Midterms

**HTML**

* What is HTML?
* Hypertext Markup Language
* Language used to create webpages
* What is HTML’s role in modern web development?
* To represent structure and content
* Who invented HTML?
* Tim Berners-Lee

- lead the invention of HTML

- also founded W3C, which defines the standards of the HTML

* Who oversees the development of HTML?
* WWW
* What are the versions of HTML?
* HTML 1.0 (1989-1990)
* HTML 1.3 - defined by TBL
* HTML 2.0 – RFC 1866 (1995)

-W3C was established

- HTML 3.0 – first W3C standard, defined by W3C through Recommendations

- XHTML

XML - standard information interchange format

- HTML 4.0 (1997)

- HTML 4.1 (1999)

* 3 versions
  + - Strict version (a way for developers to shift seamlessly to XHTML)
    - Transitional (allows use of deprecated elements)
    - Frameset
* HTML 5 (2014) - defined by WHATWG
* HTML 5.1 (2016)
* Which documents specify the HTML standards?
* W3C
* What is the syntax of HTML?

**<!DOCTYPE html>**

**<html>** -encloses all the content

**<head>** -contains meta information

**<title></title>**

**</head>**

**<body>** -actual content

**</body>**

**</html>**

**HTML 5**

* Heading content
* h1, h2, h3, h4, h5, h6
* Grouping content
* p element
* Root elements
* head and body

**TAGS**

**<html>**

**<head>** : title, base , link, meta , style

**<body>**

article, aside, nav, section

header, footer html 5 tags

main - main content

address – contact information

div – generic divider

h1, h2, h3, h4, h5, h6

p – paragraphs

hr – horizontal line

pre – pre-formatted text

blockquote- quotations (larger blocks)

**lists**

ol, ul

li

**description lists**

dl

dt, dd

a

em, strong, small, s

cite

q – quotations (single lines)

dfn –define

abbr –abbreviation

ruby, rb, rt, rtc , rp

data, time

code, var, samp, kbd

sup, sub

span

bdi, bdo

br- break

wbr- word break

ins, del

**Cascading Style Sheets**

* specify presentation aspects of structurally marked up documents
* developed by Hakon Wium Lie (CHSS) and Bert Bos (SSP)

Versions:

* + CSS 1, 1996
  + CSS 2.1, 2011
  + CSS 3
* CSS Preprocessors , CSS Frameworks
* Sass, Less, 960, Grid System, Bootstraps, Foundation, Materialize, etc.
* Sass – Syntactically awesome style sheet

**HTML/XHTML Stylesheets**

* Author styles
* External stylesheets
* Embedded styles
* Inline styles
* Use <Link> to attach stylesheets for external
* Use <Style> for embedded style, used for quick prototyping
* <Style> attribute used for inline
* User styles
* Preferences of the user (browser extensions)
* User agent styles
* Example default CSS 2.1 stylesheet for HTML 4

**CSS Statements**

* At-rules
* @charset – including other characters
* @import – including other content
* @media – media queries
* @font-face – include other fonts
* @keyframes – keyframe animations
* @page
* @document
* CSS Rule Sets (a.k.a CSS rules, style rules)
* Consists of a selector, followed by a brace-enclosed

**CSS Selectors**

* Selector
* Condition in a CSS-rule to determine which elements in the document tree are matched by the selector and are thus targeted by the formatting specified in the CSS rule declaration
* Matched elements
* Selector Syntax
* Chain of one or more sequences of simple selectors separated by combinators, with one pseudo-element possibly appended to the last sequence

**div#abc>p.xyz[title] + span::first-child::after**

* Sequence of simple selectors
  + Chain of simple selectors not separated by combinators
  + Always start with a type selector or a universal selector
  + Cannot contain other type selectors or universal selectors
* Group of selectors
  + Comma-separated list of selectors representing the union of all elements selected by each of the selectors in the list
* Simple Selectors
* Type selector
* P, div, span
* Universal selector
* \*
* Attribute selector - targets an element based on the value
* [attr]
* [attr = value]
* [attr ~ = value]
* [attr |= value]
* [attr ^= value]
* [attr $= value] CSS 3
* [attr \*= value]
* Class selector - makes use of a period
* ID selector
* Pseudo-class
* Dynamic pseudo-class

-link pseudo class

* :link
* :visited

-hover/user action pseudo class

* :hover
* :active
* :focus

-target pseudo class

* :target

-language pseudo class

* :lang()

-UI element pseudo classes

* :enabled
* :disabled
* :checked
* :indeterminate

-structural pseudo classes

* :root
* :first-child
* :last-child
* :only-child
* :Nth-child()
* :Nth-last-child
* :first-of-type
* :last-of-type
* :only-of-type
* :nth-of-type()
* :nth-last-of-type()
* :empty
  + Negation pseudo-classes
* :not()
* Combinators
  + Descendant combinator (whitespace)
  + Child-combinator (>)
  + Sibling combinator

-adjacent sibling combinators (+)

-general sibling combinator (~)

* Pseudo- elements
  + ::first-letter
  + ::first-line
  + ::before
  + ::after

**CSS Rule Precedence**

* By origin and importance
* !important

Order:

* user agent important declarations
* user important declarations
* author important declarations
* author normal declarations
* user normal declarations
* user agent normal declarations
* By Specificity
* inline style
* number of ID selectors
* number of class-selectors, attribute selectors, and pseudo-classes
* number of type selectors and pseudo- elements
* By Order
* Last one wins

**CSS Declarations**

* Properties
* Shorthand properties
* Vendor specific extensions
* Values
* Keywords
* Numbers

-integers, decimals etc.

* Dimensions

-length, angle, duration, frequency, resolution

* Length units

-font-relative : em, ex ,ch ,rem

-viewport-percentage: vw, vh , vmin, vmax

-absolute lengths: cm, mm, q, in, pt, pc, px

* Angle units: deg, grad, rad, turn
* Duration units: s, ms
* Frequency units: hz, khz
* Resolution units: dpi, dpcm, dppx
* Percentages
* URLs and URIs
* Colors
* Strings
* Functions: calc(), attr(), counter(), linear-gradient(), radial-gradient(),translate(), scale(), rotate(), etc.

**CSS Preprocessors and Frameworks**

* Use advanced syntax, converts syntax into something valid
* Using existing frameworks to create
* Sass - syntactically Awesome Style Sheet, also the most popular
* SCSS
* Bootstrap

**Javascript (client side)**

1. A Script is placed in a separate file. (external) It can be placed anywhere on the HTML file
2. It can also be embedded in the HTML file (internal)
3. Inline script (not desirable)

**TYPES**

* Top-level/global code - executes on the fly, non-member of a function code
* Function - has to be called to execute

**External Attribute**

* Defer - fetch and continue rendering the page
* Async - execution becomes asyncronous, compiles the script after it’s being fetched
* No Script - notifies users if the browser doesn’t support scripting
* Window - global object, access to other objects
* Document object - gain access to any elements, allows interaction between JS and HTML
* Outer text
* Outer HTML
* Inner HTML - allows html elements to be parsed
* Inner vs outer - outer replaces the whole markup

**Document**

* getElementByID - only global
* getElementByClassName
* getElementByName
* getElementByTagName

Query - Specify a css selector

Matches - sees if the object matches the selector

Query selector/all - for retrieving

- pass CSS selector tags

* .matches - to check if the body matches the head
* .style - changes the style

- values are always a string

Node Interface - the primary data type for the entire DOM

\* node value is always null, can also be used for changing node values

**Check Node Types**

* firstElementChild
* lastElementChild
* nextSiblingElement
* previousElementSibling
* parentElement

**Dynamic Content Creation**

* createElement()
* insertBefore()
* replaceChild
* removeChild
* shallowClone
* createDocumentFragment

Java Script (the defacto client - side scripting language)

- is an imperative programming language

- weakly typed

**Variables** - ECMA script 2016

* Var - the same inside a function, becomes a property of the global object

- only exists inside the function its declared on

* Let - varies depending on which bracket

- only exists inside the block it declared in.

- if declared in the top - level, variable is just standard

* Const - cannot be reassigned/ needs initialization

- if declared in the top - level, variable is just standard

**Data types**

Simple/primitive

* boolean = simplest

“falsy” = not really false but equivalent to false

“truthy” = not really true but equivalent to true

\*0 is considered a false

\* null, undefined or empty string is false

* Numbers - are only considered numbers
* Strings - no char type

- use backslash to escape a quote

- template literals [` `] for multiple lines

- string interpolation

* Undefined
* Null

Standard Objects - all reference types begin with object

**FUNCTIONS**

* Procedural
* All functions return a value
* Type coercion

- functions can be recursive

n! = n \* (n-1)

0! = 1

\*”throw” can throw anything

* - Functions can be nested
* - “this”, this object which you are invoking

“arguments”, arguments you use to call, organized like arrays.

- only exists in the function

* Functions can have default values
* Function rest parameter

**Structures**

* Array
* Wrapper Classes

**ARRAYS**

var emptyArray = new Array ();

var alsoEmptyArray = [];

Var arrayWithLengthFIve = new Array(5);

Var arrayWithOneElementWithValue5 = [5];

Var array = new Array(5, 10, 15);

Var sameArray = [5, 10, 15];

Var mixedElementType = [10, true, ‘hello’, new Date()];

Array destructing - extract element and assign to different values

Var array = [1, 2, 3, 4, 5];

Var[a,b,c,d,e] = array;

Va[m,n,… others] = array;

Var[, x,, y] = array;

- array indices can be non - contiguous

Var array = [1,2,3,4,5]

Array [10] = 10;

Array Methods

Mutator methods - change the array value “mutate”

Iteration methods - iterate through methods one by one and apply a function.

Objects - based on a prototype of an object

Var emptyObj = new object();

Var alsoEmptyObj = {};

Ex: student.idno = ‘2150387’;

Student[‘name’] = ‘Juan Dela Cruz’;