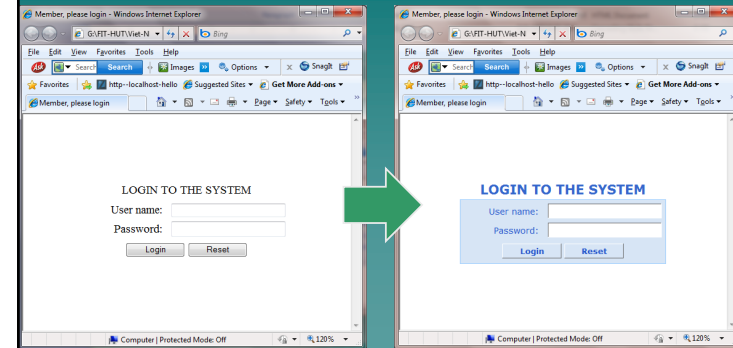


ICT 5 Web Development

Chapter 8. CSS

Before and after using CSS



2

- login.html
- ...
- <link rel="stylesheet" type="text/css" href="style.css">
- ...
- <table class="forumline" width="280" border="0" cellspacing="1" cellpadding="2">
- <tr class="formstyle"><td>

style.css

```
....
body
{
    font-family: Verdana,Tahoma,Geneva,Arial,Helvetica,sans-serif;
    font-size: 12px; ...
}
.formstyle
{
    background-color: #D7E5F5;
    font-family: Verdana,Tahoma,Geneva,Arial,Helvetica,sans-serif;
    font-size: 12px;
}
.forumline
{
    background-color: ...
} ...
```

3

Content

- ⇒ 1. Introduction to CSS
- 2. Specifying and applying style rules
- 3. Style class
- 4. Some useful properties
- 5. CSS box model

4

1. Introduction to CSS

- ◆ Cascading Style Sheet
- ◆ Created by Hakon Wium Lie of MIT in 1994
- ◆ Has become the W3C standard for controlling visual presentation of web pages

5

1.1. Benefits of CSS

- ◆ Simple syntax: easy to learn
- ◆ Powerful and flexible way to specify the formatting of HTML elements
 - Can define font, size, background color, background image, margins, etc
- ◆ Separates presentation (design elements) from content (structural logic)
 - HTML contains content and structure of a web page.
 - CSS defines a style of a web page – how the content is displayed

1.1. Benefits of CSS (2)

- ◆ Share style sheets across multiple documents or entire Web site
 - Easy to maintain consistent pages
 - Can update a common style → Reflected in all pages that use it
- ◆ Cost Savings
 - Reduced Bandwidth Costs
 - ◆ One style sheet called and cached
 - ◆ CSS require less code
 - Higher Search Engine Rankings
 - ◆ Cleaner code is easier for search engines to index
 - ◆ Greater density of indexable content

1.2. CSS Basics

- ◆ CSS defines the way that HTML elements should be presented:
 - Positioning (e.g. left, right or centered)
 - Font (size and family)
 - Text decoration (e.g. underlined)
 - Borders (solid, dashed, invisible)
 - Image usage (e.g. for backgrounds and bullets)

1.3. CSS Does Not...

- ◆ Re-order HTML
 - E.g. won't sort a table
- ◆ Perform calculations
 - Won't sum a shopping basket
- ◆ Filter
 - Won't decide what to show
 - Though JavaScript can set display or visibility of elements in order to achieve this
- ◆ These can all be done on the server
 - Or using XSLT or JavaScript on the client

1.4. Types of CSS Styles

- ◆ (Browser default)
- ◆ External styles
 - written in a separate document and then attached to various Web documents
 - External style sheets can affect any document they are attached to
- ◆ Internal styles (embedded styles)
 - embedded in the head of the document.
 - embedded styles affect only the tags on the page they are embedded in
- ◆ Inline Style
 - written directly in the tag on the document

10

Content

1. Introduction to CSS
- ⇒ 2. Specifying and applying style rules
3. Style class
4. Some useful properties
5. CSS box model

11

2.1. Specifying Style Rules

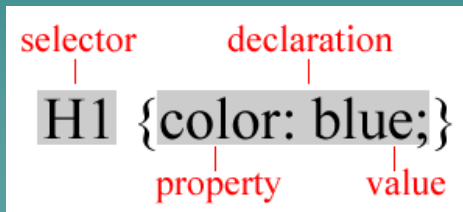
- ◆ General form of rule
`selector { property: value }`
- Or
`selector { property1: value1;
property2: value2;
...
propertyN: valueN }`
- ◆ Example
`H1 { text-align: center;
color: blue }`

Selector	Property	Value
H1	color	green
Declaration		

12

2.1. Specifying Style Rules (2)

- ◆ The *selector* is the link between the HTML document and the style. It specifies what elements are affected by the declaration.
- ◆ The *declaration* is that part of the rule that sets forth what the effect will be



13

2.1. Specifying Style Rules (3)

- ◆ Grouping selectors and rules

```
H1 { font-weight: bold }  
H2 { font-weight: bold }  
H3 { font-weight: bold }
```

→H1, H2, H3 { font-weight: bold }

→What is different?

```
b i{background-color:yellow;}  
b,i{color:blue;}
```

- ◆ A selector may have more than one declaration

```
H1 { color: green }  
H1 { text-align: center }
```

14

2.2. Applying styles to the document

- ◆ Inline style
 - Apply a style sheet to an individual element using the **style attribute**
- ◆ Embedded style
 - Apply the basic, document-wide style sheet for the document by using the **style element**
- ◆ External style
 - Link an external style sheet to the document using the **link element** or
 - Import a style sheet using the CSS @import notation.

15

2.2.1. Inline style

- ◆ Using Style attribute
- ◆ For individual elements

```
<H1 STYLE="color: blue; font-size: 20pt;">  
  A large purple Heading  
</H1>
```

16

2.2.2. Embedded style

- ◆ Using Style element
- ◆ Putting the style sheet inside a style element at the top of your document

```
<HTML>
  <HEAD><TITLE>Bach's home page</TITLE>
    <STYLE> H1, H2 { color: green } </STYLE>
  </HEAD>
  <BODY>
    <H1>Bach's home page</H1>
    <P>Johann Sebastian Bach was a prolific composer. Among his
      works are: <UL> <LI>the Goldberg Variations
        <LI>the Brandenburg Concertos
        <LI>the Christmas Oratorio </UL>
    <H2>Historical perspective</H2>
    <P>Bach composed in what has been referred to as the Baroque
      period.
  </BODY>
</HTML>
```

17

2.2.2. Embedded style (2)

```
<STYLE type="text/css">
  <!--
    H1, H2 { color: green }
  -->
</STYLE>
```

Bach's home page

Johann Sebastian Bach was a prolific composer. Among his works are:

- the Goldberg Variations
- the Brandenburg Concertos
- the Christmas Oratorio

Historical perspective

Bach composed in what has been referred to as the Baroque period.

18

Tree structures and inheritance

- ◆ Just as children inherit from their parents, HTML elements inherit stylistic properties.
- ◆ CSS property values set on one element will be transferred down the tree to its descendants

```
<STYLE TYPE="text/css">
  BODY { color: green }
</STYLE>
```



19

Overriding inheritance

- ◆ Sometimes children don't look like their parents.
- ◆ E.g.

```
<STYLE TYPE="text/css">
  BODY { color: green }
  H1 { color: navy }
</STYLE>
```

20

2.2.3. External style

- ◆ Using Link element
- ◆ This is true “separation” of style and content.
- ◆ Keeping all your styles in an external document is simpler

```
<HEAD>
<LINK REL="stylesheet" TYPE="text/css" HREF="styles/mystyles.css">
</HEAD>
```

```
/* mystyles.css - a simple style sheet */
body {
  margin-left: 10%;
  margin-right: 10%;
  color: black;
  background: white;
}
```

21

Content

1. Introduction to CSS
2. Specifying and applying style rules
- ⇒ 3. Style class
4. Some useful properties
5. CSS box model

22

3.1. Element Style Classes

- ◆ *Proceed* the HTML element by a period and a class name

```
// Define an "abstract" paragraph type
P.abstract { margin-left: 0.5in;
             margin-right: 0.5in;
             font-style: italic; }
```

- ◆ To use, supply the name of the style class in the CLASS attribute of the HTML element

```
<H1>New Advances in Physics</H1>
<P CLASS="abstract">
  This paper gives the solution to three previously
  unsolved problems: turning lead into gold,
  antigravity, and a practical perpetual motion machine.
```

23

3.2. Global Style Classes

- ◆ omit the element name

```
// Style available to all elements
.blue { color: blue; font-weight: bold; }
```

- ◆ To use, simply specify the style class in the CLASS attribute of the HTML element

```
<H2 CLASS="blue">A Blue Heading</H2>
<!-- Apply to a section of text -->
This text is in the default color, but
<SPAN CLASS="blue">this text is blue.</SPAN>
```

24

3.3. Styles through User-Defined IDs

- ◆ An ID is like a class but can be applied only once in a document

```
<HEAD>
  <TITLE>...</TITLE>
  <STYLE TYPE="text/css">
    <!--
      #foo { color: red; }
    -->
  </STYLE>
</HEAD>
<BODY>
  ...
  <P ID="foo">
    ...
</BODY>
```

25

Content

1. Introduction to CSS
2. Specifying and applying style rules
3. Style class
- ⇒ 4. Some useful properties
5. CSS box model

26

4.1. Useful Font Properties

◆ font-weight

- Relative weight (boldness) of font
 - **normal** | lighter | bold | bolder | 100 | 200 | ... | 900
- ```
H1 { font-weight : 200; }
H2 { font-weight : bolder; }
```

#### ◆ font-style

- Font face type within a family
  - **normal** | italic | oblique
- ```
P { font-style : normal; }
TH { font-style : italic; }
```

27

4.1. Useful Font Properties (2)

◆ font-size

- Either relative or absolute size of font
- pt, pc, in, cm, mm | em, ex, px, % | xx-large | x-large | large | **medium** | small | x-small | xx-small | smaller | larger

```
STRONG { font-size: 150%; }
P { font-size: 14pt; }
P { font-size: xx-large; }
```

◆ font-family

- Typeface family for the font
- ```
H1 { font-family: Arial; }
```

28



## 4.2. Useful Text Properties

### ◆ text-decoration

- Describes text additions or “decorations” that are added to the text of an element
- **none** | underline | overline | line-through | blink
- E.g. `P { text-decoration: underline }`

### ◆ vertical-align

- Determines how elements are positioned vertically
- top | bottom | **baseline** | middle | sub | super | text-top | text-bottom | %

### ◆ text-align

- Determines how paragraphs are positioned horizontally
- left | right | center | justify

29

## 4.2. Useful Text Properties (2)

### ◆ text-indent

- Specifies the indentation of the *first line of the paragraph*
- +/- pt, pc, in, cm, mm | +/- em, ex, px, %
- E.g. `P { text-indent: -25px }` /\* Hanging indent \*/

### ◆ line-height

- Specifies the distance between two consecutive baselines in a paragraph
- **normal** | number | pt, pc, in, cm, mm | em, ex, px, %
- .double { line-height: 200% }
- .triple { line-height: 3 } /\* 3x the font size \*/
- DIV { line-height: 1.5em }

30

## 4.3. Useful Foreground and Background Properties

### ◆ color

- Color of the text or foreground color
- color-name | #RRGGBB | #RGB | rgb(rrr, ggg, bbb) | rgb(rrr%, ggg%, bbb%)
- `P { color : blue }`
- `H1 { color : #00AABB }`
- `H3 { color : rgb(255, 0, 0 ) } /* red */`

### ◆ background-image

- Specifies an image to use as the background of region
- **none** | url(filename)
- `H2 { background-image: url(Bluedrop.gif); }`

31

## 4.3. Useful Foreground and Background Properties (2)

### ◆ background-repeat

- Specifies how to tile the image in the region
- **repeat** | **repeat-x** | **repeat-y** | **norepeat**
- `BODY {`
- `background-image: url(Bluedot.gif);`
- `background-repeat: repeat-x;`
- `}`

### ◆ background

- Lets you combine properties in a single entry
- `P { background: url(wallpaper.jpg) repeat-x }`

32



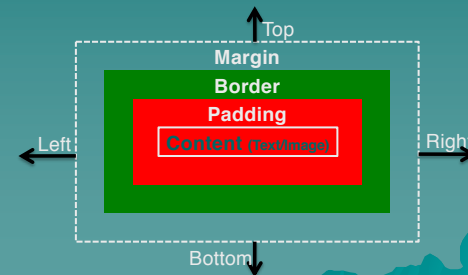
## Content

1. Introduction to CSS
2. Specifying and applying style rules
3. Style class
4. Some useful properties
- ⇒ 5. CSS box model

33

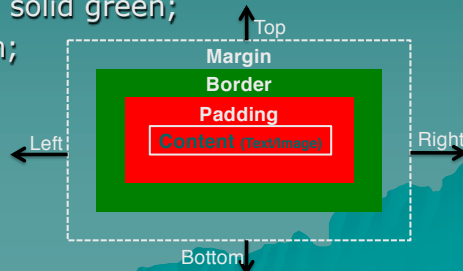
## 5. CSS Box Model

- ◆ Each HTML element have the rectangular "box"
- ◆ Each box has a content area and optional surrounding padding, border and margin area



## CSS Box Model - example

```
div#boxtest {
 background-color: red; color: white;
 padding: 1em;
 border: 1em solid green;
 margin: 1em;
}
```



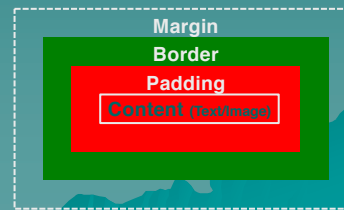
## CSS Box Model - color

- ◆ Padding - same as the element's background-color
- ◆ Border - may have its own color (border-color property)
- ◆ Margin - always transparent (same as its ancestor's background-color)



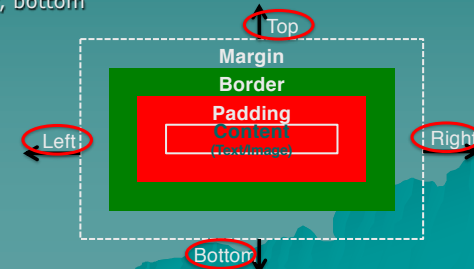
## CSS Box Model - edge sizes

- ◆ Inner edge (Black line): Content itself or CSS width and height property may define the size
- ◆ Padding edge (Red): IE + padding width
- ◆ Border edge (Green): PE + border width
- ◆ Outer edge (Dotted black line): BE + margin width



## CSS Box Model – width (1)

- ◆ margin, padding, border-width
  - Define the width for all directions at once
- ◆ margin-top, padding-top, border-top-width
  - Define the width for each specific direction
  - top, right, left, bottom



## CSS Box Model – width (2)

- ◆ Effective values for box width
- ◆ <length> - e.g. 10pt, 3px, 1.2em
  - Effective for border, padding, margin
- ◆ <percentage> - e.g. 10%
  - Effective only for padding, margin
  - Calculated with respect to the width of the generated box's containing block
- ◆ Thin, medium, thick
  - Effective only for border

## Border properties

- ◆ border-width or border-top-width (top, right, left, bottom)
  - Specify the line width
- ◆ border-color or border-top-color (top, right, left, bottom)
  - Specify the line color by the color name or RGB values
- ◆ border-style or border-top-style (top, right, left, bottom)
  - Specify the line style of box's border
  - Values: solid, dotted, dashed, double, groove, ridge, inset, outset, hidden
  - Special value "none" means width 0
- ◆ border or border-top (top, right, left, bottom)
  - shorthand property for setting the width, style, and color
  - e.g. "border: 1em solid black;"

## TIPS: Before your experiment of box model

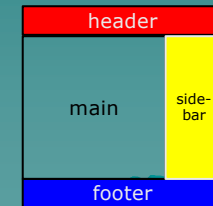
- ◆ Web browsers define their own default margin and padding width for some elements
- ◆ To override them, insert this CSS code at first

```
* {
 margin: 0;
 padding: 0;
}
```

## Page layout with CSS box and div element

- ◆ Typical page layout with four regions  
– main, header, footer, sidebar
- ◆ Enclosed by div elements with id attributes

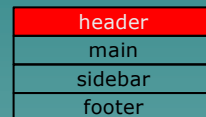
```
<div id="header">
 <p>Header content</p>
</div>
<div id="sidebar">
 <p>Sidebar content</p>
</div>
<div id="main">
 <h1>Main content</h1>
</div>
<div id="footer">
 <p>Footer content</p>
</div>
```



## Layout (1): header

- ◆ Reset default margin and padding to 0
- ◆ Specify header's property

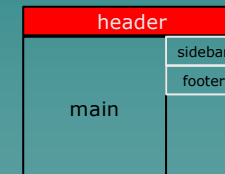
```
* {
 margin: 0;
 padding: 0;
}
body {
 background-color: white;
 color: black;
}
div#header {
 background-color: red;
 color: white;
}
```



## Layout (2): main

- ◆ Specify main region's properties
- ◆ Set its height and shift to left side

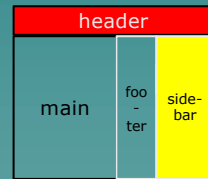
```
div#main {
 float: left;
 height: 400px;
}
```



## Layout (3): sidebar

- ◆ Specify sidebar's properties
- ◆ Set its height and shift to right side
- ◆ Restrict sidebar's width to 25% of the parent

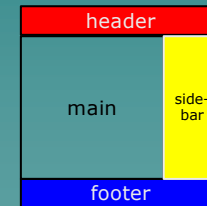
```
div#sidebar {
 float: right;
 height: 400px;
 width: 25%;
 background-color:
 yellow;
 color: black;
}
```



## Layout (4): footer

- ◆ Specify footer's properties
- ◆ Use "clear: both;" property  
– not be adjacent to an earlier floating box

```
div#footer {
 clear: both;
 background-color:
 blue;
 color: white;
}
```



Question?

