MIDTERM NOTES

CSS – language used to specify the presentation, developed by Hakon Wium Lie(CHSS) and Bert Bos(SSP)

Versions:

* CSS1
* CSS2.1
* CSS3

HTML/XHTML Stylesheets

* Author styles
* User style
* User agent styles (example default CSS 2.1 stylesheets for HTML 4)

Reset CSS - standart stylesheet

Author styles

* External stylesheets – separate document

Media types:

* Screeb
* Print
* Speech

Title attribute

* Embedded styles
* Inline CSS

CSS STATEMENTS

* At-rules
* @chatnet
* @import
* @media
* @font-face
* @key-frames
* @page
* CSS Rule Sets (a.k.a. CSS Rules, styles Rules)
* Consists of a selector, followed by a brace-enclosed declaration block, w/c contains zero or more semi colon

CSS Selector

* Selector - structured used as a condition
* Selector syntax - chain of one or more sequences of simple selectors separated by combinations, with one pseudo-element possibly appended to the last sequence
* Sequence of simple of selector

**HTML Element:**

<html>

<head> *metadata*

<title></title>

<base></base>

<link></link>

<meta> </meta>

<style></style>

</head>

<body>

<article></article>

<aside></aside>

HTML5 <nav></nav>

<section></section>

<header></header>

<footer></footer>

<main></main>

<address></address>

<div></div>

<h1>< h2>< h3>< h4>< h5>< h6>

<p>

<hr>

<pre>

<blockqoute>

<ol>

<li></li>

</ol>

<ul>

<li></li>

</ul>

<dl>

<dt></dt>

<dd></dd>

</dl>

<figure>

<figure caption>

</figure>

<a></a>

<em></em>

<strong></strong>

<small></small>

<cite></cite>

<dfn></dfn>

<abbr></abbr>

<ruby>,<rb>,<rt>,<rtc>,<rp> *ruby animations*

<continuation>

<data>,<time>*embedding machines readable data processing*

<sup></sup>

<sub></sub>

<a></a>

<i></i>,<b></b>,<u></u>,<mark></mark>

<bdi></bdi>

<bdo></bdo>*bidirectional*

<span></span>

<br>,<br>

<ins>

<del>

<table>

<caption></caption>

<colgroup></colgroup>

<col></col>

<thead></thead>

<tbody>

<tr>

<td></td>

</tr>

</tbody>

<tfoot></tfoot>

</table>

<form>

Label

Input

Button

Select

Data

</form>

MATHML – Mathematics Mark-up Language

**HTML Attribute:**

* Global Attribute
* ID
* Tittle
* Language
* Translate
* Lang
* Alt
* Src
* Src set
* Read only attribute – cannot be modified or change.
* Element Specific Attribute
* Data Attribute
* Role, ARIA Attribute
* WAI-ARIA, WCAG 20
* Index of HTML
* Index of attributes
* Content attributes

ARIA – Accessible Reach Internet Application

DOM –

Meta data element – Contain information about the document itself.

**CSS**

CSS – Cascading Style Sheet Language used to specify the presentation aspects (layout & formatting)

**Developed by :** Hokon Wium lie(CHSS), Bert Boss(SSP)

**VERSION:**

CSS 1: 1996

CSS2.1 : 2011

CSS3 – Divided in module in every specification

* CSS preprocessors – extension to CSS
* CSS framework
* SASS
* CCSS
* 960 Grid System
* Bootstrap
* Foundation
* Materialize

**HTML/XHTML Style Sheets:**

* Author Styles – Comes from the author of page
* External style sheets (recommended) – separate document for CSS and html

<head>

<link rel = “style sheet” type = “text/css” href = “ .css”>

</head>

* Attribute: media
* Media type: all, screen, print, speech
* Embedded style sheets

<head>

<style type = “text/css” media = screen (min-width : 5000px)>

body{width:90%;}

</syle>

Or

<style type = “text/css”>

@import url(.css)

@media(orientation : portrait);

@import “.css” print; /\*comment\*/

</head>

* Media queries
* Inside the html
* Inline style sheet (within element)

<p style = “border : px”>

* Inside HTML
* User style
* User agent style(Ex. Default CSS 2.1 stylesheet for HTML 4)

**CSS Statements**

* **At-Rules**

@charset

@import

@media

@font-face

@keyframes

@page

**CSS Rule Sets(aka CSS Rules, Style Rules) – consists of selector, declaration, block(i), property declaration**

**CSS Selector – Structure used as a condition in a CSS rule to determine the matched element**

* **Selector Syntax – Chain of one or more sequences of simple selectors separated by combinators with one pseudo – element**

**div.xyz > p#abc + ul :: after**

**div.xyz – simple selector**

**> + - combinators**

**::after – pseudo element**

* **Type Selector (p, h1, body)**
* **Universal Selector (\*)**
* **Attribute Selector ([attr])**

**[Attr = value] [Attr ^= value]**

**[Attr ~= value] [Attr $= value]**

**[Attr |= value] [Attr \*= value]**

* **Class Selector**
* **ID Selector**
* **Pseudo – Class**

**Dynamic pseudo-class**

**Link pseudo-classes**

**:link :visited**

**User Action pseudo-classes**

**:hover :active :focus**

**Target pseudo-class**

**:target**

**Language pseudo-class**

**:lang()**

**UI element status pseudo-class**

**:enabled :checked**

**:disabled :indeterminate**

**Structural pseudo-classes**

**:root :with-child()**

**Css2 :first-child :with-last-child()**

**:last-child :first-of-type**

**:only-child :last-of-type**

**DOM Tree – Document Object Model Tree**

**p: first-child – all <p> that is a first child**

**:only-of-type**

**:nth-of-type()**

**:nth-last-of-type()**

**:empty**

**Negation pseudo-class : not()**

* **Combinators**

**Descendant combinatory (whitespace, space, etc)**

**Child combinatory (>)**

**Sibling combinatory**

**Adjacent sibling combinator(+)**

**General sibling combinatory(~)**

**Pseudo Element**

**::first-letter :first-letter**

**::first-line :first-line**

**::before :before**

**::after :after**

**CSS RULE PRECEDENCE**

* **By origin and importance**

**User agent important declarations**

**User important declarations**

**Author important declarations**

**Author normal declarations**

**User normal declarations**

**User agent normal declarations**

* **By specifity**

**Inline style**

**Number of ID selectors, class selectors, attribute selectors, pseudo classes**

**Number of type selectors and pseudo element**

* **By order – the later declaration will be executed**

**CSS DECLARATIONS**

* **Properties**

**Shorthand properties**

**Vendor – specific extentions (aka vendor prefixes)**

* **Values**

**Declared values**

**Cascaded Value**

**Specified value**

**Keywords**

**Numbers (integers and reals in decimal notations)**

* **Dimensions**

**Length:**

**Length units:**

**Font-relative: em,ex,rem, ch**

**Viewport-percentage: vw, vh, vmin,vmax**

**Absolute lengths: cm, mm, q, in, pt, pc, px**

**Angle:**

**Angle units:**

**Deg, grad, rad, turn**

**Duration:**

**Frequency:**

**Frequency Units:**

**Hz, khz**

**Resolution:**

**Resolution Units: Dpi, dpcm, dppx**

**Percentage**

**URLs and URIs**

**Colors**

**Strings**

**Functions**

**CSS PREPROCESSORS, CSS FRAMEWORKS:**

* **SASS – Syntactically Awesome Style Sheet**

**-Allows run loops**

**-Allows variable $**

**-Ruby application**

**-Indentation to specify nesting**

* **LESS**
* **960 GRID SYSTEM**
* **BOOTSTRAP**
* **FOUNDATION**
* **MATERIALIZE**

**JAVASCRIPT**

**Client Side Javascript**

**OTHER SCRIPT:**

* **VB SCRIPT**
* **FLASH SCRIPT**

**<head>**

**<Script type = “text/javascript” src = ‘script.js’ > </script>**

**</header>**

**Inner HTML**

**Outer HTML**

**Inner text**

**Outer text**

**TYPE OF SCRIPT**

* **INLINE SCRIPT**
* **NO SCRIPT – for browsers with no javascript support**
* **EMBEDDED SCRIPT**
* **EXTERNALLY LINK SCRIPT**

**GET ELEMENTS METHOD:**

**getElements – set of nodes**

**getElementById**

**getElementByClassName**

**getElementByName**

**getElementByTagName**

**getSelection**

**getRootNode**

**QUERY SELECTOR METHODS:**

**querySelector**

**querySelectorAll**

**matches**

**body + p**

**body > p**

**childNodes – return all nodes everything is significant text = whitespace**

**children – return only element**

**parentNode**

**parentElement**

**hasChildNode**

**childElemetCount**

**firstChild**

**firstElementChild**

**lastChild**

**lastElementChild**

**firstChild.nextSibling**

**firstChild.nextElementSibling**

**lastChild.previousSibling**

**lastChild.previousElementSibling**

**appendChild – add to the end of document**

**insertBefore(new, reference)**

**replaceChild(replace, reference)**

**removeChild**

**parentNode.append() – Not fully supported in some browsers**

**cloneNode – copies the entry structure – duplicating content**

**applicable only to document**

**document.importNode()**

**document.adoptNode()**

**createDocumentFragment**

**addEventListener**

**removeEventListener**

**ELEMENT**

**NodeName**

**NodeType**

**NodeValue**

**var a – to declare variable use keyword var**

**typeOf – to know datatype**

**difference of var, let, const:**

**var – local variable**

**let – did not introduce as a global behavior , window property, stand-alone variable, function**

**const – cannot assign a value to it, not allow to change value**

**window = global object if did not declare var it will become a dynamic property**

**OBJECT TYPE – properties, method DOM**

* **ARRAY**

**Var emptyArray = new Array() :constructor**

**Var alsoEmptyArray = []**

**Var arrayWthLengthFive = new Array(5)**

**Var arrayWthOneElementWthValue= [5]**

**Array length is writable**

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

**Var sameArray = new Array[5, 10, 15]**

**Var mixedElementType = new Array (a, b, 1, 2, “asd”)**

**Var matrix = new Array(**

**new Array()**

**new Array()**

**);**

**Var multiDimensionalArray = [**

**[‘a’,’b’,’c’] [1,2,3[4,5],6,7] []];**

**ARRAY DESTRUCTURING:**

**Var [a,b**

* **BOOLEAN**
* **DATE**
* **ERROR**
* **NUMBER**
* **STRING**

**FUNCTION DECLARATION**

**Function Expression:**

**(function(a, b) {return a+b;}) (10, 20)**

**var add = new fuction (‘a, b’, ‘return a+b’);**

**var sum = add(10, 20);**

**Anonymous Function:**

**var subtract = function (a, b){**

**return a-b;**

**}**

**Arrow syntax**

**Var multiply = (a,b) => {return a\*b;}**

**Var product = multiply(10, 20);**

**Single Statement:**

**Var divide = (a, b) => a/b**

**No Arguments**

**Var zero = () => 0;**

**RECURSIVE FUNCTION**

**Function can be nested**

**Function arguments can have default values**

**FUNCTION PARAMETERS**

**Function fn(a,10,…..others)**