The DOM

What is it

Why do we need it

How to use it

Document are files like or xml

Object are the tags and the elements of that file

Model is the layout structure.

The Document Object Model (DOM) is a programming interface for HTML. It translates the content of an HTML document into a standardized object that functional programming languages like JavaScript can easily access and modify.

We are going to tap into why we need DOM.

Data comes in all different types: numbers, words, and characters, to name just a few.

to organize related pieces of data, programmers use a data type called an **object**. Objects are unique from other data types because their purpose is to hold other data. An object contains related pieces of data under a common concept, stored in a hierarchy.

Suppose you’re building a piece of software that stores information about houses. In real life, houses contain a lot of things — rooms, furniture, appliances, and important information like the neighborhood, and address.

We create a “house” object for each house in the system and put all the house things inside it. A “house” can contain data for all the information we described before.

We can even put objects inside other objects. In our example, a house object might contain a “room” object such as a kitchen, which itself can contain data for appliances, countertop material, etc.

Now, let’s look at this HTML file. Notice its parts are also structured in a hierarchy. The <html> tag acts like the topmost “house” in our example. Inside our HTML “house”, we have our <head> and <body> tags, which are like our rooms. Finally, there are the most specific elements: <title>, <h1>, <h2>, <p>, <img>, and <button>. Think of these like room-specific items, like the bed and a lamp.

Elements in an HTML document are structured this way for the same reason as objects: A hierarchical structure helps computer programs to read and process the information.

It’s important to emphasize that HTML documents by themselves are not objects. They cannot be read by JavaScript without some sort of file-to-object translation. This is where the DOM comes in.

The DOM represents an entire HTML document as a single object. To create the DOM, the web browser reading the HTML file takes in all its parts and outputs them as an object that JavaScript understands.

Javascript affects most pages we see on the web, but DOM is not specific to Javascript. Other programming languages like C++ and Python can use the DOM to modify web pages.

How to use the DOM?

In web design, JavaScript is used to control the behavior of web pages. The DOM links JavaScript to the source HTML code, which enables JavaScript to execute its functions on individual HTML elements. This JavaScript-HTML interaction creates the experience of interactive web design.

A basic example of the DOM in action.

The top panel contains the source code, and the panel below shows the code displayed as a web page. The moment you press **Click me!** the following JavaScript runs.

**Document** is JavaScript’s way of targeting the HTML file object through the DOM. It’s telling the browser, “I want to go into the HTML file to make a change.”

**getElementByID(‘myBtn’)** targets the page element with the ID “myBtn”. To get there, JavaScript starts at the highest <html> tag, then traverses through the DOM until it hits click me.

**.addEventListener** is the method we are using to assign a function on a click event.

Displaydate function runs when the event occurs. This time JavaScript is targeting the id Demo targets to display the date. The id is found in the last paragraph element.

Once the event is triggered, the current date is displayed.