**Javascript**

**Introduction:-**

A script is a list of commands that are executed by a scripting engine. Scripts are used to generate dynamic web pages on the web. Scripts can be opened and edited by using a text editor. There are a variety of scripting languages used to develop dynamic web pages. Javascript was initially created to make web pages alive. They don’t need a special preparation or a compilation to run. Using HTML one can only design a web page but cannot implement any logic on web browsers like addition of two numbers , check any condition , loping statements, decision making statements etc. This is possible by embedding Javascript block into HTML.

Javascript can be used as client side or server side scripting language. Javascript can be inserted in HTML program between <script> and </script> tag. Language attribute is used to set scripting language. Script can be placed in <body> or in <head> section of HTML or in both.

**Variables:-** It is a unit of storage in Javascript program where value is changing.

**Rules to declare variables:-**

1. They must start with an alphabet.
2. Uppercase and lowercase letters are different in Javascript.
3. It should not contain blank space or special symbol except underscore.
4. Standard keywords are not allowed as variable name.
5. Variable name is limited up to 255 characters.

**Data types:-** Following data types are used in javascript.

* **Number:-** It stores both whole number (integer) or decimal point numbers (float). Number data type can hold positive as well as negative values.
* **String:**- It stores text. Strings must be inside double or single quotes.
* **Boolean:**- It represents only two values true and false. All relational, conditional and logical operators produce Boolean values either true/false or yes/no.
* **null:**- It is just a value which means “nothing”, ”empty”, ”unknown”.

Variables in javascript are declared using keyword var or let or const. var is the oldest format to declare variables. let keyword was introduced in 2015. const keyword is used to declare variables with constant value.

Difference between let and const is that using var you can redeclare the variable but using let you cannot redeclare the variable.

**Operators:-** Operators are used to do arithmetic and logical operations. Operators that require one operand is called as unary operator and operators that require two operands are called as binary operators. Following operators are used in Javascript.

**Arithmetic operators:-**These operators are used to perform mathematical operations. Following arithmetic operators are used in Javascript.

* +:- for addition
* -:- for subtraction
* \*:- for multiplication
* /:- for division
* %(modulus):- to get remainder of division

‘+’ operator is also used as concatenation operator which is used for combining two strings.

**Relational operators:-** These operators are used to check conditions or to compare operands. Result of relational operators is Boolean value true or false. Following relational operators are used in javascript.

* **< (less than)**
* **>(greater than0**
* **<=(less than equal to)**
* **>=(greater than equal to)**
* **==(Equal to)**
* **=(assigning values)**
* **!=(not equal to)**

**Logical operators:-** Logical operators are used to verify more than one condition at a time or to negate the condition. Following logical operators are used in javascript.

* **&&(and):-** It evaluates to true only when all conditions are true.
* **||(or):-** It evaluates to true only when any one condition is true.
* **!(not):-** It returns true when condition is false.

**Increment (++) and decrement (--) operators:-** Increment operator is used to increment value of variable by one and decrement operator is used to decrement value of variable by one. They can be used in two ways:-

++x(pre-increment):- Value of variable x is incremented before it is used in expression.

x++(post-increment):- Value of variable x is incremented after it is used in expression.

--x(pre-decrement):- Value of variable x is decremented before it is used in expression.

x--(post-decrement):- Value of variable x is decremented after it is used in expression.

**Comments in Javascript:-** Comments are non-executable statements in program. Comments are used to provide information or explanation about your programming construct. It supports two types of comments:

1. Single line comment:- Single line comment begin with //. Javascript ignores everything from // to the end of line.
2. Multiline comment:- Multiline comments are used to comment on more than one line. It starts with /\* and end with \*/.

**Commonly used Built-in functions in Javascript:-**

**1. parseInt():-** It is used to parse a string and convert it into a number.

**2. parseFloat():-** It is used to parse a string and convert it into a floating point

representation.

**3. alert():-** It displays alert popup box with ok button. This is also called as message

box.

**4. prompt():-** This function is used when you want input value from user at the

time of program execution.

**5. confirm():-** It displays confirmation box with ok and cancel buttons.

**6. toLowerCase():-** Thjis string function is used to convert the given string into

lower case alphabets.

**7. toUpperCase():-**Thjis string function is used to convert the given string into

upper case alphabets.

**8. isNaN():-** It returns true is given value is not a number. It returns false if given

value is number.

**Statements in Javascript:-**

There are two types of statements used in Javascript.

1. Conditional statements/Decision making statements
2. Loop/Iteration statements

**Conditional Statements:-**

Conditional statements are of following types:-

1. if statement:-

syntax:- if(condition)

{

Statement block;

}

If the conditional expression given in the parenthesis is true, then the statements within the block will be executed, followed by execution of the remaining program. If the conditional expression evaluates to false the the statement block will not be executed.

1. if else statement:-

syntax:- if(condition)

{

Statement block;

}

else

{

Statement block;

}

If the conditional expression given in the parenthesis is true, then the statements within the block will be executed, otherwise false block that is the else part will be executed.

1. Switch statement:-

Syntax:- switch(expression)

{  
  case 1:  
    *// code block*    break;  
  case 2:  
    *// code block*    break;  
  default:  
    // code block  
}

The switch expression is evaluated once. The value of the expression is compared with the values of each case. If there is a match, the associated block of code is executed. If there is no match, the default code block is executed.

**Loop statements:-**

Loop statements execute a block of code number of times. Following loop statements are used in Javascript:-

1. For loop:-

Syntax:- for (*statement 1*;*statement 2*;*statement 3*)

{  
  // *code block to be executed*  
}

Statement 1 contains the initial value of variable, statement 2 contains comparison of variable with end value, and statement 3 increments value by 1.

1. While loop:-

Syntax:- while (condition)

{  
*// code block to be executed*

*Value++;*  
}

In this type code block will be executed till condition is true.

1. do while loop:-

syntax:- do

{  
*// code block to be executed*}  
while (condition);

This will work same as while loop only difference is that do while loop will be executed at least once before checking condition.

**Functions:-**

A JavaScript function is a block of code designed to perform a particular task.

A JavaScript function is executed when "something" invokes it (calls it).

Syntax:- function functionname()

{

Code block;

}

This is a simple function. We can pass parameters through function and also we can return value.

**Event Handling:-**

Javascript is an event-driven language. Event is an action done by the user or an application. Javascript’s interaction with HTML is handled through events that occur when user or browser manipulates page. When the page loads, it is called an event, when the user clicks a button, that click is also an event. Other examples include events like pressing any key, closing a window, resizing a window, etc.

Javascript lets you execute a code when events are detected. You can respond to any event using an event handler which is just a function that’s called when an event occurs. This event handler may cause button to close windows, messages to be displayed to users, data to be validated and virtually any other type of response. Some commonly used events are as following:

**Mouse events:-**

* onMouseOut:- This event triggers when user moves the mouse away from an element.
* onClick:- This event triggers when user clicks on element.
* onMouseOver:- This event triggers When user moves the mouse over an element.
* onMouseUp:- This event triggers When user releases a mouse button over an element.

**Keyboard events:-**

* onKeyPress:- This event triggers when user presses and releases a key from the keyboard.
* onKeyDown:- This event triggers when user presses a key.
* onKeyUp:- This event triggers when user releases a key.

**Form events:-**

* onblur:- It occurs when user leaves field or looses focus of an element.
* onfocus:- It occurs when an element gets focus.
* onchange:- It occurs when user changes content of an element or selects dropdown value.
* onselect:- It occurs when user selects some text of an element.
* onsubmit:- It occurs when user clicks submit button.
* onreset:- It occurs when user clicks reset button.
* onload:- It occurs when page or image has been loaded.
* onunload:- It occurs when page or document has been unloaded or closes.

**Javascript objects:-**

Javascript is an object based scripting language. A javascript object is an entity having properties and methods. A object can group data together with functions needed to manipulate it. Properties and methods of objects are accessed with ‘.’ operator. Javascript supports two types of objects:

1. Built-in-objects:- e.g. Math, String, Array, Date etc.
2. User defined objects:- These are created by using ‘new’ keyword. E.g. d=new Date();

**Built-in-objects:-** These objects are available regardless of window content and operates independently of whatever page browser has loaded. These objects provide different properties and methods that are useful while creating live web pages. These objects are as following:

**String Object:-** It is used to store zero or more characters of text within single or double quotes. String object is used to store and manipulate text. I has following properties and methods:-

Properties:-

* length:- Returns the number of characters in a string.

Methods:-

* charAt():- Returns the character at the specified position.
* indexOf():- Returns the index of the first occurrence of specified character in given string.
* lastIndexOf():-Returns the index of the last occurrence of specified character in given string.
* substr():- Returns the characters you specified that means (14,7) returns 7 characters from the 14th character.
* substring():-Returns the characters you specified that means (7,14) returns all characters between the 7th and the 14th.
* trim():- Removes white space from both sides of a string.
* toLowerCase():- Converts a string to lower case.
* toUpperCase():- Converts a string to upper case.

e.g. var str=”Sevenmentor Pvt Ltd”;

document.write(“Length of string is: “ + str.length);

document.write(“Substring is: “ + str.substr(12,7));

**Math Object:-** It includes mathematical constants and functions. There is no need to create the math object before using it. Math object has following methods:

* abs(x):- Returns the absolute value of a number.
* cbrt(x):- Returns the cube root of a number.
* ceil(x):- Returns the next integer greater than or equal to a given number(rounding up).
* floor(x):- Returns the next integer less than or equal to a given number(rounding down).
* max(x,y,…):- Returns the greatest number from the list.
* min(x,y,…):- Returns the smallest number from the list.
* pow(x,y):- Returns the base to the exponent power that is xy.
* sqrt(x):- Returns the square root of a number.

e.g. var x=56.899;

alert(Math.ceil(x));

**Date object:-** It is used to create date and time values. It is created using ‘new’ keyword. There are different ways to create new date object.

var currentdate=new Date();

var currentdate=new Date(milliseconds);

var currentdate=new Date(dateString);

var currentdate=new Date(year,month,day,hours,minute,seconds,milliseconds);

Date object has following methods:

* getDate():- Returns the day of the month (from 1-31)
* getDay():- Returns the day of the week (from 0-6)
* getFullYear():- Returns the year (four digits)
* getHours():- Returns the hour (from 0-23)
* getMinutes():- Returns the minutes (from 0-59)
* getMonth():- Returns the month (from 0-11)
* getSeconds():- Returns the seconds (from 0-59)
* getTime():- Returns the number of milliseconds since midnight Jan 1,1970
* now():- Returns the number of milliseconds since midnight Jan 1,1970
* setDate():- Sets the day of the month of a date object
* setFullYear():- Sets the full year of a date object
* setHours():- Sets the hours of a date object
* setMinutes():- Sets the minutes of a date object
* setMonth():- Sets the month of a date object
* setSeconds():- sets the seconds of a date object
* setTime():- Sets a date to specified number of milliseconds after/before Jan 1,1970

**Number Object:-** It has following properties and methods:

Properties:-

* MIN\_VALUE:- Returns the minimum value.
* MAX\_VALUE:- Returns the maximum value.

Methods:-

* isNaN():- It represents ‘Not a Number’ value.
* isInteger():- It determines whether the given value is a integer.
* parseFloat():- It converts the given string into a floating point number.
* parseInt():-It converts the given string into a integer number.
* isFixed():- It returns a string that represents a number with exact digits after a decimal point.

**Array object:-** It can store a collection of items. Arrays become really useful when you need to store large amounts of data of same type. You can create an array in javascript as following:

var fruits=[“Mango”,”Apple”,”Orange”,”Grapes”];

or

var fruits=new Array(“Mango”,”Apple”,”Orange”,”Grapes”);

You can access and set items in an array by referring to its index number and the index of the first element of an array is zero. arrayname[0] is the first element, arrayname[1] is the second element and so on.

e.g. var fruitname=fruits[0];

document.getElementById(“demo”).innerHTML=fruits[1];

Array object has following properties and methods:

Properties:-

* index:- The property represents zero-based index of the match in the string.
* length:- Reflect number of elements in array.

Methods:-

* concat():- Joins two or more arrays and returns the copy of the joined arrays.
* copyWithin():- Copies array elements within the array to and from specified positions.
* find():- Returns the value of the first element in an array that satisfies a condition.
* forEach():- Calls a function for each array element.
* indexOf():- Search the array for an element and returns its position.
* isArray():- Checks whether an object is an array.
* pop():- Removes last element of an array and returns that element.
* push():- Adds new element to the end of an array and returns the new length.
* reverse():- Reverses the order of elements in an array.
* sort():- Sorts the element of an array.