
<Semester II>

<Web Development for Designers>

Week 6

Semantic Tags in HTML

A semantic element clearly describes its meaning to both the browser and the developer.

In programming, Semantics refers to the *meaning* of a piece of code – or "what purpose or role does that HTML element have" (rather than "what does it look like?".)

According to the W3C: "A semantic Web allows data to be shared and reused across applications, enterprises, and communities."



Importance of Semantic Tags

HTML should be coded to represent the *data* that will be populated and not based on its default presentation styling. Presentation (how it should look), is the sole responsibility of CSS. Some of the benefits from writing semantic markup are as follows:

- ❑ Search engines will consider its contents as important keywords to influence the page's search rankings.
- ❑ Screen readers can use it as a signpost to help visually impaired users navigate a page
- ❑ Finding blocks of meaningful code is significantly easier than searching through endless divs with or without semantic or namespaced classes
- ❑ Suggests to the developer the type of data that will be populated
- ❑ Semantic naming mirrors proper custom element/component naming

Web Browser

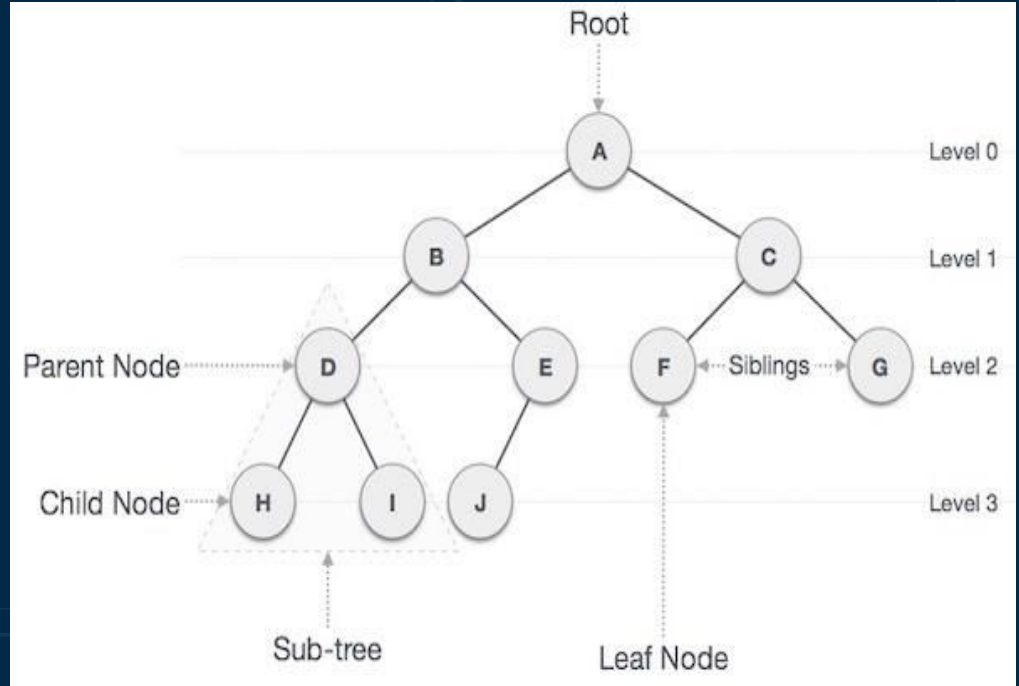
A web browser is an application for accessing websites. When a user requests a web page from a particular website, the browser retrieves its files from a web server and then displays the page on the user's screen. The first web browser World Wide Web was invented in the year of 1990 by Tim Berners-Lee. Later, it becomes Nexus. Virtually all URLs on the Web start with either *http:* or *https:* which means they are retrieved with the Hypertext Transfer Protocol (HTTP). When the web browser fetches data from an internet connected server, it uses a piece of software called a rendering engine to translate that data into text and images. This data is written in Hypertext Markup Language (HTML) and web browsers read this code to create what we see, hear and experience on the internet.



"Tree" Data Structure

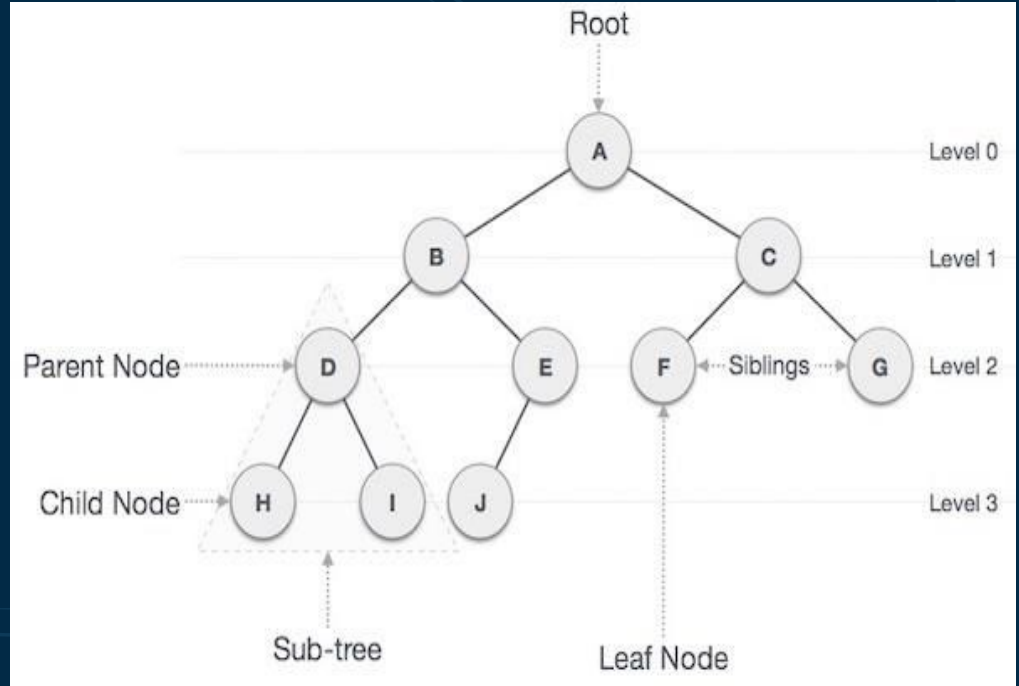
Tree data structure is a hierarchical structure that is used to represent and organize data in the form of parent child relationship. The following are some real world situations which are naturally a tree.

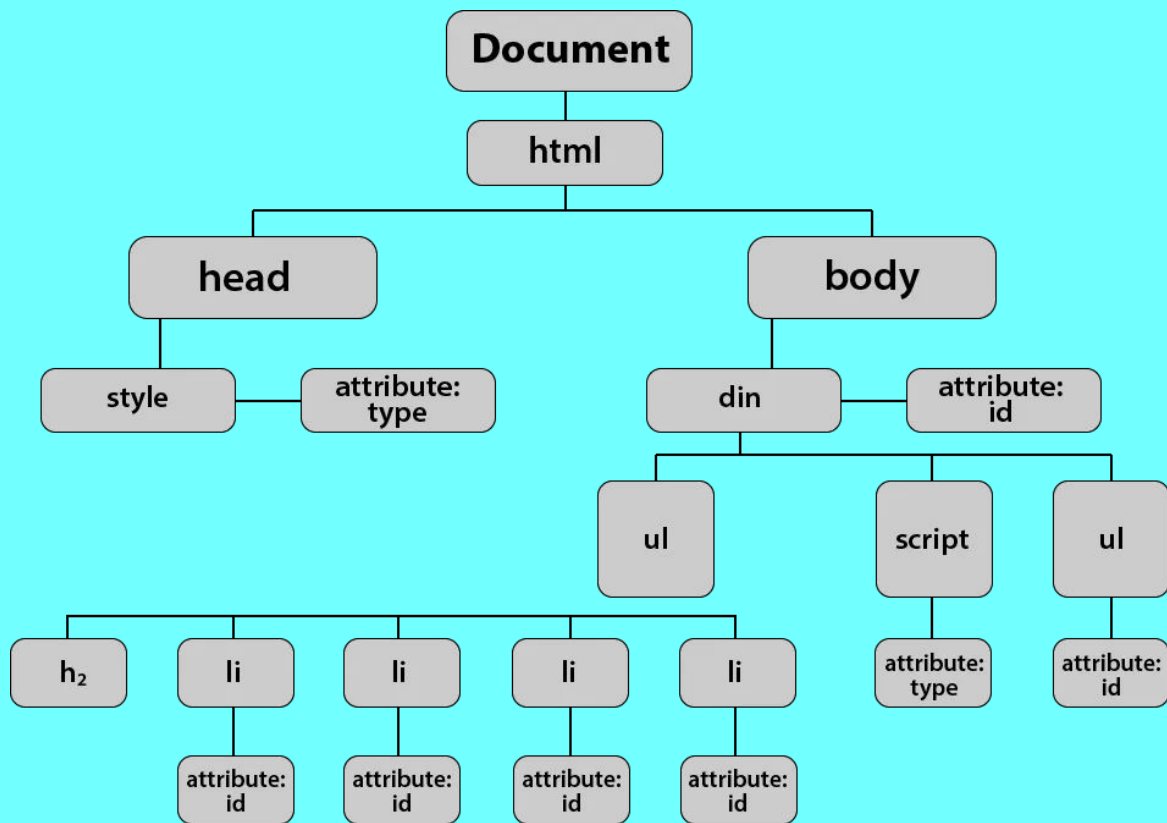
- ❑ Folder structure in an operating system.
- ❑ Tag structure in an HTML (root tag the as html tag) or XML document.



"Tree" Data Structure

The topmost node of the tree is called the root, and the nodes below it are called the child nodes. The data in a tree are not stored in a sequential manner i.e., they are not stored linearly. Instead, they are arranged on multiple levels or we can say it is a hierarchical structure. For this reason, the tree is considered to be a non-linear data structure.





Document Object Model

- ❑ When a browser loads a webpage, it creates a model of that webpage. This is called a DOM tree and it is stored in the browser memory.
- ❑ Every element, attribute, and piece of text in the HTML is represented by its own DOM node.
- ❑ Every element in on a web page is accessible in JavaScript through the DOM: Document Object Model.
- ❑ The DOM is a tree of nodes corresponding to the HTML elements on a page. DOM is the API used by JavaScript to manipulate HTML docs.
- ❑ With JavaScript, you can change/remove/add the HTML elements, attributes, and CSS styles in a webpage. It can react to all existing HTML events in the page and create new ones.

Getting Started with JavaScript

- ❑ JavaScript (JS) is a lightweight interpreted (or just-in-time compiled) programming language.
- ❑ While it is most well-known as the scripting language for Web pages, many non-browser environments also use it, such as Node.js, Apache CouchDB and Adobe Acrobat. JavaScript is a dynamic language, supporting object-oriented, imperative, and declarative (e.g. functional programming) styles.
- ❑ It is a case sensitive language.
- ❑ JavaScript is an **event-driven language**. This means that its execution flow is determined by events like user interactions or system notifications, allowing for dynamic and interactive web applications.

Application Programming Interface

An API or *Application Programming Interface* is a messenger or a middleman that lets computer programs securely access data from one another. A very common example of an API is a flight booking website that compares prices of flights from multiple airlines. The application requests data from multiple third-party APIs, which in turn connect to the actual data source (like databases). Once the API passes the response to the requesting application, the client user interface (UI) displays the information in a single view. People make APIs of dynamic data.

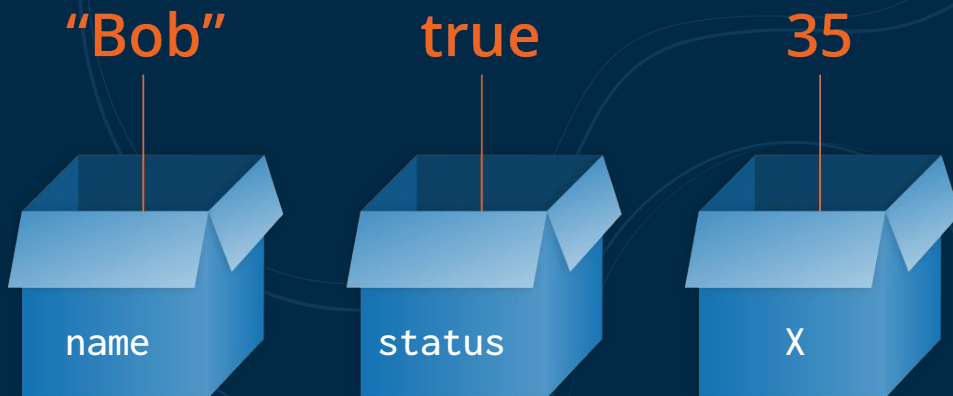
Public APIs are made to be accessed by the general public, like individuals, entrepreneurs, developers, and third-party systems. Some APIs don't have any authentication at all. Most of the APIs are built as public, so that many people can benefit from them—for example, programming language standard library APIs, Google Maps, and OpenWeatherMap.

API Working



JavaScript Variables

- ❑ In JavaScript, variables are named containers that store data, allowing you to manipulate and work with that data within your code. The value placed inside a variable can be changed.
- ❑ In contrast, a constant variable is fixed throughout the program.
- ❑ All JavaScript variables must be identified with unique names.



JavaScript Variables

JS

VARIABLES

JS

```
let name = "Bro Code"
```

```
let age = 25
```

```
let isHuman = false
```

Writing your first JavaScript code

1. Open up your LCA1 assignment on Visual Studio Code
2. Inside your first-website folder, create a new folder named scripts.
3. Download the extension called “Live Server” by Ritwick Dey
4. Within the scripts folder, create a new text document called main.js, and save it.
5. Go to your index.html file and enter this code on a new line, just before the closing `</body>` tag:
6. `<script src="scripts/main.js"></script>`
7. This is doing the same job as the `<link>` element for CSS. It applies the JavaScript to the page, so it can have an effect on the HTML (along with the CSS, and anything else on the page).
8. Type: `window.alert(5 + 6);` OR `window.alert(“Welcome to my webpage”);`

JavaScript Functions

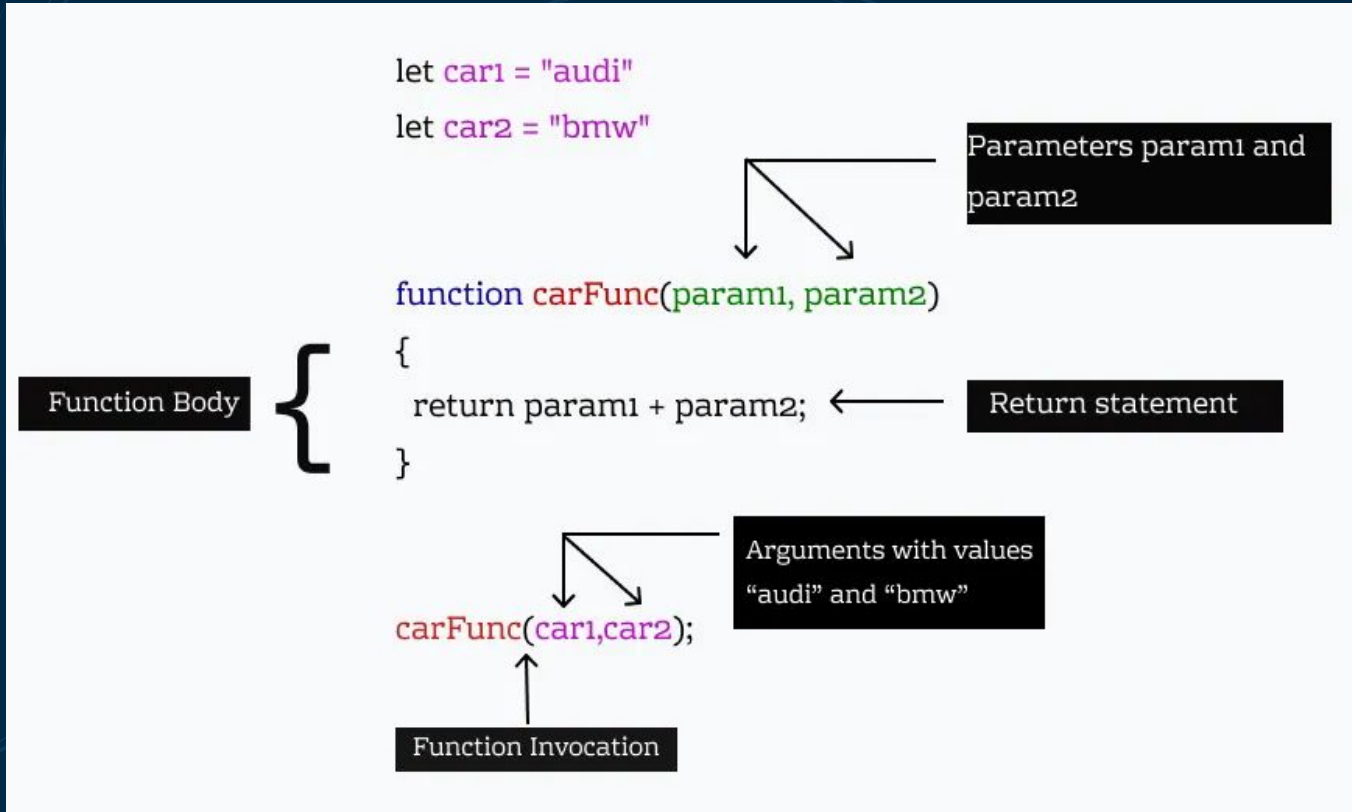
- ❑ A JavaScript function/method is a block of code designed to perform a particular task. A JavaScript function is executed when "something" invokes it (calls it).
- ❑ **There are some predefined functions in JavaScript, and you can create your own custom functions also.**
- ❑ A JavaScript function is defined with the function keyword, followed by a name, followed by parentheses ().
- ❑ Function names can contain letters, digits, underscores, and dollar signs (same rules as variables).
- ❑ The parentheses may include parameter names separated by commas: (*parameter1*, *parameter2*, ...) The code to be executed, by the function, is placed inside curly brackets: {}
- ❑

```
function name(parameter1, parameter2, parameter3) {  
    // code to be executed  
}
```
- ❑ **Custom Functions are first defined then called when you want to execute it**

Learn
Functions
In 10 Minutes



Structure of a JavaScript Function





**Homework: Start the HTML + CSS portion of your
LCA2 project.**

[W3Schools JavaScript Tutorial](#)
[JavaScript for Web | MDN](#)

Reference



1. [Tree Data Structure Geeks for Geeks](#)
 2. [Intro to the DOM - MDN](#)
 3. [HTML JavaScript DOM - Geeks for Geeks](#)
 4. [What are APIs - RedHat](#)
 5. [Application Programming Interface - Geeks for Geeks](#)
 6. [What are Variables - MDN](#)
 7. [JavaScript Variables - W3Schools](#)
 8. [JavaScript Functions - MDN](#)
 9. [Functions in JavaScript - GeeksforGeeks](#)
-

Fonts & colors used

This presentation has been made using the following fonts:

Blinker

(<https://fonts.google.com/specimen/Blinker>)

Inconsolata

(<https://fonts.google.com/specimen/Inconsolata>)

#ffffff

#022a46

#72ffdd

#72fff

#0d3a58