Java Script Notes

Decentral Developers

**Why we can’t run JS on Vs code terminal?**

JavaScript is primarily designed to run in web browsers. Browsers have built-in engines (like Chrome's V8) that interpret and execute JavaScript code.

However, when using VS Code's terminal, you're typically interacting with your computer's operating system, not a web browser. The terminal doesn't have a built-in JavaScript engine to run JavaScript code directly.

To run JavaScript outside a browser, you need an environment like Node.js. Node.js includes the V8 engine, allowing you to run JavaScript in the terminal.

Function of alert:

An alert message in JavaScript is a simple way to display a message to the user.

**Sample code:**

|  |
| --- |
| **<!DOCTYPE html>**  **<html>**  **<head>**  **<title>Alert Example</title>**  **</head>**  **<body>**  **<script>**  **alert("Hello, World!");**  **</script>**  **</body>**  **</html>** |

**Difference between window. alert and alert:**

There is no difference; window.alert and alert are the same because alert is a method of the window object, and JavaScript allows you to call it without specifying window. explicitly**.**

**Use of semicolon; + parenthesis () and inverted commas “ ” :**

* Semicolon (;): Ends statements to separate them, although JavaScript can automatically insert them in many cases.
* Parentheses (()): Used for function calls, function definitions, and to group expressions.
* Quotes (" “or ‘ ‘): Used to define string literals.

**String:**

Strings in JavaScript are used to represent text. They can be created using single quotes ('), double quotes ("), or backticks (`).

**Sample code:**

|  |
| --- |
| **// Creating strings**  **let singleQuoteString = 'Hello, World!';**  **let doubleQuoteString = "Hello, JavaScript!";**  **let templateString = `Hello, Template Literals!`;**  **// Logging strings to the console**  **console.log(singleQuoteString);**  **console.log(doubleQuoteString);**  **console.log(templateString);** |

**Creating Strings**:

* let singleQuoteString = 'Hello, World!'; uses single quotes.
* let doubleQuoteString = "Hello, JavaScript!"; uses double quotes.
* let templateString = Hello, Template Literals!; uses backticks, which allow for template literals (strings that can include embedded expressions).

**Logging Strings:**

* console.log(singleQuoteString); prints the string to the console.
* console.log(doubleQuoteString); prints the string to the console.
* console.log(templateString); prints the string to the console.

**Prompt:**

The prompt function in JavaScript is used to display a dialog box that prompts the user for input.

**Sample code:**

|  |
| --- |
| **// Using prompt to get user input**  **let userInput = prompt("Please enter your name:");**  **// Displaying the input in an alert**  **alert("Hello, " + userInput + "!");** |

1. Prompt: let userInput = prompt("Please enter your name:");
   * This line displays a dialog box with the message "Please enter your name:" and stores the user's input in the variable userInput.
2. Alert: alert("Hello, " + userInput + "!");
   * This line displays an alert box with the message "Hello, [userInput]!", where [userInput] is the name entered by the user.

**Variable:**

In JavaScript, you can create variables and assign them values.

|  |
| --- |
| // Using let to declare a variable  Var gamelevel=1; |

**Exchange value of variable:**



****



Container one Container two Empty container

Let,

a=3; b=5; empty

a=b; empty☺ container3=a

a=b; b=a; empty☺

**Sample code:**

|  |
| --- |
| **// Initial values**  **let container1 = 3;**  **let container2 = 5;**  **let container3;**  **// First operation: a = b;**  **container1 = container2;**  **// Second operation: a = b; b = a;**  **container1 = container2;**  **container2 = container1;**  **// Resulting values**  **console.log("Container 1:", container1); // Should print 5**  **console.log("Container 2:", container2); // Should print 5**  **console.log("Container 3:", container3); // Should print undefined or null** |

**Naming variables:**

Naming variables with **meaningful names** in JavaScript is crucial for writing clean, readable, and maintainable code.

**String concatenation:**

String concatenation in JavaScript involves combining multiple strings into a single string.

**Sample code:**

|  |
| --- |
| **let firstName = "John";**  **let lastName = "Doe";**  **let fullName = firstName + " " + lastName;**  **console.log(fullName); // Output: John Doe** |

**Length of string:**

The .length property in JavaScript is used to retrieve the number of characters in a string.

**Sample code:**

|  |
| --- |
| let message = "Hello, World!";  let lengthOfMessage = message.length;  console.log(lengthOfMessage); // Output: 13 |

**.slice :**

The slice method in JavaScript is used to extract a portion of a string and returns a new string without modifying the original string.

**Sample code:**

|  |
| --- |
| let message = "Hello, World!";  let slicedMessage = message.slice(7, 12);  console.log(slicedMessage); // Output: "World" |

**Upper and lower case:**

In JavaScript, you can convert strings to uppercase and lowercase using the toUpperCase() and toLowerCase() methods, respectively.

**Sample code:**

|  |
| --- |
| let message = "Hello, World!";  let upperCaseMessage = message.toUpperCase();  let lowerCaseMessage = message.toLowerCase();  console.log(upperCaseMessage); // Output: "HELLO, WORLD!"  console.log(lowerCaseMessage); // Output: "hello, world!" |

**Declaration and assignment:**

|  |
| --- |
| // Using let to declare and assign a variable  Let, age; (Declaration)  age = 30; (Assignment)  // Using const to declare and assign a constant  const PI = 3.14;  console.log(age); // Output: 30  console.log(PI); // Output: 3.14 |

**Arithmetic operations:**

Arithmetic operations in programming refer to basic mathematical calculations performed on numbers. These operations include addition (+), subtraction (-), multiplication (\*), and division (/).

**Sample code:**

|  |
| --- |
| let a = 10;  let b = 5;  let addition = a + b; // Addition  let subtraction = a - b; // Subtraction  let multiplication = a \* b; // Multiplication  let division = a / b; // Division  console.log("Addition:", addition); // Output: 15  console.log("Subtraction:", subtraction); // Output: 5  console.log("Multiplication:", multiplication); // Output: 50  console.log("Division:", division); // Output: 2 |

Increment and decrement:

Increment (++) and decrement (--) operators are used to increase or decrease the value of numeric variables by 1, respectively.

Sample code:

|  |
| --- |
| let x = 5;  // Increment operator (++)  x++; // Equivalent to: x = x + 1;  // Decrement operator (--)  let y = 10;  y--; // Equivalent to: y = y - 1;  console.log("After increment:", x); // Output: 6  console.log("After decrement:", y); // Output: 9 |

Let:

It declares a variable that is block-scoped, which means it is only accessible within the block (a pair of curly braces { }) where it is defined. Variables declared with let are not hoisted to the top of their scope, unlike variables declared with var.

**Sample code:**

|  |
| --- |
| function demoFunction() {  let x = 10;  if (true) {  let y = 20;  console.log(x); // Output: 10  console.log(y); // Output: 20  }  // console.log(y); // Error: y is not defined  }  demoFunction(); |

Example for practice:

|  |
| --- |
| // Define the radius of the circle  let radius = 5;  // Calculate the circumference  let circumference = 2 \* Math.PI \* radius;  // Print the circumference  console.log("Circumference of the circle:", circumference); |