

IBM FULL STACK DEVELOPMENT (Module 2)

1. HTML Hyper Text Markup Language

- Enable documents as web pages
- Tags represent element of HTML page
- DOM tree : IN memory representation

Using HTML or XHTML

XHTML

- Tags need to be in lowercase
- Codes must be well-formed
- XML parser will stop processing if it encounters a situation where the syntax is not well-formed

HTML

- Case used does not matter
- Unmatched quotation marks, non-terminated and uncontained elements are allowed
- Syntax is less rigorous than XHTML syntax

Recap

- HTML5 includes features for:
 - Categorizing sections of web pages
 - Managing data, video, and audio tools
 - Developing cross-browser and cross-platform applications
 - Creating engaging user experience
- DOM tree is an in-memory representation of a document
 - Contains nodes that define the document type and structure, such as:
 - Headers and paragraphs
 - Text nodes
 - Comment nodes

Recap

- HTML5 themes provide:
 - Syntax that are compatible with HTML4 and XHTML1
 - New and refined APIs for video and audio elements, offline web apps, and drag and drop
- HTML5 web Apps have:
 - Improved search indexing with meta tags
 - Better page load times
 - Enhanced user experience
- HTML5 elements enable you to define web page structure and content

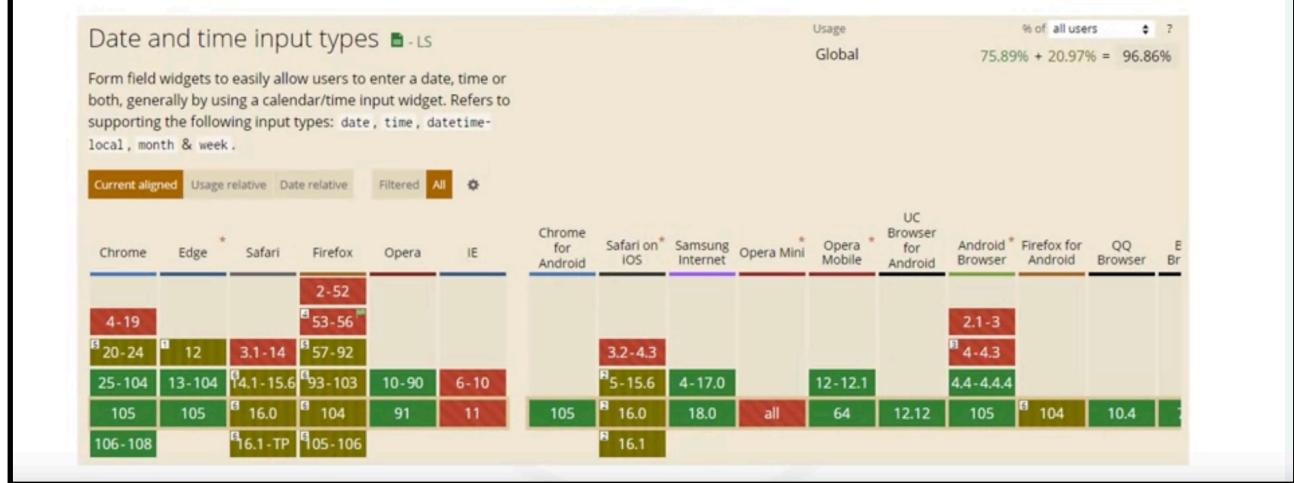
HTML document API - DOM tree accessors

PROPERTY	DESCRIPTION
head	Returns the <i>head</i> element
title	Sets or returns the <i>title</i> of the document
images	Returns an <code>HTMLCollection</code> of the <i>img</i> elements in the document
lastModified	Returns the date of the last modification to the document

HTML document API - DOM tree methods

PROPERTY	DESCRIPTION
getElementById('id')	Accesses the first element with the specific id
getElementsByName('tag')	Returns a nodelist of all elements with the specified HTML tag name
open()	Opens an output stream to collect the output from <code>document.write()</code>
write()	Writes JavaScript code to the document
close()	Closes the output stream previously opened with <code>document.open()</code>

HTML5 browser support tables



HTML5 specific tags (1 of 2)

TAG	DESCRIPTION
<article>	Content from an external source - news article, blog, or forum
<aside>	Content aside from the page content
<audio>	Used to embed sound content
<canvas>	Used to draw graphics
<datalist>	Provides a list of predefined options for input controls
<details>	Used to show or hide contents
<embed>	Embeds an external application or interactive content into page
<figcaption>	Caption for the figure tag
<figure>	Specifies self-contained content
<footer>	The footer of a document or section
<header>	Specifies a group of introductory or navigational elements
<keygen>	Specifies a key-pair generator field used in forms

- Tags provide control within an HTML5 document
- Tags provide structural elements, such as:
- <div> tag separates areas in a document into divisions
- <article>, <section>, <header>, and <footer> are specific elements
- The <aside>, <figure> and <figcaption> tags enables content grouping
- The <nav> tags enable grouping of navigational links

What elements can CSS control?

CSS controls a document's appearance and specifies style rules for the following web page elements:

- Fonts
- Text
- Colors
- Backgrounds
- Sizes
- Borders
- Spacing
- Positioning
- Visual effects
- Tables
- Lists

CSS format

```
html-tag-name
{
    css-property-key-1: css-value-1;
    css-property-key-2: css-value-2;
}
```

HTML Elements	Description
Tags	<ul style="list-style-type: none">• Any tags in the HTML code• For example: <a>, <div>, , or <label>
ID reference	<ul style="list-style-type: none">• Displayed with a preceding hash symbol (#)• For example: #id-of-html-tag
Class reference	<ul style="list-style-type: none">• Displayed with a preceding dot/period symbol (.)• For example: .class-of-html-tag

Base styles

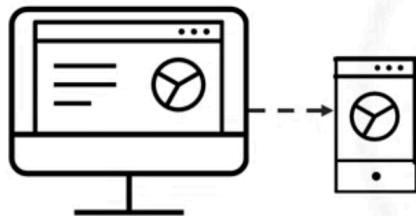
```
Body
{
background-color: #EEEEEE;
color: #000000;
margin: 0;
padding: 0;
text-align: left;
font-size: 100%;
font-family: sans-serif;
}
```

Guidelines for base styles

- Colors use Red-Green-Blue (RGB) hexadecimal light values
- Size use pixels, em, or a percentage
- Text can be aligned left, right, or center
- Floats can also be left or right
- Vertical alignments must be top, middle, or bottom
- Fonts can be any specific font or font family, such as serif, sans-serif, or monospace or even a downloadable font

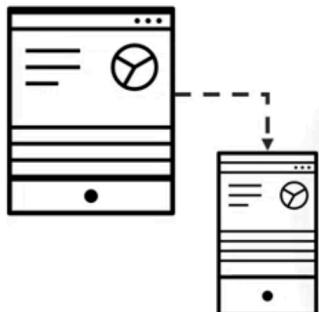
Choosing a layout

In a fluid layout:



- The height and width of elements are flexible
- The expansion or contraction is based on screen size
- The elements are specified using percentages and ems

Choosing a layout



In a fixed layout:

- You specify the height and width of elements
- Values remain the same regardless of screen size
- You specify elements using pixels

Types of CSS frameworks

Plain/Vanilla CSS	Utility-first framework	Component framework
Developers write style for all components	Provides utility classes	Provides pre-styled components
Provides complete freedom when styling	Simplifies references to CSS properties	Requires little knowledge of CSS
Requires time and effort	Provides freedom when styling components	Assists in keeping consistent styles
		Limits freedom when styling

Utility-first frameworks

Advantages	Disadvantages
Provides an easier way to reference CSS properties	Mixed styles reduces separations of concern, making HTML markup verbose
Utility classes scope to single-purpose CSS classes	Involves adding several classes to HTML markup
Implements CSS properties from HTML class attributes	Increases HTML download size and slows down web pages
Helps in consistency with color choices, spacing, typography, and shadows	

```
text-align: center; //CSS property  
text-center //Utility class
```

Tailwind CSS

Modifiers can help create responsive websites to fit any screen size

```

```

- Applies the class when user's screen is greater than 768 px wide
- Saves time when styling several different elements

Component frameworks

Advantages	Disadvantages
Provides pre-styled components and templates	Styles limited to what the framework provides
Reduces development time	Limits freedom of customization
Easier to maintain uniform style throughout website	Includes overhead code from unused components

Module Summary

In this module, you learned that:

- CSS creates a uniform look throughout each element of each page of the website.
- CSS is usually coded in external style sheets and creates base styles for a website.
- CSS frameworks assist in implementing UI elements and creating dynamic web pages.
- CSS has two types of frameworks:
 - Utility-first frameworks, which provide utility classes to help in building one's own styles and layouts.
 - Component frameworks, which provide a wide selection of pre-styled components and templates that can be implemented onto a website.
- Plain (Vanilla) CSS lets developers write the styles and layouts of a website.
- HTML5 elements provide structure and function to websites.
- HTML5 uses the <input> tag to allow users to input information.