JavaScript

Introduction

JavaScript is used **to develop interactive web applications**. JavaScript can power featured like interactive images, carousels, and forms. The language can be used with back-end frameworks like Node. js to power the mechanics behind a web page, such as form processing and payments.

World wide web

**an interconnected system of public webpages accessible through the Internet**.

HTTP

The Hypertext Transfer Protocol is an application layer protocol in the Internet protocol suite model for distributed, collaborative, hypermedia information systems. It was designed for communication between web browsers and web servers, but it can also be used for other purposes. USED TO CONNECT TO WEB SERVERS

Javascript originally runs only in browsers. Browsers have JS engine to execute JS codes. Chrome has **V8** JS engine

Node JS (Ryan Dhal)

A C++ program that can run JS code outside of browsers. With this we can build backend for our web or mobile.

ECMA script vs JavaScript

ECMA is specification or standardization JS is a programming language

Scripting vs Programming, Interpreter vs compiler

Scripting runs in time while programming must be compiled first before execution.

Scripting mostly is used for automation or repeated task

Scripting is faster since it doesn’t need to be compiled. Programming is more complex

Web says JS is interpreted and dynamic?

JS probably became a compiled language due to the introduction of Node JS

Placing scripts at the bottom of the <body> element improves the display speed, because script interpretation slows down the display.

Setting Up JS

* Install first a code editor such as Notepad++ or VS Code
* Create your Project Directory
* Open folder using code editor
* Create HTML file then type ! exclamation and press tab to generate a basic html structure

Note:

Browser parses code from top to bottom

JS can exist in body and head

Execute code in Node JS

Open cmd and go to your directory. type node and name of your js file

Node myscript.js

Getelementbyid vs getelementbyname or class

get id is specific while get by name can be multiple. Change input value using ID

JavaScript has dynamic types. This means that the same variable can be used to hold different data types

let x; // Now x is undefined  
x = 5; // Now x is a Number  
x = "John"; // Now x is a String

In Data Types, JavaScript has only one type of numbers

Arrays = square brackets

Objects = curly braces

Local variables can only be accessed from within its block

Use const when declaring objects and arrays

JS is camel cased. CamelCased(sample).

Template Literals/String interpolation allows multiline string. Allows variable in strings. Allows expression in strings.

w3schools.com/js/js\_string\_templates.asp

The JavaScript interpreter works from left to right.

First 10 + 20 is added because x and y are both numbers.

NaN = not a number

## What is **this**?

const person = {  
 firstName: "John",  
 lastName : "Doe",  
 id : 5566,  
 fullName : function() {  
 return this.firstName + " " + this.lastName;  
 }  
};

In the example above, this refers to the **person object**.

I.E. **this.firstName** means the **firstName** property of **this**.

I.E. **this.firstName** means the **firstName** property of **person**.

In JavaScript, the this keyword refers to an **object**.

**Which** object depends on how this is being invoked (used or called).

The this keyword refers to different objects depending on how it is used:

|  |
| --- |
| In an object method, this refers to the **object**. |
| Alone, this refers to the **global object**. |
| In a function, this refers to the **global object**. |
| In a function, in strict mode, this is undefined. |
| In an event, this refers to the **element** that received the event. |
| Methods like call(), apply(), and bind() can refer this to **any object**. |

## The **this** Keyword

In a function definition, this refers to the "owner" of the function.

In the example above, this is the **person object** that "owns" the fullName function.

In other words, this.firstName means the firstName property of **this object**.

HTML events

<https://www.w3schools.com/js/js_events.asp>

List of events

https://www.w3schools.com/jsref/dom\_obj\_event.asp

GET and POST