

| Javascript

It is a loosely-typed, dynamic language that is used to add interactivity and dynamic behaviour on a website.

All browsers support Javascript by default using an engine like the "V8 Engine" in Chrome.

Javascript Runtime →

It can be added to an HTML document in three ways.

1. Inline Javascript → using attributes like `onclick()` and `onblur()`.
2. Internal Javascript → using the `<script>` tag and writing inside it.
3. External Javascript → using `<script>` tag and linking the javascript using the `src=""` attribute.

| Language Features

1. Single Line Comment are given like this `//`.
2. Multi Line Comments are given using these `/* */`
3. Semicolons are optional but it's better to use them as the JS interpreter adds semicolons where there is a missing one.

String Concatenation and Templating

```
var name = "Jaish";

//String Concatenation is done using the plus symbol.
var text = "M." + name + "Khan";
console.log(text); //M. Jaish Khan

//String Templating is done using backticks.
var text = `M. ${name} Khan`;
console.log(text); //M. Jaish Khan
```

| Variables

Data Types aren't explicitly written because they are implicitly assigned at runtime. However, Javascript has 8 primitive data types → String, Number, BigInt, Boolean, Undefined, Null, Symbol, Object.

There are four ways of assigning variables in javascript.

1. `x = 10` without any keyword.
2. `var x = 10;` using the var keyword.
3. `let x = 10;` using the let keyword.
4. `const x = 10;` using the const keyword.

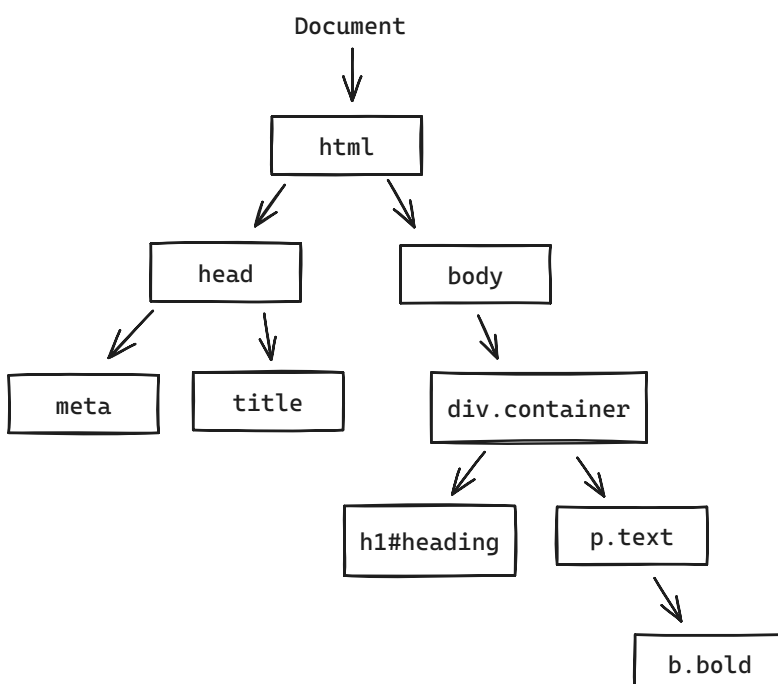
| Document Module

It is a module/extension of Javascript that allows us to interact with the HTML page.

DOM (Document Object Model) → It is a tree of objects made out of the HTML page.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <div class="container">
    <h1 id="heading">Jaish Khan</h1>
    <p class="text">Aspiring to become a <b class="bold">Web Dev</b>
  </p>
  </div>
</body>
</html>
```

The DOM Tree of the above HTML code would look like:



| DOM Traversal Functions

They are used to interact with specific elements in the DOM Tree.

```
//selects the first element that matches the id "email".
var email = document.getElementById("email");

//selects all the elements with the class "container".
var containers = document.getElementsByClassName("container");

//selects the first element that matches the given selector (can be anything).
var email2 = document.querySelector("#email2");

//selects all the elements that match the given selector (can be anything).
var boxes = document.querySelectorAll(".box");
```

The `innerHTML` property can be used to select the content of an HTML element and can also be used to change it.

```
var text = email.innerHTML; //selects the content of the selected element.
email.innerHTML = "jaishkhan771@gmail.com"; //changes the content of the element
```

| DOM Manipulation Functions

```
document.write("<h1>Jaish Khan</h1>");
//writes onto the HTML page (also creates the h1 element).

document.createElement(h1); //creates an h1 element.
document.appendChild(h1); //attaches the created h1 element to the document.
document.removeChild(h1); //removes the created h1 element from the document.
```

| DOM Events

```
<h1 onclick="this.innerHTML = '0oops';">Jaish</h1>
```

| addEventListener()

```
document.getElementById("button").addEventListener("click", function);
```

| Window Module

```
window.alert("Click Ok to continue.");  
window.
```

| Location SubModule

Property	Function
<code>window.location.hostname</code>	Sets/Returns the hostname of a URL.
<code>window.location.port</code>	Sets/Returns the port of a URL.
<code>window.location.host</code>	Sets/Returns the hostname + port of a URL.
<code>window.location.protocol</code>	Sets/Returns the protocol of a URL.
<code>window.location.origin</code>	Sets/Returns the hostname + port + protocol of a URL.
<code>window.location.href</code>	Sets/Returns the entire URL.

| Navigator SubModule

Property	Function
<code>window.navigator.appName</code>	Sets/Returns the hostname of a URL.
<code>window.navigator.appVersion</code>	Sets/Returns the port of a URL.
<code>window.navigator.cookieEnabled</code>	Sets/Returns the hostname + port of a URL.
<code>window.navigator.geolocation</code>	Sets/Returns the protocol of a URL.
<code>window.navigator.language</code>	Sets/Returns the hostname + port + protocol of a URL.
<code>window.navigator.online</code>	Sets/Returns the entire URL.

| Form Validation

*Checking if the data that is submitted by the user is valid or not is called **Form Validation**.*

This is done for Security

We can do form validation in many ways.

1. In HTML, using attributes like `required` or `type=" "` in form elements.
2. In CSS, using selectors like `:invalid` or `:disabled`.
3. In Javascript,
4. In PHP (or any other server-side language)

| PHP

It is a loosely-typed, dynamic language that is used as a server-side language to fetch requests and send responses.

A PHP file supports HTML, CSS and JS by default. The PHP code begins with the delimiter `<?php` and ends at `?>`.

| Variables

Data Types aren't explicitly written because they are implicitly assigned at runtime. However, PHP has 8 primitive data types → String, Number, BigInt, Boolean, Undefined, Null, Symbol, Object.

We assign variables in php using the dollar sign `$name = "Jaish"`. Both **"double quotes"** and **'single quotes'** can be used but string templating only works with double quotes.

String Concatenation and Templating

```
$name = "Jaish";

//String Concatenation is done using the dot symbol.
$text = "M." . name . "Khan"
echo text //M. Jaish Khan

//String Templating is done using double quotations (doesn't work in
single quotes).
$text = "M. $name Khan"
echo text //M. Jaish Khan
```

Arrays and Associative Arrays

```
$arr = array(1, 2, 3, 4, 5)

$assoc_arr = array("one"⇒1, "two"⇒2, "three"⇒3, "four"⇒4, "five"⇒5)
```

| PHP Superglobal Variables

PHP has some special variables that are always accessible and are system-defined. They all start with a `$_` and are uppercase. There are **9** superglobals in PHP.

Variable	What it stores
<code>\$GLOBALS</code>	
<code>\$_SERVER</code>	

Variable	What it stores
<code>\$_REQUEST</code>	
<code>\$_POST</code>	
<code>\$_GET</code>	
<code>\$_FILES</code>	
<code>\$_ENV</code>	
<code>\$_COOKIE</code>	
<code>\$_SESSION</code>	