Writing Web Pages

Lecture 5 :

Meta Tags and Cascading Style Sheets

Nishanthi Dasanayaka
nishanthid@dcs.ruh.ac.lk

<meta> Tag

- Meta elements provide information about a given webpage, most often to help search engines categorize them correctly, and are inserted into the HTML code, but are not visible to a user looking at the site.
- Metadata is data (information) about data.
- Meta elements are typically used to **specify page description**, **keywords**, **author of the document**, **last modified**, and other metadata.
- <meta> tags always go inside the <head> element.
- Metadata is always passed as name/value pairs.
- There are two type of Meta tags.
    - <META HTTP-EQUIV="name" CONTENT="content“>
    - <META NAME="name" CONTENT="content">

HTTP-EQUIV & NAME Attributes

- HTTP-EQUIV attributes are equivalent to HTTP headers. Typically, they control the action of browsers, and may be used to refine the information provided by the actual headers.

- The NAME attribute is used to define information which is to be referenced outside of the document. This includes data passed to search engines and directories.

- Note : *If the http-equiv attribute is set, the name attribute should not be set.*

Meta name Attributes

- The name attribute specifies a name for the information/value of the content attribute.

Value	Description
application-name	Specifies the name of the Web application that the page represents
author	Specifies the name of the author of the document. Example: <meta name="author" content="Hege Refsnes">
description	Specifies a description of the page. Search engines can pick up this description to show with the results of searches. Example: <meta name="description" content="Free web tutorials">

Meta name Attributes

Value	Description
keywords	Specifies a comma-separated list of keywords - relevant to the page (Informs search engines what the page is about). **Tip:** Always specify keywords (needed by search engines to catalogize the page). Example: <meta name="keywords" content="HTML, meta tag, tag reference">

Syntax:

<meta name="*value*">

Meta name Attributes- Examples

<head>

<title>Marcia's Dog Training Page</title>

<meta name="**description**" content="This Web page contains information on dog training, including modern training techniques, training news, info on training centers and links to other dog-training sites.">

<meta name="**keywords**" content="Dog, training, puppy, techniques, centers, links, leash, teaching, commands, sit, stay, fetch, beg"> </head>

Meta name Attributes

- The description tag is often used by search engines such as Google to display a short summary of your web page in their search results pages.

- The keywords are used to help index your page, so that when people type in the word "fetch" into their favorite search engine, your page will be one of the pages returned.

<meta> http-equiv Attributes

- The name attribute specifies a name for the information/value of the content attribute.

Value	Description
content-type	Specifies the character encoding for the document. Example: <meta http-equiv="content-type" content="text/html; charset=UTF-8">
default-style	Specified the preferred style sheet to use. Example: <meta http-equiv="default-style" content="*the document's preferred stylesheet*">

<meta> http-equiv Attributes

Value	Description
refresh	Defines a time interval for the document to refresh itself. Example: <meta http-equiv="refresh" content="300"> Note: The value "refresh" should be used carefully, as it takes the control of a page away from the user

Client – Pull Document

- Client-pull documents are relatively easy to prepare.
- All you need to do is embed a <meta> tag in the header of your HTML or XHTML document.
- The special tag tells the client browser to display the current document for a specified period of time and then load and display an entirely new one, just as if the user had selected the new document from a hyperlink.

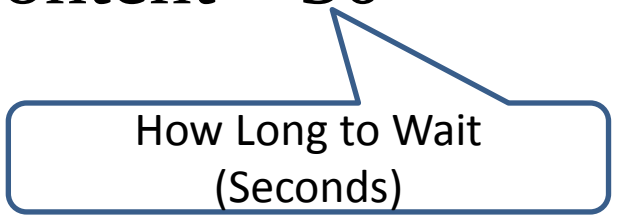
How ?

- <meta> tag is used
- Tags with an HTTP- EQUIV attribute is used for client –pull documents.

Example 1

- Refreshing with the same document

```
<meta http-equiv="refresh" content="30">
```



Refresh.html

Example 2

- Refreshing with the different document

```
<meta http-equiv="refresh" content="30 ; URL= abc.html">
```

Refresh_another.html

Drawbacks of Meta refresh tag

- Meta refresh redirects have been used by spammers to fool search engines.
- If the redirect happens quickly (less than 2-3 seconds), readers with older browsers can’t hit the “Back” button.
- If the redirect happens quickly and goes to a non- existent page, your readers may get stuck in a loop without seeing any content other than a 404 page.
- Refreshing the current page can confuse people.

Cascading Style Sheets (CSS)

CSS - Introduction

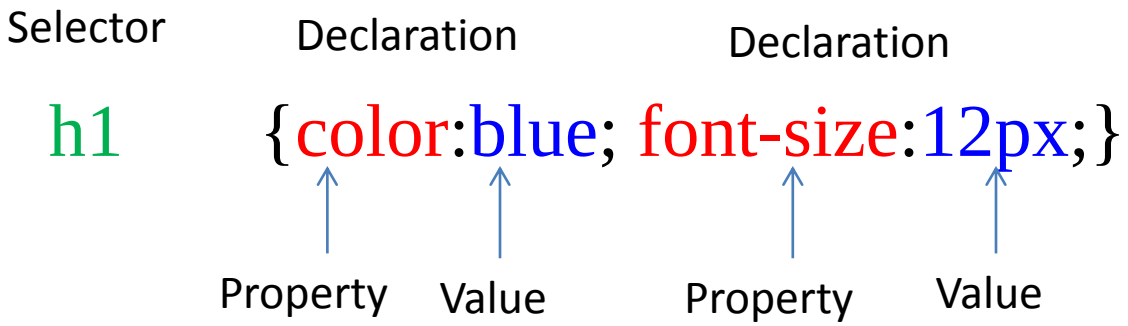
- Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation semantics (the look and formatting) of a document written in a markup language.
- CSS (cascading style sheets) is **a simple mechanism for controlling the style of a Web document separately from the structure of your document**.
- This separation of structure from content allows greater manageability and makes changing the style of your document easier
- CSS can **control text attributes** such as bold, italic
- CSS can **control page properties** such as spacing, font face, size line spacing, alignment, margins including the visibility and position of text and images.

Advantages of CSS

- Saves time
- Easy to change
- Keep consistency
- Give you more control over layout
- Use styles with JavaScript => DHTML
- Make it easy to create a common format for all the Web pages

CSS Syntax

- A CSS rule-set consists of a selector and a declaration block,



- The selector points to the HTML element you want to style.
- The declaration block contains one or more declarations separated by semicolons.
- Each declaration includes a CSS property name and a value, separated by a colon.
- A CSS declaration always ends with a semicolon, and declaration blocks are surrounded by curly braces.

CSS Selectors

- CSS selectors are to "find" (or select) HTML elements based on their element name, id, class, attribute, and more.

Types of CSS selectors

1. Element selector :

   - selects elements based on the element name

     Ex: h1 {color:blue;}

Types of CSS selectors

2. Id selector :
   - The id selector uses the id attribute of an HTML element to select a specific element.
   - The id of an element should be unique within a page, so the id selector is used to select one unique element!
   - Id selector is represented with # .

```
Ex:     #para1 {
                text-align: center;
                color: red;
            }
<p id="para1">Hello World!</p>
```

Note *: An id name cannot start with a number!*

Types of CSS selectors

3. Class selector :

- The class selector selects elements with a specific class attribute.
- To select elements with a specific class, write a period (.) character, followed by the name of the class.

```
Ex:     .center {
                text-align: center;
                color: red;
            }
<p class=“center">Hello World!</p>
```

Can specify that only specified html elements should be affected by a class

Ex: p.center { color:red;}

the <p> elements with class="center" will be color red.

Types of CSS selectors

4. Group selector :

- If we have elements with same styles then it is better to group the selectors to minimize the code.
- To group selectors, separate each selector with a comma.

Ex:

```
h1{
    text-align: center;
    color: red;
}
h2{
    text-align: center;
    color: red;
    }
```

After grouping selectors,

```
h1, h2 {
        text-align: center;
        color: red;

}
```

Apply CSS to the Document

Three ways to apply css

- **External style sheets**
- **Internal style sheets**
- **Inline style sheets**

External Style Sheets

- External style sheets keep the style rules in a **separate file**, apart form the HTML Web documents.
- The advantage of an external style sheet is that **you can use it to link to multiple pages.**
- By defining your styles in a single document and linking them to multiple pages, you only need to edit the style sheet to change the presentation style of all the pages linked to it.
- An external style sheet can be written in any text editor.
- The file should not contain any html tags.
- The style sheet file must be saved with a **.css** extension.

External Style Sheet - Example

```
body {
   background-color:lightblue;
}

h1 {
   color: navy;
   margin-left: 20px;
}

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

style1.css

```
<html>
   <head>
        <title> Web Page </title>
   </head>
   <body>
        <h1> Hello </h1>
        <p> This is first
paragraph </p>
   <body>
</html>
```

web.html

How to include css file ????

Link external CSS file to html page

- Create <link> tag element inside the html page to include a reference to external style sheet file.
- The <link> element goes inside the <head> section

  - Ex:

```
<head>

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

</head>
```

externalcss.html

Internal Style Sheets

- An internal style sheet may be used if one single page has a unique style.
- Internal styles are defined within the <style> element, inside the <head> section of an HTML page
- These style sheets are called embedded style sheets or document level style sheet.

- Syntax :

```
<html>
        <head>
                <style>
                </style>
        </head>
</html>
```

<style> Tag

- The type attribute within the style tag defines the page as one that uses embedded styles. The value is "text/css" a text application of CSS.

```
<style type="text/css">
        Selector {property1:value1; property2:value2}
</style>
```

internalcss.html

Inline Style Sheets

- An inline style may be used to apply a unique style for a single element.
- To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property

Syntax :

```
<h1 style="color:blue;margin-left:30px;">This is a heading.</h1>
```

inlinecss.html

CSS Backgrounds

- The CSS background properties are used to define the background effects for elements.

Background properties

1. background-color
    - The background-color property specifies the background color of an element.

Ex:-
```
body {
    background-color: lightblue;
}
```

backgroundcolor.html

CSS Backgrounds

Background properties

2. background-image

- 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

Ex:
```
body {
    background-image: url("image.gif");
}
```

backImage.html

CSS Backgrounds

Background properties

3. background-repeat

- Background-image property repeats an image both horizontally and vertically. But **background-repeat** property repeats an image either horizontally or vertically.

Ex:

```
body {
    background-image: url(“image.png");
    background-repeat: repeat-x;
}
```

backgroundRepeat.html

CSS Backgrounds

Background properties

4. background-position

- **Background-position** property is to change the position of the background image.

ex:

```
body {
    background-image: url("img_tree.png");
    background-repeat: no-repeat;
    background-position: right top;
    background-attachment: fixed;
}
```

backgroundposition.html

CSS Backgrounds

Background properties

5. background-attachment

- To specify that the background image should be fixed (will not scroll with the rest of the page), **background-attachment** property is used.

Ex:

```
body {
 background-image: url("img_tree.png");
 background-repeat: no-repeat;
background-attachment: fixed;
}
```

backgroundattach.html

CSS Pseudo-classes

- **Used to add special effects to some selectors**

Syntax:

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

Example

```
a:link {color:#FF0000;}      /* unvisited link */
a:visited {color:#00FF00;}  /* visited link */
a:hover {color:#FF00FF;}  /* mouse over link */
a:active {color:#0000FF;}  /* selected link */
```

pseudolinks.html

CSS Pseudo-classes and CSS classes

- **Pseudo classes can be combined with CSS classes.**

Syntax:

selector.CSS_class:pseudo-class {property:value;}

Example

a.red:visited {color:#FF0000;}

CSS Syntax

Red: class name

pseudoNcss.html

The :first-child Pseudo-class

- The **:first-child** pseudo-class matches a specified element that is the first child of another element.

Syntax:

```
selector:first-child {
    property: value;
}
```

Example

```
p:first-child{color:#FF0000;}
```

firstchild.html