

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΔΙΑΔΙΚΤΥΟΥ

HTLM – CSS

Lecture 2

ver. 2.03

Jan 2011

Δρ. Γεώργιος Φ. Φραγκούλης



ΤΕΙ Δυτ. Μακεδονίας -
Μεταπτυχιακό Τμήμα

The HTML head Element

- The head element is a container for all the head elements. Elements inside `<head>` can include scripts, instruct the browser where to find style sheets, provide meta information, and more.
- The following tags can be added to the head section: `<title>`, `<base>`, `<link>`, `<meta>`, `<script>`, and `<style>`.

The HTML title Element

- The <title> tag defines the title of the document. The title element is required in all HTML documents.
- The title element:
 - defines a title in the browser toolbar
 - provides a title for the page when it is added to favorites
 - displays a title for the page in search-engine results

Example

- `<html>`
 `<head>`
 `<title>Title of the document</title>`
 `</head>`

 `<body>`
 The content of the document.....
 `</body>`

 `</html>`

The HTML base Element

- The <base> tag specifies a default address or a default target for all links on a page:

```
<head>
```

```
<base href="http://www.mypages.com/images/" />
```

```
<base target="_blank" />
```

```
</head>
```

The HTML link Element

- The <link> tag defines the relationship between a document and an external resource. The <link> tag is most used to link to style sheets:

```
<head>
```

```
<link rel="stylesheet" type="text/css"
```

```
href="mystyle.css" />
```

```
</head>
```

The HTML style Element

- The <style> tag is used to define style information for an HTML document.
- Inside the style element you specify how HTML elements should render in a browser:

```
<head>
<style type="text/css">
body {background-color:yellow}
p {color:blue}
</style>
</head>
```

HTML head Elements

Tag	Description
<u><head></u>	Defines information about the document
<u><title></u>	Defines the title of a document
<u><base /></u>	Defines a default address or a default target for all links on a page
<u><link /></u>	Defines the relationship between a document and an external resource
<u><meta /></u>	Defines metadata about an HTML document
<u><script></u>	Defines a client-side script
<u><style></u>	Defines style information for a document

The HTML meta Element

- Metadata is information about data. The <meta> tag provides metadata about the HTML document. Metadata will not be displayed on the page, but will be machine parsable.
- Meta elements are typically used to specify page description, keywords, author of the document, last modified, and other metadata.
- The <meta> tag always goes inside the head element. The metadata can be used by browsers (how to display content or reload page), search engines (keywords), or other web services.

Keywords for Search Engines

- Some search engines will use the name and content attributes of the meta element to index your pages.
- The following meta element defines a description of a page:
 - `<meta name="description" content="Free Web tutorials on HTML, CSS, XML" />` The following meta element defines keywords for a page:
 - `<meta name="keywords" content="HTML, CSS, XML" />`
- The intention of the name and content attributes is to describe the content of a page.

Example

```
<html>
<head>
<meta http-equiv="Refresh" content="5;url=http://www.mypages.com" />
</head>

<body>
<h1>Sorry! We have moved!</h1>
<h2>The new URL is: <a
    href="http://www.mypages.com">http://www.mypages.com</a></h2>
<p>You will be redirected to the new address in five seconds.</p>
<p>If you see this message for more than 5 seconds, please click on the link
    above!</p>
</body>
</html>
```

The HTML script Element

- The `<script>` tag is used to define a client-side script, such as a JavaScript. The script element either contains scripting statements or it points to an external script file through the `src` attribute. The required `type` attribute specifies the MIME type of the script.
- Common uses for JavaScript are image manipulation, form validation, and dynamic changes of content. The script below writes Hello World! to the HTML output:
 - ```
<script type="text/javascript">
document.write("Hello World!")
</script>
```

# HTML Entities

- Some characters are reserved in HTML. It is not possible to use the less than (<) or greater than (>) signs in your text, because the browser will mix them with tags.
- To actually display reserved characters, we must use character entities in the HTML source code. A character entity looks like this:
  - `&entity_name;` OR
  - `&#entity_number;`

# HTML Useful Character Entities

| Result | Description          | Entity Name | Entity Number |
|--------|----------------------|-------------|---------------|
|        | non-breaking space   | &nbsp;      | &#160;        |
| <      | less than            | &lt;        | &#60;         |
| >      | greater than         | &gt;        | &#62;         |
| &      | ampersand            | &amp;       | &#38;         |
| ¢      | cent                 | &cent;      | &#162;        |
| £      | pound                | &pound;     | &#163;        |
| ¥      | yen                  | &yen;       | &#165;        |
| €      | euro                 | &euro;      | &#8364;       |
| §      | section              | &sect;      | &#167;        |
| ©      | copyright            | &copy;      | &#169;        |
| ®      | registered trademark | &reg;       | &#174;        |
| ™      | trademark            | &trade;     | &#8482;       |

# Advanced HTML – CSS scripting

## What is CSS?

- **CSS** stands for **Cascading Style Sheets**
- Styles define **how to display** HTML elements
- Styles were added to HTML 4.0 **to solve a problem**
- **External Style Sheets** can save a lot of work
- External Style Sheets are stored in **CSS files**

# Styles Solved a Big Problem

- HTML was never intended to contain tags for formatting a document. HTML was intended to define the content of a document, like:
- `<h1>This is a heading</h1>`
- `<p>This is a paragraph.</p>`
- When tags like `<font>`, and color attributes were added to the HTML 3.2 specification, it started a nightmare for web developers. Development of large web sites, where fonts and color information were added to every single page, became a long and expensive process.
- To solve this problem, the World Wide Web Consortium (W3C) created CSS. In HTML 4.0, all formatting could be removed from the HTML document, and stored in a separate CSS file.
- All browsers support CSS today.



# Example

```
<html>
 <head>
 <link rel="stylesheet" type="text/css" href="ex1.css" />
 </head>
 <body>
 <h1>This header is 36 pt</h1>
 <h2>This header is blue</h2>
 <p>This paragraph has a left margin of 50 pixels</p>
 </body>
</html>
```

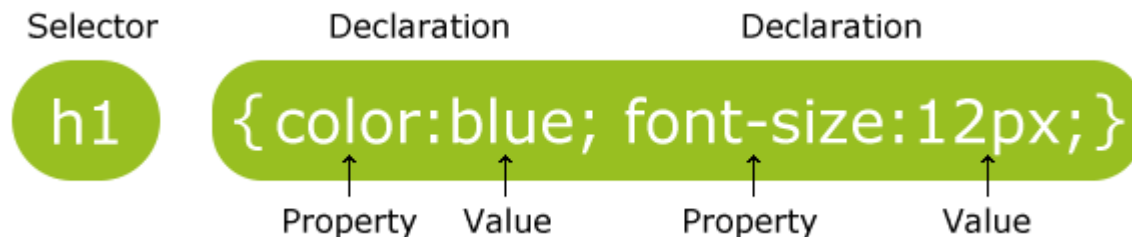
# Example (cont.)

This is the style sheet file (ex1.css):

```
body
{
background-color:yellow;
}
h1
{
font-size:36pt;
}
H2
{
color:blue;
}
p {
margin-left:50px;
}
```

# CSS Syntax

- A CSS rule has two main parts: a selector, and one or more declarations:



- The selector is normally the HTML element you want to style.
- Each declaration consists of a property and a value.
- The property is the style attribute you want to change.
- Each property has a value.

# Example

- To make the CSS more readable, you can put one declaration on each line, like this:

## Example

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

# CSS Comments

Comments are used to explain your code, and may help you when you edit the source code at a later date. Comments are ignored by browsers.

A CSS comment begins with "/\*", and ends with "\*/", like this:

```
/*This is a comment*/
p
{
 text-align:center;
 /*This is another comment*/
 color:black;
 font-family:arial;
}
```

# The id Selector

- The id selector is used to specify a style for a single, unique element.
- The id selector uses the id attribute of the HTML element, and is defined with a "#".
- The style rule below will be applied to the element with id="para1":

## Example

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

# Example

```
<html>
<head>
<style type="text/css">
#para1
{
text-align:center;
color:red;
}
</style>
</head>

<body>
<p id="para1">Hello World!</p>
<p>This paragraph is not affected by the style.</p>
</body>
</html>
```

# The class Selector

- The class selector is used to specify a style for a group of elements. Unlike the id selector, the class selector is most often used on several elements.
- This allows you to set a particular style for any HTML elements with the same class.
- The class selector uses the HTML class attribute, and is defined with a "."
- In the example below, all HTML elements with class="center" will be center-aligned:

## **Example**

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



# Example

```
<html>
<head>
<style type="text/css">
.center
{
text-align:center;
}
</style>
</head>

<body>
<h1 class="center">Center-aligned heading</h1>
<p class="center">Center-aligned paragraph.</p>
</body>
</html>
```

# Three Ways to Insert CSS

- There are three ways of inserting a style sheet:
  - External style sheet
  - Internal style sheet
  - Inline style

# External Style Sheet

- An external style sheet is ideal when the style is applied to many pages. With an external style sheet, you can change the look of an entire Web site by changing one file. Each page must link to the style sheet using the <link> tag. The <link> tag goes inside the head section:

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

An external style sheet can be written in any text editor. The file should not contain any html tags. Your style sheet should be saved with a .css extension. An example of a style sheet file is shown below:

```
hr {color:red;}
p {margin-left:20px;}
body {background-image:url("images/back40.gif");}
```

# Internal Style Sheet

- An internal style sheet should be used when a single document has a unique style. You define internal styles in the head section of an HTML page, by using the <style> tag, like this:

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

# Inline Styles

- An inline style loses many of the advantages of style sheets by mixing content with presentation. Use this method sparingly!
- To use inline styles you use the style attribute in the relevant tag. The style attribute can contain any CSS property. The example shows how to change the color and the left margin of a paragraph:

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

# Multiple Styles Will Cascade into One

- Styles can be specified:
  - inside an HTML element
  - inside the head section of an HTML page
  - in an external CSS file
- **Tip:** Even multiple external style sheets can be referenced inside a single HTML document.

# Cascading order

- What style will be used when there is more than one style specified for an HTML element?
- Generally speaking we can say that all the styles will "cascade" into a new "virtual" style sheet by the following rules, where number four has the highest priority:
  - Browser default
  - External style sheet
  - Internal style sheet (in the head section)
  - Inline style (inside an HTML element)
- So, an inline style (inside an HTML element) has the highest priority, which means that it will override a style defined inside the <head> tag, or in an external style sheet, or in a browser (a default value).
- **Note:** If the link to the external style sheet is placed after the internal style sheet in HTML <head>, the external style sheet will override the internal style sheet!

# Div

- Div (short for division) divides the content into individual sections. Each section can then have its own formatting, as specified by the CSS. Div is a block-level container, meaning that there is a line feed after the `</div>` tag.

For example, if we have the following CSS declaration:

```
.large {
 color: green;
 font-family: arial;
 font-size: 4pt;
}
```



# Div

- The HTML code  
`<div class="large">`  
This is a DIV sample.  
`</div>`  
gets displayed as

This is a DIV sample.

# Span

- Span is similar to div in that they both divide the content into individual sections. The difference is that span goes into a finer level, so we can span to format a single character if needed. There is no line feed after the `</span>` tag.

For example, if we have the following CSS declaration:

```
.largefont {
 color: red;
 font-family:arial;
 font-size: 6px;
}
```

# Span

The HTML code

Span is not at the `<span class="largefont">block level</span>`.

gets displayed as

Span is not at the **block level**.

# CSS Background

- CSS background properties are used to define the background effects of an element.
- CSS properties used for background effects:
  - background-color
  - background-image
  - background-repeat
  - background-attachment
  - background-position

# Background Color

- The background-color property specifies the background color of an element.
- The background color of a page is defined in the body selector:

## Example

```
body {background-color:#b0c4de;}
```

# Example

```
<html>
<head>
<style type="text/css">
body
{
background-color:#b0c4de;
}
</style>
</head>

<body>

<h1>My CSS web page!</h1>
<p>Hello world! This is an example.</p>

</body>
</html>
```

# Background color

- The background color can be specified by:
  - name - a color name, like "red"
  - RGB - an RGB value, like "rgb(255,0,0)"
  - Hex - a hex value, like "#ff0000"
- In the example below, the h1, p, and div elements have different background colors:

## Example

```
h1 {background-color:#6495ed;}
p {background-color:#e0ffff;}
div {background-color:#b0c4de;}
```

# Background Image

- The background-image property specifies an image to use as the background of an element. By default, the image is repeated so it covers the entire element.
- The background image for a page can be set like this:

## **Example**

```
body {background-image:url('paper.gif');}
```



# Text Color

- The color property is used to set the color of the text. The color can be specified by:
  - name - a color name, like "red"
  - RGB - an RGB value, like "rgb(255,0,0)"
  - Hex - a hex value, like "#ff0000"
- The default color for a page is defined in the body selector.

## Example

```
body {color:blue;}
h1 {color:#00ff00;}
h2 {color:rgb(255,0,0);}
```

# Text Alignment

- The text-align property is used to set the horizontal alignment of a text. Text can be centered, or aligned to the left or right, or justified.
- When text-align is set to "justify", each line is stretched so that every line has equal width, and the left and right margins are straight (like in magazines and newspapers).

## Example

```
h1 {text-align:center;}
p.date {text-align:right;}
p.main {text-align:justify;}
```

# Example

```
<html>
<head>
<style type="text/css">
h1 {text-align:center;}
p.date {text-align:right;}
p.main {text-align:justify;}
</style>
</head>

<body>
<h1>CSS text-align Example</h1>
<p class="date">Jan, 2011</p>
<p class="main">In my younger and more vulnerable years my father gave me some advice
 that I've been turning over in my mind ever since. 'Whenever you feel like criticizing
 anyone,' he told me, just remember that all the people in this world haven't had the
 advantages that you've had.'</p>
<p>Note: Resize the browser window to see how the value "justify" works.</p>
</body>

</html>
```

# Text Decoration

- The text-decoration property is used to set or remove decorations from text. The text-decoration property is mostly used to remove underlines from links for design purposes:

## Example

```
h1 {text-decoration:overline;}
h2 {text-decoration:line-through;}
h3 {text-decoration:underline;}
h4 {text-decoration:blink;}
```

# CSS Font Families

- In CSS, there are two types of font family names:
- **generic family** - a group of font families with a similar look (like "Serif" or "Monospace")
- **font family** - a specific font family (like "Times New Roman" or "Arial")

# Fonts Family

Generic family	Font family	Description
Serif	Times New Roman Georgia	Serif fonts have small lines at the ends on some characters
Sans-serif	Arial Verdana	"Sans" means without - these fonts do not have the lines at the ends of characters
Monospace	Courier New Lucida Console	All monospace characters have the same width

# Font Style

- The font-style property is mostly used to specify italic text. This property has three values:
  - normal - The text is shown normally
  - italic - The text is shown in italics
  - oblique - The text is "leaning" (oblique is very similar to italic, but less supported)

## Example

```
p.normal {font-style:normal;}
p.italic {font-style:italic;}
p.oblique {font-style:oblique;}
```

# CSS Lists

- In HTML, there are two types of lists:
- unordered lists - the list items are marked with bullets
- ordered lists - the list items are marked with numbers or letters
- With CSS, lists can be styled further, and images can be used as the list item marker.



# Different List Item Markers

- The type of list item marker is specified with the list-style-type property:

## Example

```
ul.a {list-style-type: circle;}
```

```
ul.b {list-style-type: square;}
```

```
ol.c {list-style-type: upper-roman;}
```

```
ol.d {list-style-type: lower-alpha;}
```

# CSS Tables

- The look of an HTML table can be greatly improved with CSS

## Table Borders

To specify table borders in CSS, use the border property. The example below specifies a black border for table, th, and td elements:

### Example

```
table, th, td
{
border: 1px solid black;
}
```

# Table Width and Height

- Width and height of a table is defined by the width and height properties.
- The example below sets the width of the table to 100%, and the height of the th elements to 50px:

## Example

```
table
{
width:100%;
}
th
{
height:50px;
}
```

# Table Text Alignment

- The text in a table is aligned with the text-align and vertical-align properties.
- The text-align property sets the horizontal alignment, like left, right, or center:

## Example

```
td
{
text-align:right;
}
```

# Table Color

- The example below specifies the color of the borders, and the text and background color of the elements:

## Example

```
table, td, th
{
border:1px solid green;
}
th
{
background-color:green;
color:white;
}
```