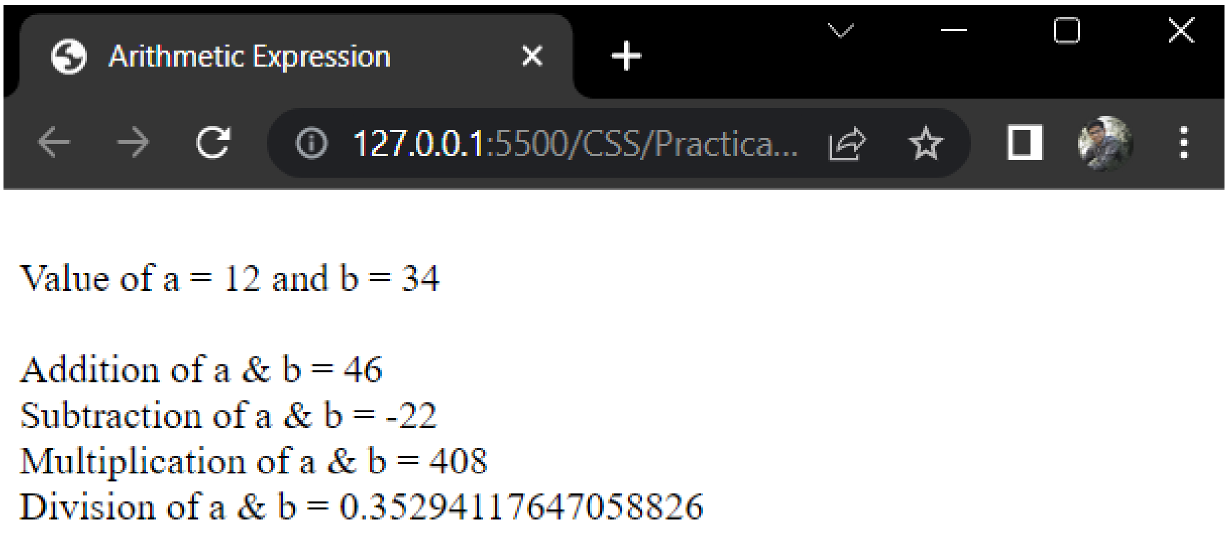
Practical No. 1

Write simple JavaScript with HTML for Arithmetic expression evaluation and message printing.

CODE:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Arithmetic Expression</title>
  </head>
  <body></body>
  <script type="text/javascript">
    var a = 12;
    var b = 34;
    var result;
    document.write("</br> Value of a = " + a + " and b = " + b);
    result = a + b;
    document.write("<br><br>Addition of a & b = " + result);
    result = a - b;
    document.write("<br>Subtraction of a & b = " + result);
    result = a * b;
    document.write("<br>Multiplication of a & b = " + result);
    result = a / b;
    document.write("<br>Division of a & b = " + result);
  </script>
</html>
```

OUTPUT:



```
Value of a = 12 and b = 34

Addition of a & b = 46
Subtraction of a & b = -22
Multiplication of a & b = 408
Division of a & b = 0.35294117647058826
```

Marks Obtained			Dated signature of Teacher
Process Related (15)	Product Related (35)	Total (50)	

Practical No. 2

Develop JavaScript to use decision making and looping statement.

Decision Making Statements:

- **Decision Making** in programming is similar to decision making in real life.
- A programming language uses control statements to control the flow of execution of the program based on certain conditions.
- JavaScript's conditional statements are:

  1) if

  2) if-else

  3) if…else…if

  4) switch

Looping Statements:

- **Looping** in programming languages facilitates the execution of a set of instructions/functions repeatedly while some condition evaluates to true.
- For example, suppose we want to print "Hello World" 10 times this is possible with the help of loops.
- There are mainly two types of loops:
  1) Entry Controlled loops: In this type of loops the test condition is tested before entering the loop body. For Loop and While Loop are entry controlled loops.
  2) Exit Controlled Loops: In this type of loops the test condition is tested or evaluated at the end of loop body. Therefore, the loop body will execute atleast once, irrespective of whether the test condition is true or false. do-while loop is exit controlled loop.
- Following are the types of loops in JavaScript:

  1) while loop

  2) do-while loop

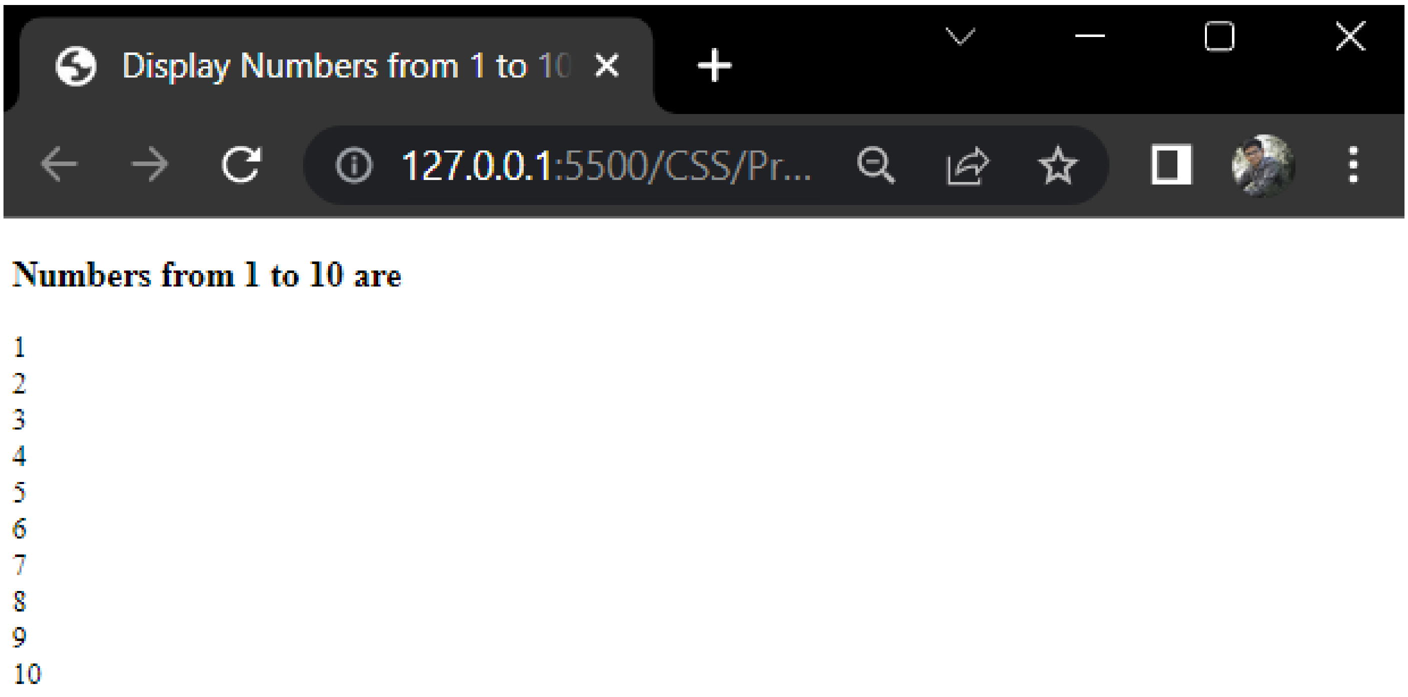
  3) for loop

  4) for…in loop

CODE:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Display Numbers from 1 to 10</title>
  </head>
  <body>
    <h3>Numbers from 1 to 10 are</h3>
  </body>
  <script type="text/javascript">
    var a = 1;
    while (a <= 10) {
      document.write(a + "<br />");
      a++;
    }
  </script>
</html>
```

OUTPUT:



```
Numbers from 1 to 10 are

1
2
3
4
5
6
7
8
9
10
```

Marks Obtained			Dated signature of Teacher
Process Related (15)	Product Related (35)	Total (50)	

Practical No. 3

Develop JavaScript to implement Array Functionalities.

JavaScript Arrays:

- An array is a special variable, which can hold more than one value at a time.
- JavaScript arrays are used to store multiple values in a single variable.

Declaring an array:

- There are 2 ways to construct an array in JavaScript
  1) By array literal
  2) By using an Array constructor (using new keyword)

JavaScript Array Methods:

- The Array object has many properties and methods which help developers to handle arrays easily and efficiently.
- You can get the value of a property by specifying **arrayname.property** and the output of a method by specifying **arrayname.method()**.
- The following are methods of the JavaScript Array:

1. length property → Returns the number of elements in an array.

2. prototype property → Used to add new properties and methods of object.

3. reverse() method → Returns the reverse order of items in an array.

4. sort() method → Returns the sort array.

5. pop() method → Remove the last item of an array.

6. shift() method → Remove the first item of an array.

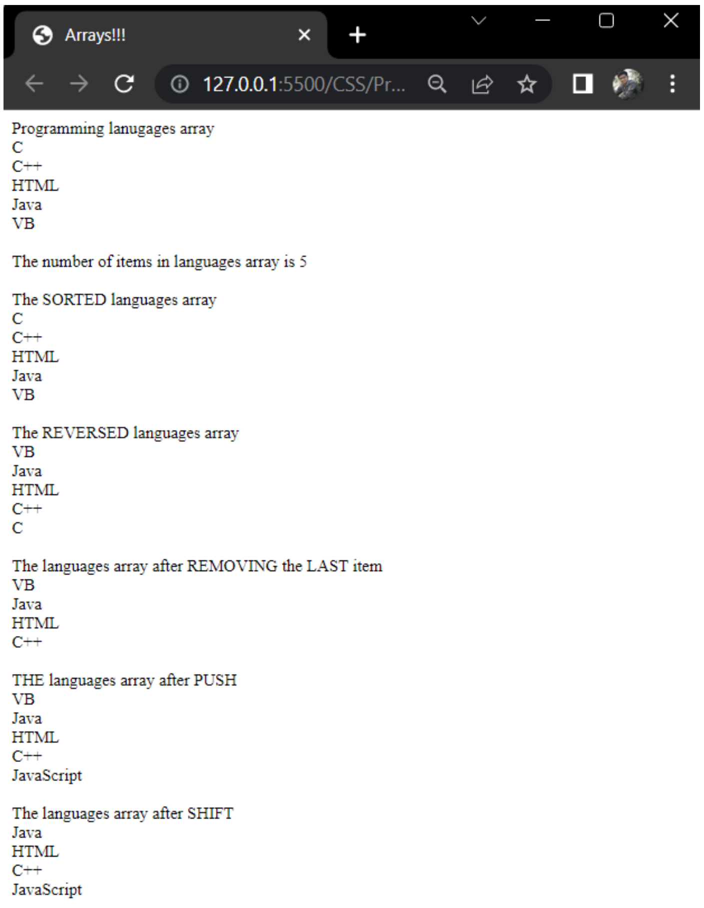
7. push() method → Adds an array element to array at last.

8. unshift() method → Adds an array element to array at beginning.

CODE:

```
<html>
<head>
 <title>Arrays!!!</title>
 <script type="text/javascript">
  var languages = new Array("C", "C++", "HTML", "Java", "VB");
  Array.prototype.displayItems=function(){
   for (i=0;i<this.length;i++){
    document.write(this[i] + "<br />");
   }
  }
  document.write("Programming lanugages array<br />");
  languages.displayItems();
  document.write("<br />The number of items in languages array is " +  languages.length +
"<br />");
  document.write("<br />The SORTED languages array<br />");
  languages.sort();
  languages.displayItems();
  document.write("<br />The REVERSED languages array<br />");
  languages.reverse();
  languages.displayItems();
  document.write("<br />The languages array after REMOVING the LAST item<br />");
  languages.pop();
  languages.displayItems();
   document.write("<br />THE languages array after PUSH<br />");
   languages.push("JavaScript");
  languages.displayItems();
   document.write("<br />The languages array after SHIFT<br />");
   languages.shift();
  languages.displayItems();
 </script>
</head>
<body>
</body>
</html>
```

OUTPUT:



```
Programming lanugages array
C
C++
HTML
Java
VB

The number of items in languages array is 5

The SORTED languages array
C
C++
HTML
Java
VB

The REVERSED languages array
VB
Java
HTML
C++
C

The languages array after REMOVING the LAST item
VB
Java
HTML
C++

THE languages array after PUSH
VB
Java
HTML
C++
JavaScript

The languages array after SHIFT
Java
HTML
C++
JavaScript
```

Marks Obtained			Dated signature of Teacher
Process Related (15)	**Product Related (35)**	**Total (50)**	

Practical No. 4

Develop JavaScript to implement Functions.

JavaScript Functions:

- A **function** is block of code designed to perform a particular task.
- It is reusable code which can be called anywhere in program.
- There are two types of functions:

  1) Built-in functions

  2) User-defined functions

Defining a function:

- A **JavaScript function** is defined with the function keyword, followed by a name, followed by parentheses ().
- Function names can contain letters, digits, underscores, and dollar signs (same rules as variables).
- The parentheses may include parameter names separated by commas: (parameter1, parameter2, ...)
- It is recommended that the function should define in <head> tag.
- A function definition consists of four parts:

  1) Function name

  2) Parenthesis

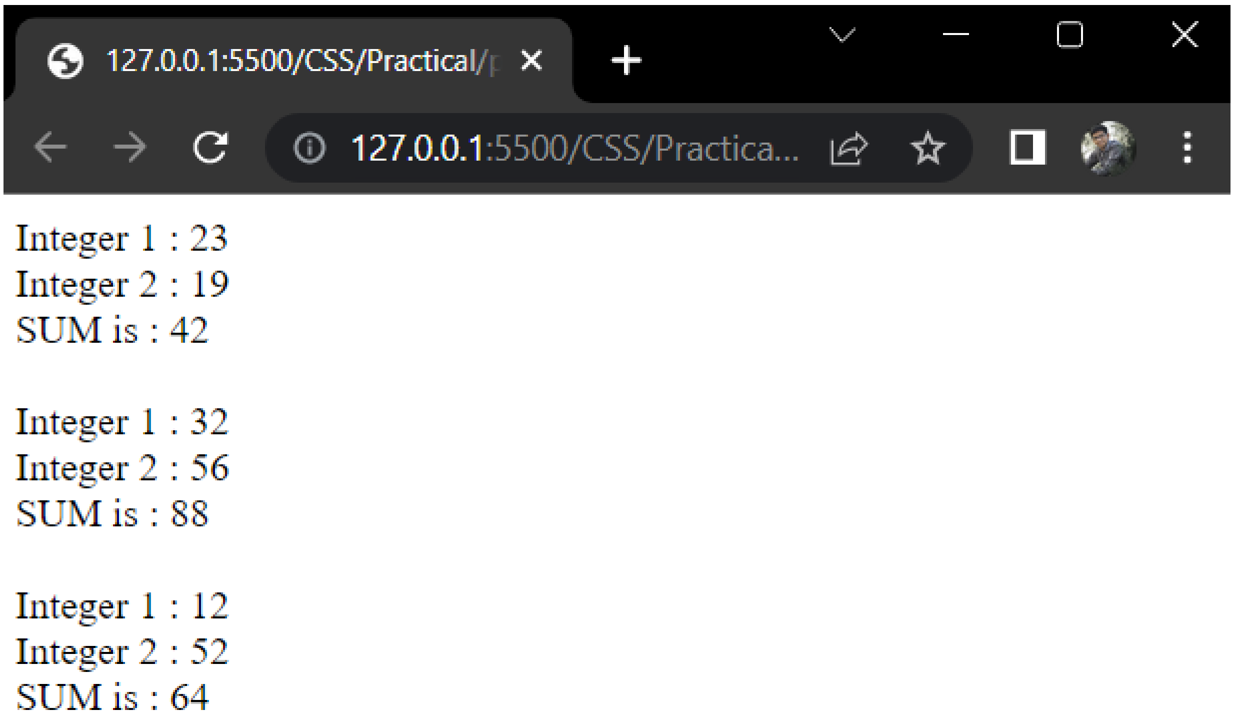
  3) Code block

  4) Return statement (Optional)

CODE:

```
<html>
<head>
    <script>
        function sum(val1, val2) {
            var result = val1 + val2;
            document.write("Integer 1 : "+ val1 + "<br/>");
            document.write("Integer 2 : "+ val2 + "<br/>");
            document.write("SUM is : " + result + "<br/><br/>");
        }
    </script>
</head>
<body>
    <script>
        sum(23, 19);
        sum(32, 56);
        sum(12, 52);
    </script>
</body>
</html>
```

OUTPUT:



```
Integer 1 : 23
Integer 2 : 19
SUM is : 42

Integer 1 : 32
Integer 2 : 56
SUM is : 88

Integer 1 : 12
Integer 2 : 52
SUM is : 64
```

Marks Obtained			Dated signature of Teacher
Process Related (15)	Product Related (35)	Total (50)	

Practical No. 5

Create a webpage to implement Form Events, Part-1.

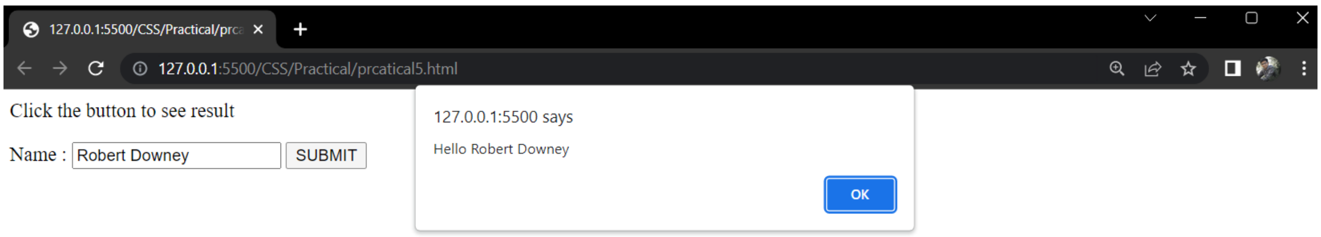
HTML Form Events:

- When JavaScript is used in HTML pages, JavaScript can “react” on these events.
- An HTML event can be something the browser does, or something a user does.
- The simple example of an event is a user clicking the mouse or pressing a key on the keyboard.
- Some examples of HTML events:
  - An HTML web page has finished loading
  - An HTML input field was changed
  - An HTML button was clicked
- When events happen, we may want to do something. JavaScript lets execute code when events are detected.

CODE:

```
<html>
  <head> </head>
  <body>
    <p>Click the button to see result</p>
    <form>
      <label for="name">Name :</label>
      <input type="text" name="name" id="name" />
      <input type="button" onclick="sayHello()" value="SUBMIT" />
    </form>
  </body>
  <script type="text/javascript">
    function sayHello() {
      let name = document.getElementById("name").value;
      alert("Hello " + name);
    }
  </script>
</html>
```

OUTPUT:


127.0.0.1:5500/CSS/Practical/prcatical5.html

Click the button to see result

Name : Robert Downey SUBMIT

127.0.0.1:5500 says

Hello Robert Downey

OK

Marks Obtained			Dated signature of Teacher
Process Related (15)	**Product Related (35)**	**Total (50)**	

Practical No. 6

Create a webpage to implement Form Events, Part-2.

Mouse Events:

- Mouse events are used to capture the interactions made by user using mouse.
- Such events may come not only from "mouse devices", but are also from other devices, such as phones and tablets.

Mouse Event Types:

- **mousedown/mouseup** : Mouse button is clicked/released over an element.
- **mouseover/mouseout** : Mouse pointer comes over/out from an element.
- **mousemove** : Every mouse move over an element triggers that event.
- **click** : Triggers after mousedown and then mouseup over the same element if the left mouse button was used.
- **dblclick** : Triggers after two clicks on the same element within a short timeframe. Rarely used nowadays.
- **contextmenu** : Triggers when the right mouse button is pressed. There are other ways to open a context menu, e.g. using a special keyboard key, it triggers in that case also, so it's not exactly the mouse event.

CODE:

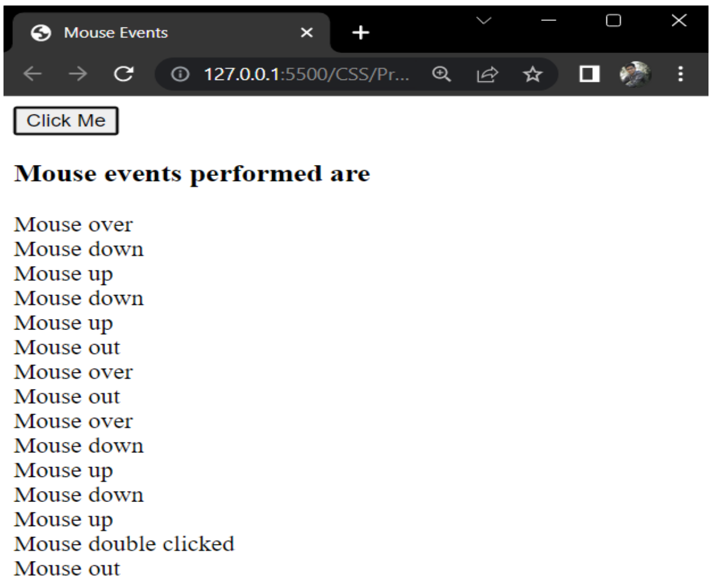
```html
<html>
  <head>
    <title>Mouse Events</title>
    <script>
      function init() {
        var panel = document.getElementById("panel");
        var btn = document.getElementById("btn");

        btn.addEventListener("dblclick", dblClick);
        btn.addEventListener("mousedown", mouseDown);
        btn.addEventListener("mouseup", mouseUp);
        btn.addEventListener("mouseover", mouseOver);
        btn.addEventListener("mouseout", mouseOut);
      }

      function click() {
        panel.innerHTML += "Mouse clicked<br/>";
      }

      function dblClick() {
        panel.innerHTML += "Mouse double clicked<br/>";
      }
      function mouseDown() {
        panel.innerHTML += "Mouse down<br/>";
      }
      function mouseUp() {
        panel.innerHTML += "Mouse up<br/>";
      }
      function mouseOver() {
        panel.innerHTML += "Mouse over<br/>";
      }
      function mouseOut() {
        panel.innerHTML += "Mouse out<br/>";
      }
      function mouseMove() {
        panel.innerHTML += "Mouse moved<br/>";
      }
    </script>
  </head>
  <body onload="init()">
    <input
      type="button"
      id="btn"
      value="Click Me"
      onclick="click()"
      onmousemove="mouseMove"
    />
    <h3>Mouse events performed are</h3>
    <p id="panel"></p>
  </body>
</html>
```

OUTPUT:



Mouse events performed are

```
Mouse over
Mouse down
Mouse up
Mouse down
Mouse up
Mouse out
Mouse over
Mouse out
Mouse over
Mouse down
Mouse up
Mouse down
Mouse up
Mouse double clicked
Mouse out
```

Marks Obtained			**Dated signature of Teacher**
Process Related (15)	**Product Related (35)**	**Total (50)**	

Practical No. 7

Develop a webpage using Intrinsic Java Functions.

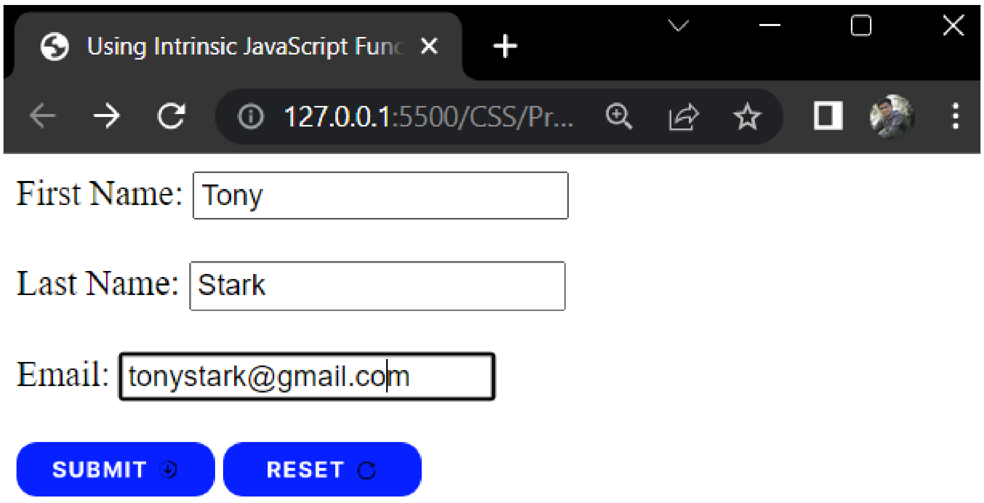
Intrinsic JavaScript Functions:

- An **intrinsic function** (or **built-in function**) is a function (subroutine) available for use in a given programming language whose implementation is handled specially by the compiler.
- **“Intrinsic”** is the way some authors refer to what other authors call **“built-in”**.
- Those data types/objects/classes are always there regardless of what environment you’re running in.
- JavaScript provides intrinsic (or “built-in”) objects. They are the Array, Boolean, Date, Error, Function, Global, JSON, Math, Number, Object, RegExp, and String objects.
- As you know JavaScript is an object oriented programming language, it supports the concept of objects in the form of attributes.
- If an object attribute consists of function, then it is a method of that object, or if an object attribute consists of values, then it is a property of that object.

CODE:

```
<html>
  <head>
    <title>Using Intrinsic JavaScript Functions</title>
  </head>
  <body>
    <form name="contact" action="#" method="post">
      <p>
        First Name: <input type="text" name="Fname" /> <br /><br />
        Last Name: <input type="text" name="Lname" /><br /><br />
        Email: <input type="text" name="Email" /><br /><br />
        <img
          src="submit.png"
          width="90px"
          onclick="javascript:document.forms.contact.submit()"
        />
        <img
          src="reset.png"
          width="90px"
          onclick="javascript:document.forms.contact.reset()"
        />
      </p>
    </form>
  </body>
</html>
```

OUTPUT:



First Name: Tony

Last Name: Stark

Email: tonystark@gmail.com

Marks Obtained			Dated signature of Teacher
Process Related (15)	**Product Related (35)**	**Total (50)**	