INTRODUCTION TO JAVASCRIPT

Why Should I learn Javascript ?

- You must learn JavaScript if you want to get into **web development**, and you must learn it well if you're planning on being a **front-end developer**or on using JavaScript for **back end development**.

- JavaScript allows you to build **interactive websites**. JavaScript has become an essential web technology along with HTML and CSS, as most browsers implement JavaScript.

What are pre-requisites/ requirements to learn JavaScript

- Basic computer knowledge.
- Basic programming knowledge's like C or java(Not mandatory).
- Should have knowledge of HTML and CSS.

What is Javascript ?

Javascript is a **high level, light weight, interpreted client side scripting language**. It is used primarily by web browsers to create dynamic web pages.

- It is computer programming language that runs inside an internet browser.
- Inside normal webpage you place javascipt code. When the browser loads the page,the browser has a built-in interpreter that reads the javascript code and runs it.

High level language- It enables development of a program in a much more user-friendly programming context and is generally independent of the computer's hardware architecture.

Interpreted Language- It is executed at the runtime. No compilation is required to translate to machine-language.

Difference between scripting language and programming language

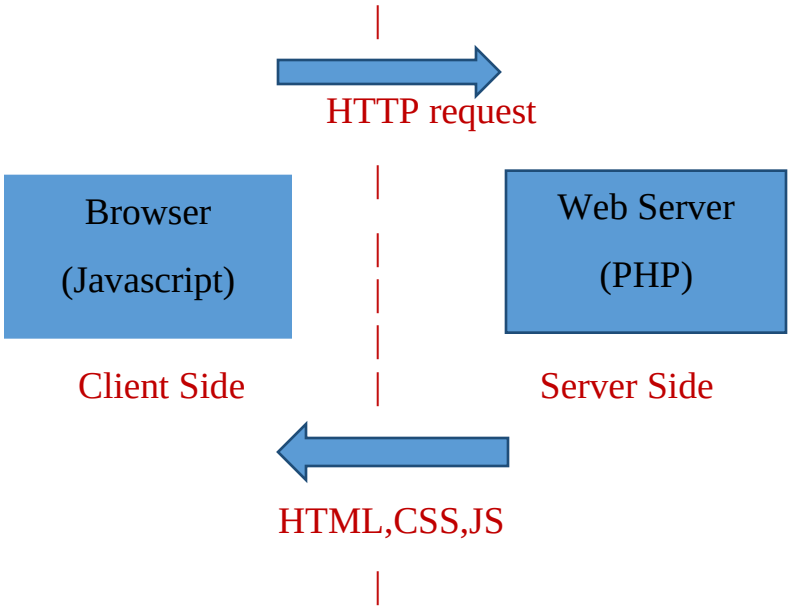
- **Scripting languages**are not compiled to machine code by the user. Rather another program called interpreter runs the program.
  Ex: Python, Perl, Shell.PHP,JSP
- **Programming languages**are compiled to machine code by user.
  Ex: C, Java.

Client side scripting language– The script or program which runs in web browser. Here web browser is called client.

Ex: Javascript, Python, Perl

Server side scripting language– The script or program which runs in web server.

Ex: PHP,JSP



Features of Javascript:

- Validating users input.
- Simple Client-side calculations.
- Easy to learn.
- Weakly or loosely typed language.
- Platform independent.
- Generating HTML content.
- Interpreter Based.
- Event Handling.
- Object-oriented scripting language.
- Case sensitive.

Software's required to write Javascript

- Any Web browser(Preferred Chrome or Firefox)
- Basic text Editor or IDE like Notepad++ or VS code

Popular text editors for Front-end development.

1. VS code (https://code.visualstudio.com/) - FREE
2. Atom (https://atom.io/) - FREE
3. Sublime (https://www.sublimetext.com/) - FREE
4. Webstorm(https://www.jetbrains.com/webstorm/) - PAID

How to add Javascript to web page or HTML

You can include Javascript in your HTML in two ways

- Including the code in your HTML

The **<script>** element is used to include JavaScript within an HTML(header or body section) document.

<script type="text/javascript">

var name=”hello”; // jsvariables

console.log(name); // js console

</script>

- Including using External javascript file

you can write JavaScript code in a separate file with .js extension and include it in a web page using <script> tag and reference the file via src attribute

you can include script tag either in head setion or body section of HTML.

<script src=”pathtoscriptfile.js”></script>

DATA TYPES AND VARIABLES

Javascript provides different data types to hold different types of values. There are two types of data types in javascript.

- Primitive data type.
- Non-primitive (reference) data type.

Primitive Data types:

String : It is a sequence of characters(it is surrounded with double or single quotes).

Ex: “Angular”, ‘Js’

Number : It represents numerical values.

Ex:20, 15.0**5**

Boolean : It has two values i.e true or false.

Undefined : It represents undefined value.

Null : It represents null value i.e nothing or no values.

Non-primitive Data types:

Object : It is a complex data type that allows you to store collection of property value pairs. Objects can have both properties and methods.

Ex :Person,Car,Movie,Pen

Ex : var car={

model:"Nexa",

color:"White",

cost;1000}

Array : It is single variable that is used to store multiple elements. Each element in array has a numeric position ,known as index. The Array index starts from 0.

Ex:var colors=[“red”,”green”,”blue”];

console.log(colors[0]) // it prints output as red

Function :A function is a block of code which will be executed only if it is called. If you have a few lines of code that needs to be used several times, you can create a function with the repeating lines of code and then call the function whenever it is required.

Ex:function greet(){

```
  console.log("hello");                    // defining a function
}
```

greet() // calling a function , it prints “Hello” in console

The typeof operator

Thetypeof operator can be used to find the data type of variable. It can be used with or without parentheses(typeof(a) or typeof x)

Examples:

- **Numbers :**

  typeof**25**// Returns :“number”

  typeof**42.7** // Returns : “number”

- **Strings :**
  typeof**‘ ’** **//** Returns : “string”

  typeof**“Hello”** // Returns : “string”

  typeof**“123”** // Returns : “string”

- **Boolean :**
  typeof**true** // Returns : “boolean”
  typeof**false** **//** Returns : “boolean”

- **Undefined :**
  typeof **undefined** **//** Returns : “undefined”
  typeof**undeclared variable** // Returns : “undefined”

- **Null :**
  typeof **null** // Returns : “object”

- **Objects :**
  typeof **{name:”hyd”,age:18}** //Returns : “object”

- **Arrays :**
  typeof **[1,2,3]** // Returns : “object”

- **Functions :**
  typeof **function(){}** // Returns : “function”

Variables

Variables are used to store data, like string,numbers etc. variables can be changed any time.

Javascript variables are divided into two types

- Local variables
- Global variables

Javascript uses reserved keyword **var** to declare variable. You can assign value to variable using **equal to (=)** operator.

There are some rules while declaring a javasciptvariable (also known as identifiers)

- Variable name must start with the letter(a to z or A to Z),underscore(_),dollar ($) sign
- Javascript variables are case sensitive, for example **name** and **Name** are different variables
- Variable names should not be reserved keywords like **var,if,switch,for**

Syntax :

var<variable-name>;

var<variable-name> = <value>

Valid Variables

var x=20; // variable stores numeric value

var_name = “venkat”; // variable stores string value

varisMarried = false; // variable stores Boolean value

var three; // declared variable without assigning value

var ten=10, five=”five”, two; // declared multiple variables in single line

Invalid Variables

var 123=300 // variable should not start with numeric vlue

var#city= “HYD” // variable should not start with special characters

Local Variables

A JavaScript local variable is declared inside block or function. It is accessible within the function or block only.

Ex: function add(){

var x=10;//local variable

}

Global Variables

A JavaScript global variable is accessible from any function. A variable i.e. declared outside the function or declared with window object is known as global variable.

Ex: varvalue=200; //gloabal variable

function a(){

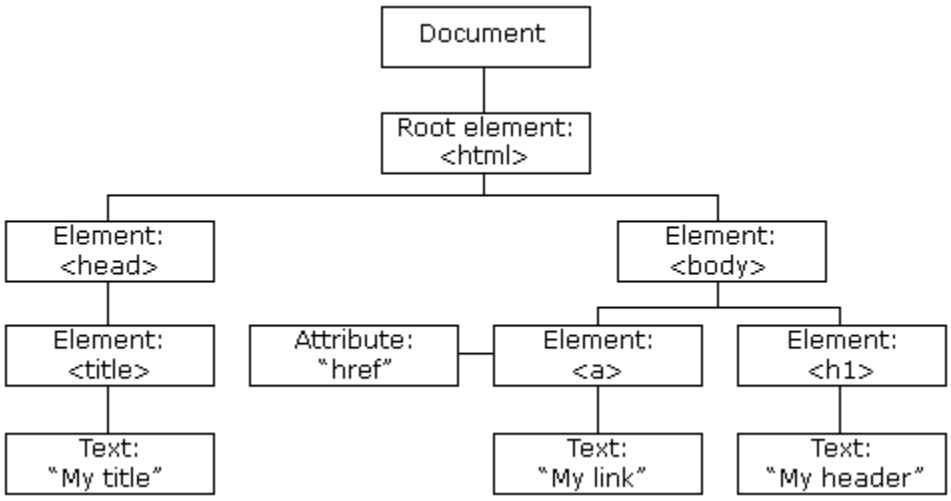
document.write(value);

}

DOCUMENT OBJECT MODAL (DOM)

Webpage is a document created by using HTML language. Where you can view the content of web page in browser. You need certain methods to access content on this document like changing the style or adding or removing the content.

DOM is a programming interface which gives you the ability to access the content present inside the document. The Dom interface interprets the document as a collection of objects.



DOM Methods

Dom methods are used to access HTML elements.

- **getElementById**
  You can access an HTML element by using id of the element.
  Ex :document.getElementById("test")
  document.getElementById("test").innerHTML = "Hello World";

  - Here document is an object
  - getElememntById is a method.
  - innerHTML is a property

  You are finding an element with id as "test" and replacing content of it using innerHTML property

- **getElementsByTagName**
  You can access an HTML element by using name of the element.

  Ex:document.getElementsByTagName('p')
  document.getElementsByTagName('p')[0].innerHTML="Hello"

  getElementsByTagName is a method which returns all paragraph elements present in HTML

- **getElementsByClassName**
  You can access an HTML element by using class name of the element.

  Ex:document.getElementsByClassName("myClass")
  document.getElementsByClassName('myClass')[0].innerHTML="Hello"

- **querySelectorAll**
  The querySelectorAll() method returns all elements in the document that matches a specified CSS selector(s)

  Ex: document.querySelectorAll(".test")[0].style.color="red";
  Here you are accessing all the elements whose class is test

OBJECTS

What is javascriptObject : It is a complex data type that allows you to store collection of property value pairs. Objects can have both properties and methods

Object Creation : In javascript object can be created in two ways

- Object Literal
- Object constructor (new keyword)

Object Literal :

The object literal is a simple way of creating an object using { } brackets.

syntax : var<object-name> = { key1: value1, key2: value2};

var**emptyObject** = {}; // object with no properties or methods

var**person** = { firstName: "kohli" }; // object with single property

//**object with single method**
var**message** = {
showMessage: function (val) {
console.log(val);
}
};

// **object with properties & method**
var**person** = {
firstName: "virat",
lastName: "kohli",
age: 31,
getFullName: function () {
return this.firstName + ' ' + this.lastName
}
};

Access JavaScript Object Properties & Methods
You can get or set values of an object's properties using dot notation or bracket. However, you can call an object's method only using dot notation.

```
var person = {
firstName: "virat",
lastName: "kohli",
            age: 25,
getFullName: function () {
                return this.firstName + ' ' + this.lastName
            }
        };

person.firstName;         // returns virat
person.lastName;          // returns kohli

person["firstName"];      // returns virat
person["lastName"];       // returns kohli

person.getFullName();   // returns viratkohli
```

Object Constructor:

The second way to create an object is with Object Constructor using new keyword. You can attach properties and methods using dot notation

```
var person = new Object();

// Attach properties and methods to person object
person.firstName = "virat";
person["lastName"] = "kohli";
person.age = 25;
person.getFullName = function () {
      return this.firstName + ' ' + this.lastName;
   };

// access properties & methods
person.firstName; // virat
person.lastName; // kohli
person.getFullName(); // viratkohli
```

Undefined Property

JavaScript will return 'undefined' if you try to access properties or call methods that do not exist.

If you are not sure whether an object has a particular property or not, then use hasOwnProperty() method before accessing properties

```
var person = new Object();

person.firstName; // returns undefined

if(person.hasOwnProperty("firstName")){
person.firstName;
   }
```

Delete Operator

You can use the delete operator to completely remove the properties from the JavaScript object

```
var person = {
      name: "Harry",
      age: 16,
      gender: "Male"
   };

   // Deleting a property completely
   delete person.age;
   alert(person.age); // Outputs: undefined
   console.log(person); // Prints: {name: "Harry", gender: "Male"}
```