JAVA SCRIPT LECTURE NOTES

INTRODUCTION

JavaScript is the world's most popular scripting language. It is the language for HTML and the web, for servers, PCs, laptops, tablets, smart phones, and more.

JavaScript is a scripting language

A scripting language is a lightweight programming language. JavaScript code can be inserted into HTML pages which can be executed by all modern web browsers.

The <script> Tag

To insert a JavaScript into an HTML page, use the <script> tag. The <script> and </script> tells where the JavaScript starts and ends. The lines between the <script> and </script> contain the JavaScript code

```
<script type=”text/javascript”>




</script>
```

We can place this JavaScript code in head part as well as body part also.

OUTPUT IN JAVA SCRIPT

We can print any text in a web page by using JavaScript in following ways.

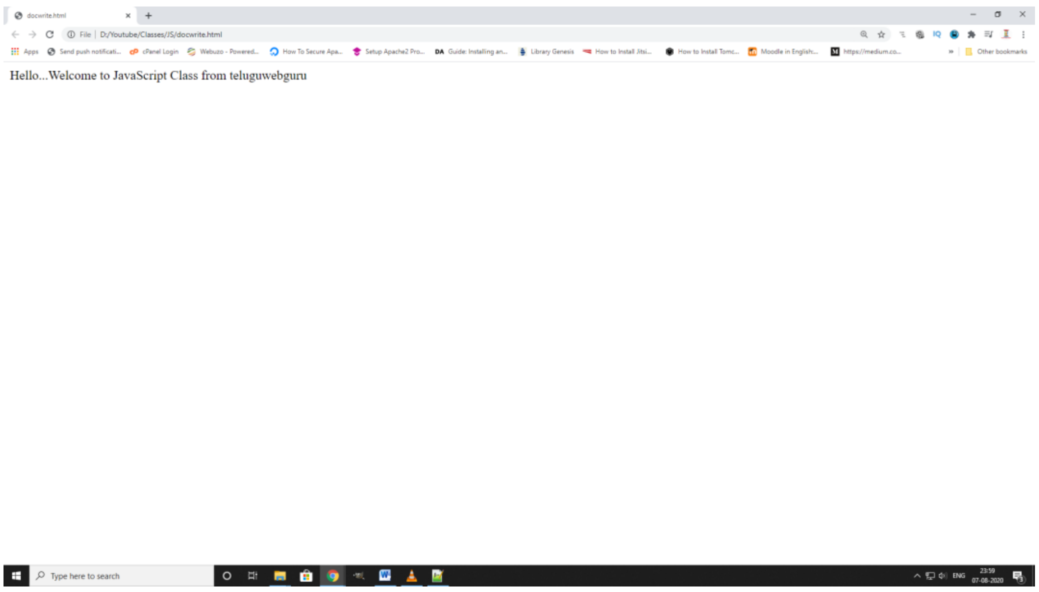
1) document.write("hello") :

This statement is equivalent to writing the text "hello" in body part.

Example Program:

```
<html>
<head>
<script type="text/javascript">
   document.write("Hello...Welcome to JavaScript Class from teluguwebguru");
</script>
</head>
<body>


</body>
</html>
```

Output:



```
Hello...Welcome to JavaScript Class from teluguwebguru
```

Here document is an object that represents html document (body part).

Whatever the text that we write within the document.write statement, that text will be directly placed within body part of the webpage.

2) innerHTML

If we want to place a text within start and end of any tag (innerHTML) then this method will be used.

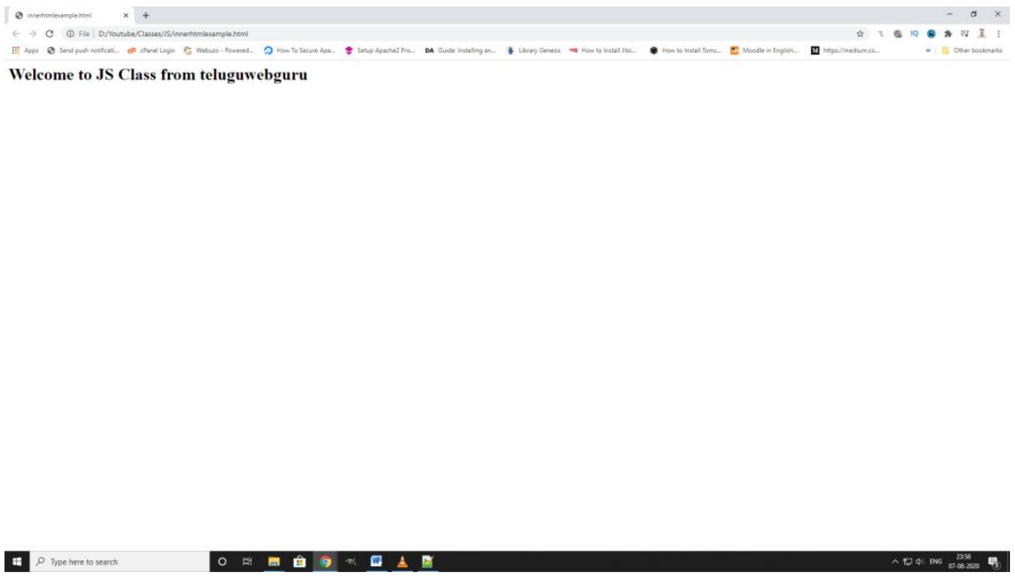
To implement this we need to assign a id to the tag so that we can place the content within this particular tag from javascript part by identifying the tag using the id assigned.

Example:

```
<html>

<head>
</head>

<body>

<h1 id="special"> </h1>

<script type="text/javascript">
document.getElementById("special").innerHTML = "Welcome to JS Class from
teluguwebguru";
</script>

</body>
</html>
```

Here as script part requires finding the tag with id “special”. JavaScript part is included after the tag h1 with id special so that system will execute the h1 tag first then only the java script can identify this tag.

Output:



```
Welcome to JS Class from teluguwebguru
```

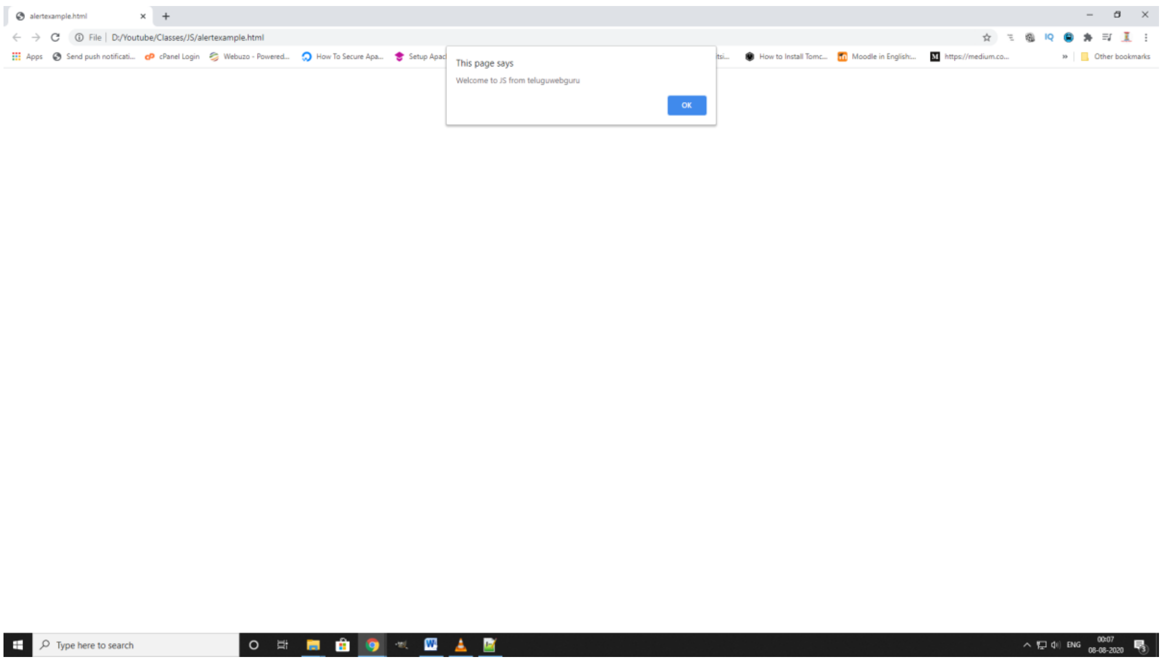
3) window.alert(“Hello”);

To display output message in separate window then we may use alert box. By using this method whatever the message that we send to browser through window.alert that will be displayed in a separate window.

Example :

```
<html>

<head>
</head>

<body>

<script type="text/javascript">
window.alert("Welcome to JS from teluguwebguru");
</script>

</body>

</html>
```

Output



This page says
Welcome to JS from teluguwebguru

COMMENTS IN JAVA SCRIPT

Comments are very useful in any programming language.

Comments will

- Improves the program understand ability and readability.
- To skip executing certain statements in a program by placing them under comments.

To write single line comments we will use //

Example :

```
<script type=”text/javascript”>
document.write(“hello”);
//document.write(“Hi”);    -- This line will never execute because it is in comments
</script>
```

To write multi line comments we simply place those particular statements witin /* */

```
<script type=”text/javascript”>
document.write(“hello”);
/*
document.write(“Hi”);
document.write(“welcome”);
*/
</script>
```

VARIABLES IN JAVA SCRIPT

JavaScript variables are containers for storing data values.

In Java Script we can declare variables by using var keyword.

Examples :

```
var a = 2; //number

var b = “teluguwebguru”;   //string

var c = true;   //boolean

var d = new Array(..) //array
```

In the above shown examples please observe that all variables are declared by using the keyword ‘ var ’. However based on the assigned values the particular datatype will be decided.

Java Script Data Types

Java Script variables can hold many datatypes: numbers, strings, objects, and more

```
var length = 16;                        // Number
var lastName = "Johnson";                  // String
var x = {firstName:"John", lastName:"Doe"};    // Object

var c = true;   //boolean

var d = new Array(..) //array
```

Javascript supports undefined data type where all the variables are belonged to undefined datatype only its values are not assigned. Once values are assigned then corresponding datatypes will be decided.

Javascript supports null datatype which is a special datatype in java script. We will discuss about this in further concepts.

Data types of Java Script variables will be decided at runtime while assigning its values. This type of languages are called as Dynamic Typed Languages.

Examples of Dynamic typed languages are Java Script, PHP, Python etc.,

POPUP BOXES IN JAVA SCRIPT

Popup Boxes in Java Script

There are three different types of popup boxes in Java Script

Alert Box	-	To give alert message to the user
Confirm Box	-	To take confirmation from the user
Prompt Box	-	To Take input from the user

Alert Box :

An alert box is often used if you want to make sure information comes through to the user.

When an alert box pops up, the user will have to click "OK" to proceed.

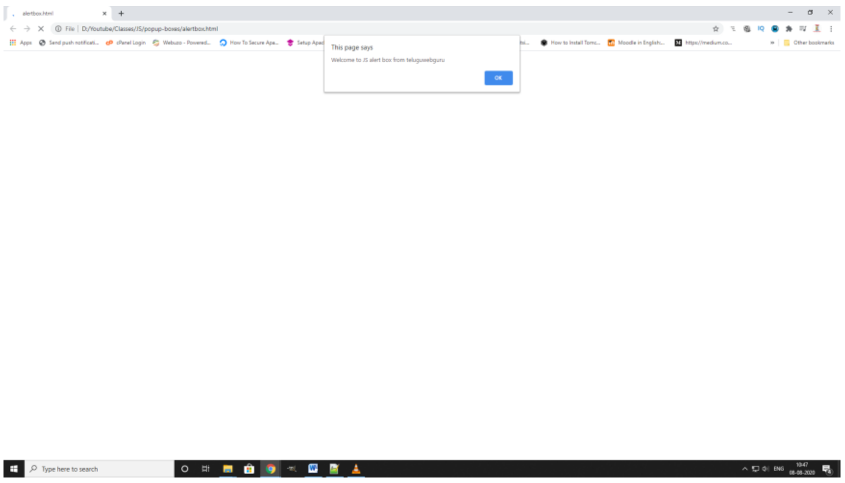
Syntax :

```
Window.alert("message");
```

Example:

```
<html>

<head>
<script type="text/javascript">
window.alert("Welcome to JS alert box from teluguwebguru");
</script>
</head>

<body>

</body>

</html>
```

Output:



This page says

Welcome to JS alert box from teluguwebguru

Confirm Box:

A confirm box is often used if you want the user to verify or accept something.

When a confirm box pops up, the user will have to click either "OK" or "Cancel" to proceed.

If the user clicks "OK", the box returns **true**. If the user clicks "Cancel", the box returns **false**.

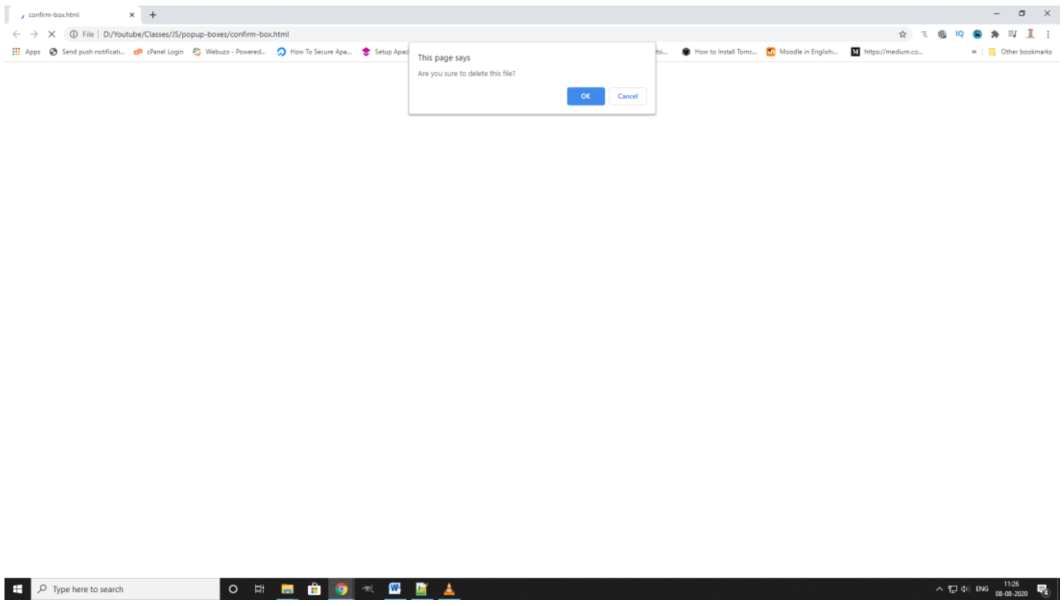
Syntax:

window.confirm("*confirmation message*");

Example:

```
<html>

<head>

<script type="text/javascript">
window.confirm("Are you sure to delete this file?");
</script>

</head>

<body>

</body>

</html>
```

Output:



This page says

Are you sure to delete this file?

Prompt Box :

A prompt box is often used if you want the user to input a value before entering a page.

When a prompt box pops up, the user will have to click either "OK" or "Cancel" to proceed after entering an input value.

If the user clicks "OK" the box returns the input value. If the user clicks "Cancel" the box returns null.

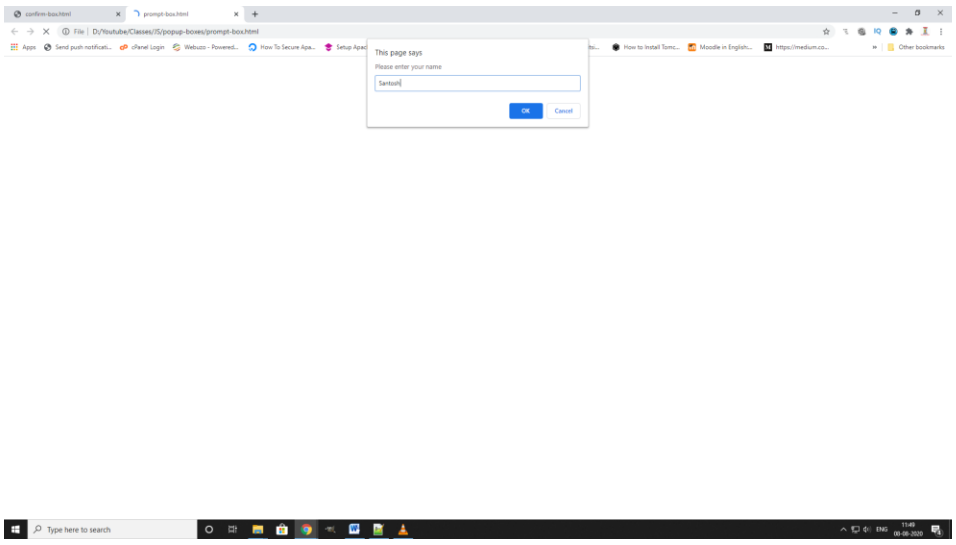
Syntax :

window.prompt("*sometext*","*defaultText*");

Example:

```
<html>

<head>

<script type="text/javascript">

var name = window.prompt("Please enter your name");

document.write("Hello "+name+", Welcome to teluguwebguru channel");

</script>

</head>

<body>

</body>

</html>
```

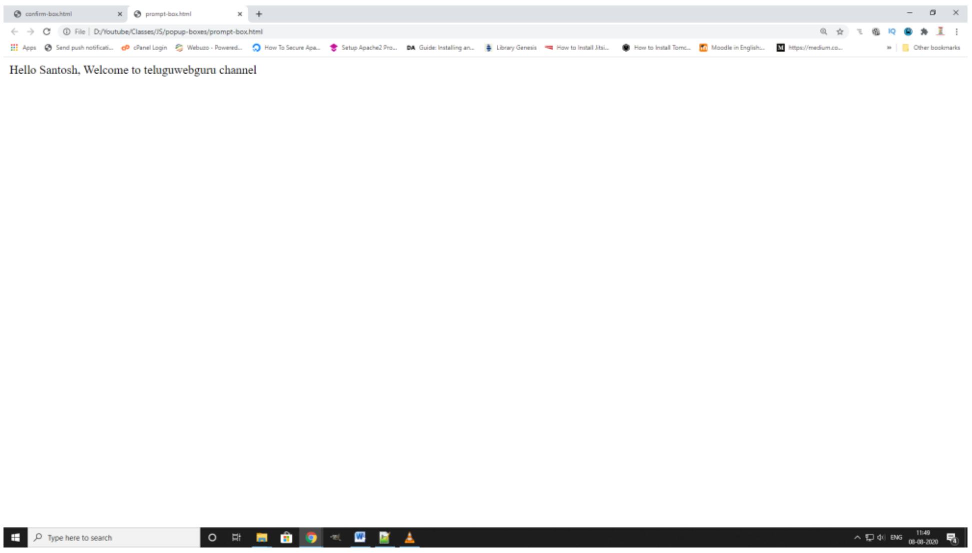
Output:



This page says

Please enter your name

Santosh



Hello Santosh, Welcome to teluguwebguru channel

OPERATORS IN JAVA SCRIPT

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

Arithmetic Operators:

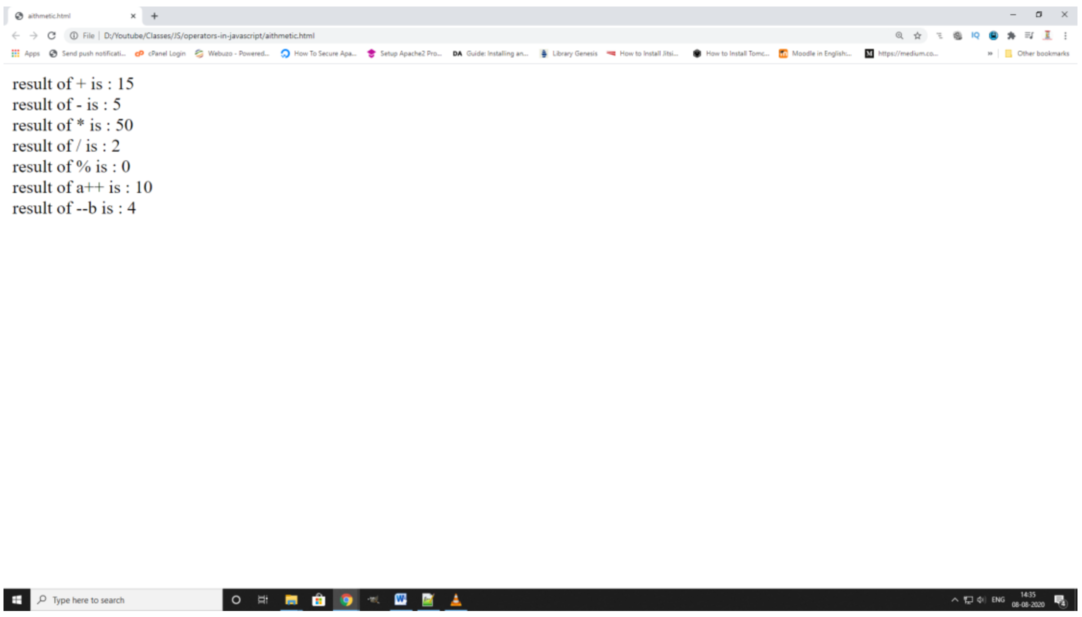
Arithmetic operators are used to perform arithmetic on numbers:

Operator	Description
+	Addition
-	Subtraction
*	Multiplication
**	Exponentiation
/	Division
%	Modulus
++	Increment
--	Decrement

Example :

```
<html>

<head>
<script type="text/javascript">
var a=10;
var b=5;
document.write("result of + is : "+(a+b));
document.write("<br/>");
document.write("result of - is : "+(a-b));
document.write("<br/>");
document.write("result of * is : "+(a*b));
document.write("<br/>");
document.write("result of / is : "+(a/b));
document.write("<br/>");
document.write("result of % is : "+(a%b));
document.write("<br/>");
document.write("result of a++ is : "+(a++));
document.write("<br/>");
document.write("result of --b is : "+(--b));
document.write("<br/>");
</script>
</head>


<body>



</body>



</html>
```

Output:



```
result of + is : 15
result of - is : 5
result of * is : 50
result of / is : 2
result of % is : 0
result of a++ is : 10
result of --b is : 4
```

Arithmetic Assignment Operators :

+=   -=   *=   /=   %=

Assignment operators assign values to JavaScript variables.

For example a+=b means a = a + b

Comparison Operators:

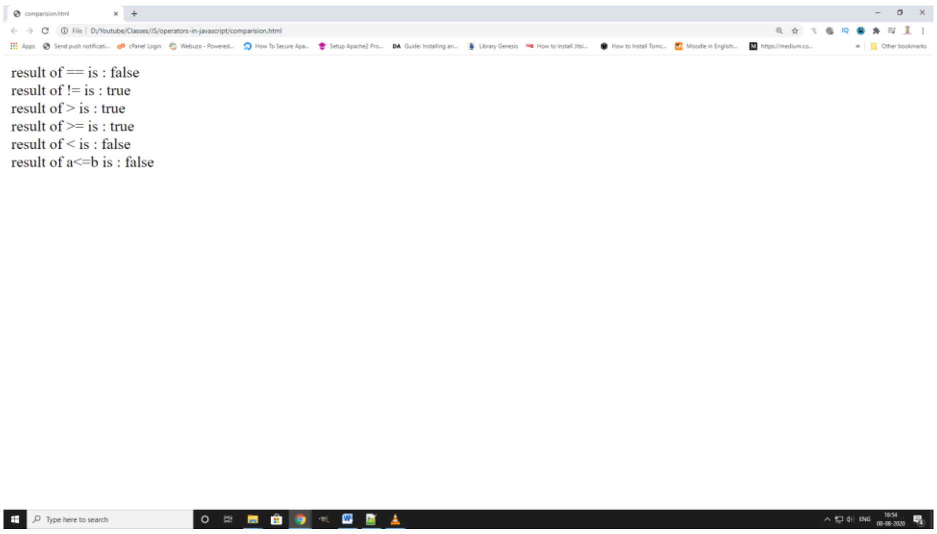
Comparison and Logical operators are used to test for true or false.

Operator	Description
==	equal to
===	equal value and equal type
!=	not equal
!==	not equal value or not equal type
>	greater than
<	less than
>=	greater than or equal to
<=	less than or equal to
?	ternary operator

Example

```
<html>

<head>
<script type="text/javascript">
var a=10;
var b=5;
document.write("result of == is : "+(a==b));
document.write("<br/>");
document.write("result of != is : "+(a!=b));
document.write("<br/>");
document.write("result of > is : "+(a>b));
document.write("<br/>");
document.write("result of >= is : "+(a>=b));
document.write("<br/>");
document.write("result of < is : "+(a<b));
document.write("<br/>");
document.write("result of a<=b is : "+(a<=b));
document.write("<br/>");

</script>
</head>
<body>

</body>



</html>
```

Output



```
result of == is : false
result of != is : true
result of > is : true
result of >= is : true
result of < is : false
result of a<=b is : false
```

Logical Operators:

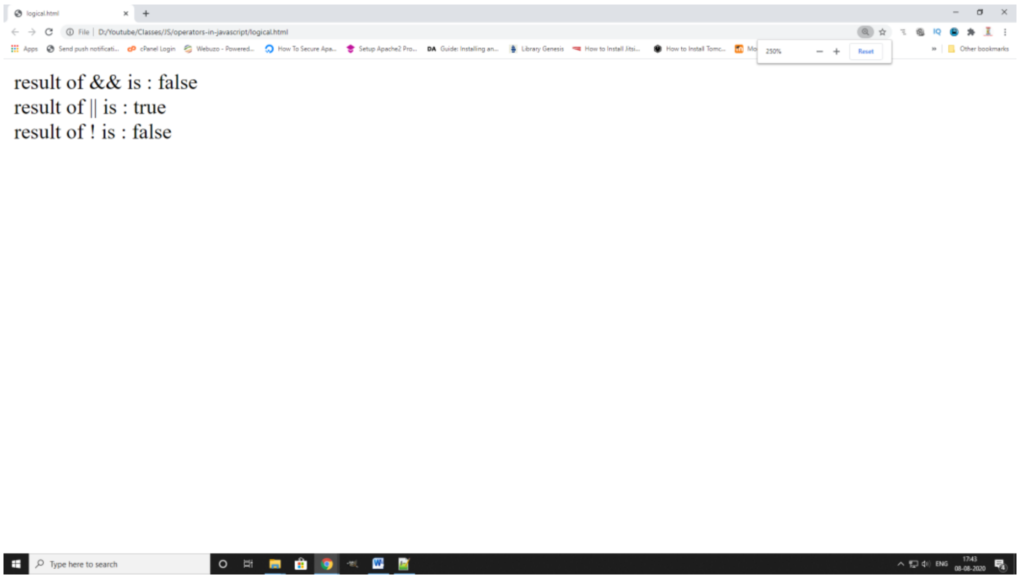
Logical operators are used to determine the logic between variables or values.

Operator	Description	Example
&&	and	(x < 10 && y > 1) is true
||	or	(x == 5 || y == 5) is false
!	not	!(x == y) is true

Example:

```
<html>

<head>
<script type="text/javascript">
var a=true;
var b=false;

document.write("result of && is : "+(a&&b));
document.write("<br/>");
document.write("result of || is : "+(a||b));
document.write("<br/>");
document.write("result of ! is : "+(!a));
</script>
</head>
<body>



</body>

</html>
```

Output



```
result of && is : false
result of || is : true
result of ! is : false
```

Bitwise Operators :

Bit operators work on 32 bits numbers.

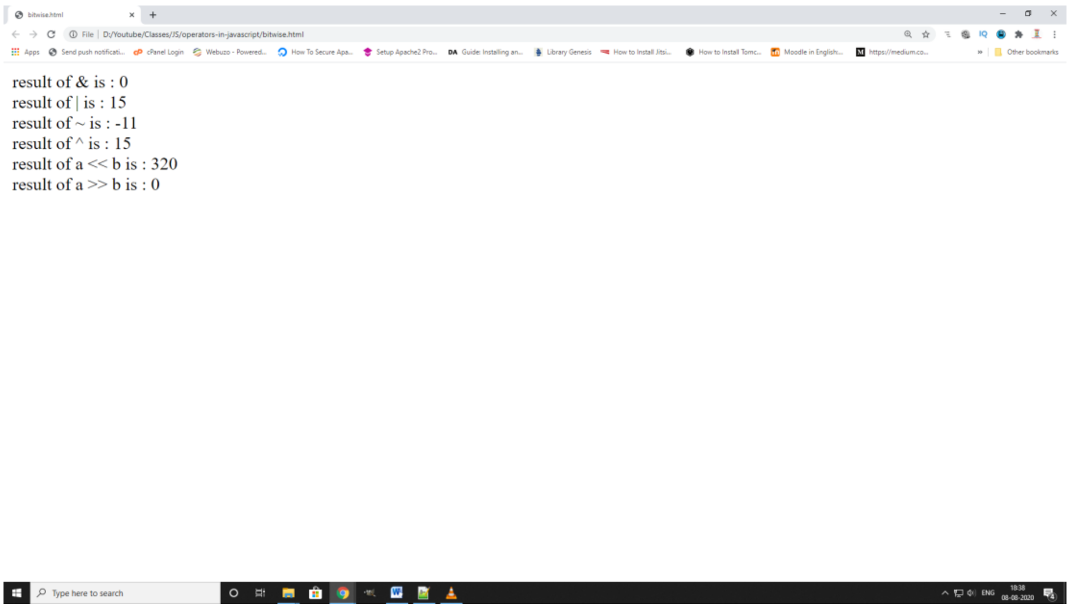
Any numeric operand in the operation is converted into a 32 bit number. The result is converted back to a JavaScript number.

Operator	Description	Example	Same as	Result	Decimal
&	AND	5 & 1	0101 & 0001	0001	1
|	OR	5 | 1	0101 | 0001	0101	5
~	NOT	~ 5	~0101	1010	10
^	XOR	5 ^ 1	0101 ^ 0001	0100	4
<<	Zero fill left shift	5 << 1	0101 << 1	1010	10
>>	Signed right shift	5 >> 1	0101 >> 1	0010	2
>>>	Zero fill right shift	5 >>> 1	0101 >>> 1	0010	2

Example

```
<html>

<head>
<script type="text/javascript">
var a=10;
var b=5;
document.write("result of & is : "+(a&b));
document.write("<br/>");
document.write("result of | is : "+(a|b));
document.write("<br/>");
document.write("result of ~ is : "+(~a));
document.write("<br/>");
document.write("result of ^ is : "+(a^b));
document.write("<br/>");
document.write("result of a << b is : "+(a<<b));
document.write("<br/>");
document.write("result of a >> b is : "+(a>>b));
document.write("<br/>");
</script>
</head>


<body>


</body>


</html>
```

Output :



```
result of & is : 0
result of | is : 15
result of ~ is : -11
result of ^ is : 15
result of a << b is : 320
result of a >> b is : 0
```

Assignment Operator (=)

This operator is used to assign values to the variables.

Example :

```
    var a  = 10;
```

Conditional (Ternary) Operator :

JavaScript also contains a conditional operator that assigns a value to a variable based on some condition.

Syntax :

variablename = (*condition*) ? *value1*:*value2*

FUNCTIONS IN JAVA SCRIPT

A function is a group or collection of reusable code which can be called from anywhere in our program.

Advantages of functions :

- Saves lot of time in writing same code for multiple times
- We can execute set of lines (function) based on event / action performed.

How to declare a function :

We can declare a function by using a keyword 'function' as follows.

```
function functionname (optional parameters) {

        function code

}
```

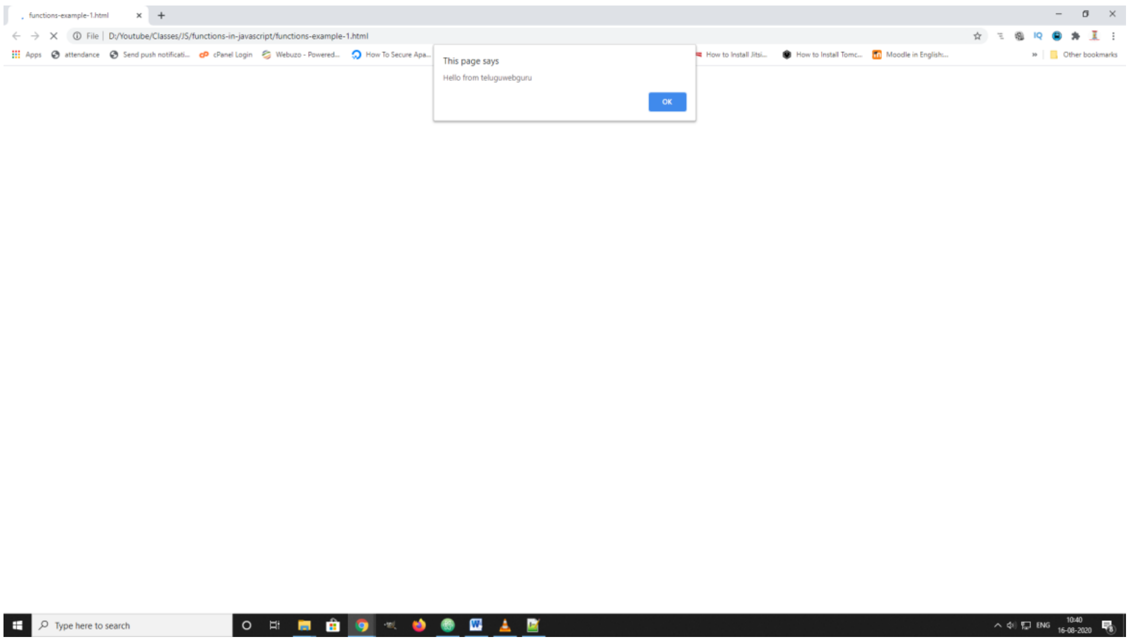
We can call the function as below.

```
functionname (parameter values if any)
```

Example :

```
<html>

        <head>
                <script type="text/javascript">
                        function f()
                        {
                                window.alert("Hello from teluguwebguru");

                        }

                        f();
                </script>
        </head>


        <body>

        </body>



</html>
```

Output:


This page says

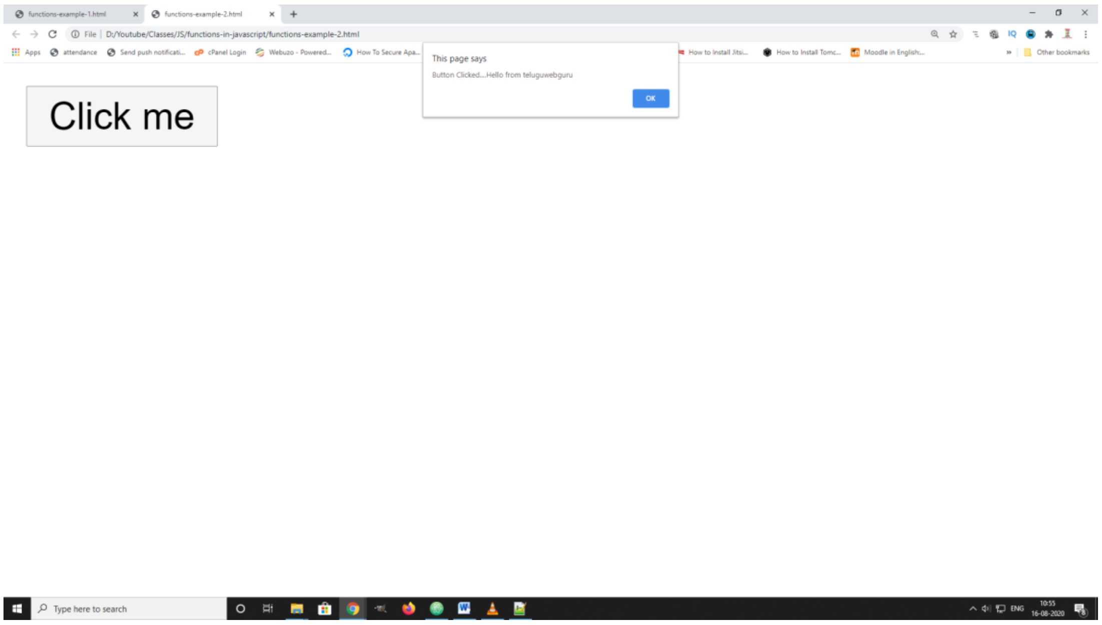
Hello from teluguwebguru

In the above example function is called immediately after declaration. But by using events we can call java script functions whenever we want.

Example:

```
<html>
        <head>
                <script type="text/javascript">
                        function f()
                        {
                                window.alert("Button Clicked....Hello from teluguwebguru");

                        }


                </script>
        </head>

        <body>
          <input type="button" value="Click me" onclick="f()">


        </body>



</html>
```

Output



```
This page says
Button Clicked....Hello from teluguwebguru
```

In the above example function f () will be executed whenever we click on the button. We made it possible with the help of javascript event onclick.

ARRAYS IN JAVA SCRIPT

Arrays are used to store multiple values in a single variable

or

Array is a datatype which can store collection of homogeneous type values.

Syntax

var arrayname = new Array(values separated by comma);

Example

var values = new Array(10,20,30);

We can access the elements of an array by using index which starts from 0. So, in above example 10 is at index 0, 20 is at index 1, 30 is at index 2. If we try to access the value at index which does not exists then it simply returns "undefined".

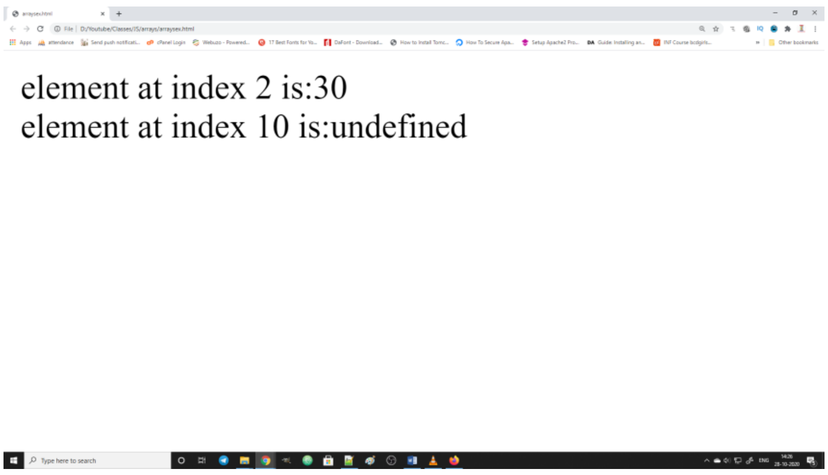
if we want to print the value at index 0 then we can write the statement as follows.

document.write(a[0]);

Example

```
<html>
<head>
 <script type="text/javascript">
    var values = new Array(10,20,30,40,50);
          document.write("element at index 2 is:"+values[2]+"<br/>");
          document.write("element at index 10 is:"+values[10]);
 </script>
</head>
<body>

</body>
</html>
```

Output



```
element at index 2 is:30
element at index 10 is:undefined
```

CONDITIONAL OR CONTROL STATEMENTS IN JAVA SCRIPT

We generally use control statements to implement various functions in our programs.

We have

1) if condition:
   if is a selection statement which allows us to execute one block of statements based on the condition returns true.

```
syntax:
   if(condition)
   {
     //statements
   }
```

Above block of statements will be executed if condition becomes true.

Under if, we have if-else statement too where one block of statements among two will be executed based on the condition result (true or false).

```
syntax:
        if(condition)
        {
          //statements
        }
        else
        {
          //else statements
        }
```

In the above code if condition satisfies then first block of statements will be executed. If condition becomes false then else block will be executed.

Under if, we have if-else-if statements also where we can check more than one condition and based on the conditions result corresponding block of statements will be selected for execution.

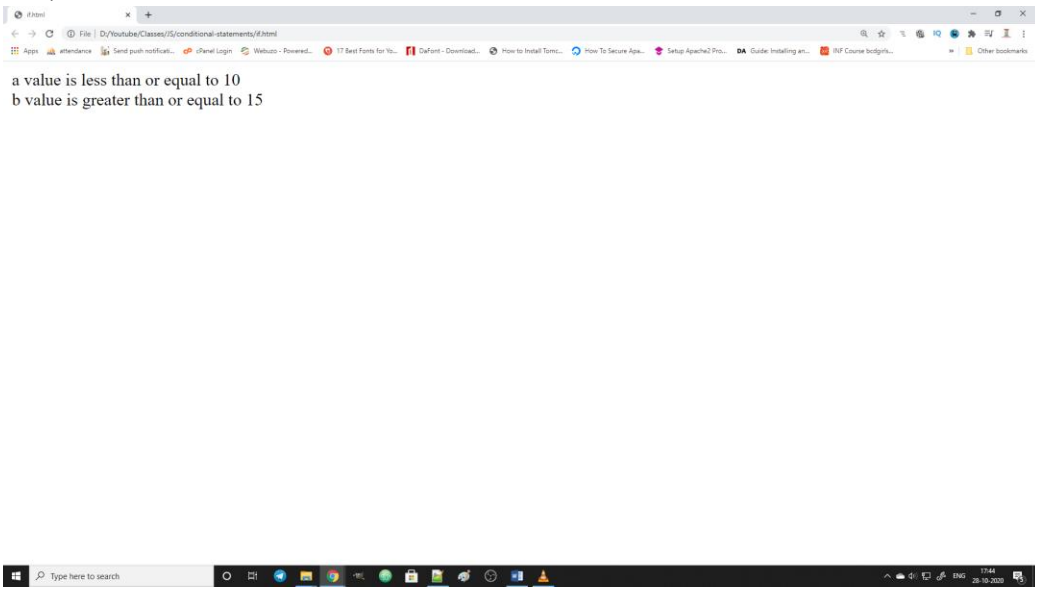
```
syntax:
        if(condition1)
        {
          //statements
        }
        else if(condition2)
        {
          //statements
        }
        else
        {
          //else statements
        }
```

In the above syntax, if first condition is true then corresponding block will be executed, other wise it evaluates the other conditions too and select the block for execution based on the condition. If all the conditions are becoming false then last else block will be executed.

Example:

```
        <html>
        <head>
                <script type="text/javascript">
                        function f()
                        {
                                var a=10;
                                var b=18;

                                if(a<=10)
                                {
                                  document.write("a value is less than or equal to 10<br/>");
                                }

                           if(b<=10)
                                {
                                  document.write("b value is less than or equal to 10<br/>");
                                }
                                else if(b<15)
                                {
                                  document.write("b value is less than 15<br/>");
                                }
                                else
                                {
                                  document.write("b value is greater than or equal to 15<br/>");
                                }

                        }

                        f();
                </script>
        </head>

                <body>
                </body>

</html>
```

Output:



```
a value is less than or equal to 10
b value is greater than or equal to 15
```

2) switch

switch case will also work same as if. It is also used to select and execute a block of statements based on a condition.

syntax:

```
switch(expression)
{
    case ____: statements;
               break;
    case ____: statements;
               break;

    case ____: statements;
               break;

    default: statements;
               break;

}
```

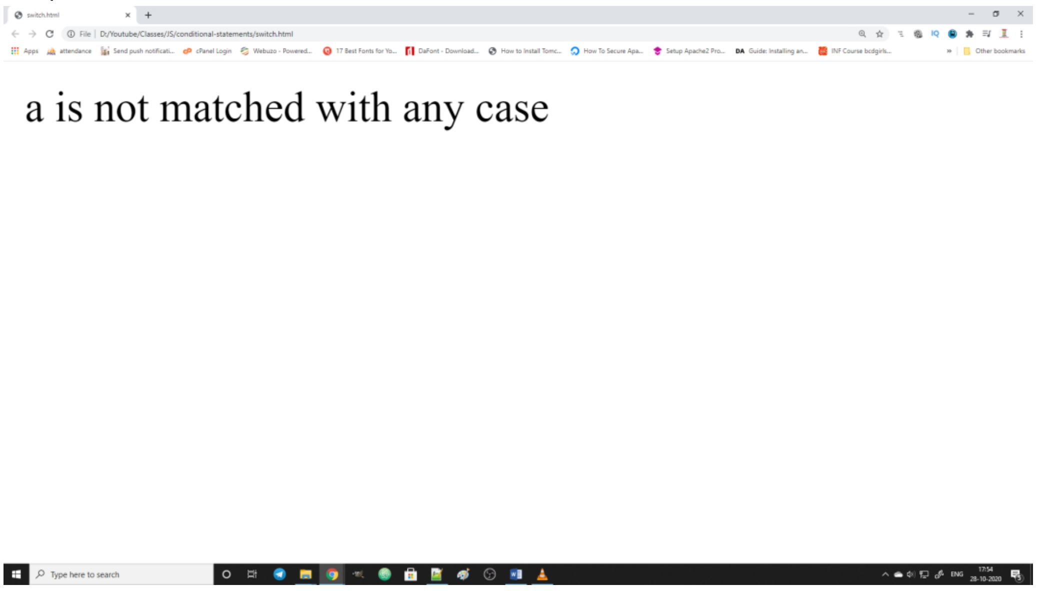
In above syntax the case that matches with expression value is selected for execution. if no case is matched with the expression then default case will be selected for execution.

Example

```
<html>
<head>
        <script type="text/javascript">
                function f()
                {
                  var a=17;
```

```
                                switch(a){
                                  case 0: document.write("a is 0");break;
                                         case 5:document.write("a is 5");break;
                                         case 10:document.write("a is 10");break;
                                         default:document.write("a is not matched with any case");
                                }
                              }

                              f();
                    </script>
          </head>

                    <body>
                    </body>

</html>
```

Output:



```
a is not matched with any case
```

3) while

By using while condition we can execute block of statements more than one time.

syntax :

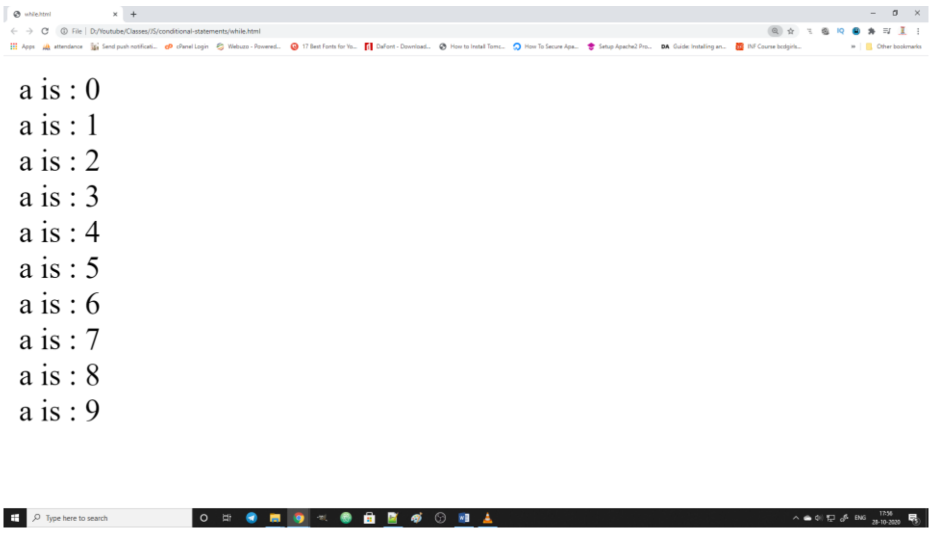
```
while(condition)

{

          statements;

}
```

above block of statements will be repeatedly executed and whenever the condition specified is failed then it stops the execution

Example:

```
<html>
        <head>
                <script type="text/javascript">
                        function f()
                        {
                          var a=0;

                          while(a<10){
                            document.write("a is : "+a+"<br/>");
                                   a++;
                          }
                        }

                        f();
                </script>
        </head>

        <body>
        </body>

</html>
```

Output



```
a is : 0
a is : 1
a is : 2
a is : 3
a is : 4
a is : 5
a is : 6
a is : 7
a is : 8
a is : 9
```

4) for

for loop also used to execute block of statements more than one time repeatedly based on a condition.

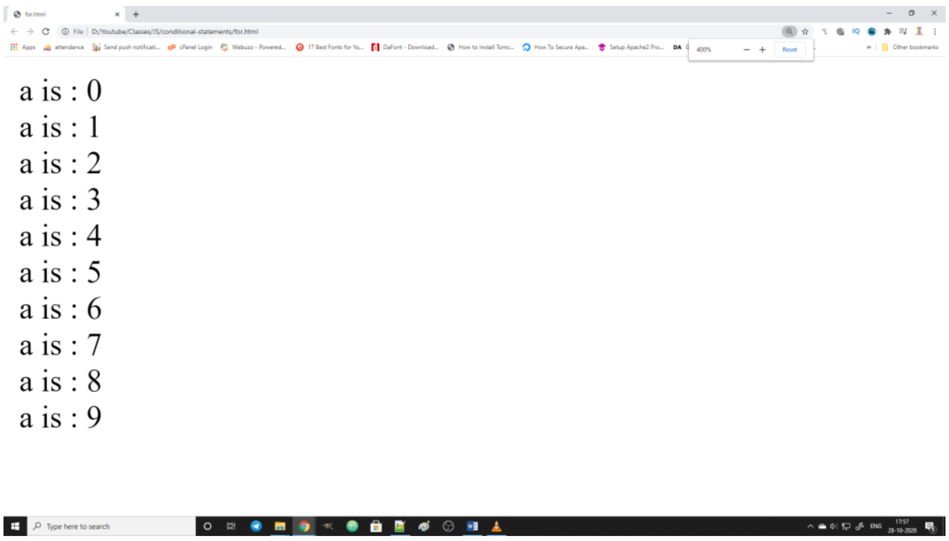
syntax:

```
for(initialization;condition;iteration)
 {

 }
```

Example :

```
<html>
        <head>
                <script type="text/javascript">
                        function f()
                        {
                          var a=0;

                          for(a=0;a<10;a++){
                            document.write("a is : "+a+"<br/>");
                          }
                        }

                        f();
                </script>
        </head>

        <body>
        </body>

</html>
```

Output



```
a is : 0
a is : 1
a is : 2
a is : 3
a is : 4
a is : 5
a is : 6
a is : 7
a is : 8
a is : 9
```

In above example, observe that unlike while where iteration statements will be written inside block, for loop allows us to write initialization, condition, iteration statements at a time.

DATE IN JAVA SCRIPT

We can use Date in java script by using following constructors.

new Date()

This is used to create date object with current date and time.

new Date(milliseconds)

This is used to create date object with our required date and time. But we need to specify the date and time in milliseconds. it adds the milliseconds that you provided to 1970 jan 01st 00:00:00 and create a date object with the result.

new Date(datestring)

We can create date object by providing date string.

new Date(year,month,date[,hour,minute,second,millisecond])

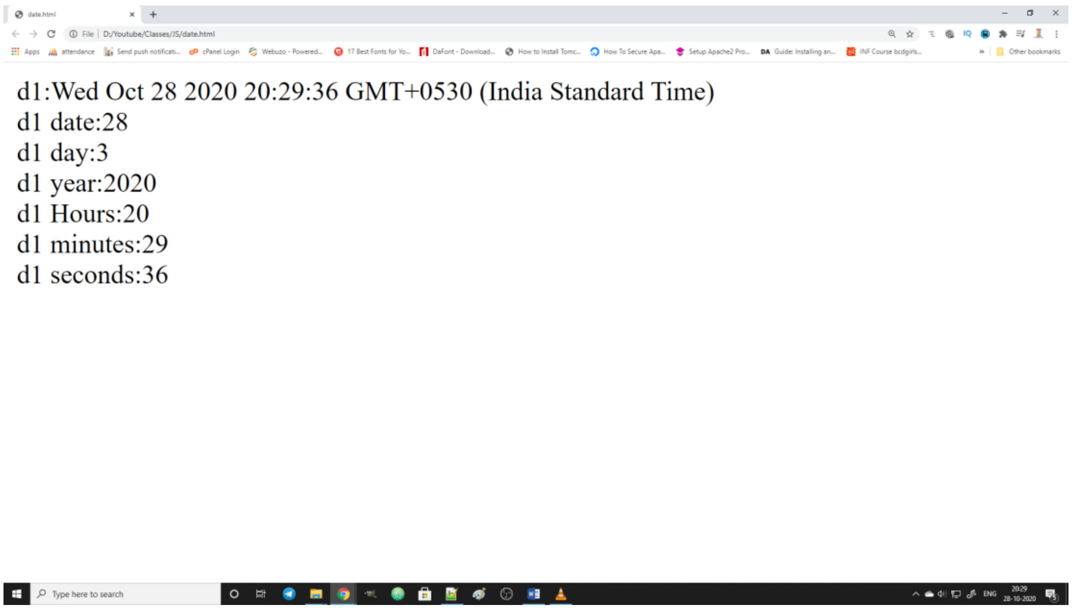
We can create date object by providing values as per above constructor.

Once a date object is created, we can retrieve independent values like date, day etc., by using methods getDate(), getDay().

Example:

```
        <html>
        <head>
                <script type="text/javascript">
                        function f()
                        {
                          var d1 = new Date();
                          document.writeln("d1:"+d1+"<br/>");
                          document.writeln("d1 date:"+d1.getDate()+"<br/>");
                          document.writeln("d1 day:"+d1.getDay()+"<br/>");
                          document.writeln("d1 year:"+d1.getFullYear()+"<br/>");
                          document.writeln("d1 Hours:"+d1.getHours()+"<br/>");
                          document.writeln("d1 minutes:"+d1.getMinutes()+"<br/>");
                          document.writeln("d1 seconds:"+d1.getSeconds()+"<br/>");
                        }

                        f();
                </script>
        </head>

                <body>
                </body>

</html>
```

output

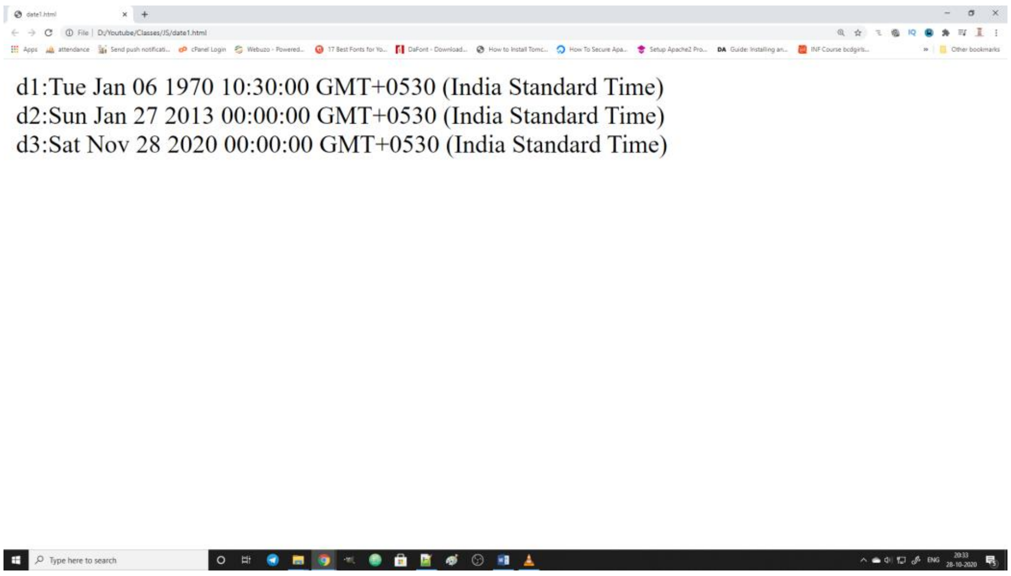


```
d1:Wed Oct 28 2020 20:29:36 GMT+0530 (India Standard Time)
d1 date:28
d1 day:3
d1 year:2020
d1 Hours:20
d1 minutes:29
d1 seconds:36
```

This example explains about first constructor and various methods available to retrieve independent values from the date object.

Example2

```
        <html>
                <head>
                        <script type="text/javascript">
                                function f()
                                {
                                  var d1 = new Date(450000000);
                                  document.writeln("d1:"+d1+"<br/>");

                                  var d2 = new Date('january 27 2013');
                                  document.writeln("d2:"+d2+"<br/>");

                                  var d3 = new Date(2020,10,28);
                                  document.writeln("d3:"+d3+"<br/>");

                                 }

                                f();
                        </script>
                </head>

                <body>
                </body>

</html>
```


```
d1:Tue Jan 06 1970 10:30:00 GMT+0530 (India Standard Time)
d2:Sun Jan 27 2013 00:00:00 GMT+0530 (India Standard Time)
d3:Sat Nov 28 2020 00:00:00 GMT+0530 (India Standard Time)
```

This example explains about remaining three objects in date object.

OBJECTS IN JAVA SCRIPT

Collection of properties is said to be an object.

We can create object as follows

```
var person = {name: ' Santosh ', age: 35, channel: ' Telugu Web Guru ',}
```

We can create properties and their values by mentioning them in curly braces as key – value pairs.

Again we can access the properties of objects as follows

```
person.age  or person['age'] will returns 35. (the value of the property age )
```

once an object is defined, we can insert any number of properties into the object as follows

```
person['city']='Visakhapatnam';

person['getdetails'] = function(){
                        //function code
                     }
```

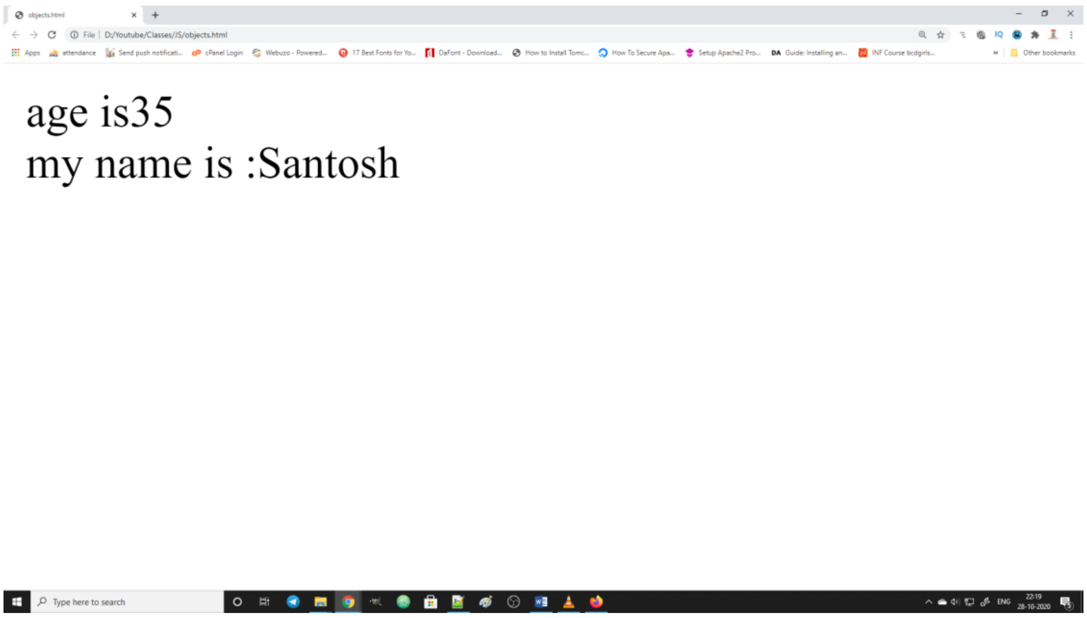
here in above property instead of value a function is given so whenever the property getdetails is called the associated function will be executed.

Example

```
<html>
        <head>
                <script type="text/javascript">
                        function f()
                        {
                          //creating object
                          var person = {name:'Santosh',
                                  age:35,
                                                channel:'Telugu Web Guru',
                                                getmydetails: function(){
                                                        return "my name is :"+this.name;
                                                     }

                                                };

                          //accessing properties
                          document.write('age is'+person.age+'<br/>');
                          document.write(person.getmydetails());


                        }

                        f();
```

```
                    </script>
          </head>

          <body>
          </body>

</html>
```

Output



```
age is35
my name is :Santosh
```

In above example this refers to the current object on which this method is executing.

EVENTS IN JAVA SCRIPT

We can decide when a java script function to be executed. For example we can execute a java script function whenever an event is triggered like when user perform mouse click, button click etc., these are called as events.

To implement this we have events concept where we can call function by using event.

Example

```
< body onload=" f()">
```

In above example function f() will be executed whenever the web page is loaded (this is called onload event).

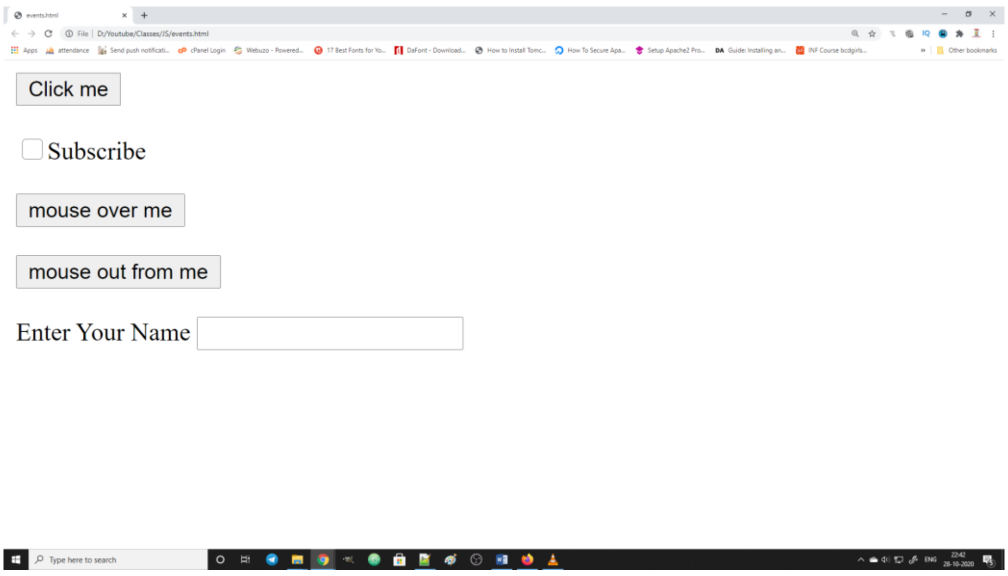
We have the following events in java script

1) onchange : changing the options of choice, list, checkbox, radio button
2) onclick : clicking any button
3) ondblclick :double clicking a button
4) onmouseover : placed mouse pointer on an element
5) onmouseout : moved mouse pointer our from an element
6) onkeydown : pressed key down on keyboard
7) onkeyup : key released in keyboard
8) onkeypress : key typed in keyboard
9) onload : whenever web page is loaded
10) onfocus : whenever an input element gets focus
11) onblur : whenever an input element lost focus

Example1

```
<html>
            <head>
                <script type="text/javascript">
                        function f()
                        {
                          document.write('Welcome to Telugu web guru');
                        }


                </script>
            </head>

            <body>
              <button onclick="f()">Click me</button> <br/><br/>
                <input type="checkbox" value="subscribe" onchange="f()">Subscribe<br/><br/>
                <button onmouseover="f()">mouse over me</button><br/><br/>
                <button onmouseout="f()">mouse out from me</button><br/><br/>
                Enter Your Name <input type="text" onfocus="f()">
                </body>

</html>
```

Output



Click me

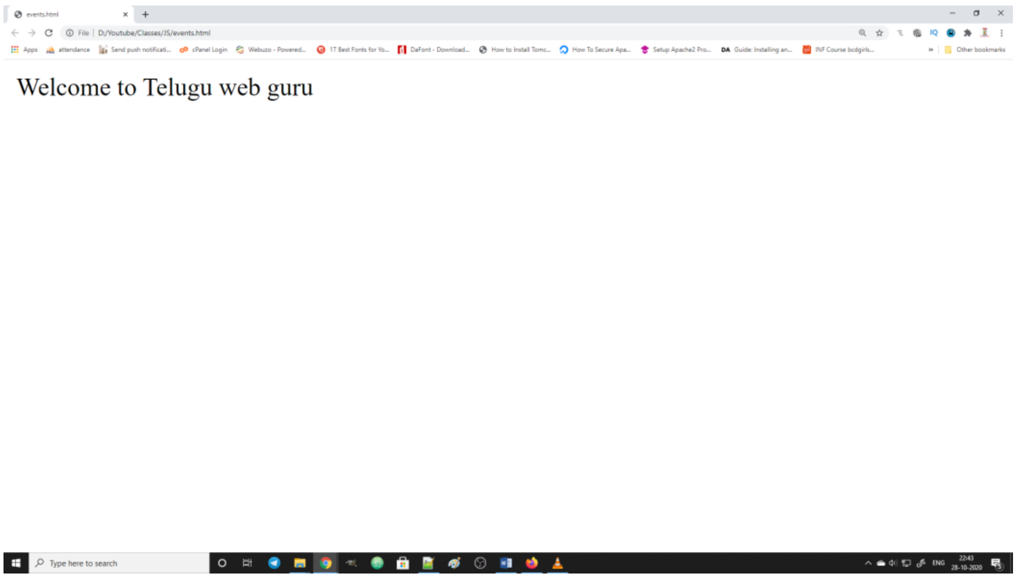
Subscribe

mouse over me

mouse out from me

Enter Your Name

For example above program calls the same function on generating different events. for example if we click on a button " Click me " then the function will be executed as follows(onclick event).



Welcome to Telugu web guru

whenever we clicked on checkbox also the event will be generated (onchange), similarly the following events are also will call the function : whenever we placed mouse pointer on "mouse over me" button , moved mouse pointer out from button “mouse out from me”, whenever we clicked on text box so that it gets focus(onfocus) .

ERRORS IN JAVA SCRIPT

Errors are statements which stops the program's execution.

we can use the following keywords to clear errors in java script

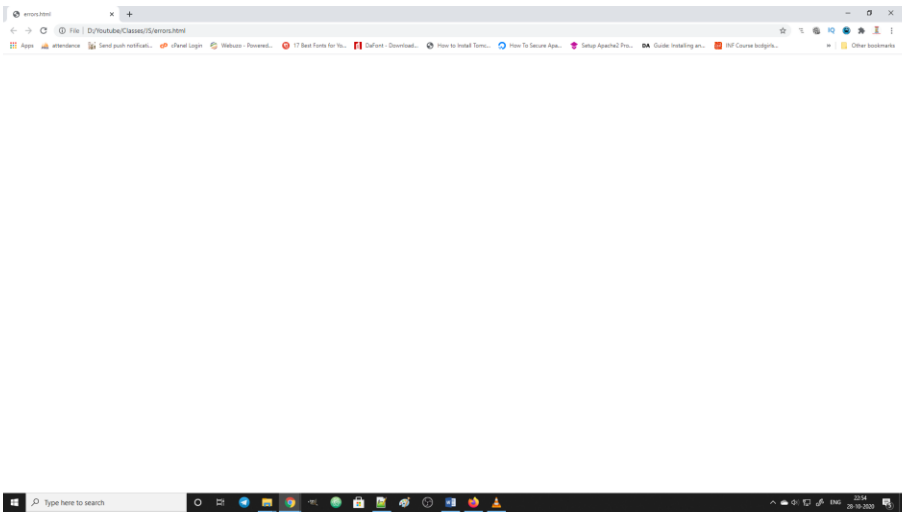
1) try : this will identify the exceptions
2) catch : this will handle the exceptions
3) throw : to manually create and throw user defined exceptions
4) finally : contains must executable statements. Statements in finally block will be executed always irrespective of existence of error

Example

```
<html>
                <head>
                    <script type="text/javascript">
                              function f()
                              {
                                document.write('Before Error Line');
                                window.alerting("Hello Welcome");
                                document.write('After Error Line');

                              }

                           f();

                    </script>
                </head>

                <body>


                </body>

</html>
```

For example in the above program intentionally we mentioned alerting instead of alert which generates the error.

Because a line in this program contains error we never get the program executed even though two lines are perfect in the program

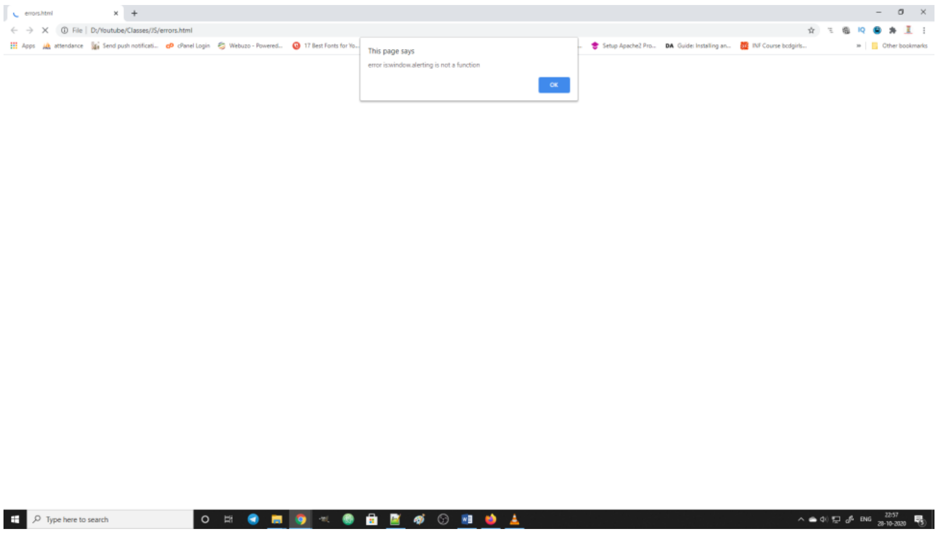


So to overcome this problem and to make the perfect lines executable, we need to use try-catch blocks as follows.

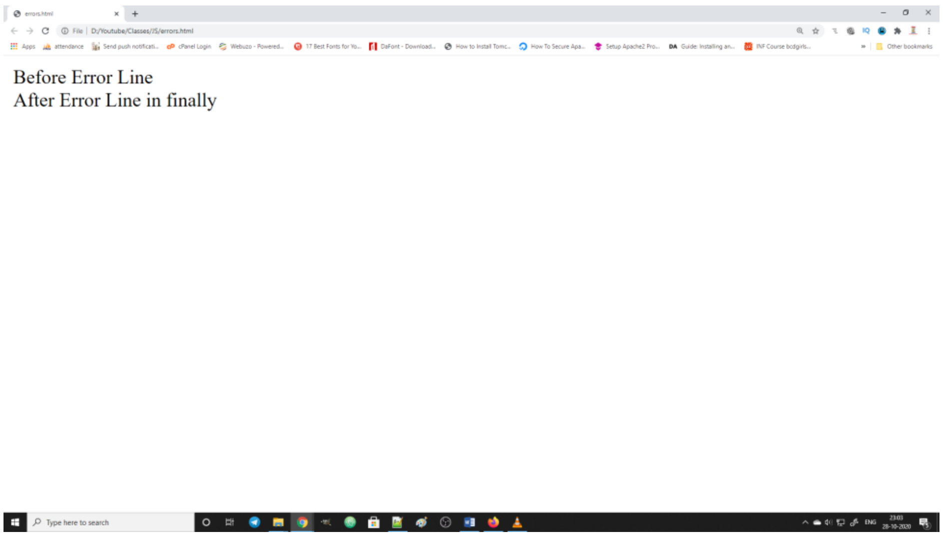
Example

```
<html>
            <head>
                <script type="text/javascript">
                        function f()
                        {
                          document.write('Before Error Line<br/>');
                          try{
                          window.alerting("Hello Welcome");
                          }
                          catch(err){
                            alert('error is:'+err.message);
                          }
                          finally{
                              document.write('After Error Line in finally');
                          }
                        }

                        f();


                </script>
            </head>

            <body>


            </body>

</html>
```

Output



```
This page says
error is:window.alerting is not a function
```



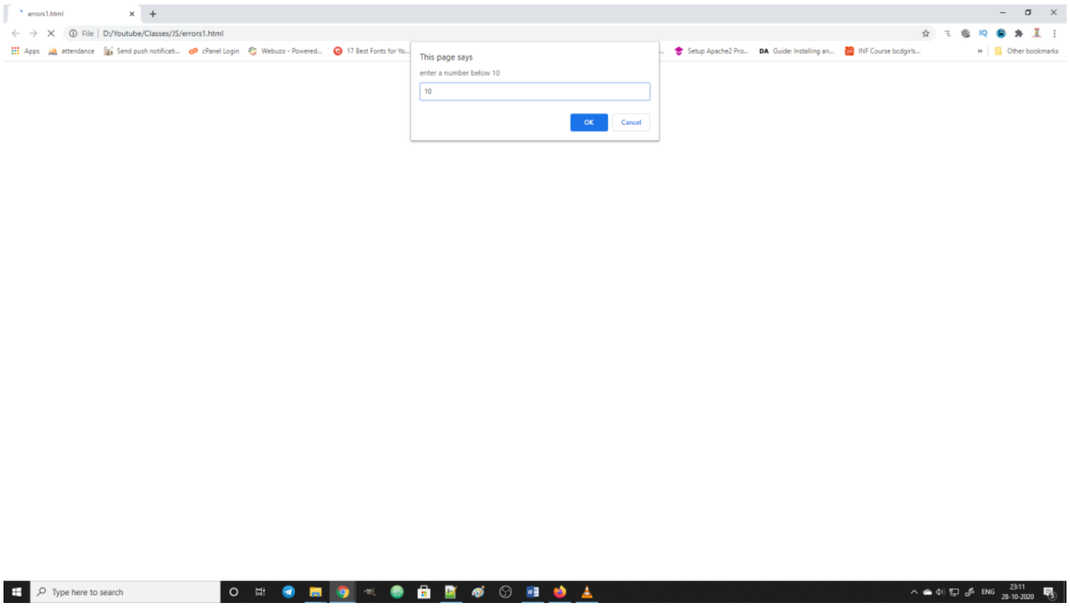
```
Before Error Line
After Error Line in finally
```

Now we are able to execute the perfect lines by handling the program using try -catch blocks. here try block identified the error occurred and forwarded to the catch block and then catch block executes it's handler code available in it and then remaining lines will get executed.

Also observe that the statement in finally block always will be executed irrespective of whether the program contains error or not.

User Defined Exception :

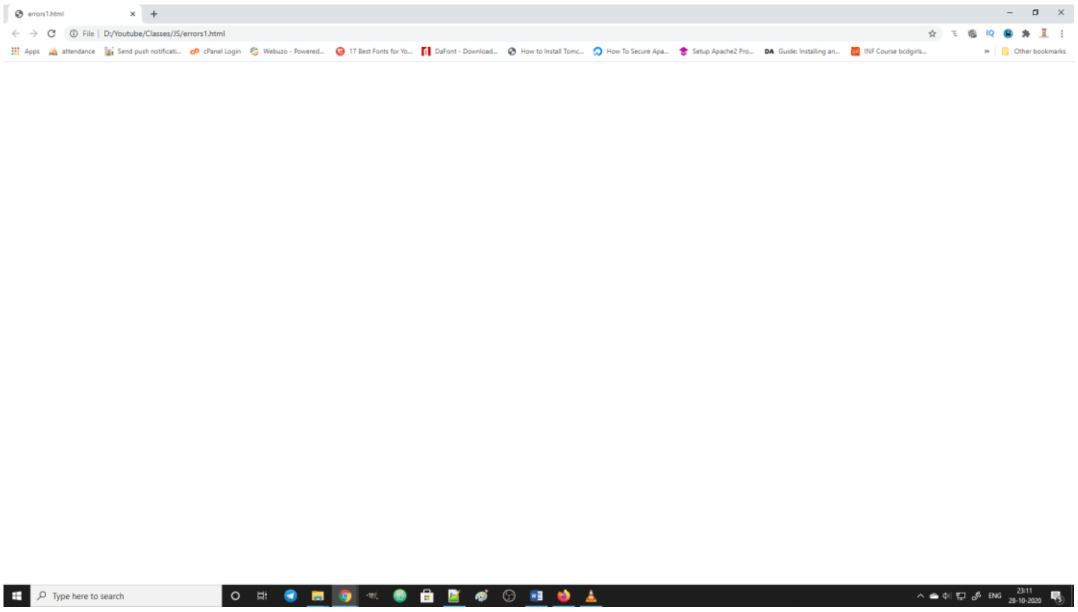
We can generate and throw errors as per our conditions or requirements in our program by using user defined exceptions.



This page says

enter a number below 10

10



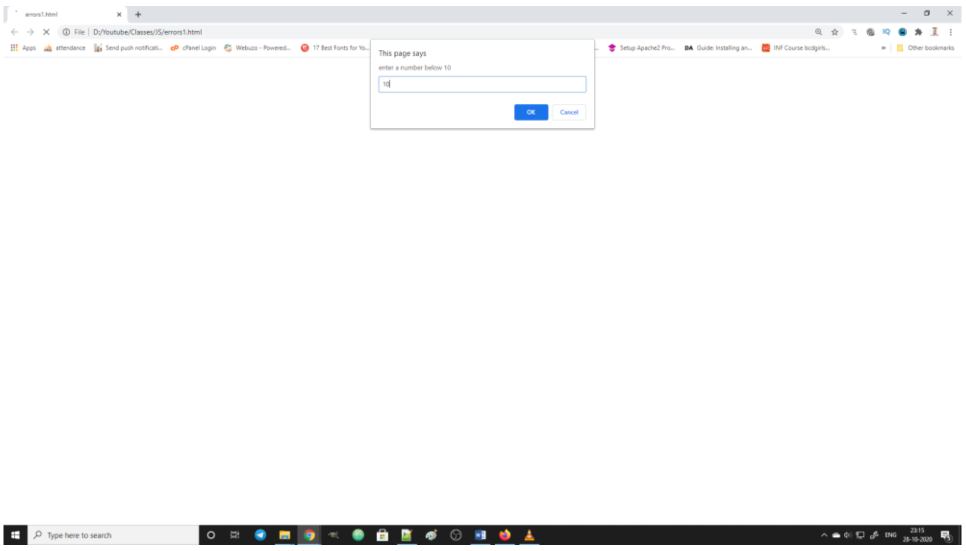
Here we entered 10 as input which raises the error so we did not get the output as the error is not yet handled by the try catch blocks

Now let us try to solve this by using try – catch blocks

```
<html>
        <head>
                <script type="text/javascript">
                        function f()
                        {
                          var input = window.prompt("enter a number below 10");
                          var num = parseInt(input);
```

```
                    if(num<10)
                      document.write("Correct input");
                          try{
                    if(num>=10)
                                  throw new Error("Wrong input");
                            }
                            catch(err){
                              alert(err.message);
                            }

                  }

                  f();


          </script>
  </head>

      <body>
      </body>

</html>
```

Now the output is


errors1.html

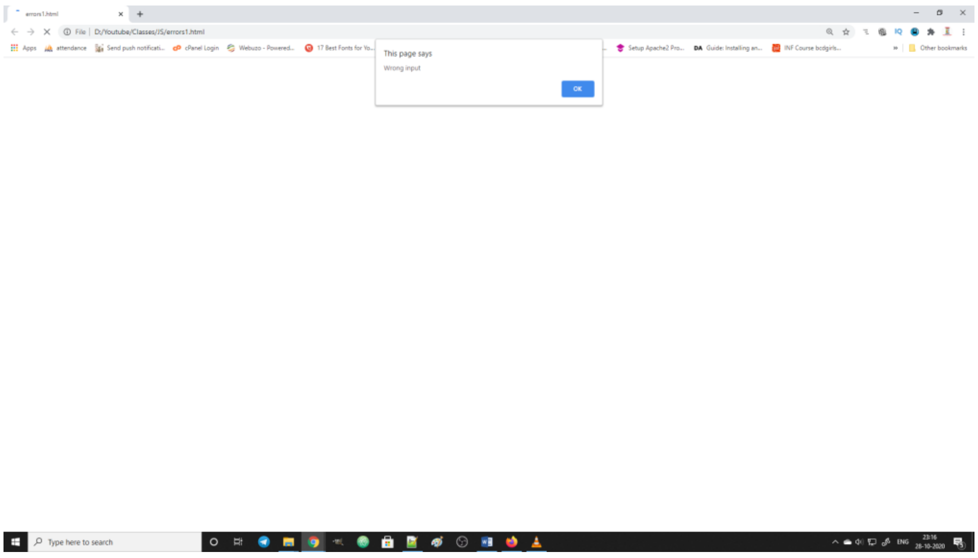
D:/Youtube/Classes/JS/errors1.html

This page says

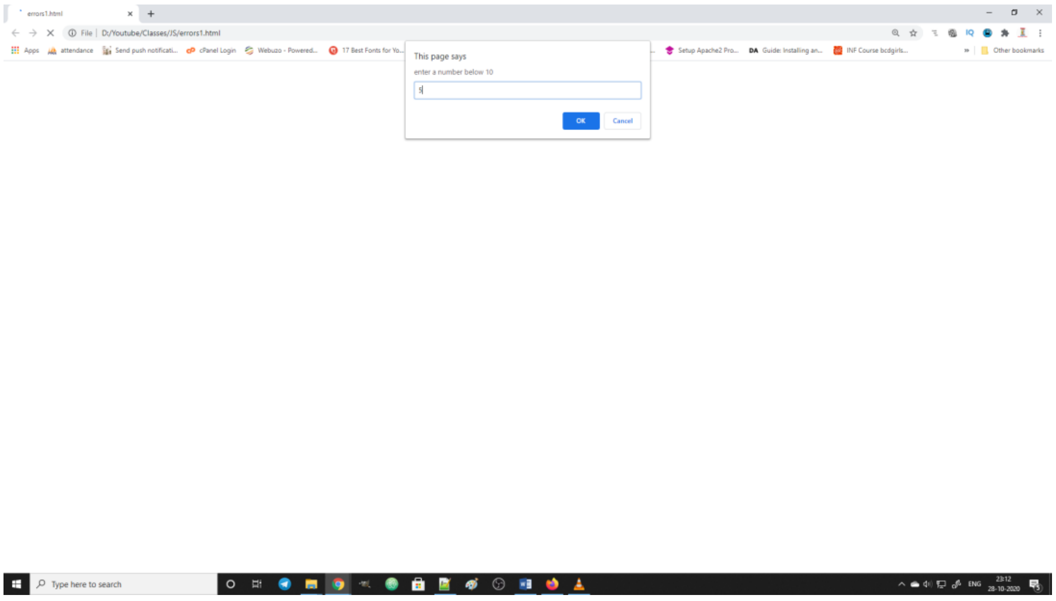
enter a number below 10

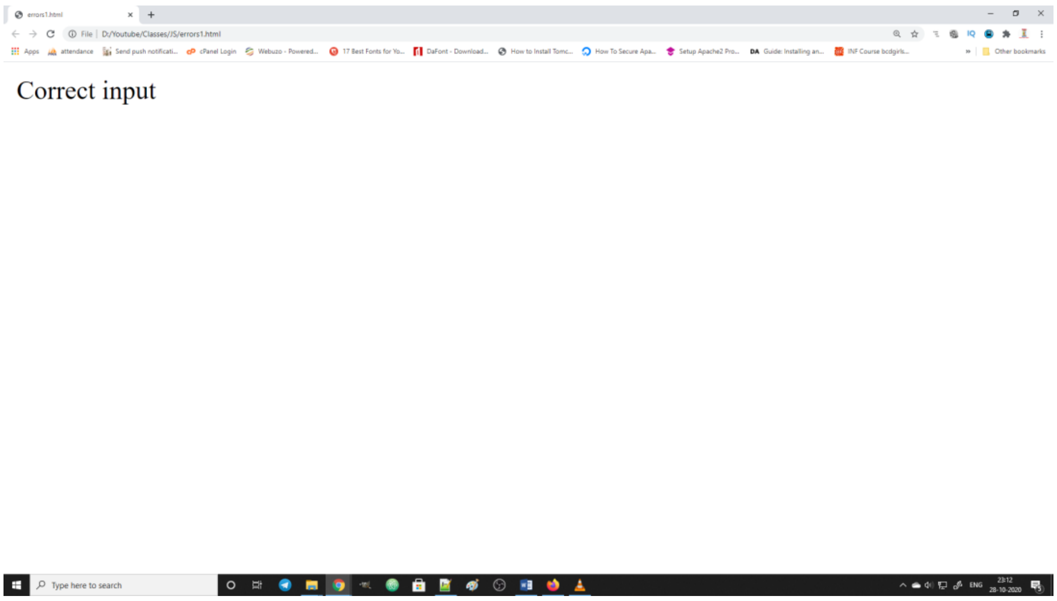
10

OK Cancel


This page says
Wrong input

Now we give the correct output also (below 10) value and check the output


This page says
enter a number below 10
5



Correct input

As we entered correct input (below 10) here no error is generated so program gets executed successfully.

This is how we can handle errors in java script by using try-catch-finally-throw keywords.