**{ JavaScript }]**

Must use emmet in web development

{ it will increase coding speed }

1. Variables:

There are thre main variables in javascrip.

1. Var
2. Let
3. Const

Variable (Var):

Variables are containers for storing data (values).

var x = 5;

var y = 6;

var z = x + y;

*2) LET* :

Variables defined with let cannot be Redeclared.

Variables defined with let must be Declared before use.

 Variables defined with let have Block Scope.

3) const:

Variables defined with const cannot be Redeclared.

Variables defined with const cannot be Reassigned.

Variables defined with const have Block Scope.

## **Events in javascript:**

The HTML DOM allows you to execute code when an event occurs.

Events are generated by the browser when "things happen" to HTML elements:

* An element is clicked on
* The page has loaded
* Input fields are changed

{Important}

JavaScript Objects:

In JavaScript, objects are king. If you understand objects, you understand JavaScript.

**Three simple ways to initialize/code object:**

**1)**

  const person = {

            firstName: "John",

            lastName: "Doe",

            age: 50,

            eyeColor: "blue"

          };

**2)**

    <script>

        const person = {};

        person.firstName = "John";

        person.lastName = "Doe";

        person.age = 50;

        person.eyeColor = "blue";

    </script>

**3)**

    <script>

        const person = new Object();

        person.firstName = "John";

        person.lastName = "Doe";

        person.age = 50;

        person.eyeColor = "blue";

    </script>

## **JavaScript Properties:**

* Properties are the values associated with a JavaScript object.
* A JavaScript object is a collection of unordered properties.
* Properties can usually be changed, added, and deleted, but some are read only.

## **Accessing JavaScript Properties**

*objectName.property*// person.age

or

*objectName*["*property*"]   // person["age"]

or

*objectName*[*expression*]   // x = "age"; person[x]

Object:

This code assigns a **simple value variable**

  <script>

        // Create an object:

        const person = {

          firstName: "John",

          lastName: "Doe",

          age: 50,

          eyeColor: "blue"

        };

        // Display some data from the object:

        document.getElementById("demo").innerHTML =

        person.firstName + " is " + person.age + " years old.";

        </script>

# Prototype:

# The prototype is a global constructor available for all JavaScript objects.

* The prototype constructor allows you to add new properties and methods to booleans.
* When constructing a new property, ALL arrays will get the property and its value.
* When constructing a new method, ALL arrays will get the method.

## **Syntax**

Boolean.prototype.*name*= value

<p id="demo"></p>

<script>

Boolean.prototype.myColor = function() {

  if (this.valueOf() == true) {

    return "green";

  } else {

    return "red";

  }

};

let a = true;

document.getElementById("demo").innerHTML = a.myColor();

</script>

# Class with static function:

The static keyword defines static methods for classes.

Static methods are called directly on the class (Car from the example Below) - without creating an instance/object (mycar) of the class.

<script>

    class Car {

        constructor(brand) {

          this.carname = brand;

        }

        static hello() {  // static method

          return "Hello!!";

        }

      }

      mycar = new Car("Ford");

      //Call 'hello()' on the class Car:

      document.getElementById("demo").innerHTML = Car.hello();

      //and NOT on the 'mycar' object:

      //document.getElementById("demo").innerHTML = mycar.hello();

      //this would raise an error.

</script>

# JavaScript Class super Keyword

* Create a class named "Model" which will inherit the methods from the "Car" class, by using the extends keyword.
* By calling the super() method in the constructor method, we call the parent's constructor method and gets access to the parent's properties and methods:
* <script>
* class Car {
* constructor(brand) {
* this.carname = brand;
* }
* present() {
* return 'I have a ' + this.carname;
* }
* }
* class Model extends Car {
* constructor(brand, mod) {
* super(brand);
* this.model = mod;
* }
* show() {
* return this.present() + ', it is a ' + this.model;
* }
* }
* mycar = new Model("Ford", "Mustang");
* document.getElementById("demo").innerHTML = mycar.show();
* </script>

**Document Arttributes [ document.getelementby… ]**

# Attributes:

# An HTML attribute always belongs to an HTML element.

getElementsByTagName:

it represent number of Tags inside the class or function.

<script>

    function myFunction() {

      var a = document.getElementsByTagName("BUTTON")[0];

      var x = a.attributes.getNamedItem("onclick").value;

      document.getElementById("demo").innerHTML = x;

    }

    </script>

# Rest parameter:

Reset parameter is use in function as a argument and it can consume more than one parameter/ argument.

    <script>

        function student(a, ...inputs){

            document.write(a);

            document.write(inputs);

        }

    student('MUHAMMAD', 'Ijaz', 'khan', 'Uet peshawar', '0068');

    </script>

In this example only “MUHAMMAD” is place in “a”, all other peramerts are handle by “…inputs” arguments.

IEFE FUNCTION: