

# FULL-STACK JAVASCRIPT UNIT 1

## VOCABULARY & REFERENCE

### HTML/CSS VOCAB

#### HTML

- **HTML:** Hypertext Markup Language - the language for defining the content and structure of a webpage.
- **HTML5:** The latest version of HTML. It has been the standard for many years now.
- **CSS:** Cascading Style Sheets - the language for styling HTML
- **Developer (Dev) Tools:** A panel built into the browser with a lot of features for investigating, debugging, and experimenting with HTML, CSS, and JavaScript on a webpage. One way to open it in Chrome is to right-click and select "Inspect".
- **HTML Element:** a piece that makes up the HTML document. There are different kinds of elements for different things (e.g. headings, paragraphs, buttons). Elements are usually defined by a start tag, some content, then an end tag. But empty (or void) elements have no content, and are defined by just a single tag.
- **HTML Tag:** used in HTML to define an element. They include the name of the element surrounded by angle brackets < >. Most elements need two tags, a start and an end.
- **Standalone Tag / Self-Closing Tag:** Tags where no closing tag is required, e.g. <hr />, <img />
- **HTML Attribute:** additional information supplied in the start tag of an element. Attributes usually consist of a name and a value. The value is usually surrounded by quotes ("").
- **Nesting:** HTML elements can be placed inside of each other. The inside (child) element's tags are placed between the start and end tags of the outside (parent/container) element.
- **Comment:** a portion of code that provides some explanation or clarity for humans reading the code but is essentially ignored by the computer.
- **HTML Form:** an element used as a container for various form input elements. It includes behavior for sending user input data to the web server, but it can also be used with JavaScript to collect the user input data.
- **Semantic Elements:** These HTML elements have names that describe their purpose. They provide great value for both readability and *accessibility* compared to using generic elements such as <div> and <span> for everything. Examples include <footer>, <nav>, <article>, and <table>.



- **Accessibility:** ensuring that a product or service is usable by people with a range of abilities. For example, consider how people with visual impairments or color blindness or who only use a keyboard to navigate will experience your website.

## CSS

- **CSS Rule:** a CSS selector combined with one or more declarations that identifies and defines styles for a set of HTML elements.
- **CSS Selector:** the part of a CSS rule that specifies which HTML elements to target. Types of selectors include element, id, class, attribute, descendant, and more.
- **CSS Declaration:** a property/value pair in a CSS rule that specifies a single style feature for the targeted elements. Multiple declarations must be separated by a semicolon (;).
- **CSS Property:** the name of a style feature. (e.g. margin, color, font-size)
- **CSS Units:** Some CSS values require units for specifying length/size. (e.g. px, rem, %, vh)
- **Specificity:** how the browser determines which CSS rule wins when there is a conflict on a given declaration.
- **Box Model:** the four adjustable parts that make up the rectangle layout of an element. From inside to outside, the parts are content, padding, border, margin.
- **Display Inline:** value of the display property that causes an element to flow in the direction of text. For English content, this is left-to-right, wrapping at each line.
- **Display Block:** value of the display property that causes an element to stack vertically, generally taking the entire width of the container and falling below the previous element and above the next element.
- **CSS Positioning:** a way of modifying or overriding the position of an element in the document using a combination of the position, left, right, top, and bottom properties.
- **Responsive:** designing and developing a website to work well on a continuum of sizes and types of devices. This includes factors such as screen size, screen orientation (landscape vs portrait), touch screen, and others.
- **Mobile-first:** responsive design and development that starts with the mobile experience (e.g. smaller screens) and then expands to the tablet, desktop and other larger device experiences.
- **Flexbox layout:** a configurable CSS layout where child elements are laid out along one dimension within their parent. This includes direction, justification, alignment, and wrapping.
- **Flexbox axis:** Flexbox includes two configurable axes--the main axis (where justify-content applies) and the cross axis (where align-items is used).
- **Media queries:** allows some CSS rules to take effect only based on specific attributes of devices. Commonly min-width and max-width are used to enable certain rules at specific screen width ranges. The exact widths specified are called *breakpoints*.
- **Viewport Meta Tag:** an HTML tag in the <head> that tells the browser how to adjust the webpage for mobile devices. We always specify this because the default behavior is



to zoom rather than giving CSS the control to use responsive development and media queries effectively.

## JAVASCRIPT VOCAB

- **JavaScript:** A programming language that allows us to give step-by-step instructions for how a webpage or other program should behave, for example when a user clicks a button or fills out a form.
- **Dynamically-Typed:** JavaScript is a dynamically-typed language. Variables don't have pre-set types. When the program runs, any kind of data can be used with each variable. In other statically-typed languages, variable types are pre-set or fixed in the code.
- **Variable:** a named container for values (data). It allows us to store data in computer memory and use it again later.
- **Declaration:** The code to create a variable, using *let*.
- **Assignment:** set the value of a variable.
- **Initialization:** the first assignment for a variable.
- **Data Type:** The various types of values that can exist in JavaScript. e.g. strings, numbers, objects, booleans, etc. Every piece of data in JavaScript has a specific type.
- **Character:** a single letter, number, punctuation mark, space, emoji, or other symbol.
- **String:** a sequence of characters. e.g. "Hello" or "As easy as 123!"
- **Undefined:** One kind of empty value--when a value does not exist or has not been set.
- **Null:** Another kind of empty value--the intentional absence of a value.
- **Concatenation:** + operator with strings. Makes a new string by joining the strings together.
- **Expression:** any piece of code that can be evaluated to a result.
- **Statement:** A statement is a complete line of code. These often end in a semicolon (;)
- **document.write():** a JavaScript function used to add text inside a document HTML
- **console.log():** used to write a message to the console. This is often used for testing purposes and debugging.
- **alert():** displays text in a dialog box that pops up on the screen.
- **prompt():** displays a dialog box that prompts the user for input. The input can be stored in a variable.
- **Array:** an ordered collection of data with numbered indexes.
- **Conditionals:** if/else statements, switch statements, and the ternary operator.
- **For Loop:** repeatedly executes a block of code using a counter variable. Useful for looping a fixed number of times.
- **While Loop:** iterates as long as a given condition evaluates to true. The condition evaluates before each iteration of the loop, so it may not execute at all if the condition starts out false.
- **Do... While Loop:** iterates as long as a given condition evaluates to true. The condition evaluates after each iteration of the loop, so it will always run at least once.



- **DOM:** Document Object Model - an interface that allows JavaScript to access and modify HTML

## VERSION CONTROL VOCAB

- **Command Line / Command Prompt / Terminal:** a program for running programs and giving the computer instructions by typing them. It is the best or only way to do many development tasks, and it allows certain tasks to be automated.
- **File Path:** a series of folder/file names, separated by slashes, that indicates the location of a file.
- **Absolute Path:** the full path to a folder starting at the root. The absolute path always starts with a slash or the drive letter such as "C". (e.g. /Users/grant/projects/lab1)
- **Relative Path:** gives the location of a file or folder relative to the current directory (e.g. images/cow.png)
- **Root Directory:** The "highest" folder. Every other folder is somewhere inside this. On Windows, it's usually "C:\". On Mac it's a single forward slash "/".
- **Home Directory:** Each user of the computer has their own home directory that's named after their username. This is the folder that has all their stuff such as documents, pictures, and desktop in it. (e.g. /Users/grant) The shortcut for this folder on the command line is tilde (~).
- **Version control:** (sometimes called source control) a system that manages history and changes to a program, website, or other collection of files.
- **Git:** an open source, distributed version control system originally developed by Linus Torvalds.
- **GitHub:** a social coding service that offers hosting for software projects that use Git as their source control.
- **Git Repository (Repo):** a database where Git stores snapshots of all the files in a project as they change over time. If you have multiple projects, each will need its own Git repository.
- **Local Repository:** a repository on your computer where Git tracks files in a folder.
- **Remote Repository:** a copy of a Git repository stored online, for example on GitHub.com.
- **Working Directory:** the folder where project files are actively worked on.
- **Commit:** a snapshot of the entire project at a point in time. e.g. `git commit -m 'did a thing'`
- **Staging Area:** where changes are queued up to be committed. Changes must be added to the staging area before a commit. e.g. `git add readme.md`
- **Git Status:** a command that displays information about the state of files and commits in the current Git repository. (e.g. `git status`)
- **Initialize (Git Init):** Turn the current folder into a brand new Git repository. (e.g. `git init`)



# COMMON HTML ELEMENTS

Where to find more elements and learn more? [W3Schools](https://www.w3schools.com/html/)

## Block Elements

- **<h1> ... <h6>** - Six levels of section heading. Typically appear bold with <h1>s being the largest.
- **<p>** - A paragraph of text. Typically has some top and bottom margin by default.
- **<div>** - A generic block element for organizing parts of the page.
- **<section>** - A section of page content, typically with a heading. For example, a portion of an article that is split into sections or a tab for pages that have tabbed content.
- **<header>** - Introductory and navigational content. For example, the main banner at the top of a webpage.
- **<main>** - The dominant contents of the page. A page should have only one visible <main> element.
- **<footer>** - Footer content typically at the bottom of the page. For example, this might include a site map, copyright, and links.
- **<nav>** - A section of the page that contains navigation links. For example the main navigation at the top or side of a site or a table of contents.
- **<article>** - A complete, self-contained portion of content. For example, an entire blog post, a news article, or a user-submitted comment.

## Inline Elements

- **<span>** - A generic inline element for organizing part of the page, typically just a small amount of content.
- **<em>** - Used to emphasize a word or phrase. It typically displays in italics.
- **<strong>** - Indicates strong importance of a word or phrase. It typically displays as bold.
- **<img>** - Defines an image. The **src** attribute provides a path for the image. The **alt** attribute provides a description of the image for users who are not able to properly load the image
- **<a>** - Anchor tag (i.e. link). The **href** attribute indicates the target page. The content between the start and end tags is what the user can click on. The optional target element indicates a different window/tab in which to open the link.

## Other Elements

- **<ol>** - Ordered list. A number list.
- **<ul>** - Unordered list. A bulleted list.
- **<li>** - List item. An item in either an ordered list or an unordered list.
- **<table>** - A table of data
- **<tr>** - A row in a table
- **<th>** - A header cell in a table



- `<td>` - A normal "data" cell in a table

## CSS SELECTORS

Selector	Description	Examples
element	elements with the given tag	p, a, section
class	elements with the given class	.primary, .top-nav
id	element with the given id	#hello, #checkout-button
combination	an element must match all of the parts	p.intro, div#results, .nav-link.active

## COMMON CSS PROPERTIES & UNITS

Where to find more properties or learn more about these? [W3Schools](#) or [MDN Web Docs](#)

Property	Description	Examples
color	color of text	<i>see colors below</i>
background-color	background color for an element	<i>see colors below</i>
font-family	typeface (i.e. font) for text	categories: serif, sans-serif, monospace, or names: Arial, 'Open Sans', 'Times New Roman'
font-size	size for text	<i>see units below</i>
font-weight	used to make text bold	bold, normal
font-style	for italic text	italic
text-decoration	for changing decorations	underline, strikethrough
text-align	for text justification	left, right, center
border-width	thickness of border around an element	<i>see units below</i>
border-style	style of the border	solid, dashed, dotted, none
border-color	color of the border	<i>see colors below</i>



border	shortcut for width, style, and color together	2px solid gray
height	height of an element	<i>see units below</i>
width	width of an element	<i>see units below</i>
margin-top margin-right margin-bottom margin-left	the area outside the element (distance from other elements)	<i>see units below</i>
margin	shortcut for setting all 4 margins	2px 2px 3px 2px 3px 4px 5px
padding-top padding-right padding-bottom padding-left	the space inside the element, between the border and the contents	<i>see units below</i>
padding	shortcut for setting all 4 paddings	10px 10px 5px 10px 5px 2px 5px

## CSS Colors

Method	Description	Examples
color name	one of the 140 predefined colors (search online for "css color names")	white, crimson, peachpuff
RGB	red, green, blue values. Each value is 0-255.	rgb(30, 168, 180)
RGBA	RGB values with an added alpha (opacity) value	rgba(255, 0, 187, 0.5)
hexadecimal	RGB values in hex form. Each value is 00-ff.	#1ea8b4

## CSS Units

Method	Description	Examples
pixel	a dot on the screen. Pixel size will vary based on the screen resolution.	30px

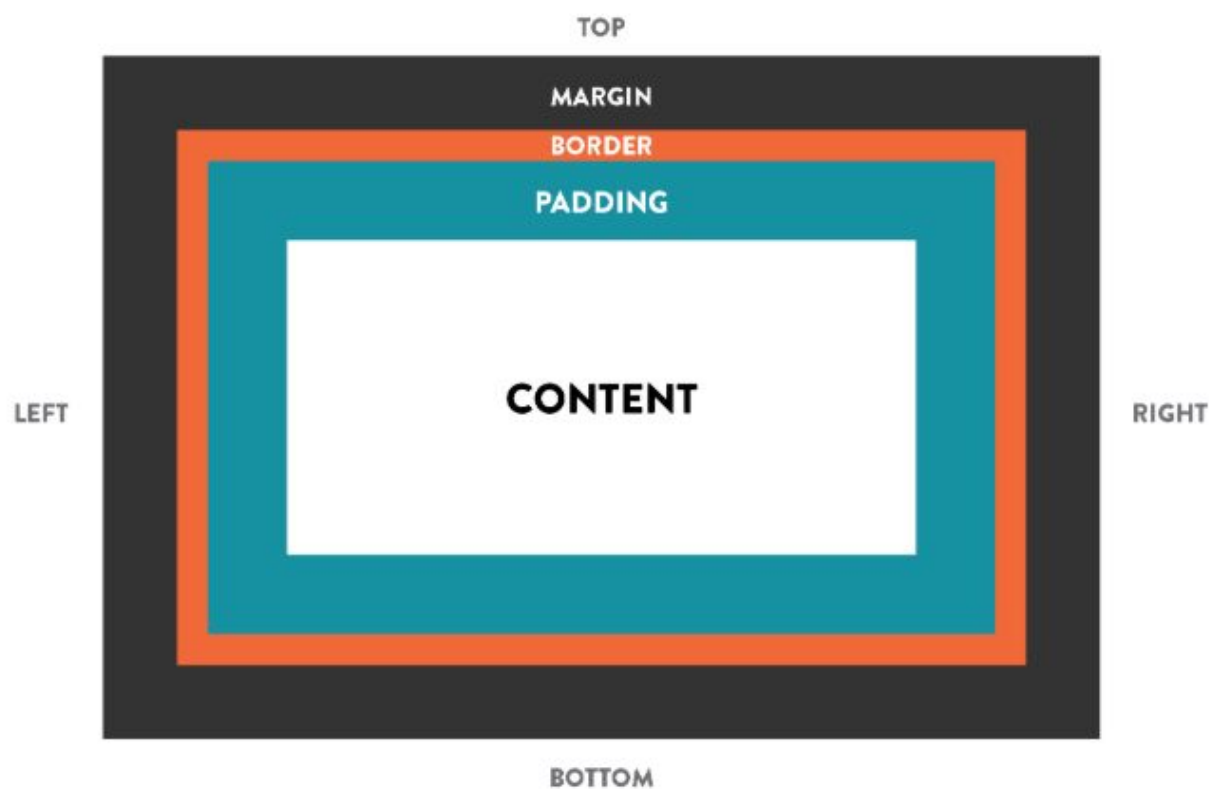


point	a typesetting term meaning 1/72 of an inch	12pt
em	a unit equal to the current font size. That means it changes depending on context and the font size on the page.	1.5em
root em	like em, but relative to the font size of the root element. The root is defined by your CSS (or by your browser if you don't define it), but rem stays the same throughout the page while em may vary.	2rem
view height	1% of the viewport (the browser window) height	50vh
view width	1% of the viewport (the browser window) width	12.5vw
percent	specifies a percentage relative to something. That something could be the element itself, the parent element, the base font size, or something else. It depends on what property is being set. Check the documentation for that property.	100% 3.5%





# BOX MODEL



# FLEXBOX PROPERTIES

All of these properties are set on the *parent container*.

Property	Description	Examples
display	set to "flex" to enable flexbox layout for this element	flex
flex-direction	affects the display order and alignment	row, row-reverse, column, column-reverse
justify-content	controls alignment in the direction of flow	flex-start, flex-end, center, space-between, space-around



align-items	controls alignment perpendicular to the flow	flex-start, flex-end, center, baseline, stretch
-------------	--	---

## MEDIA QUERIES

- **min-width:** apply these styles when the size hits a specific breakpoint and *larger*.
- **max-width:** apply these styles when the size hits a specific breakpoint and *smaller*.

Example:

```
@media screen and (min-width: 480px) {  
  p {  
    font-size: 20px;  
    color: black;  
  }  
  .nav-link {  
    padding: 10px;  
  }  
}
```

Be sure to include the Viewport Meta element in your HTML <head>:

```
<meta name="viewport" content="width=device-width, initial-scale=1" />
```

