Mobile Web-based Application Development

SYST24444

Week 2

Learning Goals

Overview of:

- HTML
- CSS
- JavaScript

Web Page: Role of HTML

HTML manages different structural types of web contents, like:

- Paragraph
- Block
- List
- Image
- Table
- Form
- Comments etc.

Web Page: Role of CSS

CSS deals with the presentation of the contents.It tells the browser how each type of element should be displayed.

- bgcolor
- textcolor
- Font style
- Paragraph style
- Table style
- Form style etc.

Web Page: Role of JavaScript

JavaScript tells the browser how to change the web page in response to events that happen.

- Clicking on something.
- Changing the value in a form input.

JavaScript introduce full programming concepts in the web page like:

- variables, arrays, objects, class
- control structures
- Functions

HTML Basics

- HTML = Hypertext Mark-up Language
- HTML is a plain-text file that can be created using a text editor like Notepad.
- When creating HTML files for the web, make sure you save them as .html – or they won't work.

Overview: Tags

- As a text document, HTML will contain *elements*, such as headers, titles, paragraphs, etc.
- These elements must be denoted in the programming script, which is done using *tags*
- HTML tags consist of a left angle bracket (<), a name, and a right angle bracket (>)
- For example: <title>
- Tags must also close. To do so, you incorporate a slash (/). A starting and ending tag would be:

<title> </title>

More Tags

- Any HTML document should contain certain tags:
  -<html>
  -<title>
  -<body>
  -headings (such as <H1>)
  -<paragraph>

Adding Attributes to Tags

- You can add attributes to tags to enhance your page.
- Added attributes go inside the brackets of the opening tag. example:
- <p align=center>

would center the paragraph

- <font color="Green">

will set the color of selected font as green

Colors

- Background and text colors are attributes of the “body” of the document.
- text=“#xxxxxx” determines your text color
- Bgcolor=“#xxxxxx” determines your background color
- Colors and codes for HTML can be found at http://hotwired.lycos.com/webmonkey/reference/color_codes/

Lists

- Lists are found inside the body, and are written as “<ul>” for an unordered list, or “<ol>” for an ordered (or numbered) list.
- List items are denoted by “<li>” and do not require closing tags.

Links

- The biggest thing that made HTML so popular was its ability to link to other documents or sections of documents.
- A link is indicated by <a> (anchor).
- The text that will become the link is identified with by <href> (hyperlink reference). For example
- Google

HTML Tables

```
<table border="1">
   <tr>
        <td>row 1, cell 1</td>
        <td>row 1, cell 2</td>
  </tr>
  <tr>
        <td>row 2, cell 1</td>
        <td>row 2, cell 2</td>
  </tr>
</table>
```

row 1, cell 1	row 1, cell 2
row 2, cell 1	row 2, cell 2

HTML Forms

```
<!DOCTYPE html>
<html>
<body>
<form>
        Login ID: <input type="text" name="loginID"><br>
        Password: <input type="password" name="pwd">
</form>
</body>
</html>
```

CSS

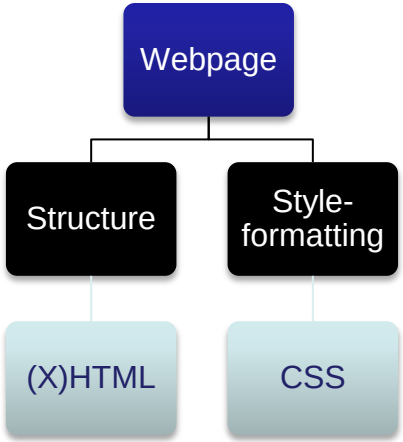
Cascading Style Sheets

CSS Basics

- **CSS** stands for **C**ascading **S**tyle **S**heets
- Styles define **how to display** (X)HTML elements
- Styles are normally stored in **Style Sheets**
- Multiple style definitions will **cascade** into one

Why to use Styles?

- Documents written with CSS are
  - more flexible
  - short
  - clear



- Basic formating tool
- Easy multiple document managment
- Save time by using selector classes
- New opportunities in formating

Basic Syntax

- Made up of three parts:

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

- The selector is normally the HTML element/tag you wish to define
- The property is the attribute you wish to change
- Every property has the value

Syntax

- If the value is multiple words, put quotes around the value

```
p {font-family: "sans serif"}
```

- To make the style definitions more readable, you can describe one property on each line

```
p
{
 text-align: center;
 color: black;
 font-family: arial
}
```

Grouping

```
h1,h2,h3,h4,h5,h6
{
color: green
}
```

- All header elements will be displayed in green text color

This is header h1

This is header h2

This is header h3

This is header h4

The class Selector

- With the class selector you can define different styles for the same type of HTML element.

```
p.right {text-align: right}
p.center {text-align: center}
```

Text color

```
<html><head>
<style type="text/css">
   h1 {color: green}
   h2 {color: #dda0dd}
   p {color: rgb(0,0,255)}
</style>
</head>
<body>
   <h1>This is header 1</h1>
   <h2>This is header 2</h2>
   <p>This is a
   paragraph</p>
</body>
</html>
```

This is header 1

This is header 2

This is a paragraph

Cascading order

1. Browser default
2. External style sheet
   - inside external *.css file
3. Internal style sheet
   - inside the <head> tag
4. Inline style
   - inside an HTML element

External Style Sheet

- Each webpage must link to the style sheet using the **<link>** tag
- Browser reads styles definitions from ***mystyle.css*** file

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

Internal Style Sheet

- Should be used when a single document has a unique style
- Defined in the head section by using the **<style>** tag

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

Exercise: Multiple Style Sheets

- An internal style sheet has following properties for the **h3** selector:

```
h3 { text-align: right;
font-size: 20pt }
```

- External style sheet has these:

```
h3 { color: red;
text-align: left;
font-size: 8pt }
```

- Your Web Browser has default formatting:

```
h3 { color: black;
font size: 10pt }
```

- What will be the format of <h3> tag?

  - `color: red;`
  - `text-align: right;`
  - `font-size: 20pt`

Background

- Control over the background color of an element
- set an image as the background,
- repeat a background image

- `background-color`
  - *`color-rgb`*
    `color-hex`
    `color-name`
- `background-image`
  - `url(URL)`
    `none`
- `background-repeat`
  - `repeat`
    `repeat-x`
    `repeat-y`
    `no-repeat`

JavaScript

JavaScript

- Interpreted language
- Code can be included in an HTML file
  - Downloaded with .html file
  - Interpreted by browser
  - Browser dependencies
- Client side, Server side
  - Client: In a browser, JavaScript embedded in html web pages
- Relation to Java
  - Similar in syntax
- JavaScript is case sensitive
  - E.g., null is not the same as Null, NULL, or any other variant.

General Uses of JavaScript

- Adds full programming language features to web scripting
  - E.g., variables, iteration, functions
- Dynamic creation of HTML code
  - HTML code can easily be output to the browser
  - Dynamically interpreted by browser

Embedding JavaScript in HTML - 1

- <script> tag
  - Can place a block of JavaScript code into HTML file
    - LANGUAGE attribute specifies version
      - E.g., <SCRIPT LANGUAGE="JavaScript1.2">
  - SRC attribute can specify name of a file containing JavaScript program code

Comments in JavaScript

```
// this is a comment
/* this starts a
   multiline comment
*/
<!-- is a single line comment also
- JavaScript does not recognize the -->
   closing bracket
```

Identifiers

- First character
  - ASCII letter
  - Underscore (_)
  - Dollar sign ($)
- Next characters
  - Letters, digits, underscores, $
- Cannot be the same as keywords (reserved words)

Primitive Types & Reference Types

- “Primitive” types represented “by value”
  - numbers
  - booleans
- Non-primitive types (e.g., arrays)
  - Represented “by reference”

Dynamic Typing & Operators

- The type of a variable can be changed
- E.g.,

  var car = “ford”

  car = 25
- String concatenation: “+”

  car = 2001 + “toyota”

Control Structures - 1

```
if (expression)
    statement
[ else statement2 ]
if (expression)
   statement
else if (expression2)
   statement2
```

Control Structures - 2

```
Switch (n) {
case:
   // statements
   break;
default:
   break;
}
```

Control Structures - 3

```
while (expression)
    Statement
do
   statement
while (expression);
for (initialize; test; increment)
   statement
// iterate through object properties
for (variable in object)
   statement
```

Event Handling

- User interface programs are often written in an event-driven style
- Program code (e.g., a function) is associated with kinds of user actions
  - E.g., mouse click
  - Tab key
  - Enter key
  - Page/frame entry/exit
- System (e.g., browser) calls function each time an event occurs

HTML Forms & JavaScript Events

- HTML forms
  - Input elements:
    - text fields, buttons, file selections etc.
- Browser JavaScript objects for each of these elements
- Event handlers
  - Each input element can have a JavaScript event handler

Events

- Events defined for HTML input elements
  - onfocus, onblur, onselect, onchange, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup
- Can attach function properties to each of these
- onclick, onchange – particularly important