**Java Script Crash Course**

**By Geetesh**

**Index**

1. **Introduction To JS**
2. **Basic Functions In JS**

**3**

**4**

**5**

**6**

**Introduction To JS**

**JavaScript is a versatile, high-level programming language that is widely used for creating dynamic and interactive web content. It is one of the core technologies of the World Wide Web, alongside HTML and CSS.**

**Key Features of JavaScript:**

1. **Client-Side Scripting:**
   * **JavaScript primarily runs in the web browser, enabling interactive web pages.**
   * **Examples include form validations, dynamic updates, and interactive content like sliders and pop-ups.**
2. **Lightweight and Interpreted:**
   * **JavaScript is designed to be lightweight and doesn't need compilation, making it quick to execute.**
3. **Cross-Platform Compatibility:**
   * **JavaScript can run on all major web browsers like Chrome, Firefox, Safari, and Edge without additional plugins.**
4. **Event-Driven Programming:**
   * **It responds to events such as user clicks, mouse movements, or keyboard input, enhancing interactivity.**
5. **Dynamic Typing:**
   * **Variables in JavaScript do not require a predefined type, allowing flexibility in coding.**

**Applications of JavaScript:**

1. **Web Development:**
   * **Enhances user interfaces by enabling dynamic updates, animations, and interactivity.**
   * **Frameworks like React, Angular, and Vue build on JavaScript for robust web apps.**
2. **Server-Side Development:**
   * **With Node.js, JavaScript is used for server-side programming, allowing full-stack development.**
3. **Mobile App Development:**
   * **Tools like React Native enable developers to build mobile apps using JavaScript.**
4. **Game Development:**
   * **JavaScript, along with HTML5, is used for creating browser-based games.**
5. **Automation:**
   * **JavaScript can automate repetitive tasks in browsers using libraries like Puppeteer.**

**Basic Syntax:**

**// Declaring a variable**

**let name = "John";**

**// Function to greet a user**

**function greet() {**

**console.log("Hello, " + name + "!");**

**}**

**// Call the function**

**greet(); // Output: Hello, John!**

**Advantages of JavaScript:**

1. **Speed: It runs quickly in the browser without needing a server round trip.**
2. **Ease of Learning: Beginner-friendly and widely supported with numerous resources.**
3. **Rich Ecosystem: A vast library of frameworks, tools, and plugins.**
4. **Interactivity: Enhances user experience by making web pages more engaging.**

**Common JavaScript Tools and Libraries:**

* **Libraries: jQuery, Lodash**
* **Frameworks: React.js, Angular.js, Vue.js**
* **Build Tools: Webpack, Babel**
* **Testing Tools: Jest, Mocha**

**Basic Functions In JS**

**1.alert()**

 **Purpose**: Displays a popup message to the user.

 **Example**: alert("Hello, World!");

**2.console.log()**

 **Purpose**: Prints messages or variables to the browser's console, useful for debugging.

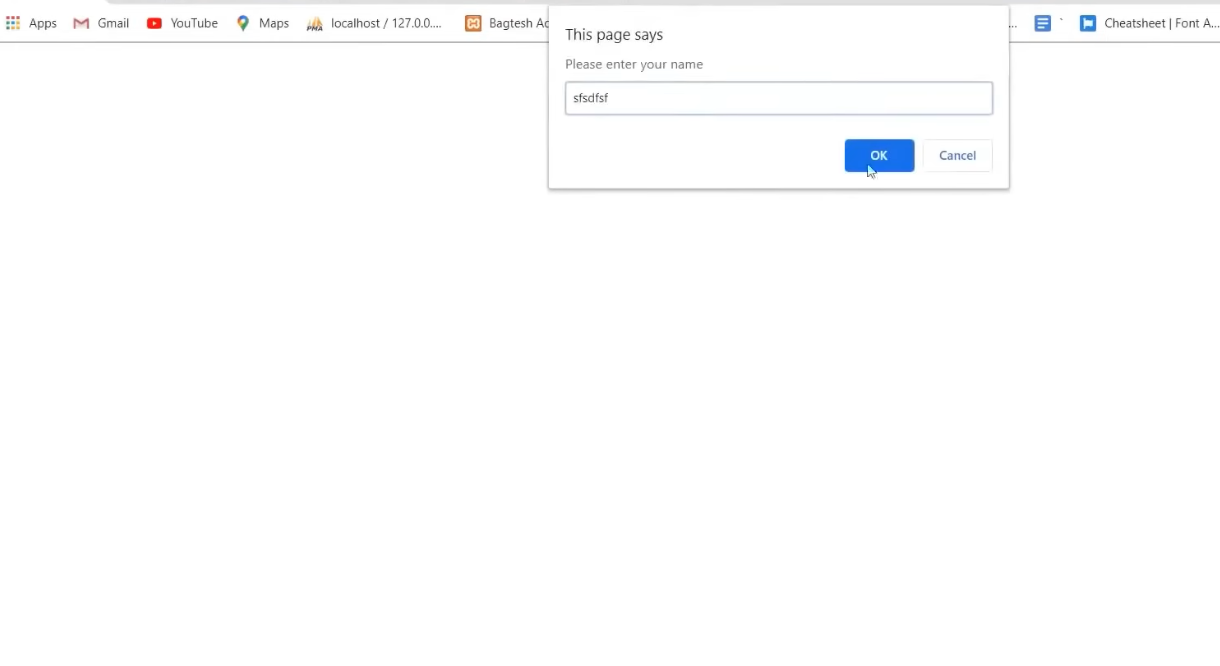
 **Example**: console.log("This is a log message!");

**3.** **prompt()**

 **Purpose**: Displays a dialog box asking the user for input.

 **Example**: let name = prompt("What is your name?");

console.log("Hello, " + name);

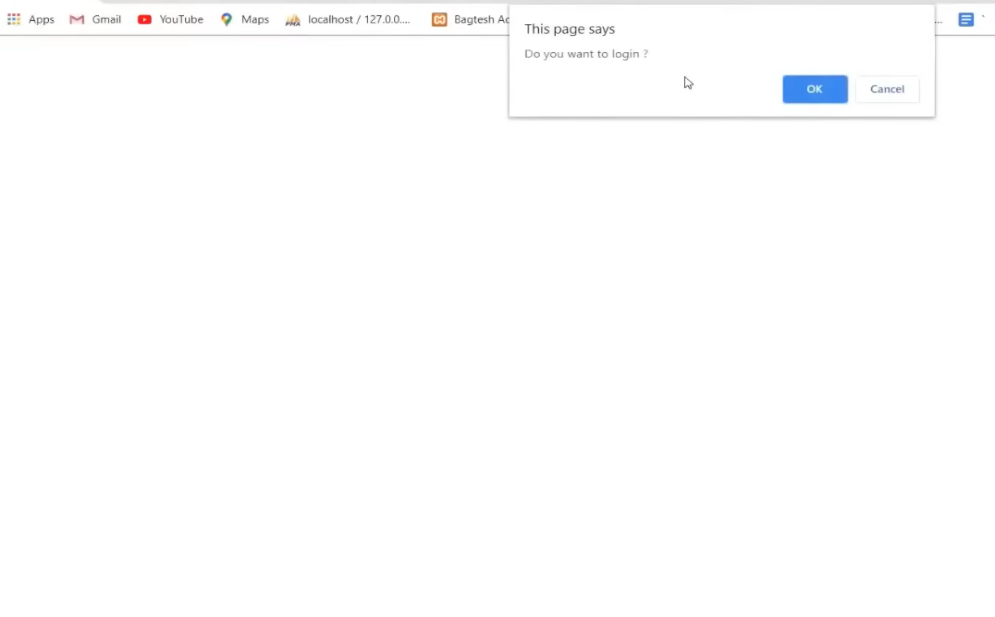


**4.** **confirm()**

 **Purpose**: Displays a dialog box with "OK" and "Cancel" buttons, returning true or false.

 **Example**: let isConfirmed = confirm("Are you sure?");

console.log(isConfirmed); // true if OK, false if Cancel



**5.** **Basic Arithmetic Functions:**

* **Addition:**

**let sum = 5 + 3;**

**console.log(sum); // 8**

* **Subtraction:**

**let difference = 10 - 4;**

**console.log(difference); // 6**

* **Multiplication:**

**let product = 6 \* 7;**

**console.log(product); // 42**

* **Division:**

**let quotient = 20 / 4;**

**console.log(quotient); // 5**

**6. parseInt() and parseFloat()**

 **Purpose**: Convert strings to numbers.

 **Example**: let intNum = parseInt("123");

let floatNum = parseFloat("123.45");

console.log(intNum); // 123

console.log(floatNum); // 123.45

**7. isNaN()**

 **Purpose**: Checks if a value is NaN (Not-a-Number).

 **Example**: console.log(isNaN("Hello")); // true

console.log(isNaN(123)); // false

**8.** **setTimeout()**

 **Purpose**: Executes a function after a specified time delay (in milliseconds).

 **Example**: setTimeout(() => {

console.log("This runs after 2 seconds!"); }, 2000);

**9.** **setInterval()**

 **Purpose**: Repeatedly executes a function at specified intervals (in milliseconds).

 **Example**: setInterval(() => {

console.log("This runs every 1 second!");

}, 1000);

**10.** **typeof**

 **Purpose**: Returns the data type of a variable or value.

 **Example**: let num = 42;

console.log(typeof num); // "number"

**11. Defining Your Own Functions**

 **Basic Function:**

**function greet(name) {**

**return "Hello, " + name;**

**}**

**console.log(greet("Alice")); // "Hello, Alice"**

 **Arrow Function:** **const greet = (name) => "Hello, " + name;**

**console.log(greet("Bob")); // "Hello, Bob"**

**12.** **Math Functions:**

 **Basic Function:**

**console.log(Math.random()); // Random number between 0 and 1**

 **Round Off:**

**console.log(Math.round(4.6)); // 5**

**#DOM FUNCTIONS IN JAVASCRIPT:**

JavaScript provides a range of functions and methods to interact with and manipulate the DOM (Document Object Model), which represents the structure of an HTML document. Here's an overview of the most commonly used DOM functions in JavaScript:

**1.Accessing Elements:**

|  |  |  |
| --- | --- | --- |
| **Function** | **Description** | **Example** |
| document.getElementById() | Selects an element by its ID. | document.getElementById("myId") |
| document.getElementsByClassName() | Selects all elements with the specified class name. | document.getElementsByClassName("myClass") |
| document.getElementsByTagName() | Selects all elements with the specified tag name. | document.getElementsByTagName("div") |
| |  | | --- | | document.querySelector() |  |  | | --- | |  | | |  | | --- | | Selects the first element matching a CSS selector. |  |  | | --- | |  | | |  | | --- | | document.querySelector(".myClass") |  |  | | --- | |  | |
| |  | | --- | | document.querySelectorAll() |  |  | | --- | |  | | |  | | --- | | Selects all elements matching a CSS selector. |  |  | | --- | |  | | document.querySelectorAll(".myClass") |

**2. Modifying Content:**

|  |  |  |
| --- | --- | --- |
| **Function** | **Description** | **Example** |
| |  | | --- | | element.innerHTML |  |  | | --- | |  | | |  | | --- | | Gets or sets the HTML content inside an element. |  |  | | --- | |  | | |  | | --- | | element.innerHTML = "<b>Hello</b>"; |  |  | | --- | |  | |
| element.innerText | |  | | --- | | Gets or sets the text content of an element. |  |  | | --- | |  | | |  | | --- | | element.innerText = "Hello"; |  |  | | --- | |  | |
| |  | | --- | | element.textContent |  |  | | --- | |  | | |  | | --- | | Similar to innerText, but includes hidden elements. |  |  | | --- | |  | | |  | | --- | | element.textContent = "Hello"; | |

**3.** **Modifying Attributes:**

|  |  |  |
| --- | --- | --- |
| **Function** | **Description** | **Example** |
| |  | | --- | | element.setAttribute() |  |  | | --- | |  | | |  | | --- | | Sets or updates an attribute of an element. |  |  | | --- | |  | | |  | | --- | | element.setAttribute("class","newClass"); |  |  | | --- | |  | |
| element.getAttribute() | |  | | --- | | Gets the value of an attribute. |  |  | | --- | |  | | |  | | --- | | element.getAttribute("class"); |  |  | | --- | |  | |
| |  | | --- | | element.removeAttribute() |  |  | | --- | |  | | |  | | --- | | Removes an attribute from an element. |  |  | | --- | |  | | |  | | --- | | element.removeAttribute("class"); | |

**4. Working with Styles:**

|  |  |  |
| --- | --- | --- |
| **Function** | **Description** | **Example** |
| |  | | --- | | element.style.property |  |  | | --- | |  | | |  | | --- | | Sets a specific CSS property for an element. |  |  | | --- | |  | | |  | | --- | | element.style.color = "red"; | |
| element.classList.add() | |  | | --- | | Adds a class to an element. |  |  | | --- | |  | | |  | | --- | | element.classList.add("active"); |  |  | | --- | |  | |
| |  | | --- | | element.classList.remove() |  |  | | --- | |  | | |  | | --- | | Removes a class from an element. |  |  | | --- | |  | | |  | | --- | | element.classList.remove("active"); | |
| |  | | --- | | element.classList.toggle() |  |  | | --- | |  | | |  | | --- | | Toggles a class on or off. |  |  | | --- | |  | | element.classList.toggle("active"); |

1. **Creating and Removing Elements:**

|  |  |  |
| --- | --- | --- |
| **Function** | **Description** | **Example** |
| |  | | --- | | document.createElement() |  |  | | --- | |  | | |  |  |  | | --- | --- | --- | | |  | | --- | | Creates a new HTML element. |  |  | | --- | |  | |  |  | | --- | |  | | |  |  |  | | --- | --- | --- | | |  | | --- | | let newDiv = document.createElement("div"); |  |  | | --- | |  | | |
| |  | | --- | | element.appendChild() |  |  | | --- | |  | | |  |  |  | | --- | --- | --- | | |  | | --- | | Appends a new child element. |  |  | | --- | |  | |  |  | | --- | |  | | |  |  |  | | --- | --- | --- | | |  | | --- | | parent.appendChild(newDiv); |  |  | | --- | |  | |  |  | | --- | |  | |
| |  |  |  | | --- | --- | --- | | |  | | --- | | element.removeChild() |  |  | | --- | |  | |  |  | | --- | |  | | |  |  |  | | --- | --- | --- | | |  | | --- | | Removes a child element. |  |  | | --- | |  | |  |  | | --- | |  | | |  |  |  | | --- | --- | --- | | |  | | --- | | parent.removeChild(child); |  |  | | --- | |  | | |
| |  |  |  | | --- | --- | --- | | |  | | --- | | element.insertBefore() |  |  | | --- | |  | |  |  | | --- | |  | | |  |  |  | | --- | --- | --- | | |  | | --- | | Inserts an element before another child element. |  |  | | --- | |  | |  |  | | --- | |  | | parent.insertBefore(newElement, referenceChild); |