**Number Object:**

The Number object in JavaScript is a built-in object that represents number. It can be used to work with numbers in JavaScript. To create a Number object, we can use the Number() constructor. The Number() constructor takes a number as its argument. For example

```
var num= new Number(10);
```

**Methods of Number object:**

**toString():** Converts a number to a string.

**toFixed():** Returns the value of the number object as a string with a specified number of decimal place.

**toPrecision():** Returns the value of the number object as a string with a specified length.

**toExponential():** Returns the value of the number object as a string in exponential notation.

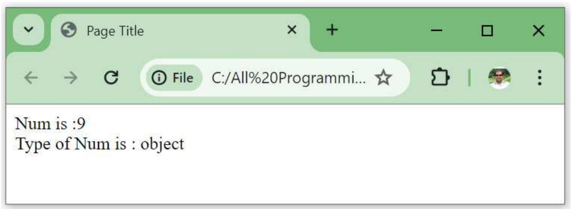
**parseInt():** Parses a string and returns an integer.

**parseFloat():** Parses a string and returns a floating point number.

**Example:**

```
<script>
        var num = new Number(9);
        document.write("Num is :" + num + "<br>");
        document.write("Type of Num is : " + typeof num);
</script>
```

**Output:**

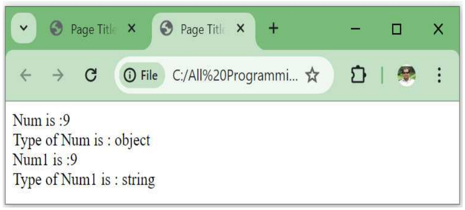


```
Num is :9
Type of Num is : object
```

**Example of toString():**

```
<script>
    var num= new Number(9);
    document.write("Num is :"+num+"<br>");
    document.write("Type of Num is : "+typeof num+"<br>");

    var num1=num.toString();
    document.write("Num1 is :"+num1+"<br>");
    document.write("Type of Num1 is : "+typeof num1);
  </script>
```

**Output:**

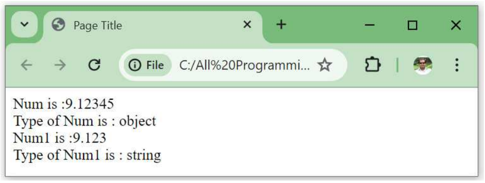


```
Num is :9
Type of Num is : object
Num1 is :9
Type of Num1 is : string
```

**Example of toFixed():**

```
<script>
    var num= new Number(9.12345);
    document.write("Num is :"+num+"<br>");
    document.write("Type of Num is : "+typeof num+"<br>");

    var num1=num.toFixed(3);
    document.write("Num1 is :"+num1+"<br>");
    document.write("Type of Num1 is : "+typeof num1);
```

**Output:**



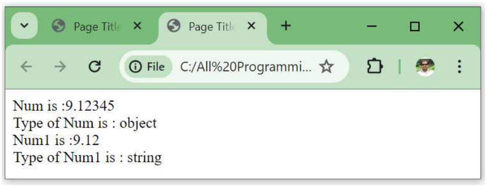
```
Num is :9.12345
Type of Num is : object
Num1 is :9.123
Type of Num1 is : string
```

```
</script>
```

**Example for toPrecision():**

```
<script>
        var num= new Number(9.12345);
        document.write("Num is :"+num+"<br>");
        document.write("Type of Num is : "+typeof num+"<br>");

        var num1=num.toPrecision(3);
        document.write("Num1 is :"+num1+"<br>");
        document.write("Type of Num1 is : "+typeof num1);
</script>
```

**Output:**

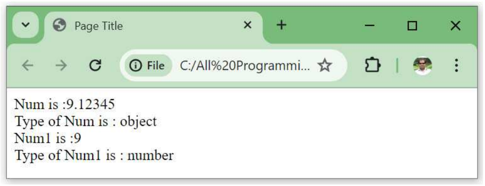


```
Num is :9.12345
Type of Num is : object
Num1 is :9.12
Type of Num1 is : string
```

**Example of parseInt():**

```
<script>
        var num= new Number(9.12345);
        document.write("Num is :"+num+"<br>");
        document.write("Type of Num is : "+typeof num+"<br>");

        var num1=parseInt(num);
        document.write("Num1 is :"+num1+"<br>");
        document.write("Type of Num1 is : "+typeof num1);
    </script>
```

**Output:**

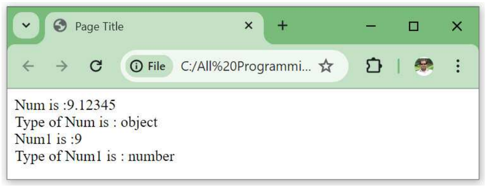


```
Num is :9.12345
Type of Num is : object
Num1 is :9
Type of Num1 is : number
```

**Example of parseFloat():**

```
<script>
        var num= new Number(9.12345);
        document.write("Num is :"+num+"<br>");
        document.write("Type of Num is : "+typeof num+"<br>");

        var num3=num.toString();
        var num1=parseFloat(num3);
        document.write("Num1 is :"+num3+"<br>");
        document.write("Type of Num1 is : "+typeof num1);
</script>
```

**Output:**



```
Num is :9.12345
Type of Num is : object
Num1 is :9
Type of Num1 is : number
```

**Boolean object:**

Boolean is an object that represents value in two value: *true or false*. You can create the JavaScript Boolean object by Boolean() constructor. For example

```
var obj = new Boolean(true);

var obj = new Boolean(false);
```

**Methods of Boolean:**

**toString():** converts Boolean into String.

**valueOf():** converts other type into Boolean.

**Math Object:**

Math is a built-in object which includes properties and methods for mathematical operations.

**Methods of Math object methods:**

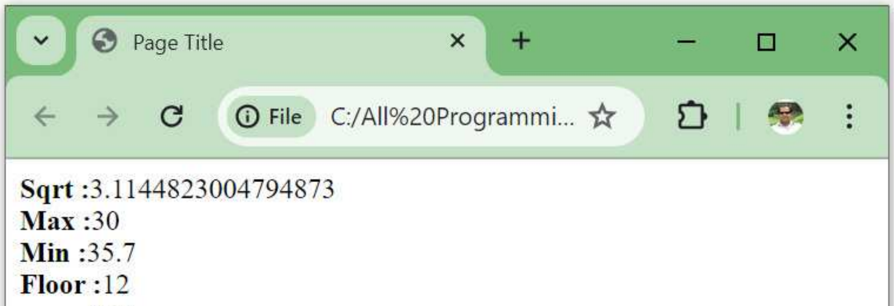
We have listed the most commonly used Math object methods with description.

**abs(x):** returns the absolute value of x
**ceil(x):** rounds up x to a nearest biggest integer
**cos(x):** returns the cosine value of x
**exp(x):** returns the value of the exponent.
**max(x,y,z,......n):** returns the highest number from the lsit.
**min(x,y,z,.......n):** returns the lowest number from the list.
**pow(x,y):** returns x to the power of y
**sqrt(x):** returns the square root of x
**random():** returns a random number between 0 and 1
**tan(x):** returns the tangent value of x
**sin(x):** returns the sine value of x.
**floor(x):** rounds up x to the nearest smallest integer.
**cos(x):** returns the cosine value of x
**log(x):**returns the logarithmic value of x

**For example:**

```
<script>
            document.write("<b>"+"Sqrt :"+"</b>"+Math.sqrt(9.7)+"<br>");
            document.write("<b>"+"Max :"+"</b>"+Math.max(9.7,10,25,30)+"<br>");
            document.write("<b>"+"Min :"+"</b>"+Math.min(99,58,90,35.7)+"<br>");
            document.write("<b>"+"Floor :"+"</b>"+Math.floor(12.7)+"<br>");
            document.write("<b>"+"pow :"+"</b>"+Math.pow(10,2)+"<br>");
            document.write("<b>"+"Log :"+"</b>"+Math.log(9.7)+"<br>");
            document.write("<b>"+"Random :"+"</b>"+Math.random()+"<br>");
            document.write("<b>"+"Round :"+"</b>"+Math.round(9.7)+"<br>");
            document.write("<b>"+"Sin:"+"</b>"+Math.sin(45)+"<br>");
            document.write("<b>"+"Exp:"+"</b>"+Math.exp(10)+"<br>");
            document.write("<b>"+"Tan:"+"</b>"+Math.tan(90)+"<br>");
 </script>
```

**Output:**



```
Sqrt :3.1144823004794873
Max :30
Min :35.7
Floor :12
```

**Date Object:**

Date object is a built-in object which is used to deal with date and time. You can use different Date constructors to create date object. It provides methods to get and set day, month, year, hour, minute and seconds.

**Date Object Methods:**

We have listed the most commonly used Date object methods with description.

**setDate():** sets the date of the month that ranges from 1 to 31.

**setHours():** set hours that range from 0 to 23

**getSeconds():** returns the seconds that range from 0 to 59

**toDateString()**:converts a date value into a string

**valueOf():** returns the primitive value of Date object.

**setMinutes():** set minutes that range from 0 to 59.

**setMonth():** set numerical equivalence of month range from 0 to 11

**setMilliseconds():** set the milliseconds that range from 0 to 999

**toTimeString():** converts time into a string.

**getDate():** returns the day of the month from 1 - 31

**getDay()**: returns the day of the week from 0 - 6

**getFullYear():** returns the year (4 digits for 4-digit years) of the specified date

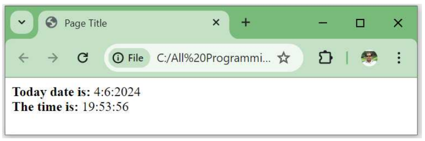
**getMinutes():** returns the minutes (0–59) in the specified date

**getMonth():** returns the month (0–11) in the specified date

**getHours():** returns the hours that that range from 0 to 23.

**For example:**

```
<script>
        var objdate = new Date();
        document.write("<b>" + "Today date is: " + "</b>" + objdate.getDate()
                                    + ":" + (objdate.getMonth() + 1) + ":" + objdate.getFullYear()
                                    + "<br>");

        document.write("<b>" + "The time is: " + "</b>" + objdate.getHours()
                                    + ":" + objdate.getMinutes() + ":" + objdate.getSeconds()
                                    + "<br>");
</script>
```



**Today date is:** 4:6:2024
**The time is:** 19:53:56

**Regular Expression:**

A regular expression is an object that describes a pattern of characters. Regular expressions are one of the most powerful tools available today for effective and efficient text processing and manipulations. For example, it can be used to verify whether the format of data i.e. name, email, phone number, etc. entered by the user is correct or not, find or replace matching string within text content, and so on.

**Syntax:**

/pattern/ Regular Expression Literal

**String methods which use regular expression**

**search():** Search for a match within a string. It returns the index of the first match, or -1 if not found.

**replace():** Search for a match in a string, and replaces the matched substring with a replacement string.

| RegExp | Describe |
|---|---|
| [abc] | Matches any one of the characters a, b, or c. |
| [^abc] | Matches any one character other than a, b, or c. |
| [a-z] | Matches any one character from lowercase a to lowercase z. |
| [A-Z] | Matches any one character from uppercase a to uppercase z. |
| [a-Z] | Matches any one character from lowercase a to uppercase Z. |

| | |
|---|---|
| [0-9] | Matches a single digit between 0 and 9. |
| [a-z0-9] | Matches a single character between a and z or between 0 and 9. |

**Modifier:**

A pattern modifier allows you to control the way a pattern match is handled. Pattern modifiers are placed directly after the regular expression, for example, if you want to search for a pattern in a case-insensitive manner, you can use the i modifier, like this: /pattern/i. lists some of the most commonly used pattern modifiers.

**g:** Perform a global match i.e. finds all occurrences.

**i:** Makes the match case-insensitive manner.

**m:** Perform multiline matching.

**Meta character (Metacharacters have special meaning)**

. : Matches any single character except newline \n.

\d: matches any digit character. Same as [0-9]

\D: Matches any non-digit character. Same as [^0-9]

\s: Matches any whitespace character (space, tab, newline or carriage return character).Same as [ \t\n\r]

\S: Matches any non-whitespace character. Same as [^ \t\n\r]

\w: Matches any word character (defined as a to z, A to Z,0 to 9, and the underscore).Same as [a-zA-Z_0-9]

\W: Matches any non-word character. Same as [^a-zA-Z_0-9]

**JavaScript's built-in methods for performing pattern-matching regular expressions.**

| **Function** | **Description** |
|---|---|
| exec() | Search for a match in a string. It returns an array of information or null on mismatch. |
| test() | Test whether a string matches a pattern. It returns true or false. |
| search() | Search for a match within a string. It returns the index of the first match, or -1 if not found. |
| replace() | Search for a match in a string, and replaces the matched substring with a replacement string. |
| match() | Search for a match in a string. It returns an array of information or null on mismatch. |
| split() | Splits up a string into an array of substrings using a regular expression. |

**Example of Regular Expression:**

```
<html>
<head>
   <title>Page Title</title>
</head>
<body>
   <script>
      var pattern=/[a-z][0-9]/i;
```

```
    var str="Hello123Welcome";
    document.write(pattern.exec(str));
    document.write("<br>")
</script>

<script>
   var pattern=/[a-z]+[0-9]+/ig;
    var str="Hello123Welcome";
    document.write(pattern.test(str));
    document.write("<br>")
</script>

<script>
    var pattern=/[a-z]+[0-9]+/ig;
    var str="Hello123Welcome";
    document.write(pattern.exec(str));
    document.write("<br>")
</script>

<script>
   var pattern=new RegExp('[a-z][A-Z][0-9]','ig');
   document.write(pattern.toString());

</script>

</body>
</html>
```

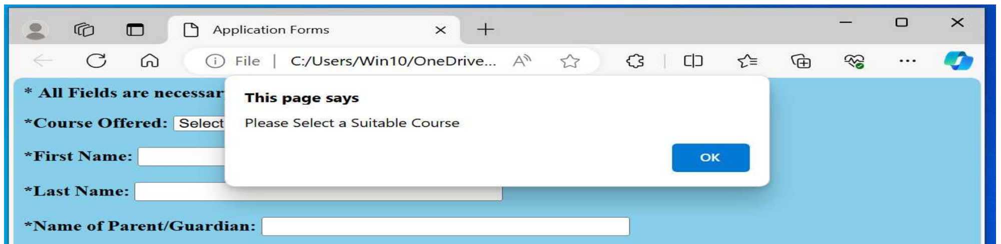
**Example of Validate Form:**

```
<html>
<head>
<title>Application Forms</title>
</head>
<body bgcolor="skyblue">

<script>
        function validatefun()
        {
        if(document.formname.course.value == "")
                {
                alert("Please Select a Suitable Course")
                document.formname.course.focus();
                 return false;
                }
        if(document.formname.fname.value == "")
                {
                alert("Please Enter Your First Name")
```

```
            document.formname.fname.focus();
             return false;
            }
if(document.formname.lname.value == "")
           {
            alert("Please Enter Your Last Name")
            document.formname.lname.focus();
             return false;
            }
if(document.formname.pname.value == "")
           {
            alert("Please Enter Your parents/Guardian's  Name")
            document.formname.pname.focus();
             return false;
            }
if(document.formname.gender.value == "")
           {
            alert("please Enter Your gender")
             return false;
            }
if(document.formname.dob.value == "")
           {
            alert("Please Enter Your Date of Birth")
            document.formname.dob.focus();
             return false;
            }
if(document.formname.nationality.value == "")
           {
            alert("Please Select a Suitable nationality")
            document.formname.nationality.focus();
             return false; }
if(document.formname.address.value == "")
           {
            alert("Please Enter your address")
            document.formname.address.focus();
            document.formname.address.select();
            return false;
            }
if ((document.formname.email.value.indexOf('@')< 1) ||

    (document.formname.email.value.lastIndexOf('.') <=
document.formname.email.value.indexOf('@')+1) ||
    (document.formname.email.value.lastIndexOf('.')  == document.formname.email.value.length -
1 ) ||
    (document.formname.email.value.indexOf(' ')!= -1))
                  {
                        alert("Please Enter Your E-mail Id ");
                        document.formname.email.focus();
                        return false;
                  }
}
```

```
</script>
        <form name="formname" onSubmit="return validatefun();">
                <p>
                        <strong>All Fields are necessary
                </p>
                <strong>Course Offered: <select name="course" size="1">
                                <option selected value>Select a course</option>
                                <option value="C.S.I.T">Bachelor of Engineering in Computer
                                        Science and Information Technology.</option>
                                <option value="B.C.A">Bachelor of Computer Applications.</option>
                                <option value="AME">AME</option>
                                <option value="D.C.S">Diploma in Computer Science Engg.</option>
                                <option value="D.E.C">Diploma in Electronics& Communication
                                        Engg.</option>
                                <option value="D.A.E">Diploma in Auotomobile Engg.</option>
                                <option value="D.M.E">Diploma in Mech. Engg.</option>
                                <option value="BCA">Bachelor of Computer Application</option>
                                <option value="B.Sc Computer">B.Sc Computer Science</option>
                                <option value="BBM">Bachelor of Business Management</option>
                                <option value="BHM">Bachelor of Hotel Management</option>
                </select> <br>
                <br><strong>First Name:</strong> <input type="text"
                        name="fname" size="40"><br>
                <br><strong>Last Name:</strong> <input type="text" name="lname"
                        size="40"><br>
                <br><strong>Name of Parent/Guardian:</strong> <input
                        type="text" name="pname" size="40"><br>
                <br><strong>Gender:</strong> <input type="radio" value="male"
                        name="gender">Male <input type="radio" value="female"
                        name="gender">Female <br>
                <br><strong>Date of Birth:</strong> <input type="text"
                        name="dob" size="40"><br>
                <br><strong>Nationality:</strong> <select name="nationality"
                        size="1">
                                <option selected value="Select nationality">Select
                                        nationality</option>
                                <option value="Nepal">Nepal</option>
                                <option value="Indian">Indian</option>
                                <option value="US">US</option>
                                <option value="Others">Others</option>
                </select><br>
                <br><strong>Permanent Address:</strong> <textarea rows="4"
                                name="address" cols="34"></textarea><br>
                <br><strong>Email:</strong> <input type="text" name="email"
                        size="40"><br>
                <br> <input type="submit" value="Submit" name="Submit"> <input
                        type="reset" value="Reset" name="Reset">
        </form>
</body>
</html>
```

**Output:**



Application Forms

C:/Users/Win10/OneDrive...

* All Fields are necessar

*Course Offered: Select

*First Name:

*Last Name:

*Name of Parent/Guardian:

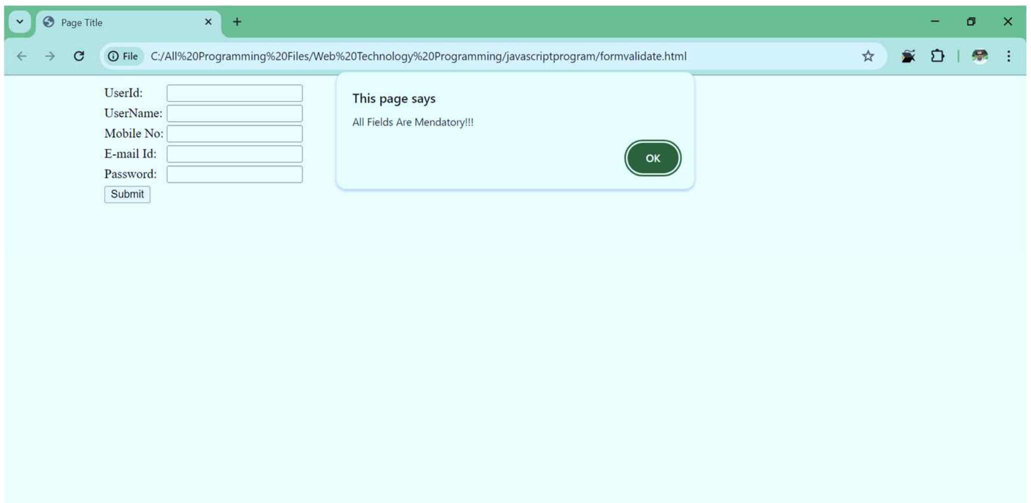
This page says

Please Select a Suitable Course

OK

```
<html>
<head>
<title>Page Title</title>
</head>
<body>
          <script>
      function validateForms()
      {
         var userid = document.getElementById("txtuserid").value;
         var username = document.getElementById("txtusername").value;
         var mobileno = document.getElementById("txtmobileno").value;
         var email = document.getElementById("txtemailid").value;
         var password = document.getElementById("txtpassword").value;

         if(userid==""||username==""||mobileno==""||email==""||password=="")
          {
            alert("All Fields Are Mendatory!!!");
            return false;
         }
          else if(mobileno.length<10 || mobileno.length>10){
                alert("Mobile Number Should be of 10 Digits!!!");
                return false;
            }
            else if(isNaN(mobileno)){
                alert("Only Numbers are allowed!!");
            }
            else
            {
                return true;
            }
        }
    </script>
        <form onsubmit="return validateForms()" action="welcome.html">
            <table align="center">
                <tr>
                    <td>UserId:</td>
                    <td><input type="text" id="txtuserid"></td>
                </tr>
                <tr>
                    <td>UserName:</td>
                    <td><input type="text" id="txtusername"></td>
                </tr>
                <tr>
                    <td>Mobile No:</td>
                    <td><input type="text" id="txtmobileno"></td>
                </tr>
                <tr>
                    <td>E-mail Id:</td>
                    <td><input type="text" id="txtemailid"></td>
                </tr>

                <tr>
                    <td>Password:</td>
                    <td><input type="password" id="txtpassword"></td>
                </tr>
                <tr>
                    <td><input type="submit" value="Submit"></td>
                </tr>
            </table>
        </form>
</body>
</html>
```

**Output:**

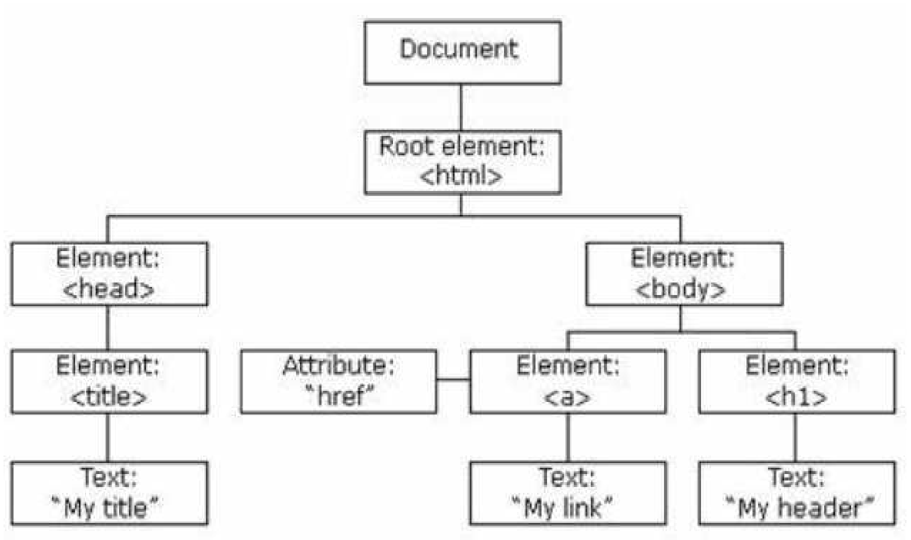


Page Title
C:/All%20Programming%20Files/Web%20Technology%20Programming/javascriptprogram/formvalidate.html
UserId:
UserName:
Mobile No:
E-mail Id:
Password:
Submit
OK

This page says
All Fields Are Mendatory!!!

**DOM (Document Object Model):**

DOM stands for **Document Object Model**. The DOM is a representation of the elements on a web page that the browser creates when it loads an HTML document. Each HTML element in the document, such as paragraphs, headings, images, etc, becomes a node in the DOM tree. These nodes can then be manipulated through JavaScript to change the content, style, and structure of the page dynamically. Here is a diagram of the HTML DOM element layout.



**For example:**

```
<html>
<head>
   <title>Page Title</title>
</head>
<body>
   <h1 id="hello">Hello</h1>
   <p class="good">Good Morning</p>
   <script src='reg-expr.js'></script>
</body>
</html>
```

**reg-expr.js**

```
document.getElementById("hello").innerText="Hello Guys, How are you?";
document.getElementsByClassName("good").innerText="Good Afternoon";
```

**Common DOM Manipulations**

- Accessing Elements: Developers can access elements in the DOM using methods like getElementById, getElementsByClassName, getElementsByTagName, querySelector, and querySelectorAll.
- Modifying Content: The content of an element can be changed using properties like textContent, innerHTML, and innerText.
- Changing Styles: Developers can modify the style of elements by accessing the style property of an element and changing its CSS properties.
- Creating and Appending Elements: New elements can be created using methods like createElement, and they can be added to the document using appendChild or insertBefore.
- Event Handling: Developers can listen for events like clicks, keypresses, mouse movements, etc., and respond to them by attaching event listeners to elements.

**Date Object in JavaScript:**

The Date object is used to work with dates and times. Date objects are created with the Date( ) constructor. We can easily manipulate the date by using the methods available for the Date object.

**Methods:**

getDate() Returns the day of the month (from 1-31)

getDay() Returns the day of the week (from 0-6)

getFullYear() Returns the year (four digits)

getHours() Returns the hour (from 0-23)

getMilliseconds() Returns the milliseconds (from 0-999)

getMinutes() Returns the minutes (from 0-59)

getMonth() Returns the month (from 0-11)

getSeconds() Returns the seconds (from 0-59) setDate() Sets the day of the month (from 1-31)

setFullYear() Sets the year (four digits)

setHours() Sets the hour (from 0-23)

setMilliseconds() Sets the milliseconds (from 0-999)

setMinutes() Set the minutes (from 0-59)

setMonth() Sets the month (from 0-11)

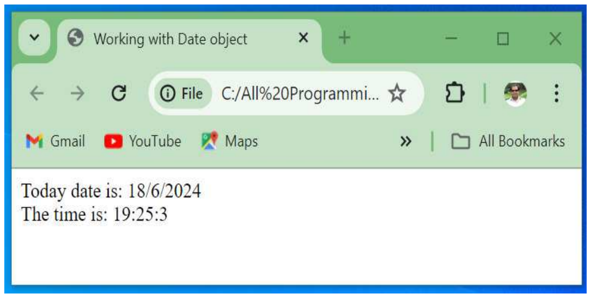
setSeconds() Sets the seconds (from 0-59)

toString() Converts a Date object to a string

**For example:**

```
<html>
<head>
<title>Working with Date object</title>
</head>
<body>
         <script>
        var mydate = new Date();
        document.write("Today date is: " +
mydate.getDate()+"/"+(mydate.getMonth()+1)+"/"+mydate.getFullYear()+"<br/>");
        document.write("The time is: " +
mydate.getHours()+":"+mydate.getMinutes()+":"+mydate.getSeconds()+"<br/>");
  </script>
</body>
</html>
```

**Output:**



```
Today date is: 18/6/2024
The time is: 19:25:3
```

**Math:**

Math is built-in object which includes properties and methods for mathematical operation. All the properties and methods are static. So, we don't need to create its object to use its property or method. Also, even if we want, we cannot create an object as **Math is not a constructor** function.

**Math Methods:**

| Method | Description |
|---|---|
| abs(x) | returns the absolute value of x |
| ceil(x) | rounds up x to a nearest biggest integer |
| cos(x) | returns the cosine value of x |
| exp(x) | returns the value of the exponent. |

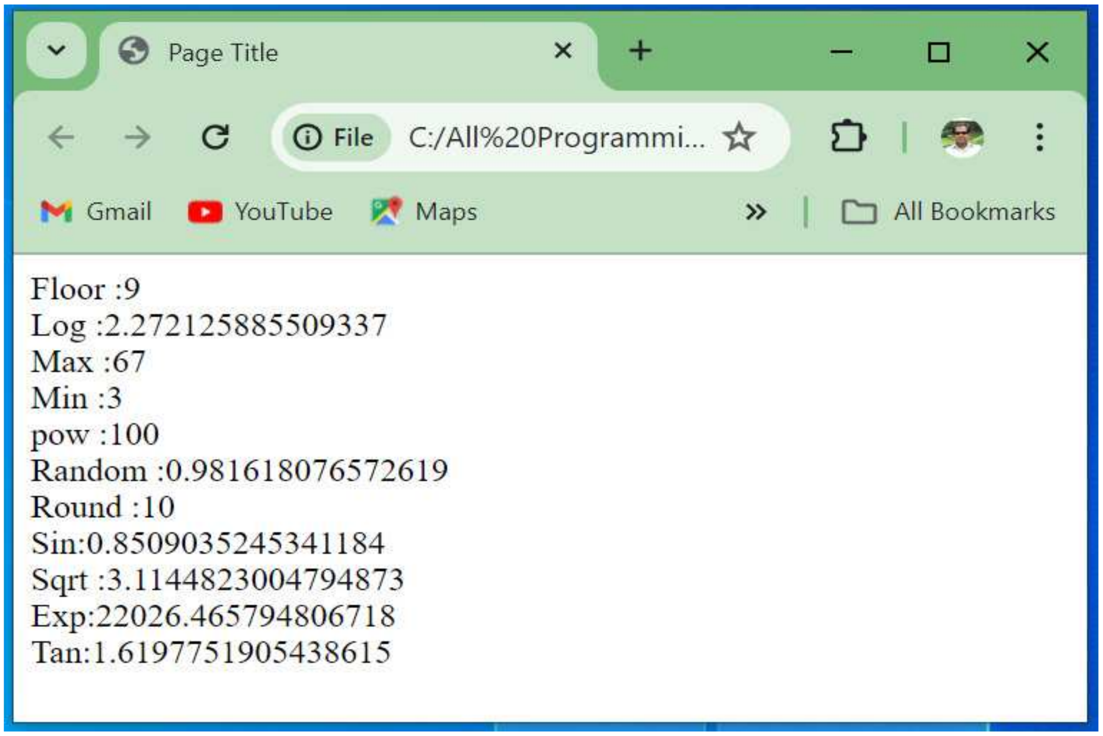
| Method | Description |
|---|---|
| random() | returns a random number between 0 and 1 |
| tan(x) | returns the tangent value of x |
| sqrt(x) | returns the square root of x |
| sin(x) | returns the sine value of x. |
| floor(x) | rounds up x to the nearest smallest integer. |
| max(x,y,z,......n) | returns the highest number from the lsit. |
| min(x,y,z,.......n) | returns the lowest number from the list. |
| pow(x,y) | returns x to the power of y |
| cos(x) | returns the cosine value of x |
| log(x) | returns the logarithmic value of x |

**For example:**

```
<html>
<head>
<title>Page Title</title>
</head>
<body>
        <script>
    document.write("Floor :"+Math.floor(9.7)+"<BR/>");
    document.write("Log :"+Math.log(9.7)+"<BR/>");
    document.write("Max :"+Math.max(9.7,11,25,67)+"<BR/>");
    document.write("Min :"+Math.min(3,78,90,12.7)+"<BR/>");
    document.write("pow :"+Math.pow(10,2)+"<BR/>");
    document.write("Random :"+Math.random()+"<BR/>");
    document.write("Round :"+Math.round(9.7)+"<BR/>");
```

```
            document.write("Sin:"+Math.sin(45)+"<BR/>");
            document.write("Sqrt :"+Math.sqrt(9.7)+"<BR/>");
            document.write("Exp:"+Math.exp(10)+"<BR/>");
            document.write("Tan:"+Math.tan(45)+"<BR/>");
        </script>
    </body>
</html>
```

**Output:**



```
Floor :9
Log :2.272125885509337
Max :67
Min :3
pow :100
Random :0.981618076572619
Round :10
Sin:0.8509035245341184
Sqrt :3.1144823004794873
Exp:22026.465794806718
Tan:1.6197751905438615
```

**Error Handling:**

**There are three types of errors in programming.**

1) Syntax Error
2) Runtime Error
3) Logical Error

**Syntax Error**: syntax error occurs when a user makes a mistake in the pre-defined syntax of a programming language, a syntax error may appear. For example.

```
<script>
        document.write("Hello")
</script>
```

**Runtime Error:** Runtime error is any error that causes abnormal program termination during execution. For example.

```
<script>
        var a = 10;
        var b = 10 / 0;
        document.write(b);
</script>
```

**Logical Error**: A logical error simply an incorrect translation of either the problem statement or the algorithm. For example.

```
<script>
        var i = 1;
        for (i = 0; i < 5; i++);
        {
                document.write(i);
        }
</script>
```

**try, catch and finally statement**

An Exception is an error that occurs at the time of execution (runtime) due to an illegal operation when a program is **syntactically correct**. So Exception or Error handling begins by identifying the code which we think may give some error/exception when we run it. Such code blocks can be enclosed within a special try block which is a special code block in which if any runtime exception occurs, then we can handle that exception in the catch block.

**Syntax try and catch:**

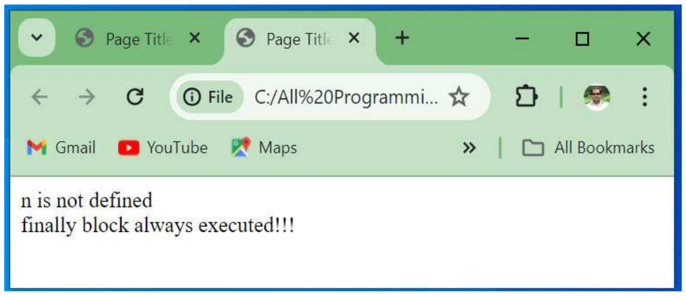
```
try{
        expression;
        }
        catch(error)
        {
                expression;
        }
        finally
        {
                expression;
        }
```

**For example:**

```
<html>
<head>
  <title>Page Title</title>
</head>
<body>
  <script>
     try
        {

          document.write(n);
```

```
            }
            catch(err)
            {
              document.write(err.message);
            }
            finally
            {
              document.write("<br/> finally block always executed!!!");
            }
    </script>
</body>
</html>
```

**Output:**



```
n is not defined
finally block always executed!!!
```

**Handling Cookies:**

A cookie is a small piece of information as a text file stored on client's computer machine by a web application. The maximum file size of a cookie is 4KB. It stores only the "String" data type. A cookie has a name, a single value and optional attributes such as comment, path and domain qualifiers, a maximum age, and a version number. The information stored within cookies is not secure because this information is stored in text-format on the client-side, which can be read by anyone.

For example:

```
<html>
<head>
<script>
        function createCookie() {
          if( document.myform.uname.value == "" ) {
            alert("Enter Any Name!");
            return;
          }
          var now = new Date();
           now.setMonth(now.getMonth() + 1);
          cookievalue = escape(document.myform.uname.value);
          document.cookie = "name=" + cookievalue;
          document.write ("Setting Cookies : " + "name=" + cookievalue );
          document.cookie = "expires=" + now.toUTCString();

           alert("Cookie is Created........");
           alert("Value of Cookie : " + cookievalue);
           alert("Expiry Date is : " + now);
          }
    </script>
</head>
<body>
         <form name="myform" action="">
                  Enter Any Name: <input type="text" name="uname">
                  <input type="button" value="Set Cookie" onclick="createCookie();" >
         </form>
</body>
</html>
```

**Example of Read Cookie:**

```
<html>
  <head>
    <script>
      function readCookie() {
        var allcookies = document.cookie;
        document.write ("All Cookies : " + allcookies );

        // Get all the cookies pairs in an array
        cookiearray = allcookies.split(';');

        for(var i=0; i<cookiearray.length; i++) {
          name = cookiearray[i].split('=')[0];
```

```
            value = cookiearray[i].split('=')[1];
            document.write ("Key is : " + name + " and Value is : " + value);
         }
      }
   </script>
</head>

<body>
   <form name = "myform" action = "">
      <input type = "button" value = "Get Cookie" onclick = "readCookie()"/>
   </form>
</body>
</html>
```

**jQuery:**

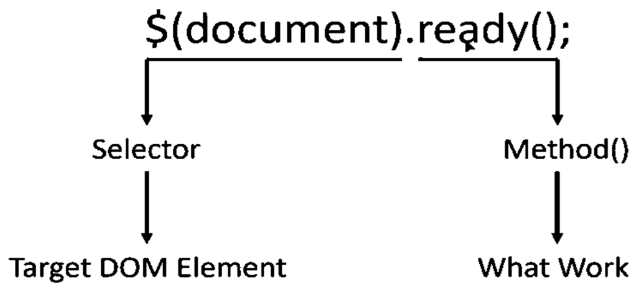
jQuery is a fast, small, and feature-rich JavaScript library developed by John Resig in August 26, 2006. It is Open-source JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript.

**Syntax of jQuery:**

$(selector).action()

- $ sign to define/access jQuery.
- Selector to "query or find" HTML elements.

- Action() to be performed on the elements.



```
document.getElementById('idName');
                    $('#idName');

document.getElementsByClassName('className');
                    $('.className')
```

**For example:**

**$(this).hide()** :hides the current element.
**$("p").hide()** : hedes all <p> elements.
**$(".test").hide()**: hides all elements **with class**="test".
**$("#test").hide():** hides the elements **with** id=test.

https://jquery.com

```
<html>
<head>
<title>Page Title</title>
<script src="jquery-migrate-1.4.1.min.js"></script>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>

</head>
<body>
        <script>
                $(document).ready(function(){
                $("p").click(function(){
                 $(this).hide();
                 });
                });
          </script>
           <p>Wecome to jQuery</p>
           <p>Click me away!</p>
</body>
</html>
```

==========================================

```
<body>
           <script>
                      $(document).ready(function() {
                                 $("p").click(function() {
                                            $(this).css("color", "red");
                                 });
                      });
           </script>
           <p>Wecome to jQuery</p>
           <p>Click me away!</p>
</body>
```

=========================================

```
<body>
           <script>
                      $(document).ready(function() {
                                 $("p").click(function() {
                                            $(this).append("Hello World!");
                                 });
                      });
           </script>
           <p>Wecome to jQuery</p>
           <p>Click me away!</p>
</body>
```

============================================

```
<body>
           <script>
                      $(document).ready(function() {
                                 $("p").click(function() {
                                            $(this).remove();
                                 });
                      });
           </script>
           <p>Wecome to jQuery</p>
           <p>Click me away!</p>
</body>
```

**jQuery Events:**

Events are often triggered by the users interaction with the web page, such as when a link or button is clicked, text is entered into an input box or textarea, selection is made in a select box, key is pressed on the keyboard, the mouse pointer is moved etc.

**Mouse Event:**

- click()
- dbclick()
- hover()
- mouseenter()
- mouseleave()

**Keyboard Event:**

- keypress()
- Keydown()
- Keyup()
- Mouseenter()
- Mouseleave()

**Form Event:**

- Change()
- Focus()
- Blur()
- Submit()
- Mouseleave()

**Document/Window Event**

- Change()
- Focus()
- Blur()
- Submit()
- Mouseleave()

**For example:**

```
<html>
<head>
<title>Page Title</title>
<script src="jquery-migrate-1.4.1.min.js"></script>
<script
        src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>

</head>
<body>
        <script>
```

```
                $(document).ready(function() {
                        $("p").click(function() {
                                alert("Hello jQuery");
                        });
                });
        </script>
        <p>Click me away!</p>
</body>
</html>
```

**JSON:**

JavaScript Object Notation (JSON) is a way of storing information in an organized and easy manner. The data must be in the form of a text when exchanging between a browser and a server. You can convert any JavaScript object into JSON and send JSON to the server. In JSON, data is represented in **key-value pairs**, and **curly braces** hold objects, where a colon is followed after each name. The comma is used to separate **key-value pairs**. Square brackets are used to hold arrays, where each value is comma-separated.

**Syntax of JSON:**

```
{ "name":"Nawaraj" }
```

**Data types:**

The most commonly used JSON data types are

- String
- Number
- Object (JSON object)
- Arrays
- Boolean
- Null

**String:** A string is always written in double-quotes. It may consist of numbers, alphanumeric and special characters.

**Number**: Number represents the numeric characters.

**Boolean**: It can be either true or false.

**Null:** It is an empty value.

**Objects:** JSON objects are surrounded by curly braces {}. They are written in key or value pairs in the following way:

```
{"name":"Nawaraj", "age":43}
```

Arrays in JSON are almost the same as arrays in JavaScript. In JSON, array values are of type string, number, object, array, boolean or null.

```
{
        "name":"Nawaraj",
        "age":43,
        "fruits":[ "Apple", "Mango", "Banana","Orange"]
}
```

**For example:**

```
<html>
<head>
<title>Page Title</title>
</head>
<body>
        <script>
                let student = {
                        name : "Nawaraj",
                        age : 43
                };

                let res = JSON.stringify(student);
                console.log(res);

                let obj = JSON.parse(student);
                console.log(obj);
        </script>
</body>
</html>
```

**Difference between json and xml:**

| JSON | XML |
|---|---|
| JSON is simple to read and write. | XML is less simple as compared to JSON. |
| It also supports **array**. | It doesn't support array. |
| JSON files are more **human-readable** than XML. | XML files are **less human readable**. |
| It supports only **text** and **number** data type | It supports many data types such as **text, number, images, charts, graphs**, etc. |