

UNIT - III

Java Script

Write about importance of Java Script, with benefits (or) How to add JavaScript in HTML (insert)

- JavaScript is an interpreter language
- It is embedded within the HTML.
- It can be written in any text editor like notepad and save with .html
- Javascript can read and write HTML elements.
- It can perform events like keyboard events and mouse events.
- There is no compilation steps and it is completely interpreted by the browser.

Rules of JavaScript:-

- Javascript contain variables, objects and functions
- Each line of code is terminated by semicolon (;)
- It is a case-sensitive
- Variables are declared by using var statement.
- It supports block of statements and it is represented as curly brackets " { } ".
- It doesnot have main function.

SM Inserting Java Script into HTML

- We can insert JavaScript into HTML using `<script>` tag.
- The `<script>` tag contains various properties such as:
 - * language
 - * type
 - * src
- JavaScript can be represented in various locations such as:
 - * Javascript in head section
 - * Javascript in body section
 - * JavaScript in both head & body
 - * JavaScript in external file
- The general structure / form of Javascript is represented as follows:

```
<html>
  <body>
    <script language="javascript">
      - - -
      - - -
      - - -
    </script>
  </body>
</html>
```

Example:-

```
<html>
<body>
<script language="javascript">
var a,b,c;
a=10;
b=20;
c=a+b;
document.write(c);
</script>
</body>
</html>
```

Q. Write about Operators in JavaScript.

To perform an operation between any two operands is called an operator.

In Javascript, it contains various types of operators. For example they are:

- * Arithmetical Operators
- * Comparison Operators
- * Logical Operators
- * Assignment and Compound Assignment Operators
- * Conditional Operators

1. Arithmetical Operators:-

It is used to perform all arithmetical operations. It can produce one numeric value i.e.,

$$a=4$$

$$b=2$$

Operator	Meaning	Example	Result.
+	Addition	$a+b$	6
-	Subtraction	$a-b$	2
*	Multiply	$a*b$	8
/	Divide	a/b	2
%	Remainder	$a \% b$	0
++	Increment by 1	$a++$	5
--	Decrement by 1	$a--$	3

d. Comparison Operator:-

It is also called as Relational Operators
It returns either true or false. i.e,

$$a=4 \quad b=2$$

Operator	Meaning	Example	Result
<	Less than ($4 < 10$)	$a < b$	false
>	Greater than	$a > b$	true
\leq	Less than or equals	$a \leq b$	false
\geq	Greater than or equals	$a \geq b$	true
$= =$	equals	$a == b$	false
\neq	not equals	$a != b$	true

3. (d) Logical Operators:-

These are used to combine two or more relational expressions. It returns either true or false. i.e,

logical and (&) :-
It returns true, if all the conditions returns true.

logical or (||) :-
It returns true, if any one of the condition returns true.

logical not (!) :-
It returns opposite status of given condition.

4. Assignment and Compound Assignment Operators:

These are used to assign the value or expressions. i.e,

$$a=4, b=2$$

Operator	Meaning	Example	Result.
=	assignment	$a=4$	4
+=	addition with right operand & assign to left operand.	$a+b$	6
-=	Subtraction with right operand & assign to left operand	$a-b$	2

$* =$	Multiply with right operand & assign to left operand	$a * b$	8
$/ =$	divide with right operand & assign to left operand	a / b	9

5. Conditional Operator:

Depends on the condition, the statement is executed. It contains ? and :

i.e.,

```
a > b ? document.write("a is big") : document.write ("b is big");
```

3. Write about Control statements

→ The control statements are divided into 3 types. They are:-

- * Conditional Statements
- * Unconditional Statements
- * Looping / Iteration Statements

→ It is represented as follows.

Control Statements

Conditional Statements

- if statement
- switch statement

Looping / Iteration Statements

Entry control loop

- for
- while

Exit control loop

- do-while

Unconditional Statements

- break
- continue

Conditional Statements:-

- It contain if and switch statement.
- (If) if is a two-way statement and switch is a multi-way branching statement.

if Statement:-

- There are four types of if statements.
They are:

- * Simple if
- * Nested if
- * if-else
- * else-if ladder

Simple if:-

In this functionality the statements are executed only if the condition is true. i.e,

Syntax:-

```
if (condition)  
    Statements;
```

Ex:-

```
if (a==0)  
document.write ("zero");
```

Nested if

- It is also called as Multiple if st
- In this functionality one if st contain another if statement. i.e.,

Syntax:

```
if (cond1)  
if (cond2)  
:  
if (condn)  
statements;
```

Ex:-

```
if (a==0)  
if (b==0)  
document.write ("zero");
```

if - else:

- It contain both true block and false block.
- True block is called as if block and false block is called as else block. i.e.,

Syntax:-

```
if (cond)
    statements;
```

```
else
    statements;
```

Example:-

```
if (a == 0)
    document.write ("zero");
```

```
else
    document.write ("not zero");
```

else-if ladder:-

→ In this functionality, an else block also contains another if-statement.

Syntax:-

```
if (cond)
    statements;
```

```
else if (cond)
    statements;
```

Example:-

```
if (a == 0)
```

```
    document.write ("zero");
```

```
else
```

```
if (a == 1)
```

```
    document.write ("one");
```

Switch Statement:-

- It is a multi-way branching statement
- It depends on the value of the variable the corresponding case statement is executed.
- If any one of the case statement is not matching then default will be executed. i.e,

Syntax:-

switch (var)

{

case val1: statements; break;

case val2: statements; break;

: : : :

case valn: statements; break;

default: statements;

}

Example:-

switch (ch)

{

case 1: c=a+b; break;

case 2: c=a-b; break;

case 3: c=a*b; break;

case 4: c=a/b; break;

default: document.write ("invalid")

}

Looping statements:-

- The statements are repeated upto finite number of times is called looping statements.
- It contain entry controlled loop and exit controlled loop.

Entry Control loop

- In this functionality first it checks the condition after that it enters into the block and execute the statements upto finite number of times.
- It contain for and while statements.

Syntax:- For loop

```
for (init; cond; incr/decr)  
statements;
```

Example:-

```
for (i=1; i<=5; i++)  
document.write(i);
```

Syntax:- While loop

```
=> init;  
while (cond)  
{  
statements;  
incr/decr;  
}
```

Example:-

```
i=1;
while (i<=5)
```

```
{  
    document.write(i);
    i++;
}
```

Exit Control loop:-

→ In this functionality first it will enter into the block and execute the statements after that check the condition

→ So the statements are executed once
→ It contains do-while statement

Syntax:-

```
init;
do
{
    Statements;
    incr/decr;
} while (cond);
```

Example:-

```
i=1;
do
{
    document.write(i);
    i++;
} while (i<=5);
```

Unconditional Statement:-

- It contains break and continue statements.

(Syntax:-)

break;

- It terminates from either block or loop.

continue:-

It performs the next iteration.

4. Write about Mathematical functions in JavaScript
(or)

Write about Numerical functions in Javascript

Ans:- The mathematical functions are used to perform mathematical operations. for example it is represented as follows:

- sqrt()
- pow()
- sin()
- cos()
- tan()
- log()

Sqrt:-

The functionality of sqrt() is used to display the square root of given number. i.e,

math.sqrt(25)

pow:- It is used to display the m^n value i.e,

math.pow(2,3)

sin():- It is used to display the sin value of i.e., `math.sin(30)`

cos():- Display the cosine value i.e,
`math.cos(90)`

tan():- Display the tangent value i.e,
`math.tan(45)`

log():- Display the logarithm value i.e,
`math.log(8)`

5. Write about string and its functions?

In general, string is defined as collection of characters or combination of characters.

Javascript support various functions to operate on strings. For example they are:

```
var s = new String ("Webtech");
        toLowerCase()
        toUpperCase()
        concat()
        charAt()
        substr()
        toString()
```

1. toLowerCase():- It is used to convert the string into lowercase. i.e., `s.toLowerCase()`

2. toUpperCase():- It is used to convert the string into uppercase. i.e., `s.toUpperCase()`

3. concat():- It is used to combine the strings i.e., `s.concat(t)`

4. charAt():- It is used to display the character at given specific position.

i.e., `s.charAt(3)`

5. substr():- It is used to display the substrings i.e., s.substr(1, 3)
6. replace():- It is used to replace the given string if it is found.
7. toString():- It is used to convert the data into string i.e., s.toString()

6. Write about datatypes in JavaScript.

In general, data type represent the type of the data. In JavaScript, the datatypes are divided into 4 types. They are:

1. Numbers
2. String
3. Boolean
4. Null.

Number:- It contain both integer and floating point numbers. for example the numbers are represented by 25, 3.14, 55 etc

String:- String is defined as collection of characters or set of characters or combination of characters or group of characters. In JavaScript the string is enclosed in between ' ', " ". for example, it is denoted by

'Webtech'; "Webtech"

Boolean:- In Javascript, the boolean variables contain either true or false.

Null:- It is a specific type of data in the Javascript. It is not referred to zero. In general, it is represented by empty.

Variable:- The value which can be changed during the program execution is called Variable. It is an identifier.

Rules:

- The first letter must be an alphabet.
- Operators are not allowed except underscore.
- Blank spaces are not allowed.
- Keywords are not allowed.

In general, it can be created by Var keyword. i.e,

Var a = 10;

Var s = "Webtech";

7. Define array and write about its functions in JavaScript.

It is defined as collection of similar data and have a common name. The starting index of array is zero and ending with size-1.

There are two ways to create an array in the JavaScript. They are:

Method -1 :-

Var array-name = [list of values];

Var a = [10, 20, 30, 40];

Method -2 :-

Var array-name = new Array (size);

Var a = new Array (10);

Accessing array elements:

Array elements are accessed through index. It is starting from zero and ending with size-1.

The length property is used to represent the number of elements in an array.

functions in an array:-

There are different types of functions to support in an array. For example, they are

1. push():- It is used to insert an element into an array.
2. pop():- It is used to delete an element from an array.
3. sort():- It is used to arrange the elements in an order.
4. concat():- It is called concatenation. It is used to combine the arrays.
5. reverse():- It is used to perform swapping i.e., the last position value is swapped into first position value and vice-versa.

Example Program:-

```
<html>
<body>
<script language = "Javascript">
    var a = [10, 20, 30, 40, 50];
    var i;
    for (i=0; i<5; i++)
        document.write(a[i]);
</script>
</body>
</html>
```

Output:-

10

20

30

40

50

8. Write about functions in JavaScript.

A function is a self contained block of statements and it performs a specific task. It is a sub-program. In generally, the functions are divided into two types. they are:

1. Pre-defined Functions
2. User-defined Functions.

The predefined functions are defined by the developers. for example, they are `alert()`, `confirm()`, `prompt()` and so on. Whereas user defined functions are defined by the user.

Advantages:-

- clarity of a program
- Reduce the redundancy
- User can develop their own library functions.

Steps to create User-defined functions:

- An user defined function can be defined using `function` keyword.
- The rules of a function name is same as Variable rules.
- The function may contain parameters.
- The return statement is used to return a value from the function.
- For example, the definition of a function is represented as follows
`function functionname (args)`

----- Statements;

Example Program Using Functions:-

```
<html>
<body>
<script language = "JavaScript">
    big(10, 20);
    function big(a,b)
    {
        if (a>b)
            document.write ("a is big");
        else
            document.write ("b is big");
    }
</script>
</body>
</html>
```

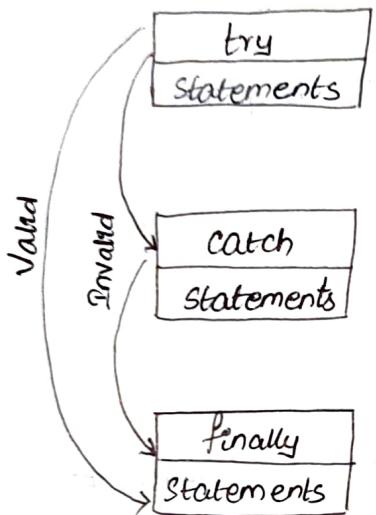
9. Write about exception handling in JavaScript

(Or)

How to handle runtime errors in JavaScript.

- Ans: → An exception is an object in Object oriented programming language.
- It is created dynamically in the program.
- The purpose of exception handling is used to handle the runtime errors.
- These errors are occurred due to invalid data.
- In general, an exception handling contain three blocks. They are
 - * try block
 - * catch block
 - * finally block.

- The try block represents where the exception is occurred, catch block represents the handle the exception, and finally block executes if the exception is occurred or not.
- The throw statement is used to create an user-defined exception.
- The representation of exception handling is as follows:



Example:-

```

<html>
<body>
<script language="Javascript">
try
{
    var rno;
    rno = parseInt(prompt("enter rno"));
    if(rno <= 0)
        throw "Invalid"
    else
        document.write("Valid");
}
catch(exception)
{
    document.write(exception);
}
</script>
</body>
</html>
  
```

Output:

enter rno 5

valid

enter rno 0

Invalid.

10. Write about regular expression in Javascript.

A regular expression is an object. It is used to perform pattern-matching and search and replace functionality. The following are the examples of declaring regular expression.

Method-1:-

```
var s1 = "is";
var s2 = "WT is Webtech";
s2.match(s1);
```

Method-2:-

```
var s1 = new RegExp("is");
var s2 = "WT is Webtech";
s1.exec(s2);
```

In those methods, the method-1 is represented as static whereas method-2 is represented as dynamic type of regular expression.

The general form of regular expression is represented as follows:

```
var x = new RegExp(pattern, modifier);
```

where,

x represents regular expression

pattern represents a string

modify represents the symbolic template

for example, the symbolic templates are represented as follows:

Symbolic template	Meaning
----------------------	---------

?

Match 0 or 1 time

+

≥ 1 time

i|j

Match i or j

1d

Match a single digit

1w

Match alphanumeric character or '-'

2u3

underscore

Match the given string x times

Method supported by RegExp

-test():-

It contain an argument as string and find the given string is exists or not.

-exec():-

It contain an argument as string and find the given string exist or not. It returns an array as result.

Method supported by String:-

-search():-

It contain an argument as string and finds the given string is exists or not.

-split():-

It contain an argument as string and split the given string as user requirement.

-replace():-

It contain two strings as an argument. If the method executes successfully then string1 is replaced by string2.

11. Write about data and objects in Javascript.

Java is fully object oriented programming language. Javascript is a markup language with object-oriented features. The creation of object is represented by using new statement i.e,

```
var object = new classname();
```

Each object contain both data and code. These are accessing through member operator (.). i.e,

```
object.membervariable;  
object.methodname();
```

Window object:-

It is one of the object in Javascript. It contain different properties and methods. for example, it is represented as follows:

Alert():-

This method is used to display the message with 'OK' button i.e,

```
Alert("Javascript");
```

confirm():-

It is used to display the message with 'OK' and 'cancel' button i.e,

```
confirm("Javascript");
```

Prompt():-

It is used to display the message with 'OK' and 'cancel' buttons along with 'textbox' i.e,

```
prompt("Javascript");
```

The window object also contain various properties like height, width, status and so on.

Document object:-

It is also one of the object in Javascript. It also contain various method and properties.

Example:-

```
document.write ("Javascript");
```

In this example, document is an object and write is a method.

Q8. Write about built in objects in Javascript?

Ans:-

Javascript contain a set of built-in objects that can provide information for the objects.

- Each object contains various properties and methods (functions).
- It contains hierarchy for the objects.
- the navigator object hierarchy is built from a single base object is called window object.
- for example, built in objects are represented as follows:

1. document object
2. window object
3. form object
4. date object
5. Browser object and so on.

1. Document object:-

- A document is a webpage
- It represents the entire page on displaying the (Webbrowser) Web browser
- It contains no. of properties that can be accessed by JavaScript Program.
- for example, the properties and method is represented as follows:

1. BG color
2. FG color
3. image
4. Title
5. write
6. written
7. Open
8. Close

2. Window object:-

- It is the top level object for each document.
- It contains location and history object.
- It contains various properties and methods for representing Javascript program.
- For example, they are:
 1. Name
 2. Status
 3. History
 4. Location
 5. Alert
 6. confirm
 7. prompt and so on.

3. form object:-

- It is a container for all GUI objects.
- It also acts as the mechanism for Grouping different components.
- It is denoted by <form> tag.
- It contains various properties and methods.
- For example, they are:
 1. Name
 2. Method
 3. Action
 4. On_click
 5. OnSubmit
 6. Onreset

4. Date Object:-

- It is also one of the objects in Javascript.
- It will display date along with time format.
- It contains universal time system and local time system.
- It contains various methods and properties.

→ for example, it is represented as follows:

1. getDate
2. getDay
3. getMonth

4. getYear
5. setDate
6. setMonth and so on

6. Browser Object:-

- It is also one of the object in Javascript
- It is actually called as navigator object.
- It also contains various properties and methods
- for example, it is represented.

1. ~~getDate()~~
2. ~~getDay()~~
3. ~~getMonth()~~

4. ~~getYear()~~
5. ~~setDate()~~
6. ~~setMonth()~~ and so on.

1. Language
2. Platform
3. App name
4. App version and so on

```
<html>
<body>
<script language = "Javascript">
document.write ("Javascript");
document.write ("html");
document.write ("MCCS");

</script>
</body>
</html>
```

