w3schools.com/css/>