**Programming with JavaScript**

**Introduction to JavaScript:**

* JavaScript is a Scripting Language
* Many used in web pages to improve design, validate forms, detect browsers, and create cookies.
* By default, all web browsers have in built JavaScript support.
* JavaScript is the most popular programming language in the world and that makes it a programmer’s great choice. Once you learnt JavaScript, it helps you developing great front-end as well as back-end software’s using different JavaScript based frameworks like jQuery, Node.JS etc.

There are many useful **JavaScript frameworks** and libraries available:

* Angular ------------- **framework**
* React ------------- **library**
* jQuery ------------- **library**
* Vue.js ------------ **library**
* Node.js ------------- **environment or backend**

**Website VS Web Application**

**Server Side VS Client Side Script**

**How Use JS?**

**JavaScript object document?**

**document.write(“Hey”);**

**how to comment?**

**Implementation of HTML.**

**Variables : var , let , const … difference between.**

**Data types. String, number, Boolean, Array, object, function, null, undefined.**

**Chrome console.**

**Global Variable VS local Variables.**

**Arithmetic Operator: + , - , \* , / , \*\* , % , ++ , --**

**Assignment Operator : = , += , -= , \*= , /= , %=**

**Comparison Operator : == , === , != , !== , > , < , >= , <=**

**If statement in details.**

**Logical operator. It is use in if statement. && , || , !**

**Concatenation**

**Template String.**

**Alert.**

**Confirm box**

**Prompt**

**Prompt with parseint();**

**Class Assignment: make marksheet with parseint();**

**Function.**

**All Loops.**

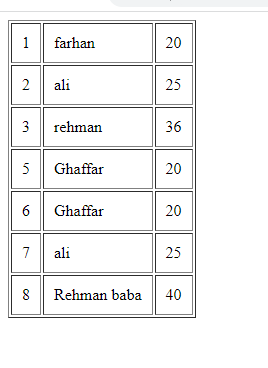
**For loop**

**While loop**

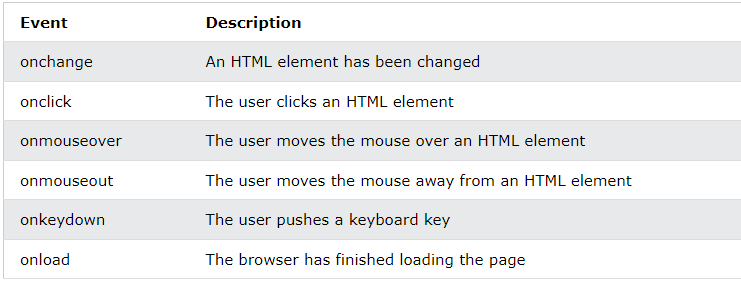
**Do while loop**

**Assignment .. find odd, even numbers.**

**Create this using table , loop and array**

****

**JavaScript Events**

****

**DOM**

Document Object Model:

The **HTML DOM** model is constructed as a tree of **Objects**:

* JavaScript can change all the HTML elements in the page
* JavaScript can change all the HTML attributes in the page
* JavaScript can change all the CSS styles in the page
* JavaScript can remove existing HTML elements and attributes



**HTML DOM Target Methods:**

* getElementById
* getElementsbyClassName
* getElementsbyTagName

use however you want.

**Get with DOM**

* Html
* Text
* Attribute

**For Get:**

* innerText **important**
* innerHTML **important**
* getAttribute for find attribute value
* getAttributeNode for find attribute
* Attributes for find all Attributes.

**For Set:**

* innerText
* innerHTML
* Setattribute
* Attributes **use in array syntax**
* Removeattributes.

**JavaScript Query Selector:**

* QuerySelector
* QuerySelectorAll **in the sense of array**

**DOM Styling In javascirpt**

* style
* className **for add class use with =**
* classList **return in array. Add()**

document.getElementById("heading").style.backgroundColor = "red";

**Javascript Add Event Listener Method:**

It is use for events.

Syntax:

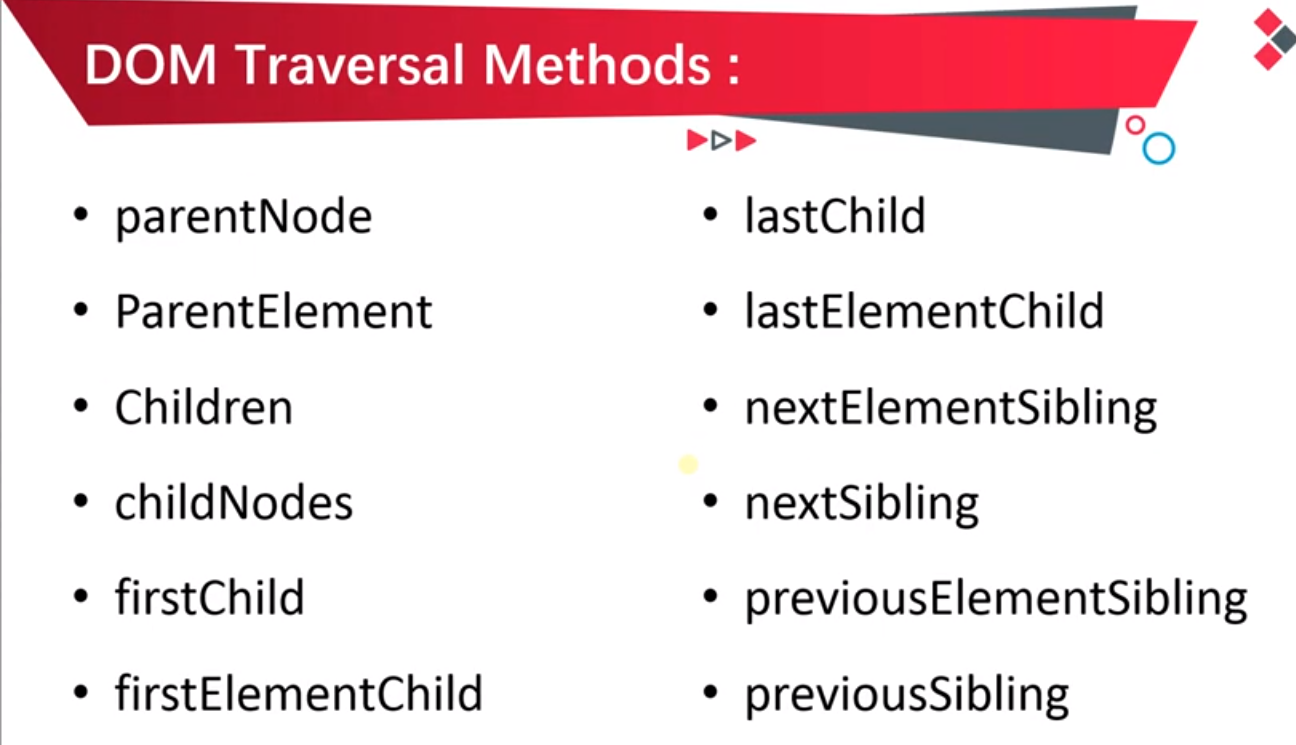
document.getElementById(“head”). addEventListener(“event”,functionname);

**Example:**

document.getElementById("hl").addEventListener("click",function(){ this.style.backgroundColor = "red";

})

**Traversal Method in JavaScript:**



**This All use in Practical.**

**DOM Create Methods:**

* createElement
* createTextNode
* createComment

**Example:**

console.log(document.createElement("p"))

console.log(document.createTextNode("Hey i am farhan"))

console.log(document.createComment("end of website"))

**DOM Append Method:**

* appendChild or append
* insertBefore

**appendChild:**

Example:

console.log(document.getElementById("para").append("HEy jf   is append text"))

**Big Example:**

var element = document.createElement("h1")

var node = document.createTextNode("Hey This Awesome place");

element.append(node)

document.getElementById("para").append(element)

**insertbefore**

it is like appendchild but it is add child in start.

How to get value from input:

**Array Methods**

Sort

Reverse

Push to add value

Pop to remove value

**indexOf ..** it is find to data in array and return index num. if don’t so return -1;

**includes** method for name but it is return true or false.

**Call back functions:**

**Find** : to find a first index number for find start values.

**FindIndex**: same as a find method. a first index number for find index number.

**Filter** to find values in array and create another array.

Syntax;

 var age = [20,50,20,1,4,14]

   console.log(age)

   var b = array.filter(function(nam){

     return nam>=18

    }

   )

 console.log(b)

**For each Loop:**

Just like another but It is very easy as compare to for loop.

**Syntax:**

     var array = ["farhan","ali","arif","rehman"]

     array.forEach(function(vl){

       document.write(vl + " <br>")

     })

Array.forEach(function(val,index){

Document.write(val + index);

})

**For In LOOP:**

It is use only for object.

      var obj = {id :1 , name: "farhan",age : 20}

     for (const key in obj) {

        console.log(obj[key])

     }

**Map Method In JS:**

It is use for creating gallery or other things.

     array.map(function(val,index){

       console.log(val + index)

     })

**Date Use in JS:**

     var now = new Date();

     console.log(now.toDateString());

     console.log(now.toTimeString());

we can fix the date.

     var now = new Date("June 19 2021");

     console.log(now.toDateString());

**Set Interval & Clear Interval:**

     setInterval(function(){

       document.write("farhan");

     }, 1000);

Set Clear is use for clear the interval but in any condition:

Syntax:

SetClear(setInterval variable name);

**Set TimeOut & set clear:**

It is running like a set interval but it is run only one time:

**Syntax:**

     setTimeout(function(){

      console.log("hello")

     },5000)

**ClearTimeOut:**

**Syntax:**

clearTimeOut(setInterval variable name);

**Example:**

   var a =  setTimeout(function(){

      console.log("hello")

     },5000)

     function stopanim(){

       clearTimeout(a)

     }

**BOM (Browser Object Model)**



**Window Height & Width:**

For Method to Get browser (window) height & width.

* innerHeight
* innerWidth
* outerHeight
* outerWidth

It is also work with events. like onscroll , onresize.

**Window Open & Close Method**

**Open:**

Window.open(URL,name,specs)

* URL: website link with protocol.
* Name or Target: give target it is great. \_blank , \_self etc.
* Spec: Width, height, left, top

Window.close()

**Location Object:**

Describe location object.

And use in console.log.

* hash
* host
* hostname
* href
* origin
* pathname
* port
* protocol
* search

**Location Methods.**

Location.reload()

Location.assign(“url”)

Location.replace(“”)

Replace like assign but it destroy history.

**History Object (length):**

History.back

History.forward

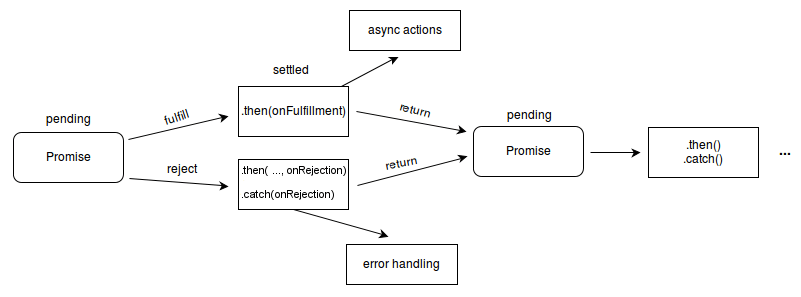
History.go it is use for both like history.go(1) or (-1)

window.scrollY with use on event.

**Fetch the API:**

**What is Promise?**

The Promise object **represents the eventual completion (or failure) of an asynchronous operation and its resulting value**.



**Fetch in method:**

The fetch() method in JavaScript is **used to request to the server and load the information in the webpages**. The request can be of any APIs that returns the data of the format JSON or XML. This method returns a promise. Syntax: fetch( url, options ).

fetch('https://fakestoreapi.com/products')

.then(res=>res.json())

.then(json=>console.log(json))

**Simple with normal function:**

fetch("https://jsonplaceholder.typicode.com/users").then(function(res){

        return res.json();

    }

).then(function(resp){

        console.log(resp)

    }

)

**Another Example:**

fetch("https://jsonplaceholder.typicode.com/users").then(function(res){

        return res.json();

    }

).then(function(resp){

  console.log(respa)

    for(var i=0;i<resp.length;i++){

        document.write(resp[i].id +") " + resp[i].name + " ---- "+ resp[i].username + "<br>" );

    }

    }

).catch(function(error){console.error("You have a error")})