Announcements

- Class Web Site:
  - http://www.cs.umd.edu/projects/passport/Classes/Spring2008/
  - You can find this link at the end of the main passport site
    - http://www.cs.umd.edu/projects/passport/webPage/
- E-mail Account
  - Get your own e-mail account if you don't have one
- Announcements' section in web site
- Rules regarding Forum Use
- Academic Integrity
- Reminder to your parents
- Be on time
- Slides

Firefox

- Browser we will use
  - http://www.mozilla.com/en-US/firefox/?from=getfirefox
- Extensions we would like to have
  - Error Console

Validation

- You can use W3C Markup Validation Service (http://validator.w3.org/) to validate your html.
- Also through firefox you can use ***tidy*** for html validation. Tidy also provides suggestions for code that cannot be validated.

HTML Basic Skeleton

- An html document has two main parts.
  - **Header** – provides information about the document
  - **Body** – contents of the page
- **Example 1 (htmlDoc1.html)**

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
    <head>
    <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-
    1" />
    <title>Template</title>
    </head>

    <body>
            <!--HTML CODE HERE-->
    </body>
</html>
```

- Let's validate the above document

CSS (Cascading Style Sheets)

- Official W3C standard for controlling presentation
- Specification: http://www.w3.org/TR/CSS21/
- Style Sheets
  - Text file with rules. It includes no html.
  - Style sheets files use a .css extension
  - Allows you to apply typographic styles (font size, line spacing, etc.)
  - Allows you to apply spacing instructions
  - Allows you to have page layout control
  - Allows you to generate smaller html files by avoiding redundancy in style specification
  - Allows you to easily update a collection of pages by updating only a single file
- Why CSS? Demo

Rules

- Rule - Basic element of a style sheet
- Rule - describes the formatting associated with a page element
- Rule format

selector declaration

selector – identifies what should be styled in a web document (e.g., h1, p)
declaration – what and how that portion of the web document should be modified.

- declaration - consists of *property: value* pair(s) enclosed in { }
- Examples:

```
h1 {color: green}
p  {font-size: 10px,
    color: red;
}
```

- Notice there is a space after the colon (;)
- Popular properties – color, font-family, font-size, text-decoration
- HTML Dog CSS Properties –
- http://www.htmldog.com/reference/cssproperties/

Types of Style Sheets

- **Inline**
  - Style information applied to specific tag (e.g., <p style=…”)
  - Avoid if possible.
- **Internal**
  - Using the <style> tag in the header of the html document
  - Convenient to provide own style to a specific page
  - Example: internalStyle.html
- **External**
  - External style sheet which web pages link to
  - Preferred approach
  - Example: externalFile.html and externalFile.css

CSS

- Why cascading?
  - Rules can come from different sources (inline, external file, etc.). The final set of rules that apply to a document comes from cascading all the sources.
- Rule Conflict Resolution
  - To resolve conflicts, styles defined at a specific level override those set at a higher level

    Example: you can set the color of body text to be blue but you can override to red the text in a list
  - When multiple style files are linked or imported the last will take precedence
- A child element inherits the same properties of its parent element (unless otherwise specified).

CSS Validator

- http://jigsaw.w3.org/css-validator/
- Notice you have three choices
  - by URI
  - by File Upload
  - by direct input

Kinds of Selectors

- **Type Selectors** – Those based on the name of an HTML tag
  - p { color: red; }
- **Pseudo-classes** – attached to selectors to specify a state. Four popular pseudo-classes are
  - a:link – initial color of a link
  - a:visited – color for a visited link
  - a:hover – color when mouse hover over link
  - a:active – color during the clicking of the link
- **Example:** selectors.html, selectors.css

Kinds of Selectors

- **Class Selectors** – Allow us to apply the same CSS rule to different elements
  - Use when you need to apply a style to many times in your document
  - Created with a period (also known as full stop)
  - Example: selectors.html, selectors.css
- **ID Selectors** – Like class selectors but appear only once in the document
  - Used when you need to apply a style only once in your document
  - Created using #
  - Example: selectors.html, selectors.css
- **Others** (will see them later on)

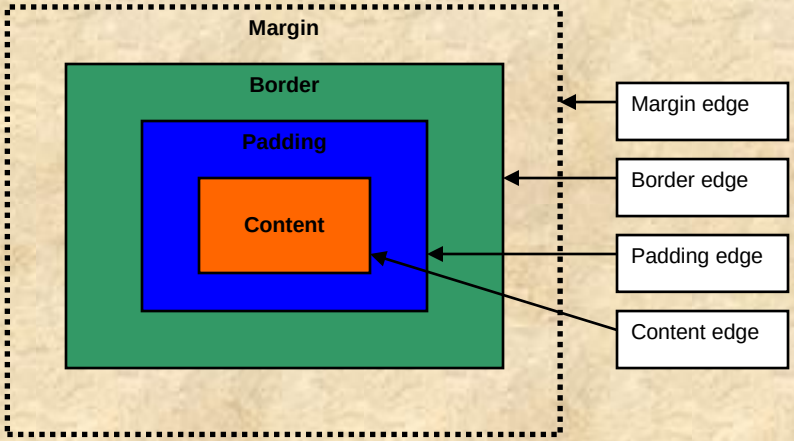
  Descendant Selectors, child selectors, attribute selectors, universal selectors
- **Example:** selectors.html, selectors.css

Additional HTML Elements

- and -
  - Allow you to delimit a section of the HTML body
-
  - Used to wrap inline content (e.g., text sequence)
- You can apply style to the sections defined by span and div.
- **Example:** spanDiv.html, spanDiv.css

Box Model

- Each block element (e.g., p) contains four edges (top, bottom, right, and left) defining a box
- Four sections can be identified with a block element
  - Content – what lies in the middle of the box (text, image, etc.)
  - Padding – surrounds the content
  - Border – surrounds the padding and represents the box border
  - Margin – surrounds the border



is a block element

- Defines a block-level entity
  - Browser starts a div element's content on its own line
- <body> also defines a block-level entity
- You can use your box model knowledge to add more style to your pages.
- **Example:** boxModel1.html, boxModel1.css

Box Model

- Let's explore more of the box model with the following examples
- **Example:** boxModel2.html, boxModel2.css
- **Example:** padding.html, padding.css
- The margins, borders, padding, and background properties of block elements (e.g., body, p, etc.) are not passed to its child block-level elements

Setting Size

- Percentages – size of the font is based on the size of the parent element
- Length units
  - centimeters (cm)
  - millimeters (mm)
  - points (pt) - 1 pt → 1/72 inch
  - picas(pc) - 1pica → 12 pts
  - inches (in)
- Relative
  - ex – height of the lowercase x in the font
  - px – pixels
  - em – refers to font size of parent element
    - Example: 2 em → twice the font size of the parent element
    - If the parent is body tag and no font size is specified then the size is looked in the user's preferences specified in the browser
    - Allow you to define scalable style sheets

Shorthand Property

- Shorthand Property- allows you to specify several properties by using only one.
- If you don't specify one of the properties a default value will be used.
- Commonly used shorthand properties
  - background
  - font
  - list-style
  - margin
  - border
  - padding
- **Example**: noShorthandProp.html, noShorthandProp.css, shorthandProp.html, shorthandProp.css

Font-size Keywords

- xx-small
- x-small
- small
- medium
- large
- x-large
- xx-large
- larger
- smaller
- **Example:** fontsizeKeywords.html,fontsizeKeywords.css

Background

- Background properties
  - background-color
  - background-image – location of image
  - background-repeat – how image repeats. Possible values
    - no-repeat – one instance of the image
    - repeat – tile
    - repeat –y → repeats on the y-axis
    - repeat –x → repeats on the x-axis
  - background-attachment – indicates attachment of the image to the containing element. Possible values are:
    - scroll → default value.
    - fixed → image will stay stationary as the scrolling takes place
  - background-position – Possible values (combination of are valid)
    - top, bottom, center, left, right
- Background images can be used in elements other than body
- **Example:** background.html, background.css
- Shorthand property: backgroundShorthand.html, backgroundShorthand.css

Generic Font Families

- sans-serif – (e.g., Verdana, Helvetica, Arial)
- serif – (e.g., Times New Roman, Georgia, Times)
- monospace – (e.g., Courier, MS Courier New)
- cursive – (e.g., Lucida Handwriting)
- fantasy – (e.g., Whimsey, Comic Sans)

JavaScript

- JavaScript – programming language that can appear in html pages.
- It allow us to:
  - To dynamically create web pages
  - To control a browser application
    - Open and create new browser windows
    - Download and display contents of any URL
  - To interact with the user
  - Ability to interact with HTML forms
    - You can process values provided by checkbox, text, textarea buttons

Execution of JavaScript Programs

- HTML parser – Takes care of processing an html document
- JavaScript interpreter – Takes care of processing JavaScript code
- HTML parser – must stop processing an html file when JavaScript code is found (JavaScript interpreter will then be running)
  - This implies a page with JavaScript code that is computationally intensive can take a long time to load

JavaScript

- Unlike html, JavaScript is a case-sensitive language
- JavaScript relies on the Unicode character set
- Let's go over several basic constructs that allow us to define JavaScript programs.
- Some definitions
  - string – Any set of characters in double quotes (" ")
  - function/method – An entity that completes a particular task for us. It can takes values necessary to complete the particular task and it can return values.
- Generating output with the document.writeln method
  - Allow us to add text to the html file (see **Example:** Writeln.html) by providing the required text in " "
  - You can specify html code and results of JavaScript constructs

JavaScript (Output)

- **Example:** Table.html
  - Illustrates how we can create a table using document.writeln
  - Notice how we can use the Date() to specify a particular date format. Date() is part of JavaScript and it is a method.
  - The + allow us to concatenate strings
    - Example: “Mary” + “Land” → “MaryLand”
    - Example: “Time is: “ + new Date()
  - Notice how we have specified the border size. If you use “ “ then the table borders will not be generated. You need to use single quotes.
  - Keep in mind that this example could have been written without using JavaScript. However you will see how by extending code similar to the one provided you can dynamically decide what your final html will look like

JavaScript (Variables)

- Variable – A memory location that can store a value. In JavaScript variables are declared using **var**

  var temperature;
- Variables names must start with a letter, underscore or dollar sign and can be followed by any number of letters, underscores, dollar signs or digits.
- Variables must be declared before they are used.
- A variable can hold different type of values
- Values we can assign to variables
  - Integer – 0, 10, 40, 6, -7
  - Floating-point – 3.2, .67, 1.48E-20
  - String literals – “hello”, “goodbye”
- Operators
- Assignment operator (=)
  - Typical arithmetic operators (+, -, *, /)
- **Example**: Variables.html)

Reserved Words

- Reserved words – words you cannot use as identifiers
- Some of them are:
  - break
  - do
  - If
  - catch

Spaces, Semicolons and Comments

- JavaScript ignores spaces, tabs, and newlines between tokens
- Use spaces to create nicely indented code
- The rules are usually one tab for indentation or three spaces. You need to satisfy this requirement in programming assignments.
- A semicolon is generally used to mark the end of a statement and is optional when a statement appears on a separate line. For example, the following two set of statements are equivalent

```
x = 1;
y = 2;

x = 1
y = 2
```

- In this course we will always use a semicolon to mark the end of a statement.

Comments

- Comments in JavaScript
  - Used to provide information to the programmer
  - Used to identify sections in your code
  - Ignore by the JavaScript interpreter
- Two types of comments
  - Inline comment - // This is a comment until the end of the line
  - Block comment –

    /* The following is a comment
    that spans several
    lines */
  - We can use a block comment for as a single-line comment
  - Block comments cannot be nested.

JavaScript (Dialog Boxes)

- We can perform input and output via dialog boxes
- Input via **prompt, Example:** InputOutput.html)
  - Notice we can define several variables at the same time
  - prompt is a function that reads that displays a dialog box with the specified titled. It can be used to read any data.
  - You can read numbers and strings via prompt

JavaScript (Loosely Typed Language)

- JavaScript is a loosely typed language
  - You do no need to specify a type for a variable
  - A variable can assume different types of values (not all at the same time)
- Code Snippet where variable first assumes a string value and then a numeric value

```
var value, input, ratePerHour = 8.0;
value = "University of Maryland College Park";
document.writeln("Check will be sent to: " + value + "<br />");
input = prompt("Enter number of hours:");
value = input * ratePerHour;
document.writeln("Total amount is: " + value);
```

Data Types

- Primitive data types in JavaScript
  - Numbers
  - Strings
  - Booleans
- Composite Data Types
  - Objects
  - Arrays
- All numbers are represented as floating-point values.
- To represent a single character using a string of length 1
- You can use ‘ ‘ or “ “ for strings although we will use “ “ in this class.
- Remember floating-point values are approximations.
- Special numeric values
  - Infinity
  - NaN – Not a Number
  - Number.MAX_VALUE – maximum value possible
  - Number.MIN_Value – smallest (closest to zero) number
  - Number.Nan – Not a Number
  - Number.POSITIVE_INFINITY
  - Number.NEGATIVE_INFINITY