**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3.0 – superseded by 3.2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**

* Is a client-side scripting language that dynamically provides interaction to web applications such as
  + games
  + events
  + carousels
  + image galleries
  + responses to button clicks
* 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

Researches:

**CSS**

Cascading Style Sheets (CSS) is a style sheet language written in a markup language, is used for formatting Web pages layout. While HTML is for the structure and content of a Web pages, CSS is a language used to specify the presentation aspects of web pages. It is used to design HTML; define text styles, layout display, format table sizes and other formatting aspects for HTML.

**History**

CSS like HTML, W3C standardized CSS and make it open for all and released in 1996. It was first proposed and developed by Hakon Wium Lie (CHSS) on October 10, 1994 and Bert Bos (SSP) who become the co-author of CSS. But it was not a new idea that time, Tim Berners-Lee on 1990 wrote his NeXT browser/editor specify a simple style sheet.

**Versions**

**CSS 1**

Hakon Wium Lie and Bert Bos developed the first CSS and was standardized and recommended officially by W3C. It is the CSS level 1 and it was published December 17, 1996. It is capable of: formatting font properties; Changing of color of some elements such as text and backgrounds; Layout of text, tables, and images; Capable of making and editing paddings, border, margin and positioning of elements. It can also specify foreground and background color, as well as background images.

**CSS 2**

Developed by W3C, CSS level 2 was published and recommended by W3C on May 1998. It added some capabilities like making boxes that behaves like tables, availability of powerful selectors, specification of positioning of elements, availability of media types, style rules, added some font properties and bidirectional text. It was revised and called CSS 2.1, due to the fixing of errors and removes poorly supported features. CSS 2.1 become a Candidate Recommendation first on Febraury 25, 2004 and was developed further more until it was become a W3C recommendation on June 7, 2011.

**CSS 3**

CSS3 was divided into what they called ‘modules,’ meaning it was separated into several documents. Each module can be a recommendation by different people, but some specifications are implemented by any browsers. Some module have new capabilities, some have extended features. Drafts of CSS 3 was released June of 1999. 50 CSS 3 modules was released as of June 2012, but only four of these are considered formal recommendations such as: Media Queries, Namespaces, Selectors Level 3 and Color.

**CSS 4**

It doesn’t have a single CSS 4 specification, and there is no standard name CSS 4, but it is divided into modules, and it does have level 4 modules. These level 4 modules are preceding the functionalities of level 3 modules, like the modules Selectors, Image Values and Background & Borders.

References:

<https://www.w3.org/Style/LieBos2e/history/Overview.html>

http://www.corelangs.com/css/basics/versions.html