UNIT-4

CLIENT –SIDE SCRIPTING USING JAVA SCRIPT

4.1 DESCRIBE THE NEED FOR CLIENT-SIDE SCRIPTING LANGUAGE:

Scripting language

- A script refers to a small non-compiled program that is created by end-users.
- Scripts are written using different types of languages called scripting languages.
- It is a special type of programming language in which program written can be interpreted by another program at run-time.
- These languages are embedded with in HTML document so as to increase the functionality of web page.
- The various type of scripting languages arejavascript, vbscript, php, perl, bash etc.

Types of scripting languages: Scripting languages are two types. They are
1: Client-side scripting language,
2: Server-side scripting language.

Client-side scripting language

This type of scripting performs execution of programs/scripts written on client-side web browser. Client side scripts are written using languages like java script, css, vb script etc., and are embedded in HTML or XHTML documents. It mainly deals with user interface which enables user interaction on web.

Executing client-side script: The steps for executing client side scripts are given below:

- A script is written using scripting languages like java script on end user system.
- Upon receiving a request, web server transfers the requested file to user computer.
- Later these, files are executed by web browser on end user system.

The most commonly used client side scripting languages are Java script, CSS and VB script

Need for client side scripting language

1. It reduces the burden of the server to maximum extent.
2. It can be effectively used for validating the user. Hence save the number of interactions to the server.
3. It possess faster execution capabilities.
4. It enables user to play online games, quiz, online shopping and so on.

4.2 LIST VARIOUS CLIENT SIDE SCRIPTING LANGUAGES:

Various types of client-side scripting languages are as follows:-

1.Java script

- ✓ Java script is a programming language with direct support to object oriented methodologies.
- ✓ It facilitates inclusion of certain executable data along with it. Hence it can be said that, with the usage of java script, a given web page no longer remain a store house of static data but it can also maintain dynamic data which can interact with the users.
- ✓ It can dynamically create HTML context and also control the browser actions.

2.VBScript

- ✓ VBScript is an object based scripting language developed by Microsoft. It is modeled on visual basic as a lightweight/active language containing fast interpreters which can run on wide variety of Microsoft environments.
- ✓ It is mainly used as a client side script in HTML documents. This helps in increasing the functionality of web pages displayed on web browser.
- ✓ Internet explorer is the most commonly used web browser that contains VBScript interpreter to execute VBScript code.
- ✓ These scripting languages makes use of component object model (COM)in order to access the elements of run time environment in which it is executed.

3.Cascading Style Sheet(CSS)

- ✓ Cascading Style Sheets(CSS),forms one of the most interesting concepts of HTML as it specifies the rules for organizing elements of a given web documents.

- ✓ It not only extends its features in controlling colors and sizes of fonts ,but also controls spaces between various elements, the color and width of a given line etc.
- ✓ Thus, CSS are used to determine the style and layout of web pages. Generally, style defines the way in which the HTML elements are shown on the web pages.
- ✓ A style can be used in the following format, Selector (property. value) Cascading Style Sheets are of three types. they are,
  a.Inline style sheets   b.Embedded style sheets c.External style sheets

INTRODUCTION TO JAVA SCRIPT

You can insert javascript code in an HTML document by using the **scrip**t element. When an HTML document with the SCRIPT element is loaded in a web browser, the browser processes the content enclosed within the SCRIPT element as javascript code. The script element contains five attributes: async, type, charset, defer and src

Attributes and its values :

1. TYPE : text/javascript, text/vbscript,
2. SRC : URL

The SCRIPT element in a webpage can be declared in the following three ways

1. **In the HEAD element**
2. **In the BODY element**
3. **As an external script file**

1. JavaScript in the HEAD element

```
<HEAD>
<SCRIPT TYPE="TEXT/JAVASCRIPT">
 SCRIPT CODE HERE
</SCRIPT>
</HEAD>
```

2. JavaScript in the BODY element

```
<BODY >
<SCRIPT TYPE="TEXT/JAVASCRIPT">
 SCRIPT CODE HERE
</SCRIPT>
</BODY >
```

3. JavaScript in an EXTERNAL FILE

```
<HEAD>
<SCRIPT SRC="URL of the external file">
 SCRIPT CODE HERE
</SCRIPT>
</HEAD>
```

Features of java script

- ➢ **Imperative and structured:**Implies that javascript supports all the structured programming language C, such as the if statement, loops, and the switch statement. The only syntactical difference between C and javascript is that, in javascript semicolon is not necessary to terminate a statement, whereas in c, semicolon is necessary to terminate a statement.
- ➢ **Functional:** implies that javascript does not support the classes. Instead of using classes, objects are created from the constructor functions. Each constructor function represents a unique object type
- ➢ **Platform-independent:**implies that javascript supports platform-independency or portability. This means that you can write script once and run it anywhere at any time. In other words, you can write your javascript application and run it on any platform or any browser without affecting the output of the script

EXPLORING VARIABLES

Data can be temporarily stored in variables, which are the named locations in the memory. A variable has a name, value, and memory address

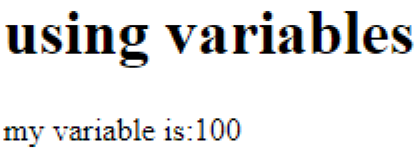
Variable declaration Syntax: Var variable-name;

The syntax to assign value to a variable at the time of declaration is as follows
varvariable_name=value;

Program:

```
<html>
<head>
<title> using variable </title>
</head>
<body>
<h1> using variables </h1>
<script type="text/javascript">
var myvar1=100;
document. write("my variable is:"+myvar1);
</script>
</body>
</html>
```

Output:



```
using variables

my variable is:100
```

4.3 USE OF VARIOUS OPERATORS

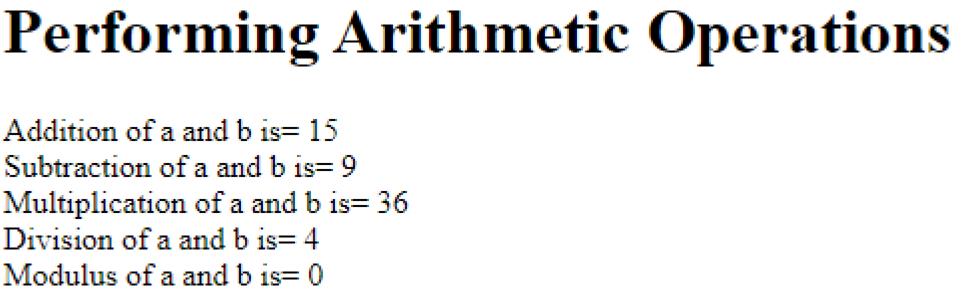
An operator is an symbol are word that is reserved for special task or action

a)Arithmetic operators:

operator	Description
+	Adds two numbers
-	Subtract two numbers
*	Multiplies two numbers
/	Divides two numbers
%	Divides two numbers and returns the reminder
++	Increments the value of a number by 1
--	Decrements the value number by 1

Sample program:

```
!DOCTYPE html>
<html>
<head>
   <title>Javascript Arithmetic
Operators</title>
</head>
<body>
    <h1>Performing Arithmetic
Operations </h1>
<script type="text/javascript>
   var a = 12, b = 3;
   document.write("Addition of a and b is= " + (a+b) + "<br/>");
   document.write("Subtraction of a and b is= " + (a-b) + "<br/>");
  document.write("Multiplication of a and b is= " + (a*b) + "<br/>");
  document.write("Division of a and b is= " + (a/b) + "<br/>");
  document.write("Modulus of a and b is= " + (a%b) + "<br/>");
</script>
</body>
</html>
```



```
Performing Arithmetic Operations

Addition of a and b is= 15
Subtraction of a and b is= 9
Multiplication of a and b is= 36
Division of a and b is= 4
Modulus of a and b is= 0
```

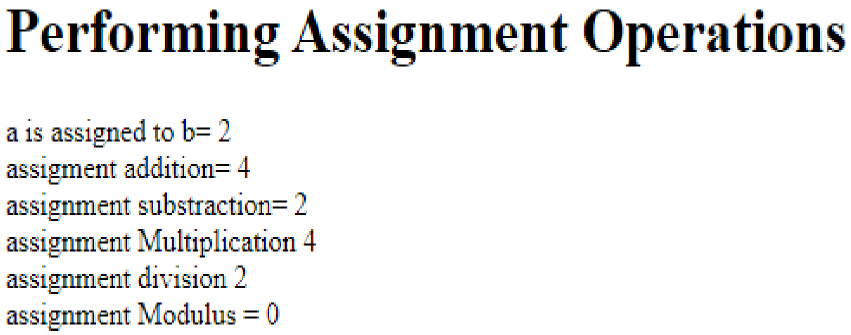
b)Assignment operator:

operator	**Description**

=	assign the value to the left hand side variable
+=	adds the right hand side operand to the left hand side operand and assign result to the left hand side operand
-=	subtract the right hand side operand to the left hand side operand and assign result to left hand side operand
*=	multiplies the right hand side operand to the left hand side operand and assign result to the left hand side operand
/=	divides the left hand side operand to the right hand side operand and assign quotient to the left hand side operand
%=	divides the left hand side operand to the right hand side operand and assign reminder to the left hand side operand

Sample program:

```
<!DOCTYPE html>
<html>
<head>
   <title>Javascript Assignment
Operators</title>
</head>
<body>
   <h1>Performing Assignment Operations
</h1>
<script>
  var a=2,b;
  document.write(" a is assigned to b= " + (b=a) + "<br/>");
  document.write("assigment addition= "+ (a+=b) + "<br/>");
  document.write(" assignment substraction= " + (a-=b) + "<br/>");
  document.write(" assignment Multiplication " + (a*=b) + "<br/>");
  document.write(" assignment division  " + (a/=b) + "<br/>");
  document.write(" assignment Modulus = " + (a%=b) + "<br/>");
</script>
</body>
</html>
```



```
Performing Assignment Operations

a is assigned to b= 2
assigment addition= 4
assignment substraction= 2
assignment Multiplication 4
assignment division 2
assignment Modulus = 0
```

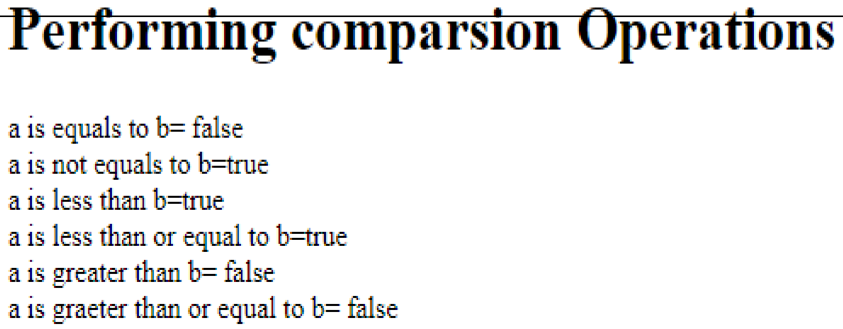
c)Comparison operator:

operator	**Description**
==	Returns true if both the operands are equal; otherwise it returns false
!=	Returns true if both the operands are not equal; otherwise it returns false
<	True if the left hand side operand is less than right hand side operand
<=	True if the left hand side operand is less than or equal to right hand side operand
>	Returns true if the left hand side operand is greater than right hand side operand

Sample program:

```
<!DOCTYPE html>
```

```
<html>
<head>
   <title>Javascript comparsion
Operators</title>
</head>
<body>
   <h1>Performing comparsion
Operations </h1>
<script>
  var a=2,b=5;
 document.write(" a is equals to b= " + (a==b) + "<br/>");
 document.write("a is not equals to b="+ (a!=b) + "<br/>");
 document.write(" a is less than b=" + (a<b) + "<br/>");
 document.write(" a is less than or equal to b=" + (a<=b) + "<br/>");
 document.write("  a is greater than b= " + (a>b) + "<br/>");
 document.write("a is graeter than or equal to b= " + (a>=b) + "<br/>");
</script>
</body>
</html>
```



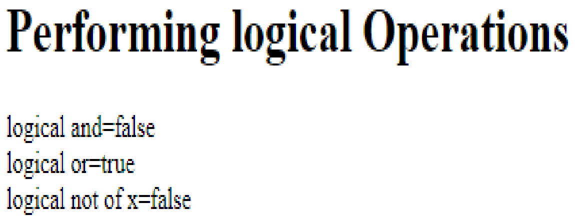
```
Performing comparsion Operations

a is equals to b= false
a is not equals to b=true
a is less than b=true
a is less than or equal to b=true
a is greater than b= false
a is graeter than or equal to b= false
```

d)Logical operators:

operator	**Description**
||	Returns true only if either of the operand is true it returns false when both the operands are false
&&	Returns true only if both the operands are true otherwise it returns false
!	Negates the operand

Sample program:

```
<html>
     <head>
        <title>Javascript logical Operators</title>
     </head>
     <body>
        <h1>Performing logical  Operations </h1>
     <script>
     var x=true,y=false;
     document.write("logical and="+(x&&y)+"</br>");
     document.write("logical or="+(x||y)+"</br>");
     document.write("logical not of x="+(!x)+"</br>");
     </script>
     </body>
     </html>
```



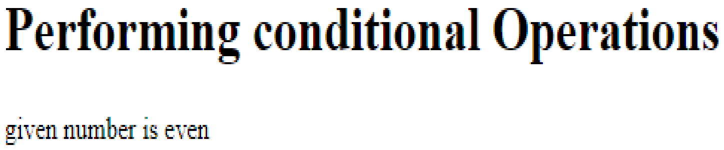
```
Performing logical Operations

logical and=false
logical or=true
logical not of x=false
```

e)Conditional operators:

? :   returns the second operand if the first operand is true. However if the first operand is false it returns third operand

Sample program:

```
<!DOCTYPE html>
<html>
<head>
   <title>Javascript conditional Operator</title>
</head>
<body>
   <h1>Performing conditional  Operations </h1>
```



```
Performing conditional Operations

given number is even
```

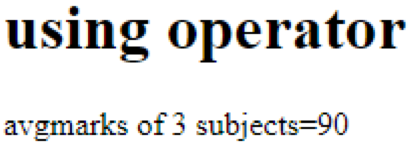
```
<script>
   var n=20;
(n%2==0)?document.write(" given number is even"):document.write(" given number is odd");
</script>
</body>
</html>
```

/* Write a javascript code to find the average of three Numbers*/

Operator:

```
<html>
<head>
<title> using operator </title>
</head>
<body>
<h1> using operator </h1>
<script type="text/javascript">
var a=80,b=95,c=95,avgmarks;
avgmarks=(a+b+c)/3;
document.write("avgmarks of 3 subjects="+avgmarks);
</script>
</body>
</html>
```

Output:



```
using operator

avgmarks of 3 subjects=90
```

4.4 USE IF, IF-ELSE AND SWITCH CONDITION STATEMENT: Scripts written in JavaScript are executed sequentially, which means the first statement in which script is the first to be executed and the last statement in the script is the last to be executed this is the simplest and most straight forward way to execute scripts

If statement : The if statement is one of the most basic and simplest control flow statements. you can use the if statement when you want to execute a group of one or more script statements only when a particular condition is met.

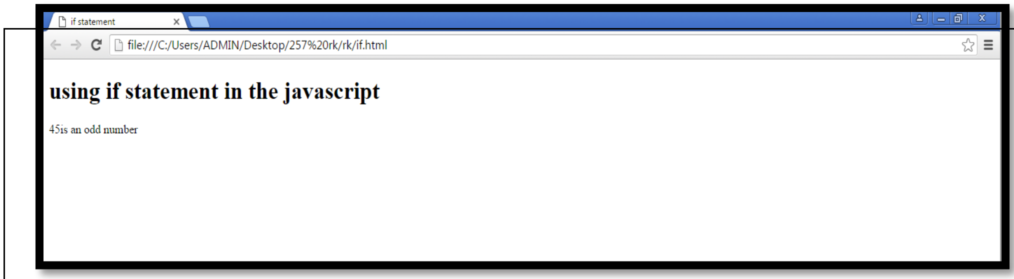
syntax

```
if(condition)
{
  Statements;
 }
```

/*write a java script code to check whether given no is odd*/

```
<html>
<head>
<title>  if statement </title>
</head>
<body>
<h1> using if statement in the javascript</h1>
<script type="text/javascript">
var number=45;
if((number%2)!=0)
document.write(number+"is an odd number");
</script>
</body>
</html>
```

Output:-



using if statement in the javascript

45is an odd number

If-else statement: However if you want to execute another set of statements when the condition is false then you can use if-else statement

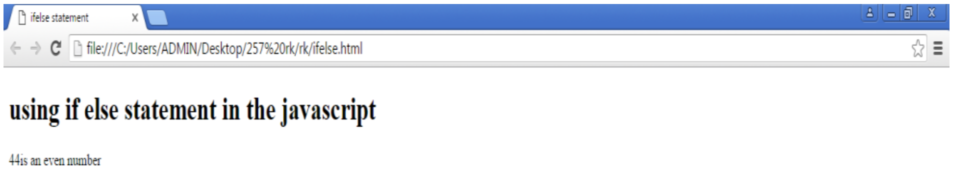
Syntax

```
 if(condition)
{
Statment1;
}
else
{
Statment2;
}
```

/*write a java script code to check whether given no is even or odd*/

```
<html>
<head>
<title>ifelse statement </title>
</head>
<body>
<h1> using if else statement in the javascript</h1>
<script type="text/javascript">
var n=44;
if(n%2==0)
{
document.write(n+"is an even number");
}
else
{
document.write(n+"is an odd number");
}
</script>
</body>
</html>
```

Output:-



using if else statement in the javascript

44is an even number

Switch statement: A switch statement is used to select a particular group of statement to be executed among several other group of statements on the basis of numeric or string expression.

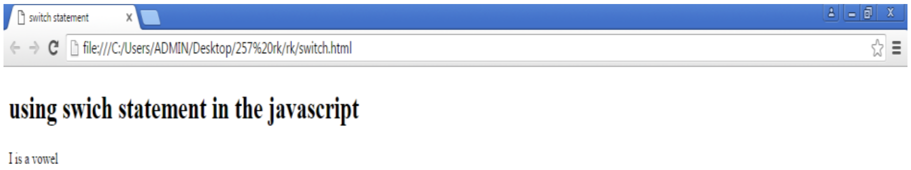
Syntax

```
switch(expression)
{
case value1: statement1;
break;
case value2: statement1;
break;
default:statement-default;
break:
```

}

/*write a java script code to check whether given character is vowel or consnant*/

```
<html>
<head>
<title> switch statement </title>
</head>
<body>
<h1> using swich statement in the javascript</h1>
<script type="text/javascript">
var letter="I";
switch(letter)
{
default:document.write("consnant");
break;
case "A":document.write("A is a vowel");
break;
case "E":document.write("E is a vowel");
break;
case "I":document.write("I is a vowel");
break;
case "O":document.write("O is a
vowel");
break;
case "V":document.write("V is a
vowel");
break;
}
</script>
</body>
</html>
```



using swich statement in the javascript

I is a vowel

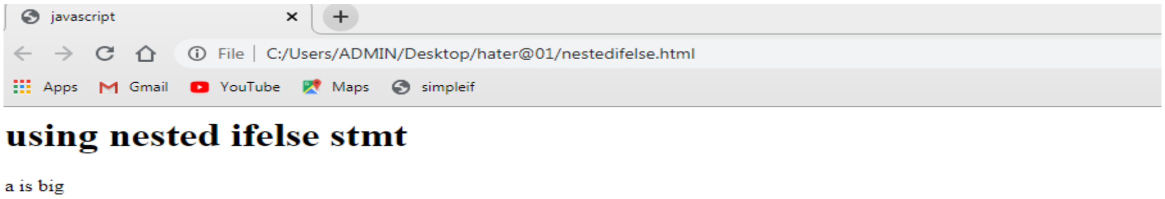
Nested if else statement: In java script you can also define one if…..else statement into another. Such statements are known as nested if-else statement. They are used in situation when additional checking are validation is required

Syntax:

```
if(condition1)
{
    if(condition2)
        {
        statement1;
        }
        else
        {
         Statement2;
        }
 }
else
{
Statement 3;
}
```

/*write a java script code to check whether given number is big among three numbers*/

```
<html>
<head>
<title>javascript</title>
 </head>
       <body>
<h1>using nested ifelse stmt</h1>
  <script type="text/javascript" >
          var a=5,b=3,c=2;
         if(a>b)
     {
           if(a>c)
        {
         document.writeln( "a is big");
        }
      else
        {
          document.writeln( "c is big");
        }
     }
     else
    {
   if(b>c)
     {
          document.writeln( "b is big");
     }
  else
    {
         document.writeln("c is big");
     }
   }
</script>
</body>
</html>
```



using nested ifelse stmt

a is big

Else if ladder:

Here, a user can decide among multiple options.The if statements are executed from the top down. As soon as one of the conditions controlling the if is true, the statement associated with that if is executed, and the rest of the ladder is bypassed. If none of the conditions is true, then the final else statement will be executed.

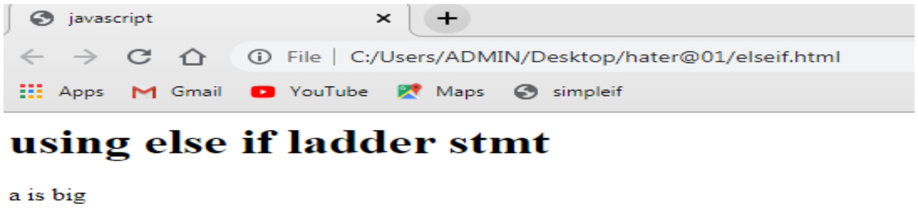
Syntax:

```
if(condition)
{
          statement;
}
else  if(condition2)
{
```

```
Statement2;
}
…….
-------
else
{
Statement n;
}
```

/*write a java script code to check whether given number is big among three numbers*/

```
<html>
<head>
<title>javascript</title>
 </head>
      <body>
     <h1>using else if ladder stmt</h1>
        <script type="text/javascript" >
         var a=5,b=3,c=2;
        if(a>b && a>c)
         document.writeln( "a is big");
        else if(b>a && b>c)
         document.writeln( "a is big");
         else
         document.writeln( "c is big");
  </script>
</body>
</html>
```



```
using else if ladder stmt

a is big
```

4.5 USE WHILE, DO- WHILE, AND FOR ITERATIVE STATEMENTS & SAMPLE PROGRAMS

Understanding loops: loops or iteration statements or control flow statements that allow you to execute a particular group of statements repeatedly the number of times group of statements is executed depends on a particular condition . loop statements are :

1. While loop
2. Do-while loop
3. For loop

While loop: check the condition at the start of the loop. The group of statements that is executed is specified after the condition

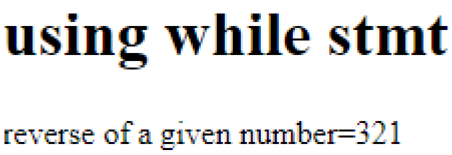
Syntax: while(condition)

```
Syntax:  while(condition)
              {
                  body of the loop;
               }
```

/* write a javascript code to print the the reverse of a given number*/

```
<html>
<head>
<title>javascript</title>
 </head>
      <body>
<h1> using while stmt</h1>
        <script type="text/javascript" >
         var n=123,rem,rev=0;
          while(n!=0)
      {
         rem=n%10;
```

output:



```
using while stmt

reverse of a given number=321
```

```
          rev=rev*10+rem;
        n=(n-rem)/10;
          }
 document.writeln("reverse of a given number="+rev);
  </script>
</body>
</html>
```

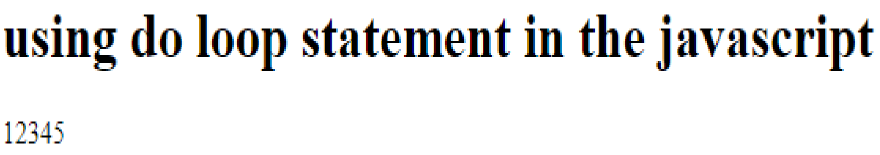
Do while loop: the group of statements to execute at least once even if the condition is false then you can use the do while loop

```
Syntax:    do{
                body of the loop;
             }while(condition);
```

/* write a javascript code to print 1 to n natural numbers*/

```
<html>
<head>
<title> do loop statement </title>
</head>
<body>
<h1> using do loop statement in the javascript</h1>
<script type="text/javascript">
var i=1;
do
{
document.write(i);
i++
}
while(i<=5);
</script>
</body>
</html>
```

Output:



using do loop statement in the javascript

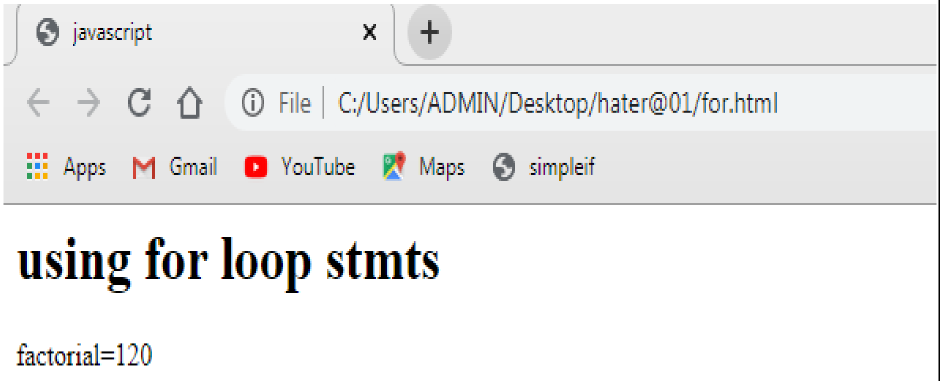
12345

For loop:- It executes a group of statements for a pro determined no of times

```
Syntax: for(initialization; condition; updation statement)
    {
    body of the loop;
    }
```

***write a javascript code to find the factorial of the given number using for loop*/**

```
<html>
<head>
<title>javascript</title>
 </head>
      <body>
        <h1> using for loop stmts</h1>
        <script type="text/javascript" >
         var i,fact=1,n=5;
          for(i=1;i<=5;i++)
      {
            fact=fact*i;
      }
document.write("factorial="+fact);
```



using for loop stmts

factorial=120

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