Flow Controls:

There are three types of non-looping conditionals, the if statement, else statement and switch statement.

If statement:

The if statement is used to check some given condition. If condition is true then the statement inside the body of if is executed. Otherwise if condition is false then the statement inside the body of if is skipped.

Syntax of if statement:

```
jf(condition)
{
        Block of statements;
        ……..
}
```

Example of if statement:

```
<html>
<head>
<title>Page Title</title>
</head>
<body>
    <script>
        var a = 10;
        if (a > 5)
        {
            document.write("Value of a is greater than 10");
        }
    </script>
</body>
</html>
```

Output:

Value of a is greater than 10

else statement:

The else statement is used to perform two operations for a single condition. The else statement evaluates the specified condition. If it is true, it executes a block of statements. If the condition is false, it executes another block of statements. We must notice that if and else block cannot be executed simultaneously.

Syntax of if else statement:

```
if(condition)
{
        true block of statements;
}
else
{
        false block of statements;
}
```

For example:

```
<html>
<head>

<title>Page Title</title>

</head>
<body>
        <script>
                var a = 10;
                if (a % 2 == 0)
                {
                        document.write("A is even number");
                }
                else
                {
                        document.write("A is odd number");
                }
        </script>
</body>
</html>
```

Output:

A is even number

if-else-if statement:

The if-else-if conditional Statement is used to execute one code from multiple conditions. It is also known as a multi-path decision statement. It is a sequence of if-else statements where every if statement is associated with else if Statement and last would be an else statement.

Syntax of if-else-if statement:

```
if(condition1)
{
        true block of statements1;
}
else if(condition2)
{
        false block of condition1;
and
        True block of condition2
}
```

For example:

```
<html>
<head>
<title>Page Title</title>
</head>
<body>
        <script>
                var a = 50;
                if (a == 10)
                {
                        document.write("A is equal to 10");
                }
                else if (a == 15)
                {
                        document.write("A is equal to 15");
                }
                else if (a == 50)
                {
                        document.write("A is equal to 50");
                }
                 else
                {
                        document.write("A is not equal to 10, 15 or 50");
                }
        </script>
</body>
</html>
```

Output:

A is equal to 50

switch...case Statement:

Switch statement is used to compares it with multiple cases. Once the case match is found, a block of statements associated with that particular case is executed. Each case in a block of a switch has a different choice which is referred to as an identifier. When the user enters a value, then the value provided by the user is compared with all the cases inside the switch block until the exact match is found. If a case match is not found, then the default statement is executed, and the control goes out of the switch block.

Syntax of switch...case:

```
switch (expression)
{
        case value1:
                //Statements
        break;
        case value 2:
                //Statements
        break;
        case value n:
                //Statements
        break;
        Default:
          //Statements
}
```

For example:

```
<html>
<head>
<title>Page Title</title>
</head>
<body>
    <script>
        var a = 5;

        switch (a) {
        case 1:
                document.write("Case 1 executed");
                break;
        case 2:
                document.write("Case 2 executed");
                break;
        case 3:
                document.write("Case 3 executed");
                break;
        case 4:
                document.write("Case 4 executed");
                break;
        default:
                document.write("Default case executed");
        }
    </script>
</body></html>
```

Output:

Default case executed

Looping Statements:

Looping statements repeatedly execute a statement or a block of statements.

- for loop
- while loop
- do-while loop
- for-in loop

for loop statement:

The for loop statement is used to execute a single statement or a block of statements repeatedly as long as the given condition is true.

Syntax of for loop statement:

```
for(initialization; condition; increment/decrement)
{
        statement-block;
}
```

- **Initialization:** First, a value is assigned to the variable.
- **Condition:** The following step is to check the condition; if it returns false, the for loop is ended. However, if the condition returns true, the statements inside the for loop's body are executed.
- **Increment/decrement:** In the last step, variable is incremented or decremented. Again the condition, statement inside body of loop and increment/decrement operation repeated until the condition expression is false and the loop terminates for the final time

For example:

```
<html>
<head>
<title>Page Title</title>
</head>
<body>
        <script>
                var i;
                for(i=0;i<5;i++)
                {
                        document.write("<br>",i);
                }
        </script>
</body>
</html>
```

Output:

0
1
2
3
4

while Loop statements:

When the condition is true, then the while loop's statements will be executed, otherwise not.

Syntax of while Loop statements:

```
variable initialization;
while(condition)
{
        statements;
        variable increment/decrement;
}
```

For example:

```
<html>
<head>
<title>Page Title</title>
</head>
<body>
        <script>
                var i = 5;
                while (i <= 10)
                {
                        document.write(i + "<br>");
                        i++;
                }
        </script>
</body>
</html>
```

Output:

5
6
7
8
9
10

do while loop statement:

do while loop will be executed at least once, even though the condition is false.

Syntax of do while loop:

```
do {
        // code to be executed
} while (condition);
```

For example:

```
<html>
<head>
<title>Page Title</title>
</head>
<body>
    <script>
        var i = 10;
        do {
            document.write(i + "<br>");
            i++;
        }
        while (i <= 5);
    </script>
</body>
</html>
```

for-in loop:

The for-in loop is a special type of a loop that iterates over the properties of an object, or the elements of an array.

Syntax of for-in loop:

```
for(variable in object)
{
    // Code to be executed
}
```

For example:

```
<html>
<head>
<title>Page Title</title>
</head>
<body>
    <script>
        var parson = {name : 'Nawaraj',surname : 'Prajapati',age : '42'};
        for ( var v in parson) {
            document.write(parson[v], "<br>");
        }
    </script>
</body>
</html>
```

Output:

Nawaraj
Prajapati
42

Functions:

A function is a group of reusable code, which can be called anywhere in program. The declaration of a function start with the **function keyword**, followed by the name of the function you want to create, followed by parentheses i.e. **()** and finally pace your function code between curly brackets **{}.**

Syntax of function:

```
function functionName()
{
        // Code to be executed
}
```

For example:

```
<html>
<head>
<title>Page Title</title>
</head>
<body>
    <script>
        function sum()
        {
            var a = 10, b = 10, res;
            res = a + b;
            document.write("Totla Sum :", res);
        }
        sum();
    </script>
    </body>
</html>
```

Parameters to Functions:

We can call function by passing arguments.

Syntax of parameters of function:

```
function functionName(parameter1, parameter2)
{
   // Code to be executed
}
```

For example:

```
<html>
<head>
<title>Page Title</title>
</head>
<body>
        <script>
                function sum(num1, num2) {
                        var res = num1 + num2;
                        document.write("Sum of two nos : ", res)
                }
                sum(6, 20);
        </script>
</body>
</html>
```

Popup Boxes:

JavaScript supports three important types of dialog boxes. Dialog boxes are also called as Popup Boxes.

- Alert box
- Confirm box
- Prompt box

Alert box:

An alert dialog box is mostly used to inform or alert the user by displaying some messages in a small dialogue box.

Syntax of alert box:

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

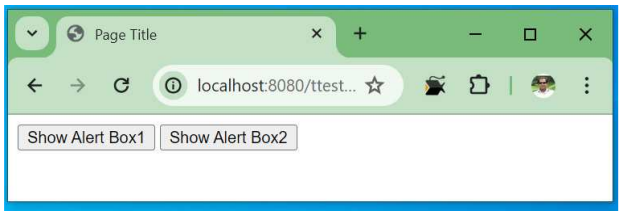
For example:

```
<html>
<head>
<title>Page Title</title>

</head>
<body>
	<script>
		alert("Click me to proceed!");
		document.write("Have a Nice day!!!")
	</script>
</body>
</html>
```

Another example:

```
<html>
<head>
<title>Page Title</title>
<script>
	function display() {
		alert("Hello alert box");
	}
	function show() {
		alert("Welcome to alert box");
	}
</script>
</head>
<body>
	<input type="button" value="Show Alert Box1" onclick="display()" />
	<input type="button" value="Show Alert Box2" onclick="show()" />
</body>
</html>
```

Output:



Confirm Box:

A confirmation dialog box is mostly used to take user's consent on any option. It displays a dialog box with two buttons: OK and Cancel. If the user clicks on the OK button, the confirm() will return true. If the user clicks on the Cancel button, then confirm() returns false.

Syntax of confirm dialog box:

```
confirm("message");
window.confirm("message");
```

For example:

```
<html>
<head>
<title>Page Title</title>
</head>
<body>
	<script>
		var a = confirm("Are you ready to confirm?");
		document.write("Your have clicked on : " + a);
	</script>
</body>
</html>
```

Another example:

```
<html>
<head>
<script>
	function isConfirmed()
	{
		var conValue = confirm("Are you ready to confirm?");

		if (conValue == true) {
			document.getElementById("result").innerHTML = "Confirmed !";
		}
		 else
		{
			document.getElementById("result").innerHTML = "Cancelled !";
		}
	}
</script>
</head>
<body>
	<form>
		<input type="button" onclick="isConfirmed()" value="Want to confirm ?" />
	</form>
	<p id="result"></p>
</body>
</html>
```

Prompt Box:

Prompt Box can be used when we want to get some user input. When Javascript displays a prompt box, the user will see a popup box with an input field and buttons "OK" or "Cancel" to proceed after entering an input value.

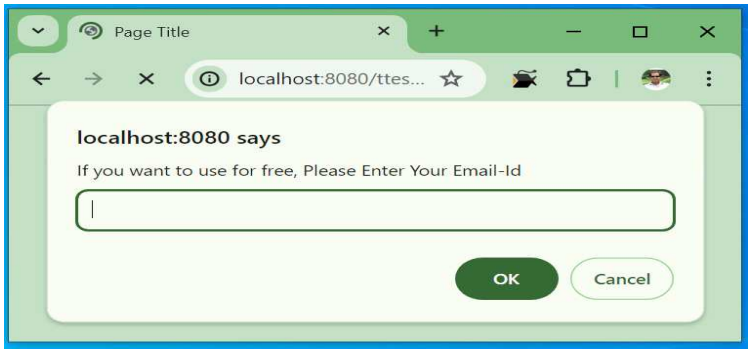
Syntax of prompt box:

```
prompt("Message","defaultValue");
window.prompt("sometext","defaultText");
```

For example:

```
<html>
<head>
<title>Page Title</title>
</head>
<body>
	<script>
		a = prompt("If you want to use for free, Please Enter Your Email-Id");
		document.write("Your have entered Email-id is : " + a);
	</script>
</body>
</html>
```

Output:



Objects and properties:

A JavaScript object is an entity having state and behavior (properties and method). JavaScript objects have a special property called prototype. The property of an object is a **key:value** pair, where **key** refers to a variable, and **value** refers to any type of value associated with the key. The value can be of any type like a number, string, or even an array, or another object.

We can create an object in JavaScript either by using the **constructor** function or the object **literal**.

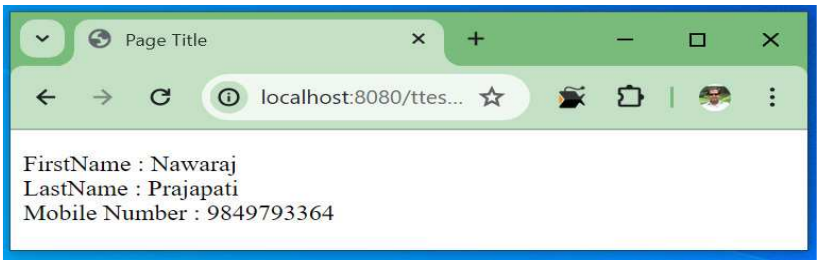
Creating a JavaScript Object:

```
//Using Object constructor
var obj1 = new Object();
let obj1 = new Object();

//Using the object literal syntax - empty object
var obj2 = {};
let obj2 = {};

//Using the object literal syntax - with properties
var obj3 = { key1:value1, key2:value2, ...};
```

Example of using object constructor:

```
<html>
<head>
<title>Page Title</title>
</head>
<body>
	<p id="objectDemo"></p>
	<script>
		function Person(firstName, lastName, mobile) {
			this.firstName = firstName;
			this.lastName = lastName;
			this.mobile = mobile;
		}
		const objectPerson = new Person("Nawaraj", "Prajapati", 9849793364);

		document.getElementById("objectDemo").innerHTML = "FirstName : "
				+ objectPerson.firstName + "<br>" + "LastName : "
				+ objectPerson.lastName + "<br>" + "Mobile Number : "
				+ objectPerson.mobile;
	</script>
</body>
</html>
```

Output:



Example of using object literal:

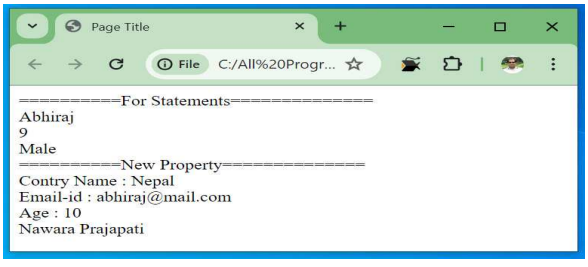
```
<html>
<head>
<title>Page Title</title>
</head>
<body>
        <script>
                let person = {
                        firstname : "Abhiraj",
                        lastname : "Prajapati",
                        mobile : 9808811777
                }
                document.write("FirstName: " + person.firstname);
                document.write("<br>" + "LastName: " + person.lastname);
                document.write("<br>" + "Mobile No: " + person.mobile);
        </script>
</body>
</html>
```

Output:

Example of new setting object properties:

```
<html>
<head>
<title>Page Title</title>
</head>
<body>
        <script>
      let person = {
      name: "Abhiraj",
      age: 9,
      gender: "Male"
   };
        document.write("==========For Statements==========="+"<br>");
   for(let i in person){
        document.write(person[i]+"<br>");
   }
   document.write("==========New Property===========");
        person.country = "Nepal";
        document.write("<br>"+"Contry Name : "+person.country);

        person["email"] = "abhiraj@mail.com";
        document.write("<br>"+"Email-id : "+person.email);

        person.age = 10;
        document.write("<br>"+"Age : "+person.age);

        person["name"] = "Nawara Prajapati";
        document.write("<br>"+person.name);
   </script>
</body>
</html>
```

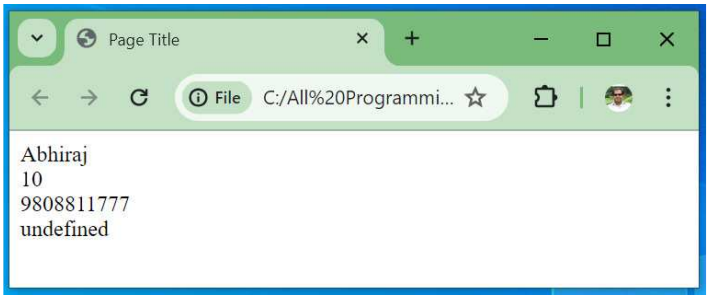
Output:



Example of deleting object properties:

```
<html>
<head>

<title>Page Title</title>
</head>
<body>
        <script>
                let person = {name: "Abhiraj", address: 10, mobile: 9808811777};
        for(let i in person)
        {
                document.write(person[i]+"<br>");
        }

        // Deleting property
        delete person.mobile;
        document.write(person.mobile);
   </script>
</body>
</html>
```

Output:



Arrays:

An array is a special variable which can hold more than one value at a time. It's like a container which is used to store values in a single variable. We can use an array to store values of string type, integer type, an object or any other valid data type in JavaScript.

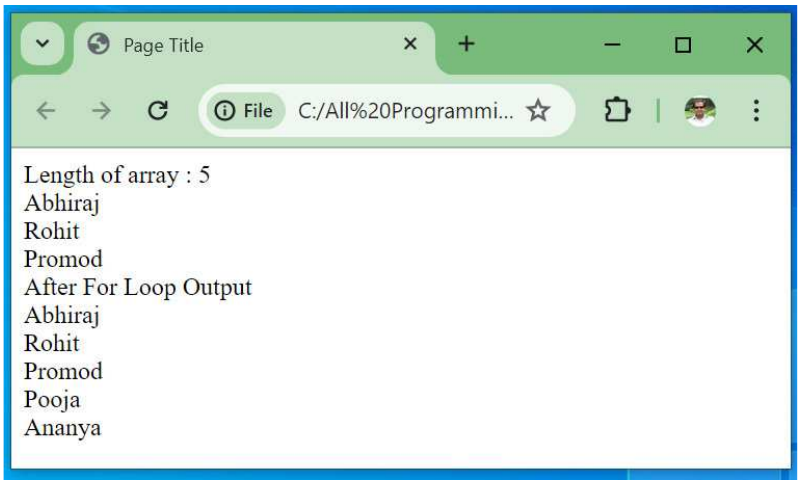
Syntax of array:

```
var arr = [ ];

var myArray = [element0, element1, ..., elementN];

var myArray = new Array(element0, element1, ..., elementN);
```

Example of array:

```
<html>
<head>
<title>Page Title</title>
</head>
<body>
        <script>
                let parson = ["Abhiraj", "Rohit", "Promod", "Pooja", "Ananya"];

                document.write("Length of array : "+parson.length)
                document.write("<br>"+parson[0]);
                document.write("<br>"+parson[1]);
                document.write("<br>"+parson[2]);
                document.write("<br>"+"After For Loop Output"+"<br>");
                for(let i in parson)
                {
                        document.write(parson[i]+"<br>");
                }
   </script>
</body>
</html>
```

Output:



Array Object Methods:

Array object methods have some of the popular methods that are used for performing various operation on array in JavaScript like adding a new element, removing an element, searching an index of an element.

- **concat():** Joins two or more arrays and returns the joined array.
- **join():** Join all the elements of an array into a string.
- **pop():** Removes the last element of an array and returns that element.
- **reverse():** Reverse the order of a list of elements in an array.
- **shift():** Removes the first element of an array and returns that element.
- **slice():** Selects a part of an array and return that part as a new array.
- **sort():** Sort the elements of an array.
- **push():** Adds new element as the last element and returns the length of the new array.
- **unshift():** Adds new elements to the array and returns the new length.
- **splice():** Adds or removes elements of an array.
- **fill():** Fill the elements into an array with static values.
- **every():** It determines whether all elements of an array are satisfying the provided function conditions.
- **isArray():** To check if the value is an array or not.
- **indexOf():** To find the index of an element in the array.
- **forEach():** To loop over the array values.

Example of cancat() method:

```
<script>
        Let parsonName = [ "Amit", "Rohit", "Abhiraj" ];
        Let parsonAddress = [ "Kalaiya", "Kathmandu", "Bhaktapur" ];

        var person = parsonName.concat(parsonAddress);
        document.write(person);
</script>
```

Output:

Amit,Rohit,Abhiraj,Kalaiya,Kathmandu,Bhaktapur

Example of join() method:

```
<script>
        let parsonName = ["Amit", "Rohit", "Abhiraj"];

        document.write(parsonName.join() + "<br>");
        document.write(parsonName.join(",") + "<br>");
        document.write(parsonName.join(" ") + "<br>");
        document.write(parsonName.join("*") + "<br>");
</script>
```

Output:

Amit,Rohit,Abhiraj
Amit,Rohit,Abhiraj
Amit Rohit Abhiraj
Amit*Rohit*Abhiraj

Example of join() method: Add an element as the Last Element of Array

```
<script>
    let parsonName = ["Amit", "Rohit", "Abhiraj"];
    document.write(parsonName+"<br>");
    parsonName.push("Nawaraj")
    document.write(parsonName);
</script>
```

Output:

```
Amit,Rohit,Abhiraj
Amit,Rohit,Abhiraj,Nawaraj
```

Example of unshift() method: Add an element as the First Element of Array.

```
<script>
    let parsonName = ["Amit", "Rohit", "Abhiraj"];
    document.write(parsonName+"<br>");
    parsonName.unshift("Nawaraj")
    document.write(parsonName);
</script>
```

Output:

```
Amit,Rohit,Abhiraj
Nawaraj,Amit,Rohit,Abhiraj
```

Example of indexOf() method: Find the index of an Array Element.

```
<script>
    let parsonName = ["Amit", "Rohit", "Abhiraj"];
    document.write(parsonName+"<br>");
    document.write(parsonName.indexOf("Abhiraj"));
</script>
```

Output:

```
Amit,Rohit,Abhiraj
2
```

Example of reverse() method: Reverse the Array.

```
<script>
    let parsonName = ["Amit", "Rohit", "Abhiraj"];
    document.write(parsonName+"<br>");
    document.write(parsonName.reverse());
</script>
```

Output:

```
Amit,Rohit,Abhiraj
Abhiraj,Rohit,Amit
```

String Methods in JavaScript:

Some of the important JavaScript string methods are

- **charAt()**: It provides the char value present at the specified index.
- **charCodeAt():** It provides the Unicode value of a character present at the specified index.
- **concat():** It provides a combination of two or more strings.
- **indexOf():** It provides the position of a char value present in the given string.
- **lastIndexOf():** It provides the position of a char value present in the given string by searching a character from the last position.
- **search():** It searches a specified regular expression in a given string and returns its position if a match occurs.
- **match():** It searches a specified regular expression in a given string and returns that regular expression if a match occurs.
- **replace():** It replaces a given string with the specified replacement.
- **substr():** It is used to fetch the part of the given string on the basis of the specified starting position and length.
- **substring():** It is used to fetch the part of the given string on the basis of the specified index.
- **slice():** It is used to fetch the part of the given string. It allows us to assign positive as well negative index.
- **toLowerCase():** It converts the given string into lowercase letter.
- **toLocaleLowerCase():** It converts the given string into lowercase letter on the basis of host?s current locale.
- **toUpperCase():** It converts the given string into uppercase letter.
- **toLocaleUpperCase():** It converts the given string into uppercase letter on the basis of host?s current locale.
- **toString():** It provides a string representing the particular object.
- **valueOf():** It provides the primitive value of string object.
- **split():** It splits a string into substring array, then returns that newly created array.
- **trim():** It trims the white space from the left and right side of the string.

For example:

```
<script>
    var str="Nawaraj";
    document.write(str.charAt(2));
</script>
```

Output:

w

```
<script>
        var str1 = "Hello";
        var str2 = "JavaScript";
        var str3 = "    Programming     ";
        var str4 = "hello my dear friends"

        var concattowstring = str1.concat(' ', str2);
        document.write(concattowstring + "<br>");

        var converttouppercase = str1.toUpperCase();
        document.write(converttouppercase + "<br>");

        var removewhitespace = str3.trim();
        document.write(removewhitespace + "<br>");

        var convertstringtoarray = str4.split(' ');
        document.write(convertstringtoarray + "<br>");

        var slicestring = str1.slice(1, 3);
        document.write(slicestring + "<br>");
</script>
```

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.

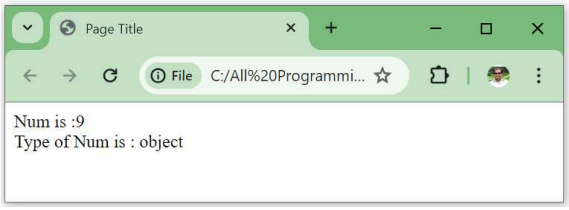
parseInt(): Parses a string and returns an integer.

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>
```

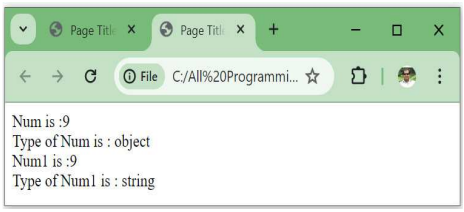
Output:



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:



Example of toFixed():

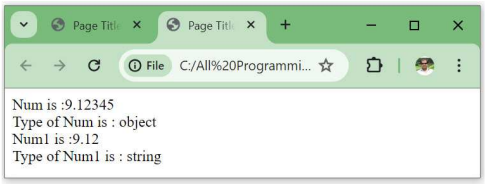
```
<script>
      var num= new Number(9.12345);
      document.write("Num is :"+num+"<br>");
      document.write("Type of Num is : "+typeof num+"<br>");

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

Output:

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>
```

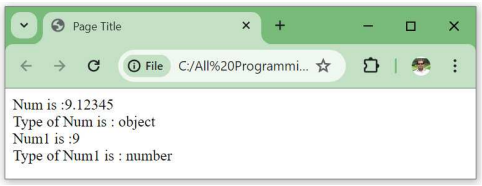
Output:



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>
```

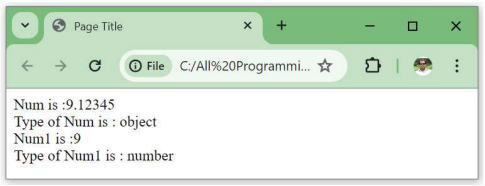
Output:



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:



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:

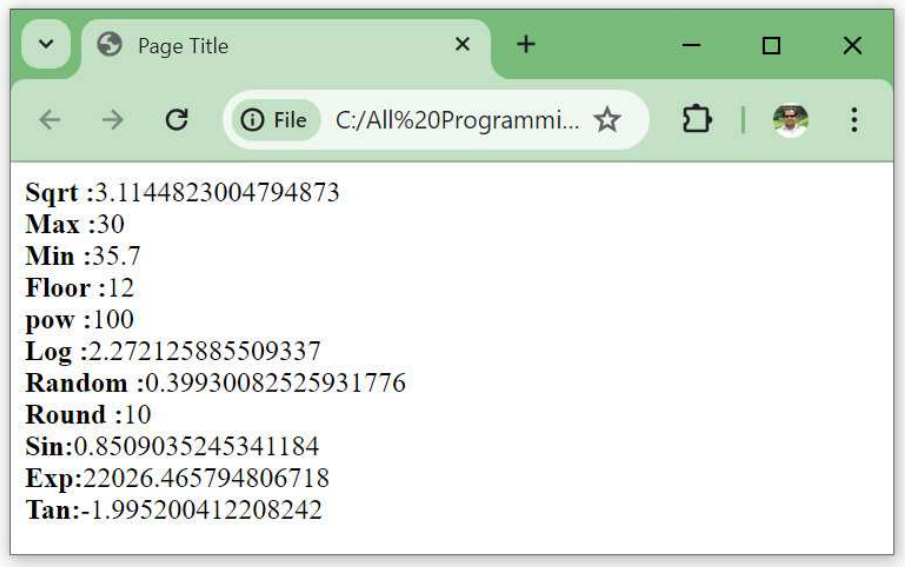
We have listed the most commonly used Math object methods with description.

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:



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

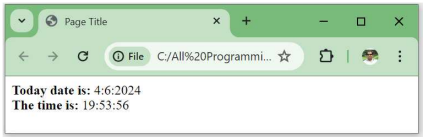
getMinutes(): returns the minutes (0–59) in the specified date

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>
```



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]

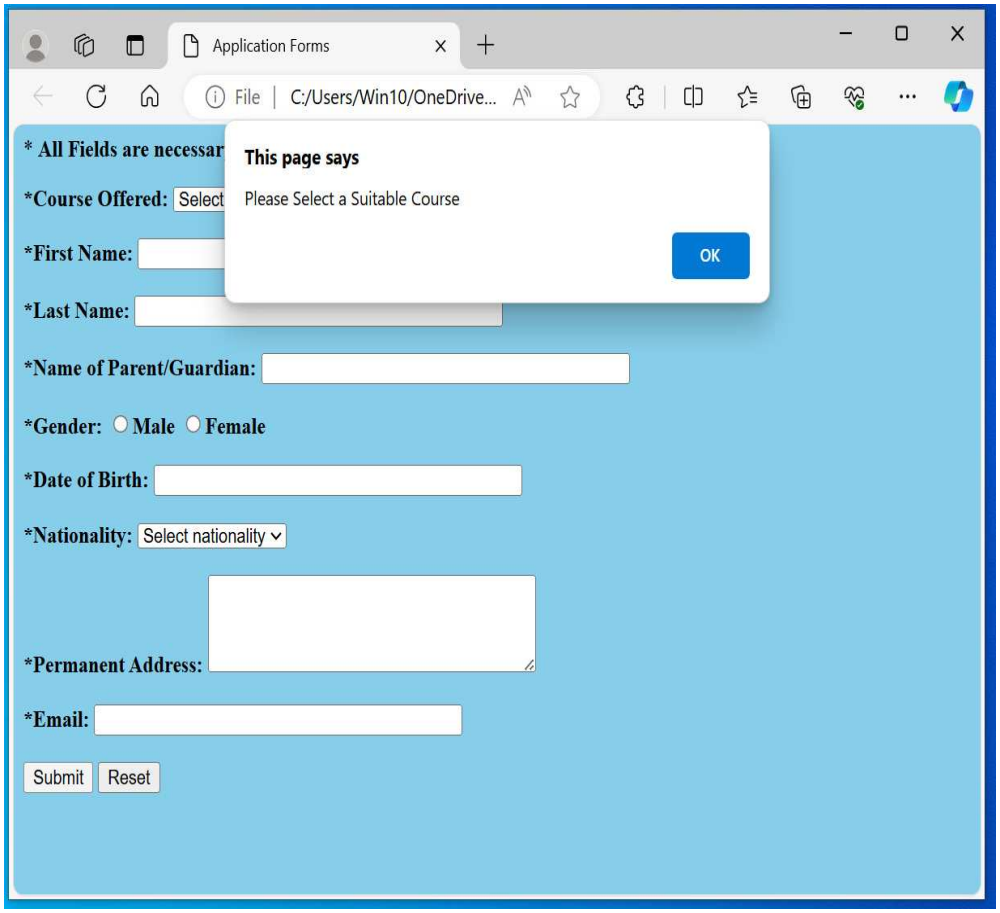
Example of RegExp:

```
<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:



```
<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: