**JavaScript**

Is a programming language that executes on the browser. It turns static HTML web pages into interactive web pages by dynamically updating the content.

**JavaScript PopUp message**

Javascript basically provides built in global functions to display pop up message boxes for different purposes.

1. alert : Displays a popup box with the specified message with ok button.
2. confirm(): Displays a popup box with the specified message with OK and cancel button
3. Prompt(): Display a pop up box to take the user’s input with the Ok and cancel button.

**Javascript Variables**

Variable means anything that can verify. In Java script, stores data.

In Javascript variable, a variable can be declared using var, let, const keywords

* Var keyword : It is used to declare variables when Javascript was developed.
* let keyword: It is new and more recommended way of declaring variables in Javascript.

**Variable Scope**

In Javascript, a variable can be declared either in the global scope or the local scope.

**Global Variables:**

Variables declared outside any function are basically global variables. They can be easily accessed anywhere in the javascript code, even inside any function.

**Local Variables :**

Variables declared inside the function are called local variables. There accessibility is only within that function, it cannot be accessed outside the function.

**Javascript Operators**

Javascript includes following catgories

1. **Arithmetic Operators**

Arithmetic operators are used to perform mathematical operations between numeric operands

+ 🡪 Adds two numeric operands

* 🡪 Subtract right operand from left operand

\* (This is used to multiply two operands)

/ (Divide left operand by right operand

% (Modulus operator. Returns remainder of two operands)

++ ( Increment operator. Increase operand value one by one)

--(Decrement operator. Decrease value one by one)

2. **Comparison Operators**

Javascript provides comparison operators that help us in comparing two operans and return a Boolean value.

== Compares the equality of two operands without considering type  
 === Compares the equality of two operands with type

!= Compares inequality of two operands

>(Return a Boolean value true if the left side value is greater than right side value, otherwise returns false

<(Returns a Boolean value true if the left side value is less than the right side value: otherwise it will return false

>=(Returns a Boolean value if the left side value is greater than or equals to the right side value: otherwise it will return false

<=(Returns a Boolean value if the left side value is less than or equals to the right side value: otherwise it will return false

**Ternary Operators**

Javascript provides a special operator called ternary operator :? This assigns a value based on some condition. You can consider this as short form of if else condition

<condition> ? <value1> : <value2>

**Functions In JavaScript**

Functions is the basic building block. In JavaScript, a function can be defined using function keyword, that is basically followed by the name of function.

function m1(firstName, lastName)

{

………………………………..

}

**JavaScript If…else statements**

JavaScript includes if / else conditional statements

1. If statemet
2. If else statement

In if else statement, else statement executes when if condition becomes false.

**Javascript Else If Condition**

Else if condition : When you want to apply second level condition after if statement

**JavaScript switch**

The switch is a statement when you want to execute multiple code blocks based on the return value. Switch can include multiple cases where each case represents some value. If none of the case matches with switch expression value then then the default case will be executed. Always use break keyword to stop the execution and exit from the switch.

**Syntax**

switch(expression){

case 1:

// code to be executed

break;

case 2:

// code to be executed

break;

default:

//default code to be executed

}

**JavaScript For Loop**

If you want to execute code repeatedly you can use for loop

**Syntax**

for(initializer; condition; iterator)

{

// code to be executed

}

**JavaSCript – While Loop**

Javascript used while Loop to execute code repeatedly till it satisfies a specified condition. Unlike for loop, while loop only requires condition expression.

**Syntax**

while(condition expression){

/\* code to be executed till the specified condition is true \*/

}

**do while**

do-while loop evaluates expression after the execution of code block. So do-while loop will execute the code block at least once

**Syntax**

do{

// code to be executed

}while(condition expression)

**Error handling In JavaScript**

In Javascript code you can get compile time errors. Javascript provides error-handling mechanism to catch exceptions using **try-catch-finally** block

**Syntax**

try

{

//code that may throw an error

}

catch(ex){

// code to be executed if an error occurs

}

finally

{

// code to be executed regardless of an error occurs or not.

}

**try:** Inside the try block there is code that may throw an exception or may not throw an exception

**catch:** write code to do something in catch block when an error occurs. The catch block you can give the statements that will give you error information.

**finally:** code in the finally block will always be executed in either or the other case.

**throw:** We use throw keyword to raise a custom error.

**JavaScript Strings**

In Javascript Sting is primitive data type it can be enclosed in single quotes, double quotes.

“Hello Champs”

‘Hello Champ’

var a = “Hello champ how doing in JS”;

var a = ‘Hello champ how doing in JS’ ;

Javascript string can be treated like a character array also. In a String you can access character using square brackets

“Hello Champs”

**String Objects**

Javascript allows to create a string object using **new** keyword

**String Comparison**

Two strings can be compared using **<**

**>**

**==**

**string.localeCompare(string)**

**===** operator compares the reference of String objects and not the values

**Functions In JavaScript**

Functions is a piece of code in a block its basic Syntax is

function <function-name>(arg1,arg2,arg3,,,,,)

{

// write a code

}

Function parameters

You can actually pass values to the function using parameters. A function can have one or more parameters

**Javascript Date Object**

**This** can be used to get year, month and day. You can display it on webpage by the help of Javascript date object.

**Constructor:** Constructor is invoked at the time of object creation. Constructor are of two types:

Default constructor

Paremetrized constructor

new String();

new String("Hello Champs!");

We have 4 variant of Date Constructor to create date object:

1. Date()
2. Date(milliseconds)
3. Date(dateString)
4. Date(year,month,day,hours,minutes,seconds,milliseconds)

**Methods Description**

**getDate()** It returns the integer value between 1 and 32 that represents the day for the specified date on the basis of local time.

**getMonth()** It returns month as a number (0 - 11)

**getFullYear()** It returns year as a four digit number(yyyy)

**getDay()** It returns a weekday as a number(0-6)

**getHours()** It returns hour(0-23)

**getMinutes()** It returns minute(0-59)

**getSeconds()**  It returns second(0-59)

**getTime()** GetTime(milliseconds)

let yesterday = new Date(currentDate);

yesterday.setDate(yesterday.getDate()-1)

**JavaScript Arrays**

We cannot assign multiple values to a single variable. Javascript array is a special type of variable, which can store multiple values.

let numArray = [1,2,3,4,5];

One more way of creating arrays that is using the Array() constructor

let numArray = new Array(1,2,3,4,5);

let strinArray = [“one”,”two”,”three”];

let booleanArray = [true,false,false,true]

let data = [1,”Peter”,true,56]

To get the size of an array les length

**You** can iterate an array using for Loop and for each loop

Push method in JS is used to add element at last

shift method --> Removes first element from an array

concat()🡪 Returns new array by combining values of any two arrays

indexOf() 🡪 Returns the index of the first occurrence of the specified element in an array, or -1 if its not found.

join() 🡪 Returns string of all the elements separated by the specified separator.

reverse() 🡪 Reverse the element of an array. Element at the last index will be first and element at 0 index will be last

sort() 🡪 sort the elements of an array