**MIDTERMS**

**HyperText Markup Language (HTML)**

**-** by Tim Burners Lee

- is the standard markup language for creating web pages and web applications.

- takes document using markup.

- Structure & Content (Main focus)

-Presentational / aesthetic (old format)

**Website**- is a collection of web resources such as wen pages and web applications.

- it is being host.

**World Wide Web Consortium (W3C) –** Tim Burners Lee

* + - Is an international community that develops open standards to ensure the long-term growth of the web.

Not already recommended codes in HTML:

* Character entities:
  + &nbsp –tab
  + &lt – less than
  + &gt – greater than
* <br>
* Tables –for tabular presentation.

**HTML versions:**

Current – 5.1

HTML 1.0 – 1989

2.0 – Nov 1995 RFC18661L

3.0 – superseded by 3.2

4.0 – December, 1997

4.1 – 1999

**Extensible HyperText Markup Language (XHTML)**

Version: 1.0 – January 26, 2000

Revised August 1, 2002

* + - It is used for data interchange
    - Wrappers up structure of a document

**Ian Hidson** – proposed to go back to HTML

**-** WHATWG

- later becomes HTML5 that is standardized in 2014

HTML 4.0

* **Transitional** – depreciated
* **Strict**
* **Frameset**

**The Elements of HTML**

**Index of Attributes**

Grouping

Categories

Flow:

* Heading
* Sectioning
* Interaction
* Phrasing
* Embedded
* Metadata

Content

* + - Flow content is expected
  + Content Model
    - Phrasing Content
  + Content Attribute
    - Global Attribute

Ex. Id, title, long, alt, src

**ARIA state and property attributes**

* + Global ARIA – attributes

**DOM interface**

* + - interface HTML Paragraph Element : HTML Element

Paragraph in HTML is Thematic a content

Content in w/c this element can be used: Embedded Content

Tag Omission in text HTML: No end tag

**Summary of HTML Elements**

Html  
 head  
 title, base, link, meta, style

Body  
 article, aside, nav, section

header, footer

main

address

div

h1, h2, h3, h4, h5, h6

p

hr

pre

blockquote

ol, ul

li

dl

dt, dd

figure

figcaption

a

em, strong, small

cite, q

dfn, abbr

ruby, rb, rt, rtc, rp --→ ruby anotations

data, time

code, vaar, samp, kbd

sup, sub

l, b, u, mark

bai, bdo

span

table

caption

colgroup, col

**CASCADING STYLE SHEET (CSS)**

CSS

* language used to specify the presentational aspects
* Developed by Hakon Wium Lie (CHSS) and Bert Bos (SSP)

Versions

* css 1 , css 2.1, css3

CSS Preprocessors, CSS Frameworks

* Sass, Less, 960 Grid System, Bootstrap, Foundation, Materializa, etc

**===HTML/XHTML STYLESHEETS===**

* author styles (external stylesheets,embedded styles, inline styles)
* user style
* user agent styles (example default css 2.1 stylesheet for HTML 4)

\* Reset CSS – overide the default stylesheet

**===CSS Statements===**

* At-Rules
  + @charset, @font-face @import ‘global.css’, @namespace, @document, @font-face, @key frames, @media, @page
* css rule sets (a.k.a CSS rules, style rules)
  + consults of a selector, followed by a declaration block.

**===CSS Selectors===**

* Simple Selector
  + either a type selector or universal selector followed by zero or more attribute selector , ID selectors or pseudo classes.
* Selector Syntax
  + Chain of one or more sequence of simple selectors by combinators.
* Type selector
  + matches the name of a document language element type.
* Universal selector
  + written "\*", matches the name of any element type.
* Attribute selector

[attr]

[attr=value]

[attr~=value] – space separated values

[attr|=value] –target long attribute

[attr^=value] -beginning (CSS3)

[attr$=value] –End (CSS3)

[attr\*=value] –Everywhere (CSS3)

* ID selector

- match an element instance based on its identifier.

-A CSS ID selector contains a "#" followed by ID value.

* Pseudo classes
  + classify elements on characteristic other than their name, attributes or content.
  + there's is implicit class associated with.
* Dynamic pseudo class
  + link pseudo-classes
    - link - link that have not yet visited.
    - visited - link that have been visited.
  + user action pseudo class
    - * hover
      * action - active a link
      * focus - move particular element
      * target pseudo classes
      * language pseudo classes

:lang()

* UI element states pseudo classes

-:enabled (CSS3)

-:disabled (CSS3)

-:checked (CSS3)

-:inderminate (css3)

* Structural

-: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-type

-:empty

negation

:not() - negates the logic of the selector.

* Combinators
  + descendant combinator (whitespace i.e. space, tab, line feed, carriage return, form feed)

--child combinator(>)

adjacent sibling combinator (+)

general sibling combinator (~)

Pseudo elements

::first-letter :first-letter

::first-line :first-line

::before :before

:: after :after

**===CSS Rule Precedence===**

* **by origin and importance**

1. Transition Declarations
2. Override declaration
3. User declaration
4. Animation Declaration
   * user agent important declarations
   * user important declarations
   * author important declarations
   * author normal declarations
   * user normal declarations
   * user agent nor mal declarations

* **by specificity**
  + inline – more specific
  + number of ID selectors
  + number of class selectors, attribute selectors and pseudo selectors
  + number of type selectors and pseudo elements
* **by order**

**===CSS Declarations===**

* **Properties**
  + **short hand properties**
    - background , font
  + **vendor specific extension (aka vendor prefixes)**
* **values**
  + **keywords**
    - **inherit,**
  + **numbers**
  + **measurements**
    - 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**
  + Rgb( , , ), #rgb,rgba, hsl, hsla
* **Strings**
* **Functions**
  + calc(), attr(), count(), linear-gradient(), translate(), scale(), rotate(), etc.

**Values and Units Module Level 3**

**CSS Preprocessors, CSS Framework**

* SCSS(Sassy Cascading Stylesheet), 60 Grid System, Bootstrap, Foundation, materials, etc.

# JAVASCRIPT CLIENT SIDE SCRIPTING

**Javascript**

* + - used for computations
    - It has scripting knowledge, and it is a C base language
    - It is a lightweight interpreted or JIT-compiled programming language with first-class functions
    - Standard – ECMA 262

ECMA 5 – Almost all browser support this

Different ways to script:

1. console.log(‘externally linked script….’);
2. <script> -(global)

console.log(‘embedded script….’);

function callme(){

console.log(‘you added me’);

}

</script>

1. <button onclick = ‘console.log(“inline script…”);’> click me </button>

**Script –**it should be executing in a ‘sandbox’

* **Defer**- to execute other part of the document
* **Async –** intermix in rendering, don’t wait until the whole document is executed.

**\* <no script>**  - only display if a user doesn’t have script

**Attribute**

* **Global attribute**

Example:

window.navigator.vendor(app version, etc)

window.screen.

window.document.getElementByID(‘h’)

h = document.getElementByID(‘h’)

<h1 id= “h”>

<p> Javascript</p>

</h1>

* **Certain attribute**

W3C:

Document Object Model Core

Document = Node

document extends node

**Interface Node**

**Node**- is the primary datatype for the entire Document Object Model.\

- represents a single mode in the document tree

|  |  |  |
| --- | --- | --- |
| Node Name | Node Type | Node Value |
| “#text” | 3 | “hi” |
| “#document” | 9 | null |

**Child Nodes**

Ex.

document.body.childNodes[0]

document.body.childNodes[length]

document.body.firstChild

document.body.firstChild.nextSibling

others:

insertBefore

replaceChild

removeChild

appendChild

hasChildNode()

**Variable, Let, Constant (var, let, const)**

**Variable-** associated with global content

Example:

<script>

var x = 100;

function f() {

var x =200;

{

var x = 300;

}

} </script>

**let-** not associated w/ global content

Example:

<script>

let x = 100;

function f() {

let x =200;

{

let x = 300;

}

} </script>

**Const-** cannot modify value

- have constant value

**Standard Objects**

**-** get to know standerd built-in objects Array, Booleans, Date, Error, Function, JSON, Math, Number, Object and others

**Expressions and Operatiors**

**-** learn more about the behavior of Java Scripts operators instance of typeOff, new, this, the operator precedence and more

**Array**

Example:

var emptyArray = new Array();

var alsoEmptyArray = [];

var arrayWithLengthFive = new Array(5);

var array = new Array(‘5’);

var sameArray = [‘5’];

**Methods**

* **Mutator methods**

**-** Method use to modify the target.

**-** It does change the target array.

* Accessor

- Don’t modify the target array

ex.

.pop() - remove the last element

.push() – adds into the end of array

.splice

.sort()

.every() – if a condition is satisfied, it wil return true

.map() – returns a certain values of the site.

.reduce() - accumulator