Understanding CSS

Sharad.s@imaginea.com

imaginea

Understanding CSS

Sharad.s@imaginea.com

imaginea

What is CSS

- Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation semantics (the look and formatting) of a document.
- CSS is designed primarily to enable the separation of document content from document presentation, including elements such as the layout, colors, and fonts.
- Improve content accessibility, provide more flexibility and control in the specification of

History

- December 1996 – CSS1
- May 1998 – CSS2
- April 2011 – CSS 2.1
- Nov 2011 – CSS 3 Proposal

Browser Support for CSS3

	MAC				WIN							
• Browser	SAFARI	FIREFOX	OPERA	CHROME	SAFARI	IE				FIREFOX	OPERA	CHROME
Version	5.1	11	11.62	18	5.1	6	7	8	9	11	11.61	18
RGBA	Y	Y	Y	Y	Y	N	N	N	Y	Y	Y	Y
HSLA	Y	Y	Y	Y	Y	N	N	N	Y	Y	Y	Y
Box Sizing	Y	Y	Y	Y	Y	N	N	Y	Y	Y	Y	Y
Background Size	Y	Y	Y	Y	Y	N	N	N	Y	Y	Y	Y
Multiple Backgrounds	Y	Y	Y	Y	Y	N	N	N	Y	Y	Y	Y
Border Image	Y	Y	Y	Y	Y	N	N	N	N	Y	Y	Y
Border Radius	Y	Y	Y	Y	Y	N	N	N	Y	Y	Y	Y
Box Shadow	Y	Y	Y	Y	Y	N	N	N	Y	Y	Y	Y
Text Shadow	Y	Y	Y	Y	Y	N	N	N	N	Y	Y	Y
Opacity	Y	Y	Y	Y	Y	N	N	N	Y	Y	Y	Y
CSS Animations	Y	Y	N	Y	Y	N	N	N	N	Y	N	Y
CSS Columns	Y	Y	Y	Y	Y	N	N	N	N	Y	Y	Y
CSS Gradients	Y	Y	Y	Y	Y	N	N	N	N	Y	Y	Y

Browser Support for CSS

	MAC				WIN							
Browser	SAFARI	FIREFOX	OPERA	CHROME	SAFARI	IE				FIREFOX	OPERA	CHROME
Version	5.1	11	11.62	18	5.1	6	7	8	9	11	11.61	18
CSS Reflections	Y	N	N	Y	Y	N	N	N	N	N	N	Y
CSS Transforms	Y	Y	Y	Y	Y	N	N	N	Y	Y	Y	Y
CSS Transforms 3D	Y	Y	N	Y	Y	N	N	N	N	Y	N	Y
CSS Transitions	Y	Y	Y	Y	Y	N	N	N	N	Y	Y	Y
CSS FontFace	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
FlexBox	Y	Y	N	Y	Y	N	N	N	N	Y	N	Y
Generated Content	Y	Y	Y	Y	Y	N	N	Y	Y	Y	Y	Y
DataURI	Y	Y	Y	Y	Y	N	N	Y	Y	Y	Y	Y
Pointer Events	Y	Y	N	Y	Y	N	N	N	N	Y	N	Y
Display: table	Y	Y	Y	Y	Y	N	N	Y	Y	Y	Y	Y
Overflow Scrolling	N	N	N	N	N	N	N	N	N	N	N	N
Media Queries	Y	Y	Y	Y	Y	N	N	N	Y	Y	Y	Y

imaginea

CSS Syntax

- The CSS syntax consists of a set of rules. These rules have 3 parts: a selector, a property, and a value.

```
selector { property: value }
```

- Grouping Selectors

```
h1,h2,h3,h4,h5,h6 {color:blue}
```

- Applying Multiple Properties

```
h1 { color:blue; font-family:arial,helvetica,"sans serif" }
```

- Readability

CSS Selectors

- In CSS, classes allow you to apply a style to a given *class* of an element.

```
.class-name { property:value; }

html-element-name.class-name { property:value; }

html-element-name.class-name.class-name { property:value; }

html-element-name.class-name .class-name { property:value; }

#id-name { property:value; }
```

CSS Selector

	Name	Example
`*`	star	`* {margin: 0; padding: 0;}`
`#`	hash	`#element-id {margin: 0; padding: 0;}`
`.`	Class	`.element-class{…}`
`X Y`	Descendent	`Element-1 Element-2 {…}`
`X`	Type	`P{…}`
`X:visited and X:link`	Pseudo	`X:visited and X:link`
`X+Y`	adjacent	`ul + p {color: red;}`
`X>Y`	Direct children	`div#container > ul { }`
`X~Y`	sibling combinator	`ul ~ p {…}`
`X[title]`	attributes selector	`a[title] {color: green;}`

imaginea

CSS Selector

	Name	Example
`X[href="foo"]`	attributes selector	`a[href="http://www.abc.com"] {…}`
`X[href*="nettuts"]`	* attributes selector	`a[href*="abc"] {…}`
`X[href^="http"]`	Start attributes selector	`a[href^="http"] {…}`
`X[href$=".jpg"]`	End attributes selector	`a[href$=“.jpg"] {…}`
`X[data-*="foo"]`	Data attributes selector	`a[data-filetype="image"] {…}`

`X[foo~="bar"]`	Space separated data	`a[data-info~="external"] {color: red;}`
		`a[data-info~="image"] {border: 1px solid black;}`
		`<a href="path/to/image.jpg" data-info="external image">`
		`Click Me, Fool </a>`

`X:checked`	Checked	`input[type=radio]:checked {…}`
`X:after`	after pseudo classes	`.clearfix:after { }`
`X:before`	Before pseudo classes	`.clearfix:before { }`
`X:hover`	Hover selector	`div:hover {…}`

imaginea

CSS Selector

	Name	Example
`X:not(selector)`	Not selector	`div:not(#container) {…}`
`X::pseudoElement`	fragment selector	`P::first-line {…}, P::first-letter {…}`
`X:nth-child(n)`	Nth child selector	`li:nth-child(3) {…}`
`X:nth-last-child(n)`	Nth last child selector	`li:nth-last-child(3) {…}`
`X:nth-of-type(n)`	Nth type selector	`ul:nth-of-type(3) {…}`
`X:nth-last-of-type(n)`	Nth last type selector	`ul:nth-last-of-type(3) {…}`
`X:first-child`	First child	`ul>li:first-child{…}`
`X:last-child`	Last child	`ul>li:last-child{…}`
`X:only-child`	Only child	`div p:only-child {…}`
`X:only-of-type`	Only type sibling	`li:only-of-type {…}`
`X:first-of-type`	First of type	`ul:first-of-type > li:nth-child(2) {…}`

CSS Units: Font relative length units

- rem Unit: root em -> The root part of the name refers to the root element, or the html element in HTML
- The rem value however stays consistent across the document, as 1rem is always the same size as the font-size of the html element (or root of which ever language you are using).
- ch Unit: The ch unit is equal to the width of the 0 (zero) glyph in the font used by the current element.

CSS Units: Viewport relative lengths

- There are three viewport relative lengths: vh, vw, and vm. These are relative to the size of the initial containing block, or in other words the viewport. If you resize the viewport, such as changing the size of the browser window, then the size of elements specified in these units will change.
- vw: This is relative to the width of the viewport. One vw unit is 100th of the width of the viewport. If the viewport is 1000 pixels wide then 10vw would map to 100px.
- vh: The vh unit works the same way as vw, but is relative to the height of the viewport instead.

CSS Styles

HTML5	Name	Properties

*	Font	font-family:
		`<p style="font-family:georgia,garamond,serif;">The text</p>`
		font-size:
		`<p style="font-size:20px;">The text</p>`
		font-size-adjust:
		`<p style="font-size-adjust:0.58;">The text</p>`
		font-stretch:
		`<p style="font-stretch:ultra-expanded;">The text</p>`
		font-style:
		`<p style="font-style:italic;">The text</p>`
		font-variant:
		`<p style="font-variant:small-caps;">The text</p>`
		font-weight:
		`<p style="font-weight:bold;">The text</p>`

CSS Styles

HTML5	Name	Properties

*	Text	color:
		`<p style=“color:red;">The text</p>`
		Text-align:
		`<p style=“text-align:right;">The text</p>`
		Text-indent:
		`<p style=“text-indent:50px;">The text</p>`
		Letter-spacing:
		`<p style=“letter-spacing:5px;">The text</p>`
		Word-spacing:
		`<p style=“word-spacing:10px;">The text</p>`
		Text-decoration: overline, underline, line-through, none
		`<p style=“text-decoration:blink;">The text</p>`
		Text-transform: uppercase, lowercase, capitalize
		`<p style="text-transform:uppercase;">The text</p>`
		Text-direction: rtl, ltr
		`<p style="text-direction:rtl;">The text</p>`
		Text-shadow:
		`<p style="text-shadow:4px 4px 8px blue;">The text</p>`
		Text-overflow: ellipsis-word
		`<p style="text-overflow:ellipsis-word;">The text</p>`
		White-space: normal, pre, and nowrap
		`<p style=“white-space:pre;">The text</p>`

CSS Styles

HTML5	Name	Properties
*	Background	Background-color: <p style="background-color:yellow;">Background</p> Background-image: <p style="background-image:url(/pix/smile.gif);">Background</p> Background-repeat: repeat x, y , norepeat <p style="background-image:url(/pix/smile.gif) no-repeat; ">Background</p> Background-position: <p style="background-image:url(/pix/smile.gif) no-repeat;background-position: 100px;">Background</p> Background-attachment: <p style="background-image:url(/pix/smile.gif) no-repeat; background-attachment:fixed; ">Background</p> Background-origin: border-box, padding-box and content-box. <div style="background-origin:border;">Background</div> Background-clip: backgrounds extends into the border or not. Border-box,padding-box,content-box <div style="background-clip:padding-box;">Background</div> Background-size: <p style="background-size: 275px 125px;">Background</p> Multiple Background: background: url(decoration.png) left top no-repeat, url(ribbon.png) right bottom no-repeat, url(old_paper.jpg) left top no-repeat;

CSS Styles

HTML5	Name	Properties

*	Border	Border-width:
		`<p style="border-width:1px;border-style:solid;border-color:blue;">This text has border styles applied using the border-width, border-style, and border-color properties.</p>`
		Border-style: dotted, solid, dashed, ridge, inset, outset, hidden, dashed
		Border-color:
		`border: 8px solid #000;`
		`-moz-border-bottom-colors: #555 #666 #777 #888 #999 #aaa #bbb #ccc;`
		Border-radius:
		`border-radius: 15px;`
		Box-shadow:inset
		`box-shadow: 10px 10px 5px #888;`

CSS Styles

HTML5	Name	Properties
✱	Margin	margin: <p style="border:1px solid orange;margin:20px;">This text has a margin of 20 pixels on all four sides.</p>
✱	Padding	padding: <p style="border:1px solid orange;padding:20px;">This text has a padding of 20 pixels on all four sides.</p>
✱	List	List-style: circle, square <ul style="list-style-type:circle;"> <li>List item one</li> <li>List item two</li> </ul> List Style Image: <ul style="list-style-image:url(/pix/printer_icon.gif);"> <li>List item one</li> <li>List item two</li> </ul> List Style Position: inside, outside <ul style="list-style-position:inside;"> <li>List item one</li> <li>List item two</li> </ul>

imaginea

CSS Styles

HTML5	Name	Properties

*	Positioning	relative:
		`<div style=“position:relative;">The container</div>`
		absolute:
		`<div style=“position:absolute;">The container</div>`
		fixed:
		`<div style=“position:fixed;">The container</div>`

*	Float	float:left,right `<h1 style="float:left;margin-right:10px;">CSS float</h1>`

*	layers	Z-index: 0-9999
		`<div style="background-color:red; width:100px; height:100px;`
		`      position:relative; top:10px; left:80px; z-index:2;">`
		`</div>`
		`<div style="background-color:yellow; width:100px; height:100px;`
		`      position:relative; top:-60px; left:35px; z-index:1;">`
		`</div>`

CSS Styles

HTML5	Name	Properties
*	Opacity	opacity: 0 - 1 `background: rgb(255, 0, 0) ; opacity: 0.2;`

*	color	HSL: Hue(0 red -120 green -240 blue-360 red), saturation(0-100%) & lightness(0-100%)
		`background-color: hsl(0,100%, 50%);`
		HSLA: Alpha(0-1)
		`background-color: hsl(0,100%, 50%,0.2);`
		RGBA: red, blue, green(0-255), alpha
		`background: rgba(255, 0, 0, 0.2);`

CSS Styles- UI Elements

HTML5	Name	Properties
*	Box-Sizing	Box-sizing: border-box, content-box `box-sizing: border-box;`
*	resize	Resize: both, vertical, horizontal `resize: horizontal;`
*	outline	outline:outline-offset `outline-offset: 12px;`

*	Attribute Selectors	Attribute selector:
		`[att^=val] – the “begins with” selector`
		`[att$=val] – the “ends with” selector`
		`[att*=val] – the “contains” selector`

CSS Styles- Transitions

- CSS3 Transitions are a presentational effect which allow property changes in CSS values, such as those that may be defined to occur on :hover or :focus, to occur smoothly over a specified duration – rather than happening instantaneously as is the normal behaviour.

HTML5	Name	Properties

*	transition	transition-property
		`transition-property: background-color, height, width;`
		transition-duration
		`transition-duration: 4000ms, 8000ms;`
		transition-timing-function
		`transition-timing-function: cubic-bezier(0.6, 0.1, 0.15, 0.8);`
		transition-delay
		`transition-delay: -5s;`

CSS Styles- Multi Column Layout

- W3C offers a new way to arrange text “news-paper wise”, in columns. Multi-column layout is actually a module on its own. It allows a web developer to let text be fitted into columns, in two ways: by defining a width for each column, or by defining a number of columns.

HTML5	Name	Properties

*	Column	Column-width
		`column-width: 13em;`
		Column-gap
		`column-gap: 1em;`
		Column-count
		`column-count: 3;`
		Column-rule
		`column-rule: 1px solid black;`

CSS Styles- Speech

- The CSS 3 Speech module removes some of the old properties and adds new ones. All of them are now assigned to the speech media type.

HTML5	Name	Properties

✱	Voice	Voice-volume
		`#voice-volume { -xv-voice-volume: x-soft; -xv-voice-balance: right; }`
		`#voice-balance { -xv-voice-balance: left; }`
		`#speech-cue { cue-after: url(ding.wav); }`
		`#voice-rate { -xv-voice-rate: x-slow; }`
		`#voice-family { voice-family: female; }`
		`#voice-pitch { -xv-voice-pitch: x-low; }`
		`#speech-speak { speak: spell-out; }`

CSS Styles- Media Queries

- CSS2 added support for the media="screen" way of defining which stylesheet to use for which representation of the data. CSS3 adds a new feature to this functionality, by adding media queries.
- Basically, this means you can change stylesheets based on for instance the width and height of the viewport. In a broader sense, this means as the spec puts it: “by using Media Queries, presentations can be tailored to a specific range of output devices without changing the content itself.”

What Next

Exercise