Client Side Scripting

CO1

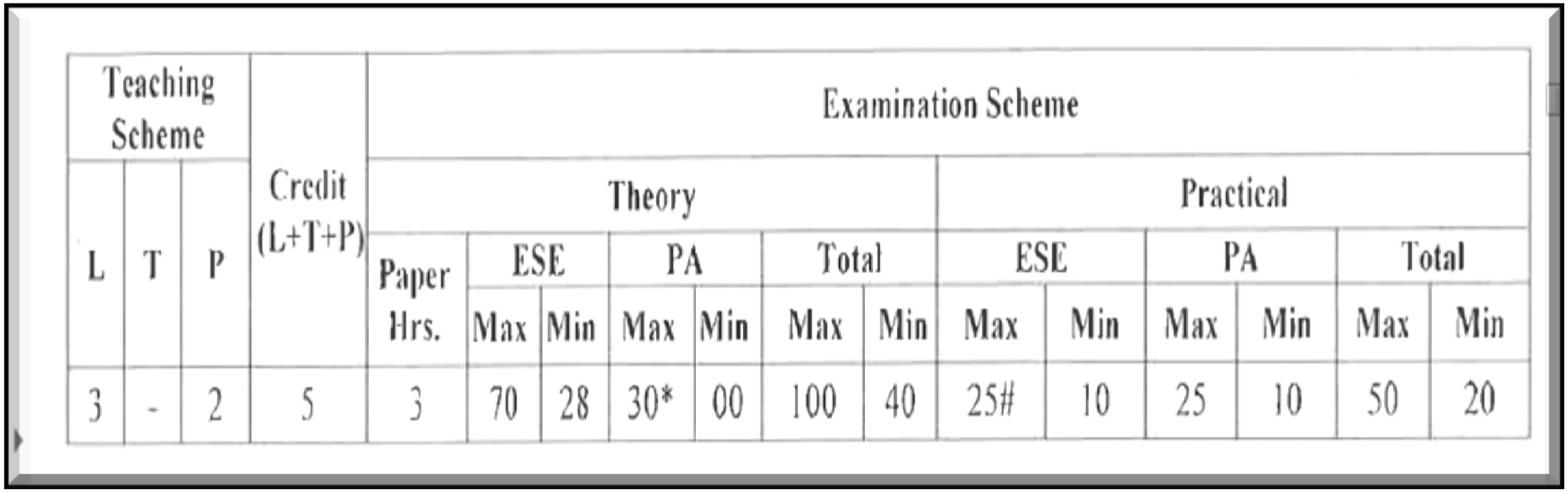
- **Create interactive web pages using program flow control structure**

LO1

LO5

- Create Object to solve the given problem
- Develop program using basic features of JavaScript to solve the given problem

TEACHING AND EXAMINATION SCHEME



Teaching Scheme			Credit (L+T+P)	Examination Scheme												
L	T	P		Theory							Practical					
				Paper Hrs.	ESE		PA		Total		ESE		PA		Total	
					Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min
3	-	2	5	3	70	28	30*	00	100	40	25#	10	25	10	50	20

What is JavaScript Programming?

What are the features of JavaScript?

CONTENTS

1. Features of JavaScript
2. Object Name, Property , Method ,Dot syntax, Main event
3. Values and Variables
4. Operators and Expressions

JAVASCRIPT

1.Concept of JavaScript

2. Features of JavaScript

3. Limitations of JavaScript

4. Advantages of JavaScript

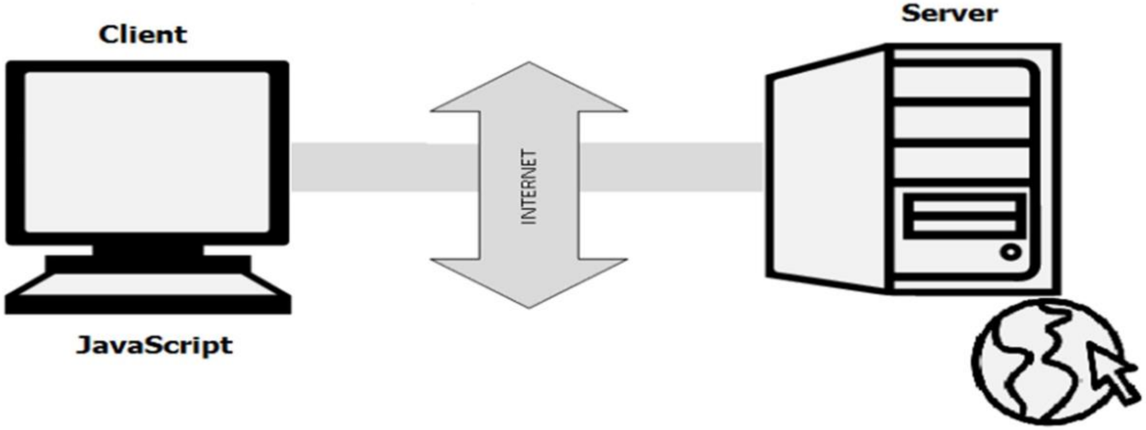
5. Simple program in JavaScript

6. Object Name, Property , Method ,Dot syntax, Main event

7. Operators and Expressions

JAVASCRIPT

- JavaScript is an open source & most popular client side scripting language supported by all browsers.
- JavaScript is a very powerful client-side scripting language.
- JavaScript is used mainly for enhancing the interaction of a user with the webpage.
- You can make your webpage more lively and interactive, with the help of JavaScript.



FEATURES OF JAVASCRIPT

1. JavaScript is an object-based scripting language.
2. Giving the user more control over the browser.
3. It Handles dates and time.
4. It detects the user's browser and OS.
5. It is light weighted.
6. JavaScript is interpreter based scripting language.
7. JavaScript is case sensitive.
8. JavaScript is object based language as it provides predefined objects.
9. Every statement in JavaScript must be terminated with semicolon (;).
10. Most of the JavaScript control statements syntax is same as syntax of control statements in C language.

LIMITATIONS OF JAVASCRIPT

1. Client-side JavaScript does not allow the reading or writing of files. This has been kept for security reason.

2. JavaScript cannot be used for networking applications because there is no such support available.
3. JavaScript doesn't have any multi-threading or multiprocessor capabilities.

ADVANTAGES OF JAVASCRIPT

1. Less server interaction − you can validate user input before sending the web page to the server. This saves server traffic, which means less loads on your server.
2. Immediate feedback to the visitors − they don't have to wait for a page reload to see if they have forgotten to enter something.
3. Increased interactivity − you can create interfaces that more interactive.
4. Richer interfaces − you can use JavaScript to include such items as drag-and-drop components and sliders to give a Rich Interface to your site visitors.

JAVASCRIPT TAG

- JavaScript can be implemented using JavaScript statements that are placed within the <script>...

  </script> HTML tags in a web page.
- You can place the <script> tags, containing your JavaScript, anywhere within your web page, but it is normally recommended that you should keep it within the <head> tags.
- The <script> tag alerts the browser program to start interpreting all the text between these tags as a script.

A simple syntax of your JavaScript will appear as follows.

```
<script ...>
 JavaScript code
   </script>
```

JAVASCRIPT CODE

<script> tag has following attributes :

1.Language – This attribute specifies what scripting language you are using. Typically, its value will be JavaScript

2.Type – this attribute is what is now recommended to indicate the scripting language in use and its value should be set to "text/javascript".

Example

```
 <script  language  =  "javascript"  type  =  "text/javascript">
JavaScript  code

 </script>
```

JavaScript Code:

```
<html>
<body>
<script  language  =  "javascript"  type  =  "text/javascript">
document.write("Hello  World!")
</script>
</body>
</html>
```

OBJECT NAME, PROPERTY , METHOD ,DOT SYNTAX, MAIN EVENT

Object Name :

- A JavaScript object is a collection of named values and these named values are properties of object .
- Each object should be uniquely identified by a name or IDs in web page to distinguished between them.
- JavaScript supports various objects like document , form ,button , window etc
- In JavaScript , object can be created with curly brackets { } with an optional list of properties.

- **Example:**

```
  <script language="Javascript" type="text/javascript">
     var student= {
     name:"vijay",
     age:21,
      year:"TY"
       };
    </script>
```

PROPERTY

- A property is a value associated with an object.
- Every object has its own set of properties for example window has properties like width, height, background color.
- A property is a “key: value” pair,

  1. key (property name) is always a string
  2. value (property value) can be any data type like strings, numbers, Booleans

Program: Write a Javascript code to declare property

```
<html>
<head>
<title>object property</title>
<script language=“Javascript” type =“text/javascript”>
Var student ={
name:”vijay”,
age:21,
year:”TY”
};
document.write(student.name);
</script>
</head>
<body>
</body>        </html>
```

METHOD

- A method is a set of instructions performed by an object when it receives a message.
- On the form when you click a submit button, the form is submitted to the server-side application. It means, clicking the submit button causes the button to process a some set of instructions (method)

Program: Write a Javascript code to illustrate the use of method

```
<html>
<head>
<title>Method</title>
<script language="Javascript" type ="text/javascript">
Var student ={
name:"vijay",
age:21,
year:"TY",
display: function()   {
return(this.name);   }              };
document.write("name=" +student.display());
</script>                </head>
<body>                   </body>          </html>
```

DOT SYNTAX AND MAIN EVENT

Dot syntax:

- One can access an object’s properties and methods by using the dot syntax along with the object name and its property and method.
- For example, the write method for a document: document. write()
- The first part is the name of the object (document) and second part is either a property or method (write) of the object.

Main event:

- In Javascript, an event is the way to start executing your code like on mouse click, button click etc.
- Javascript reacts to event with the help of event handling.
- Event handling is the execution of code on occurrence of event. For example- when click on submit button then the event handler for that click event should process the information.
- The function written for event handling are known as event handler.

VALUES AND VARIABLES

- Javascript is a dynamic language because a variable can hold value of any data type at any point of time.
- Var keyword is used to specify the data type. It can hold any type of values such as numbers, strings etc.

Following are the values used by Javascript

1. Number- A number is a numeric value. for example: var a=20;
2. String- A string a sequence of characters that is enclosed within quotation marks. For example: var city="pune";
3. Boolean- A Boolean is a value-either false or true. for example: var b=true;
4. Null- Null value means no value at all. For example: var i=null;
5. Objects- An object is a instance through which we can access members.

   for example: var person={first name : "john", last name :"doe"};

OPERATORS

- Operators are the symbols which indicate operations to be performed.
- Operands are the variable on which operation to be performed.
- The systematic arrangement of operators and operands is known as Expression.

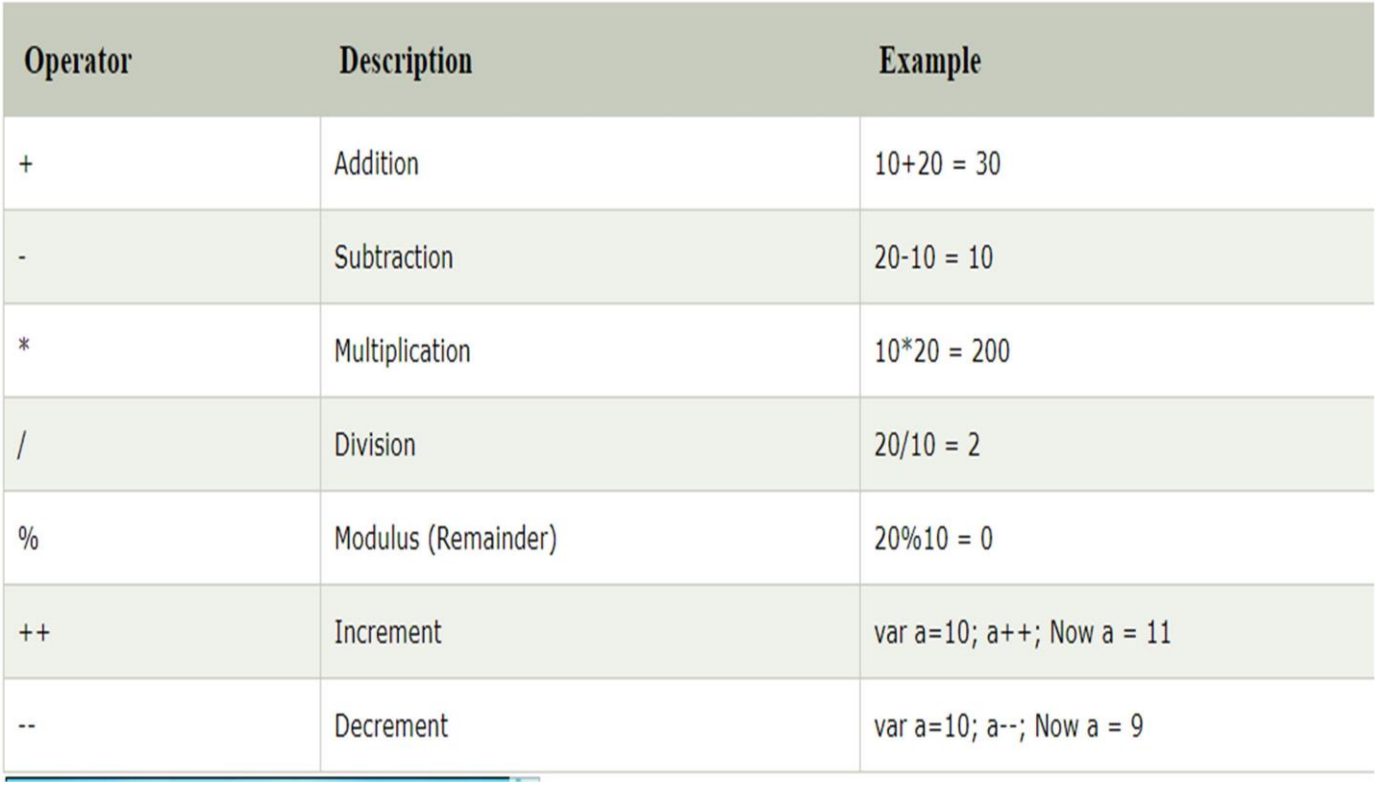
There are following types of operators in JavaScript.

1. Arithmetic Operators
2. Comparison (Relational) Operators
3. Bitwise Operators
4. Logical Operators
5. Assignment Operators
6. Conditional Operators

ARITHMETIC OPERATORS

- Arithmetic operators are used to perform arithmetic operations on the operands.

The following operators are known as JavaScript arithmetic operators.



Operator	Description	Example
+	Addition	10+20 = 30
-	Subtraction	20-10 = 10
*	Multiplication	10*20 = 200
/	Division	20/10 = 2
%	Modulus (Remainder)	20%10 = 0
++	Increment	var a=10; a++; Now a = 11
--	Decrement	var a=10; a--; Now a = 9

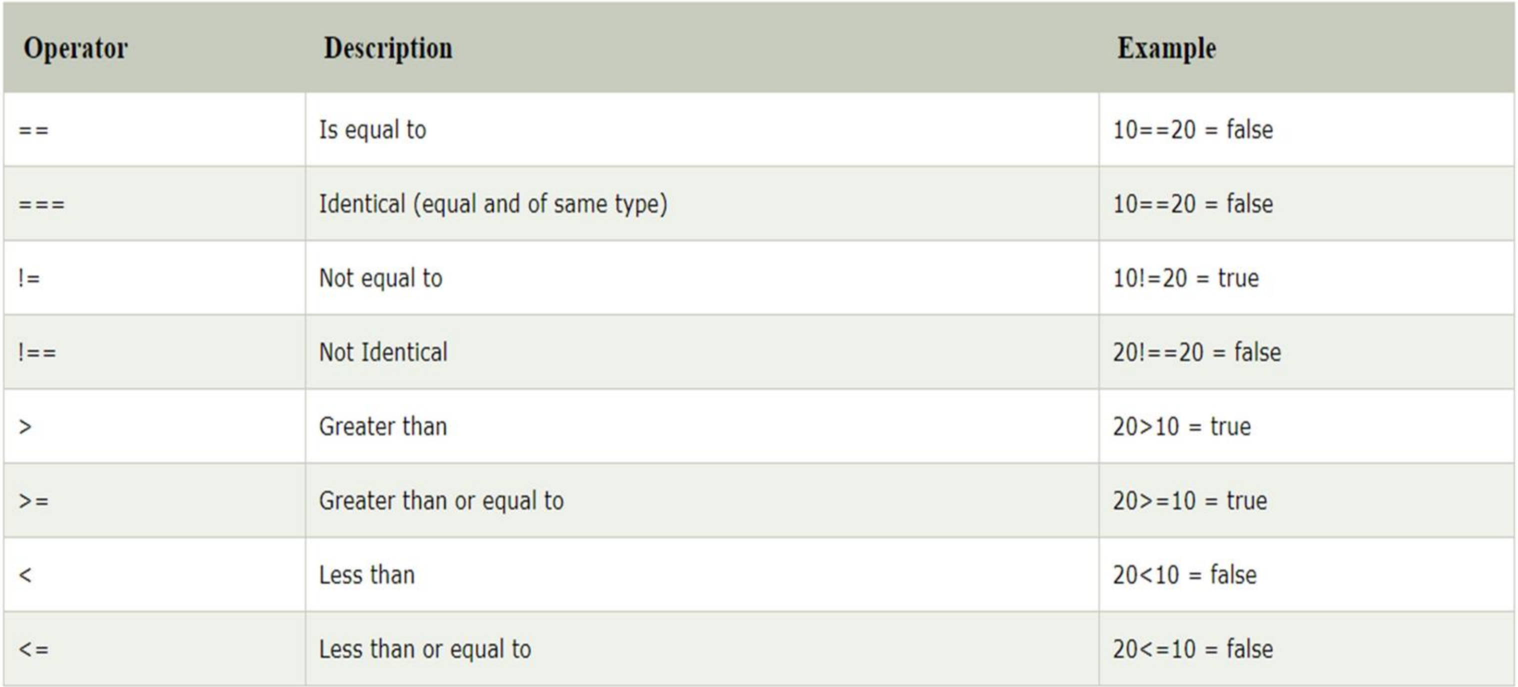
ARITHMETIC OPERATORS- PROGRAM

```
<html>
    <body>
        <script type = "text/javascript">
                var a = 33;
                var b = 10;
                var c = "Test";
                var linebreak = "<br>";

                document.write("a + b = "); result = a + b; document.write(result);
                document.write(linebreak);

                document.write("a - b = "); result = a - b; document.write(result);
                document.write(linebreak);

                document.write("a / b = "); result = a / b; document.write(result);
                document.write(linebreak);

                document.write("a %  b = "); result = a %  b; document.write(result);
                document.write(linebreak);

                document.write("a + b + c = "); result = a + b + c; document.write(result);
                document.write(linebreak);

                a = ++a; document.write("++a = "); result = ++a; document.write(result);
                document.write(linebreak);

                b = --b;
                document.write("--b = "); result = --b; document.write(result); document.write(linebreak);
                </script>
                     </body>
                          </html>
```

COMPARISON OPERATORS

- The JavaScript comparison operator compares the two operands.

The comparison operators are as follows:



Operator	Description	Example
==	Is equal to	10==20 = false
===	Identical (equal and of same type)	10==20 = false
!=	Not equal to	10!=20 = true
!==	Not Identical	20!==20 = false
>	Greater than	20>10 = true
>=	Greater than or equal to	20>=10 = true
<	Less than	20<10 = false
<=	Less than or equal to	20<=10 = false

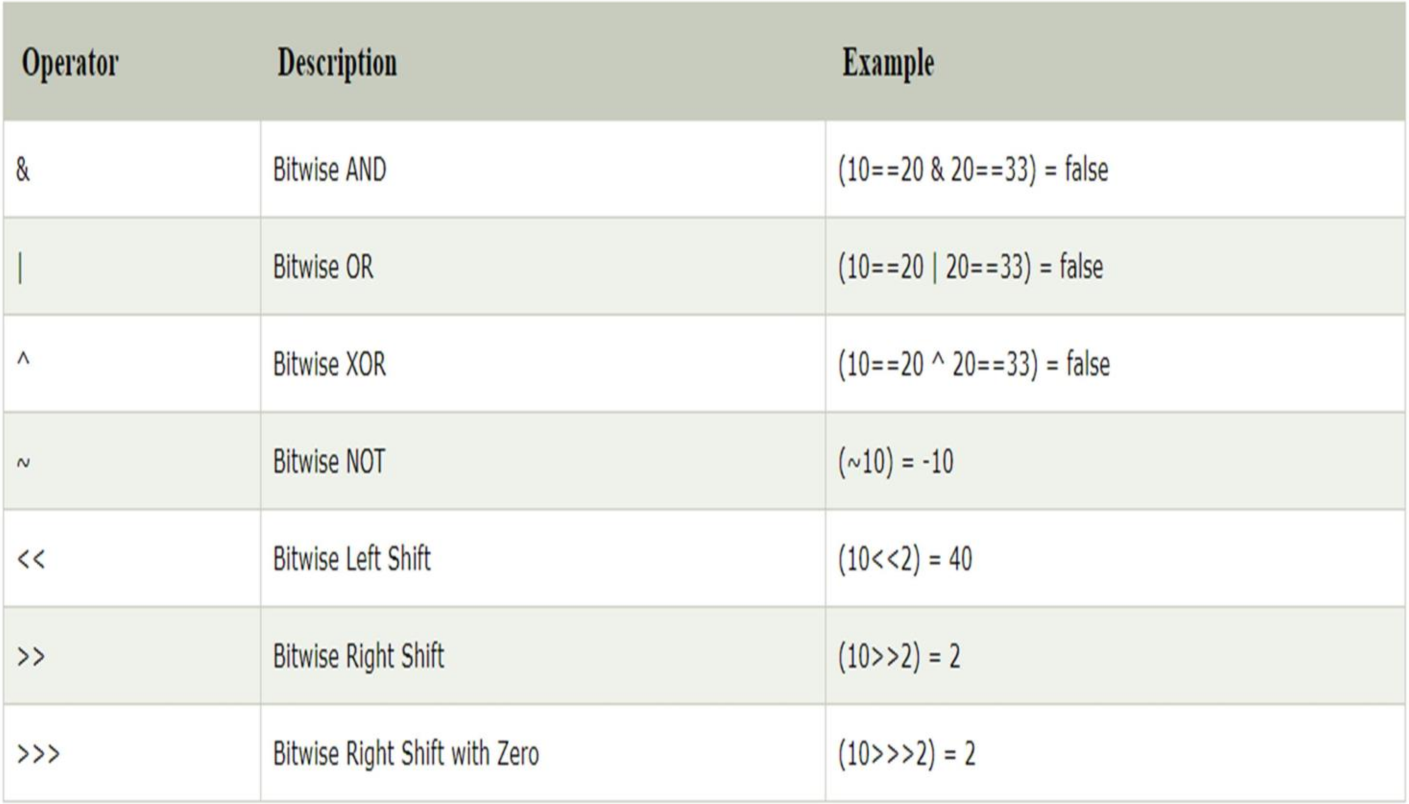
COMPARISON OPERATORS- PROGRAM

```
<html>
<body>
<script type = "text/javascript">
var a = 10;
var b = 20;
var linebreak = "<br>";

document.write("(a == b) => "); result = (a == b); document.write(result); document.write(linebreak);

document.write("(a < b) => "); result = (a < b); document.write(result); document.write(linebreak);

document.write("(a > b) => "); result = (a > b); document.write(result); document.write(linebreak);

document.write("(a != b) => "); result = (a != b); document.write(result); document.write(linebreak);

document.write("(a >= b) => "); result = (a >= b); document.write(result); document.write(linebreak);

document.write("(a <= b) => "); result = (a <= b); document.write(result); document.write(linebreak);
</script>
</body>
</html>
```

BITWISE OPERATORS

- The bitwise operators perform bitwise operations on operands.

The bitwise operators are as follows:



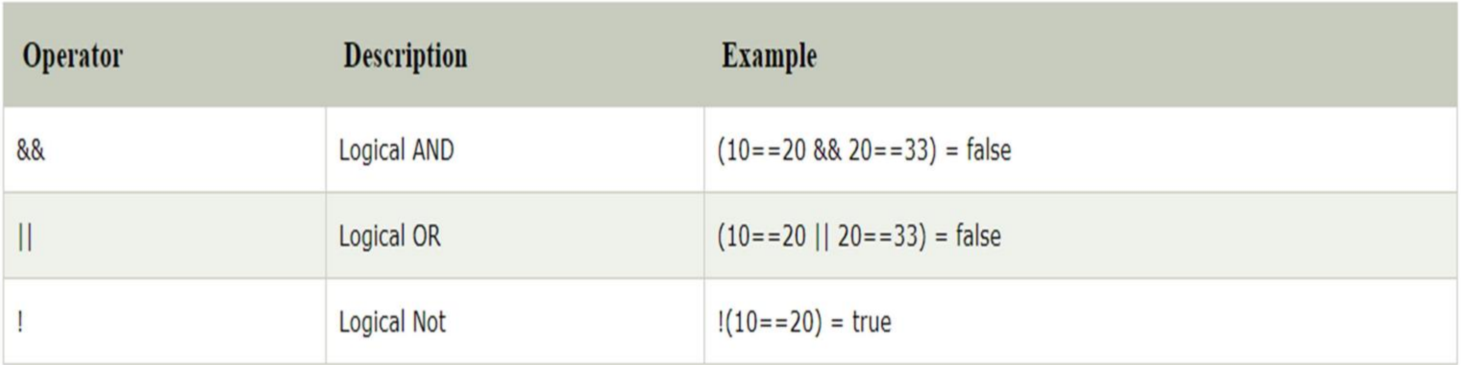
Operator	Description	Example
&	Bitwise AND	(10==20 & 20==33) = false
|	Bitwise OR	(10==20 | 20==33) = false
^	Bitwise XOR	(10==20 ^ 20==33) = false
~	Bitwise NOT	(~10) = -10
<<	Bitwise Left Shift	(10<<2) = 40
>>	Bitwise Right Shift	(10>>2) = 2
>>>	Bitwise Right Shift with Zero	(10>>>2) = 2

BITWISE OPERATORS: PROGRAM

```
<html>
<body>
<script type = "text/javascript">
var a = 2;
var b = 3;
var linebreak = "<br>";

document.write("(a & b) => "); result = (a & b); document.write(result); document.write(linebreak);

document.write("(a | b) => "); result = (a | b); document.write(result); document.write(linebreak);

document.write("(a ^ b) => "); result = (a ^ b); document.write(result); document.write(linebreak);

document.write("(~b) => "); result = (~b); document.write(result); document.write(linebreak);

document.write("(a << b) => "); result = (a << b); document.write(result); document.write(linebreak);

document.write("(a >> b) => "); result = (a >> b); document.write(result); document.write(linebreak);
</script>
</body>
</html>
```

LOGICAL OPERATORS

The following operators are known as JavaScript logical operators



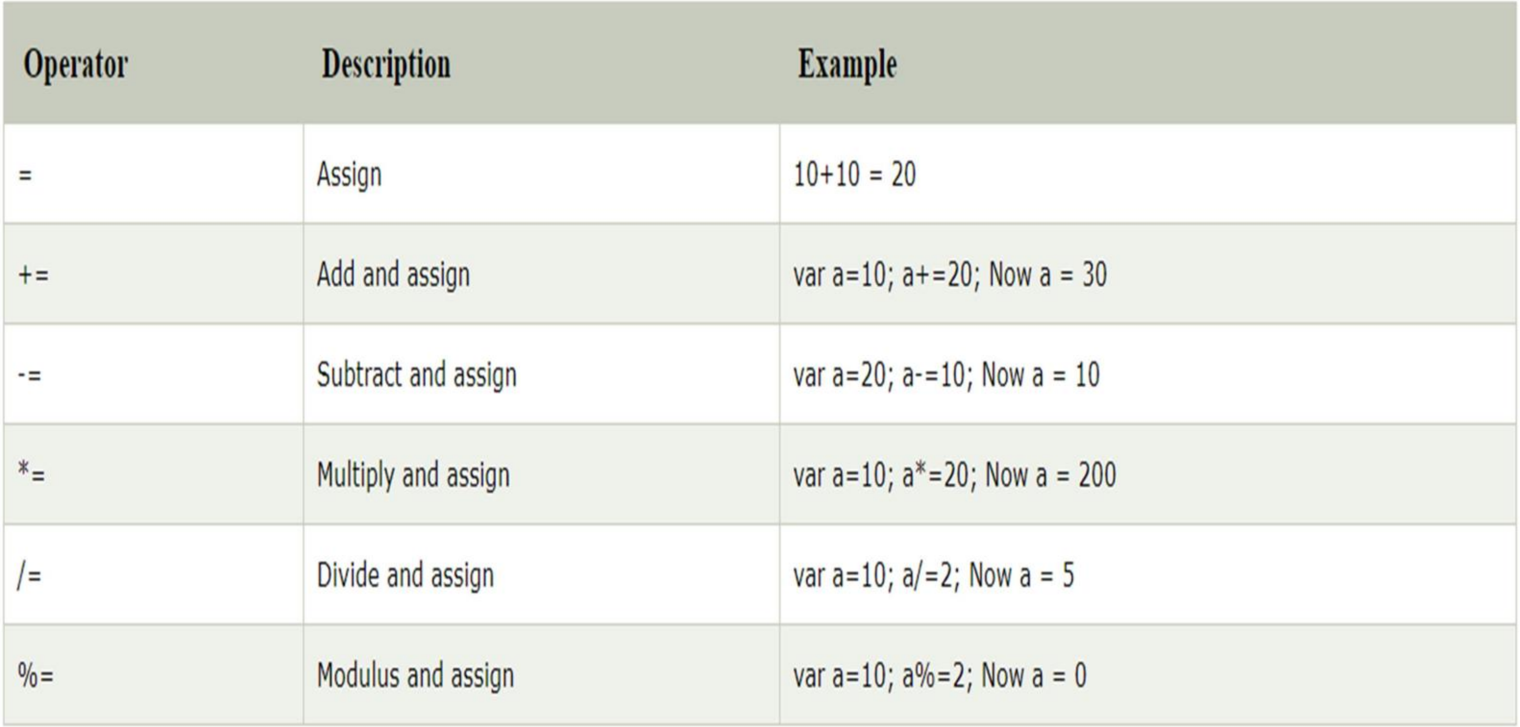
Operator	Description	Example
&&	Logical AND	(10==20 && 20==33) = false
||	Logical OR	(10==20 || 20==33) = false
!	Logical Not	!(10==20) = true

PROGRAM:

```
<html>
<body>
<script type = "text/javascript">
var a = true;
var b = false;
var linebreak = "<br>";
document.write("(a && b) => "); result = (a && b); document.write(result); document.write(linebreak);

document.write("(a || b) => "); result = (a || b); document.write(result); document.write(linebreak);

document.write("!(a && b) => "); result = (!(a && b)); document.write(result); document.write(linebreak);
</script>
</body>
</html>
```

ASSIGNMENT OPERATORS

The following operators are known as JavaScript assignment operators.



Operator	Description	Example
=	Assign	10+10 = 20
+=	Add and assign	var a=10; a+=20; Now a = 30
-=	Subtract and assign	var a=20; a-=10; Now a = 10
*=	Multiply and assign	var a=10; a*=20; Now a = 200
/=	Divide and assign	var a=10; a/=2; Now a = 5
%=	Modulus and assign	var a=10; a%=2; Now a = 0

ASSIGNMENT OPERATORS: PROGRAM

```
<html>
<body>
<script type = "text/javascript">
var a = 33; var b = 10; var result;
var linebreak = "<br>";
document. Write("Value of a => (a = b) => ");
result = (a = b); document.write(result);
 document. Write(linebreak);
document. Write("Value of a => (a += b) => ");
result = (a += b); document.write(result); document. Write(linebreak);
document. Write("Value of a => (a -= b) => ");
result = (a -= b); document.write(result);
document. Write(linebreak);
document. Write("Value of a => (a *= b) => ");
```

```
result = (a *= b); document.write(result); document.write(linebreak);

document.write("Value of a => (a /= b) => ");
result = (a /= b); document.write(result); document.write(linebreak);

document.write("Value of a => (a %= b) => ");
result = (a %= b); document.write(result); document.write(linebreak);

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

CONDITIONAL OPERATOR

- The operator (?:) is known as conditional operator.
- It is also called as Ternary operator.
- Syntax: Condition? Expression-1 : Expression-2;
- In above syntax, if condition is true then it will execute Expression-1 otherwise Expression-2.

PROGRAM:

```
<html>
<body>
<script  type  =  "text/javascript">

var  a  =  100; var  b  =  20; var  result;

result  =  (a >  b)  ?  a  : b;
document.write("Greatest  Number="+result);

</script>

</body>
</html>
```

QUIZ TIME

Q1. Which JavaScript is also called client-side JavaScript?

a) Microsoft
b) Navigator
c) LiveWire
d) Native

- Ans: b. Navigator

Q2 . What should appear at the very end of your JavaScript?
The <script LANGUAGE="JavaScript">tag

A. The </script>
B. The <script>
C. The END statement
D. None of the above

- Ans: a. The </script>

QUIZ TIME

Q3. Which of the following best describes JavaScript?

A. a low-level programming language.
B. a scripting language precompiled in the browser.
C. a compiled scripting language.
D. an object-oriented scripting language.

- Ans. D. an object-oriented scripting language

- **Q4. What is the correct JavaScript syntax to write "Hello World"?**

A. System.out.println("Hello World")
B. println ("Hello World")
C. document. Write("Hello World")
D. response. Write("Hello World")

- Ans: C. document.write(“Hello World”)

Thank You