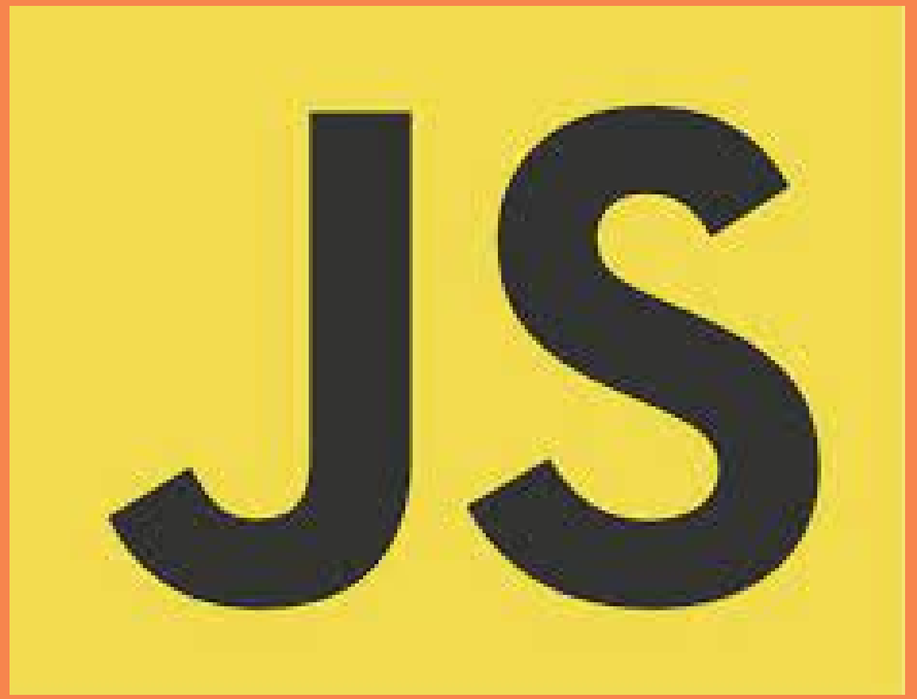
JAVASCIRPT



How to run javascript

```
<script>
  // JavaScript code goes here
</script>
```

```
<script>
console.log("hello world");
</script>
```

Try

Ctrl+shift+C

Comments in JavaScript: Comments in JavaScript are used to annotate the code for better understanding.
There are two ways to add comments:

```
// This is a single-line comment


/*
This is a multi-line comment
spanning multiple lines.
*/
```

DOM ELEMENTS

Finding HTML elements by id

document.getElementById

DOM ELEMENTS

Finding HTML elements by tag name

document.getElementByTag

HOW DO WE USE CSS inside JAVASCRIPT?

Try

style.property = new style

What is function?

function is a block of reusable code that performs a specific task

SYNTAX of function

```
function functionName(parameter1, parameter2)
 { // Function body
}
```

lets make our first function

```
function greet(name) {
return "Hello, " + name + "!";
}
console.log(greet("Alice"));
```

variables

Using var

Using let

Using const

const

- The const keyword was introduced in ES6 (2015)
- Variables defined with const cannot be Redeclared
- Variables defined with const cannot be Reassigned

let

- **Variables declared with let have Block Scope**
- **Variables declared with let must be Declared before use**
- **Variables declared with let cannot be Redeclared in the same scope**

using ‘+’ with strings

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

Lets do form validation

Objects in js

```
// Create an object:
const person = {
  firstName: "John",
  lastName: "Doe",
  age: 50,
  eyeColor: "blue"
};
```

String:

The String object represents a sequence of characters and provides methods for working with strings.

```
const message = 'Hello, World!';
console.log(message.length); // Output: 13 (length of the string)
console.log(message.toUpperCase()); // Output: HELLO, WORLD! (converts to uppercase)

console.log(message.toLowerCase()); // Output: "hello, world!”
console.log(message.indexOf('World')); // Output: 7 (index of the substring 'World')

console.log(str.charAt(0)); // Output: "H"
console.log(str.charAt(3)); // Output: "l"

console.log(str.indexOf('o')); // Output: 4
console.log(str.lastIndexOf('o')); // Output: 8
```

The Date object is used to represent a specific moment in time. It provides methods to work with dates and times, allowing you to perform various operations like creating, formatting, and manipulating dates.

```
const currentDate = new Date();
console.log(currentDate); // Output
```

get month

```
const currentDate = new Date( );
const month = currentDate.getMonth( );
console.log(month); // Output: Current month (0 to 11)
```

get date

```
const currentDate = new Date();
const day = currentDate.getDate();
console.log(day); // Output: Current day of the month (e.g., 29)
```

get Day

```
const currentDate = new Date();
const dayOfWeek = currentDate.getDay();
console.log(dayOfWeek); // Output: Current day of the week (0 to 6)
```

tolocaltimestring

```
const currentDate = new Date( );
const formattedTime = currentDate.toLocaleTimeString( );
console.log(formattedTime);

// Output: Time in local format (e.g., "12:34:56 PM" )
```

tosettime

setInterval( ) : function is used to form an event that runs in the background and calls a particular function with a fixed time delay.

The clearInterval( ) : this is a method used to find the event with the specified ID and remove it.

setTimeout()

- It is a function in JavaScript that allows you to execute a specified function or code snippet once after a specified delay.
- It's used to schedule a single execution of a function after a certain amount of time has passed.

```
setTimeout(function, delay);
```

```
// Call setInterval() to execute the function sayHello() every 3 seconds
(3000 milliseconds)

function sayHello() {
  console.log("Hello, World!");
}

let h=setInterval(sayHello, 3000);
setTimeout(() => {
 console.log(h);
 clearInterval(h);
}, 10000);
```

The addEventListener( ) method attaches an event handler to the specified element.

The addEventListener( ) method attaches an event handler to an element without overwriting existing event handlers.

```
document.getElementById("myBtn").addEventListener("click", displayDate);
```