UNIT 3 : JAVASCRIPT

3.1 Static, Dynamic and Active Page

3.2 DHTML Events

3.2.1 Window, Form, Keyboard, Mouse

3.3 Javascript

3.3.1 Overview of Client and Server Side Scripting

3.3.2 Structure of Javascript

3.3.3 Data Types and Variables

3.3.4 Operators (Arithmetic Operator, Assignment Operator, Comparison Operator, Logical Operator, Conditional Operator)

3.3.5 Control Structure

3.3 Javascript

3.3.1 Overview of Client and Server Side Scripting

3.3.2 Structure of Javascript

3.3.3 Data Types and Variables

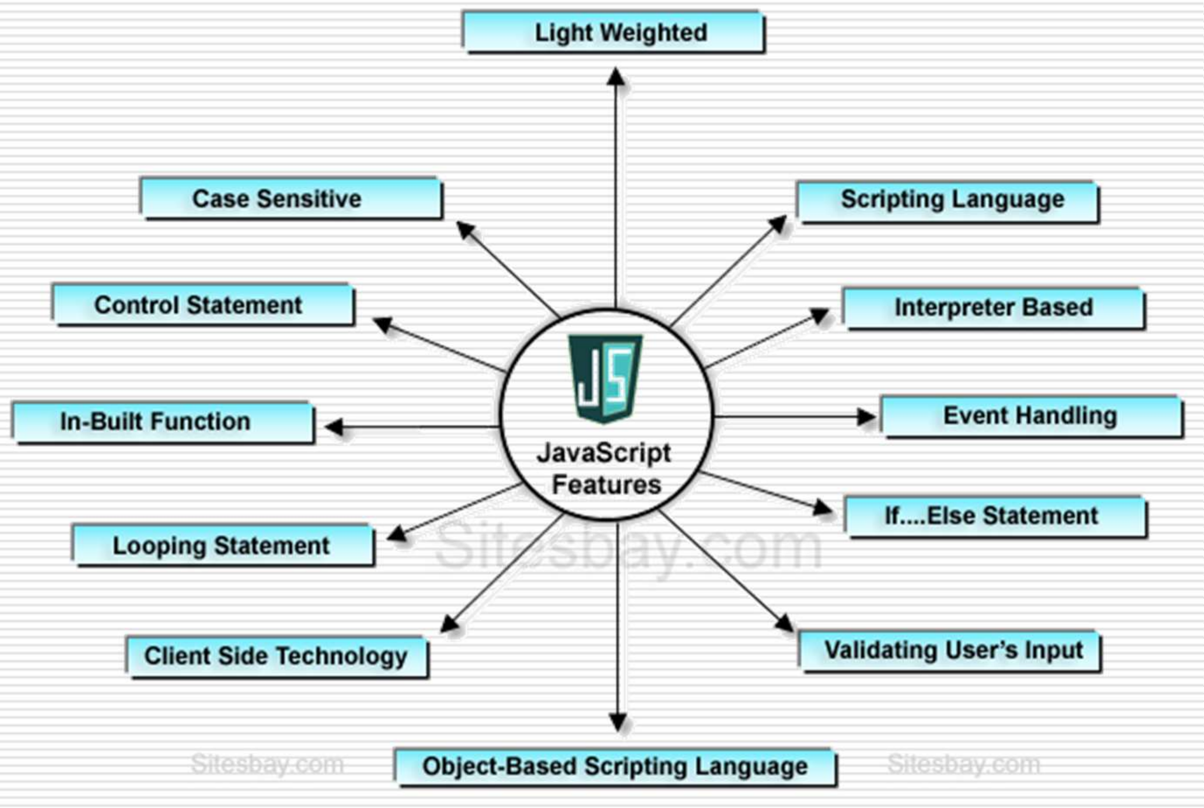
3.3.4 Operators

- Arithmetic Operator
- Assignment Operator
- Comparison Operator
- Logical Operator
- Conditional Operator

Javascript

- **JavaScript** was developed by **Brendan Eich** in **1995**, which appeared in Netscape, a popular browser of that time.
- The language was initially called LiveScript and was later renamed **JavaScript**.
- There are many programmers who think that **JavaScript** and Java are the same. In fact, **JavaScript** and Java are very much unrelated.

- JavaScript is a very powerful **client-side scripting language**. JavaScript is used mainly for enhancing the interaction of a user with the webpage.
- In other words, you can make your webpage more lively and interactive, with the help of JavaScript.
- JavaScript is also being used widely in game development and Mobile application development.


Light Weighted
Case Sensitive
Scripting Language
Control Statement
Interpreter Based
In-Built Function
JavaScript Features
Event Handling
If....Else Statement
Looping Statement
Client Side Technology
Validating User's Input
Object-Based Scripting Language

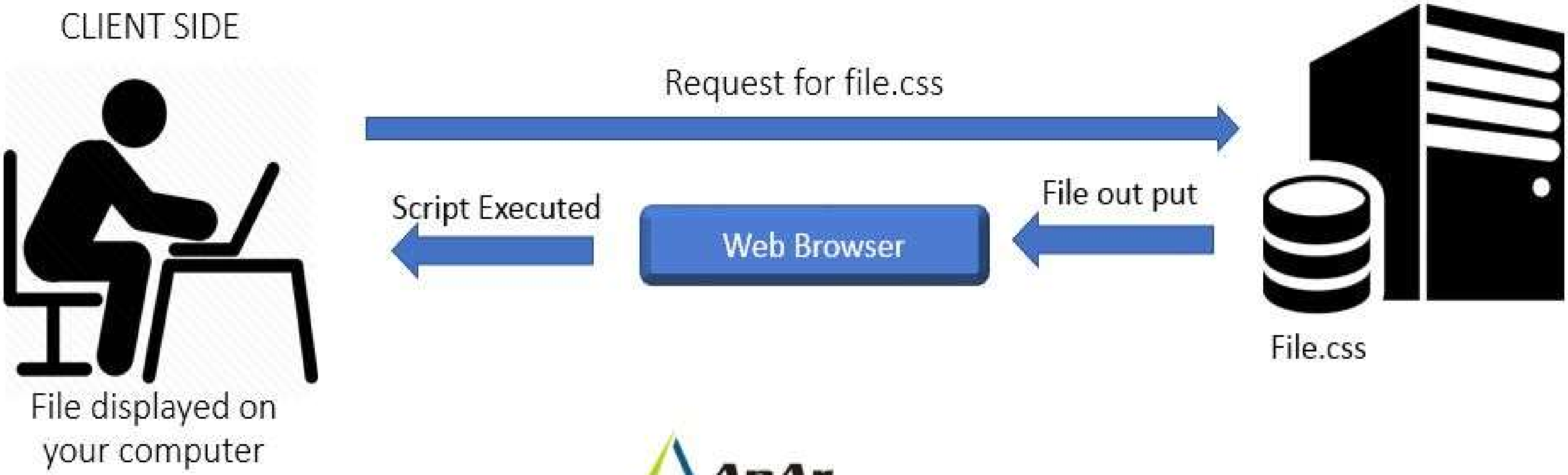
3.3.1 Overview of Client and Server Side Scripting

- A scripting or script language is a programming language for a special run-time environment that automates the execution of tasks; the tasks could alternatively be executed one-by-one by a human operator. Scripting languages are often interpreted.

Client Side Scripting

- **Client-side scripting**
- (embedded **scripts**) is code that exists inside the **client's** HTML page. This code will be processed on the **client** machine and the HTML page will NOT perform a PostBack to the web-**server**. Traditionally, **client-side scripting** is used for page navigation, data validation and formatting.

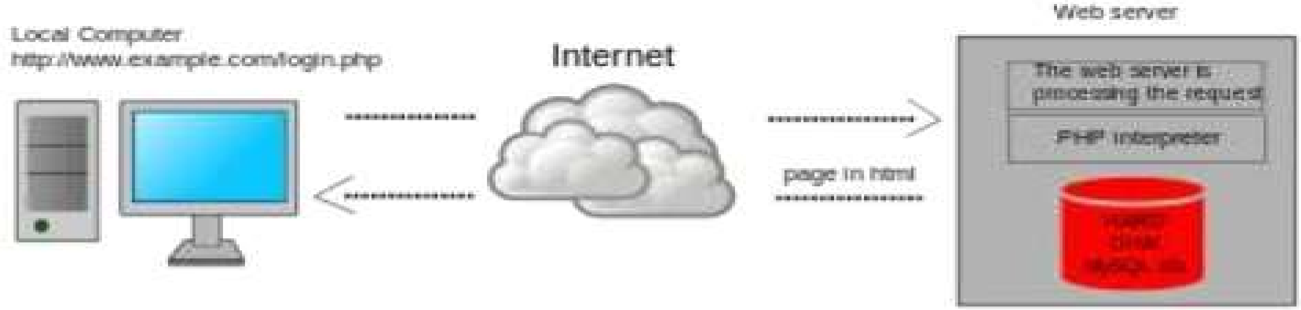
The process of client side scripting



Server Side Scripting

- Server-side scripting is a technique used in web development which involves employing scripts on a web server which produce a response customized for each user's request to the website. The alternative is for the web server itself to deliver a static web page.

SERVER SIDE SCRIPTING



3.3.2 Structure of Javascript

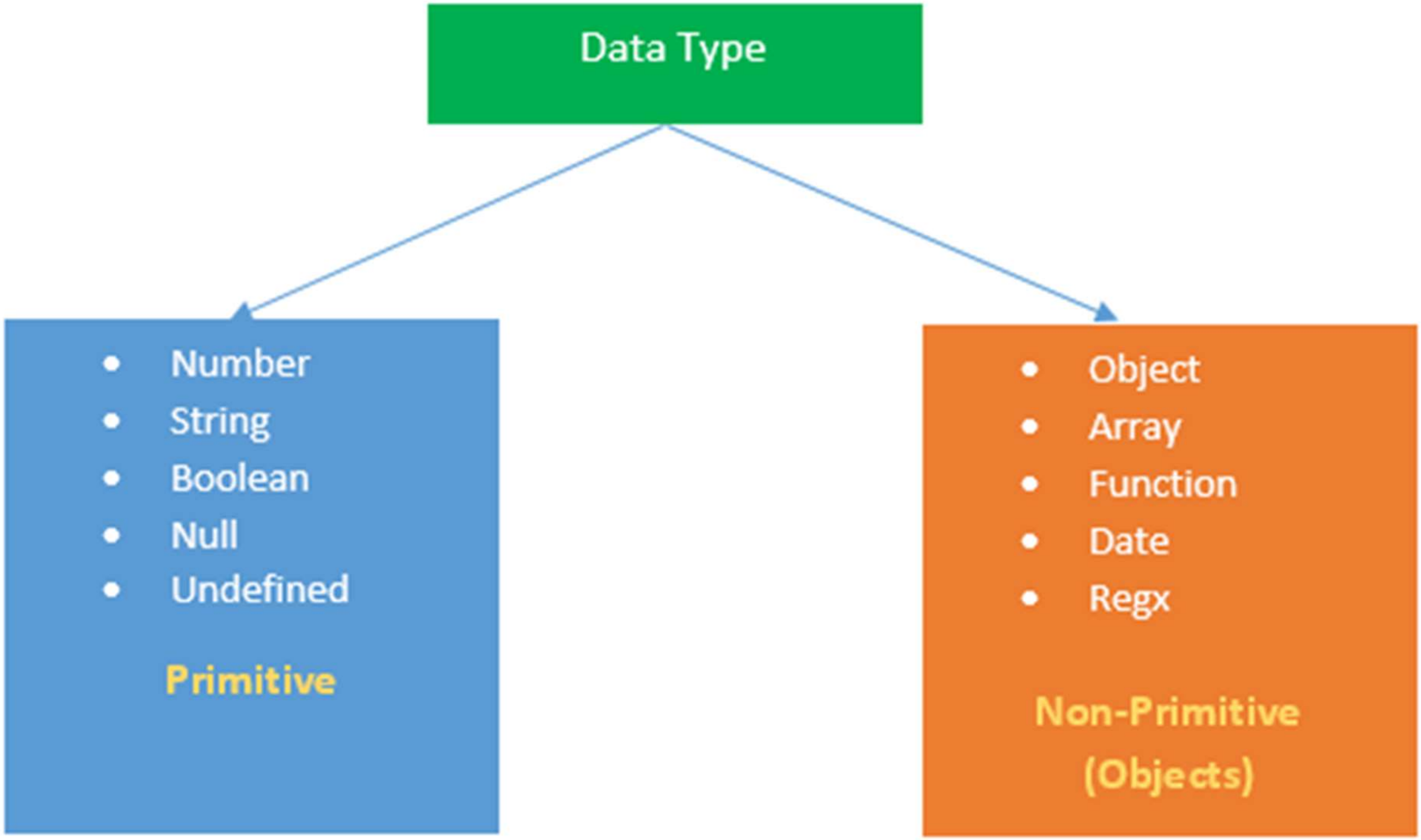
```
<HTML>
        <HEAD>
                <TITLE>JAVASCRIPT STRUCTURE</TITLE>
                <SCRIPT LANGUAGE="JAVASCRIPT">
                        //WRITE JAVASCRIPT CODE HERE
                </SCRIPT>
        </HEAD>
        <BODY>
                <FORM>

                </FORM>
        </BODY>
</HTML>
```

- Javascript is case sensitive language
- It is loosely typed language

Comments

- Single line Comment
  - // is used for Single Line Comment
- Multi line Comment
  - /* and */ is used for Multi Line Comment

3.3.3 Data Types and Variables



Data Type

- Number
- String
- Boolean
- Null
- Undefined

Primitive

- Object
- Array
- Function
- Date
- Regx

Non-Primitive (Objects)

JavaScript Data Type

Number:

- JavaScript has only one type of numbers.
- Numbers can be written with, or without decimals:
  - var x1 = 34.00; // Written with decimals
  - var x2 = 34; // Written without decimals

String:

- A string (or a text string) is a series of characters like “SYBCA THE GREAT CLASS".
- Strings are written with quotes. You can use single or double quotes:
  - var carName1 = "Volvo XC60";// Using double quotes
  - var carName2 = 'Volvo XC60'; // Using single quotes

Boolean:

- Booleans can only have two values: true or false.
  - var a=true;
  - var flag=false;

Null:

- In JavaScript null is "nothing". It is supposed to be something that doesn't exist.
- Unfortunately, in JavaScript, the data type of null is an object.
  - var a = null;

undefined:

- In JavaScript, a variable without a value, has the value undefined. The type is also undefined.
  - var car; // Value is undefined, type is undefined

Variables

- A **JavaScript variable** is simply a name of storage location.
- There are two types of variables in JavaScript : local variable and global variable.
- There are some rules while declaring a JavaScript variable (also known as identifiers).
  1. Name must start with a letter (a to z or A to Z), underscore(_), or dollar($) sign.
  2. After first letter we can use digits (0 to 9), for example value1.
  3. JavaScript variables are case sensitive, for example x and X are different variables.

- “var” is used to declare variable.
- Syntax :
  - var varname=value”;
- Example:
  - var n=10;
  - var s=“SYBCA”;
  - var flag=“true”;
  - var a=null;

3.3.4 Operators

1. Arithmetic Operator
2. Assignment Operator
3. Comparison Operator
4. Logical Operator
5. Conditional Operator
6. String Operator

1. Arithmetic Operator

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

2. Assignment Operator

Operator	Example	Same As
=	x = y	x = y
+=	x += y	x = x + y
-=	x -= y	x = x - y
*=	x *= y	x = x * y
/=	x /= y	x = x / y
%=	x %= y	x = x % y
**=	x **= y	x = x ** y

3. Comparison Operator

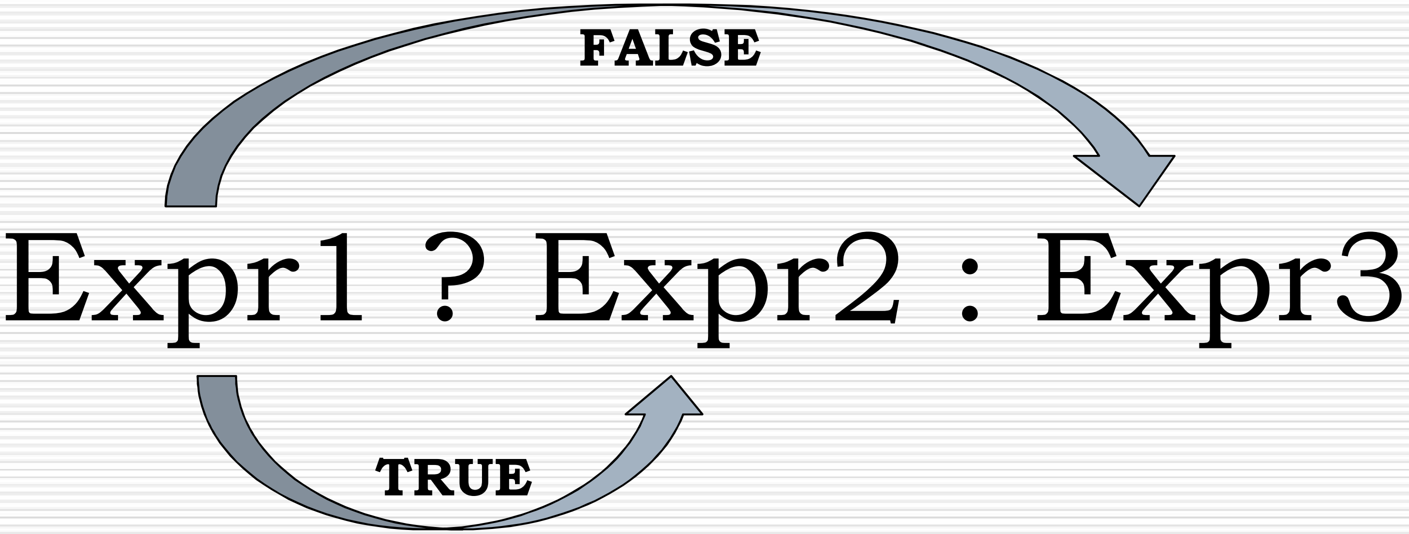
Operator	Description
==	equal to
===	equal value and equal type
!=	not equal
!==	not equal value or not equal type
>	greater than
<	less than
>=	greater than or equal to
<=	less than or equal to

4. Logical Operator

Operator	Description
&&	logical and
| |	logical or
!	logical not

5. Conditional Operator

- ? : Operator
- Also known as Ternary Operator



Expr1 ? Expr2 : Expr3

6. String Operator

- The + operator can also be used to add (concatenate) strings.

3.3.5 Control Structures

1. Branching
2. Looping
3. Jumping

Branching

- Simple if Statement
- if….else Statement
- Netsted if Statement
- Ladder if or elseif Statement
- Switch…case Statement

Looping

- Entry Control Loop
  - for loop
  - while loop
- Exit Control Loop
  - do…while Loop

Jumping

- goto Statement
- break
- continue

Dialogue Boxes

- In JavaScript you can create dialog boxes or popups to interact with the user.
- You can either use them to notify a user or to receive some kind of user input before proceeding.
  - Alert Dialogue Box
  - Confirmation Dialogue Box
  - Prompt Dialogue Box

1. Alert Dialogue Box

- An alert dialog box is mostly used to give a warning message to the users. For example, if one input field requires to enter some text but the user does not provide any input, then as a part of validation, you can use an alert box to give a warning message.
- Nonetheless, an alert box can still be used for friendlier messages. Alert box gives only one button "OK" to select and proceed.

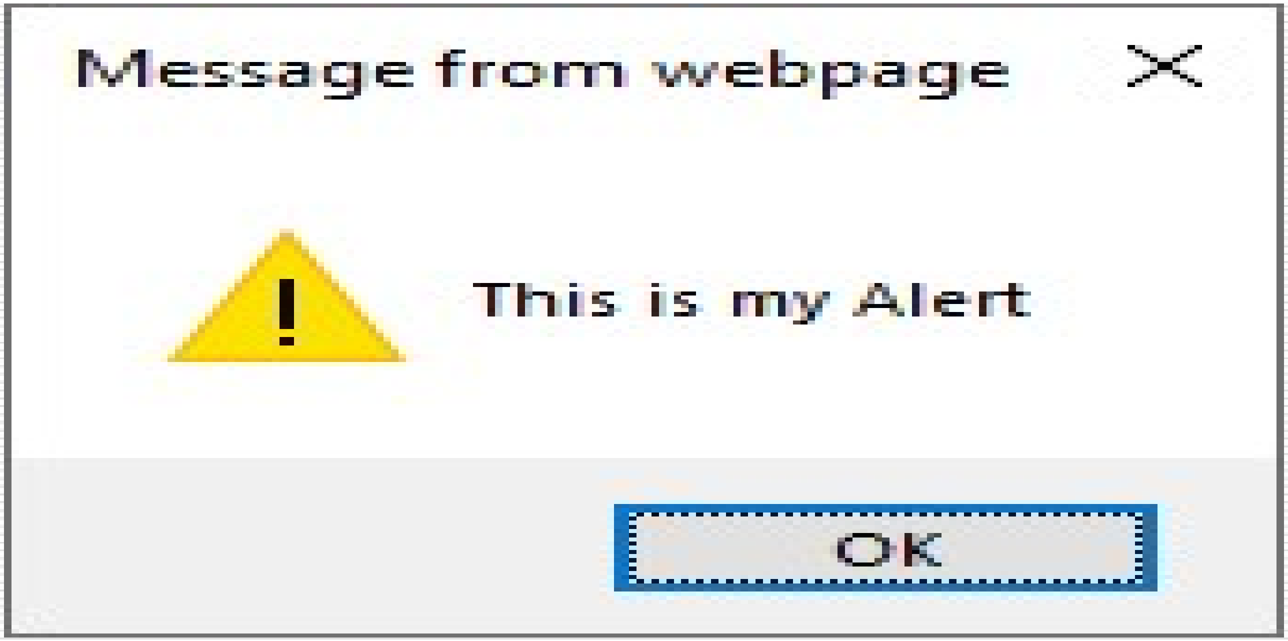
Syntax

- alert([Alert Message]);

Example

```
<HTML>
    <HEAD>
        <SCRIPT LANGUAGE="JAVASCRIPT">
            alert("This is my Alert");
        </SCRIPT>
    </HEAD>
    <BODY>

    </BODY>
</HTML>
```

Output



Message from webpage

This is my Alert

2. Confirmation Dialogue 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 window method **confirm()** will return true. If the user clicks on the Cancel button, then **confirm()** returns false.

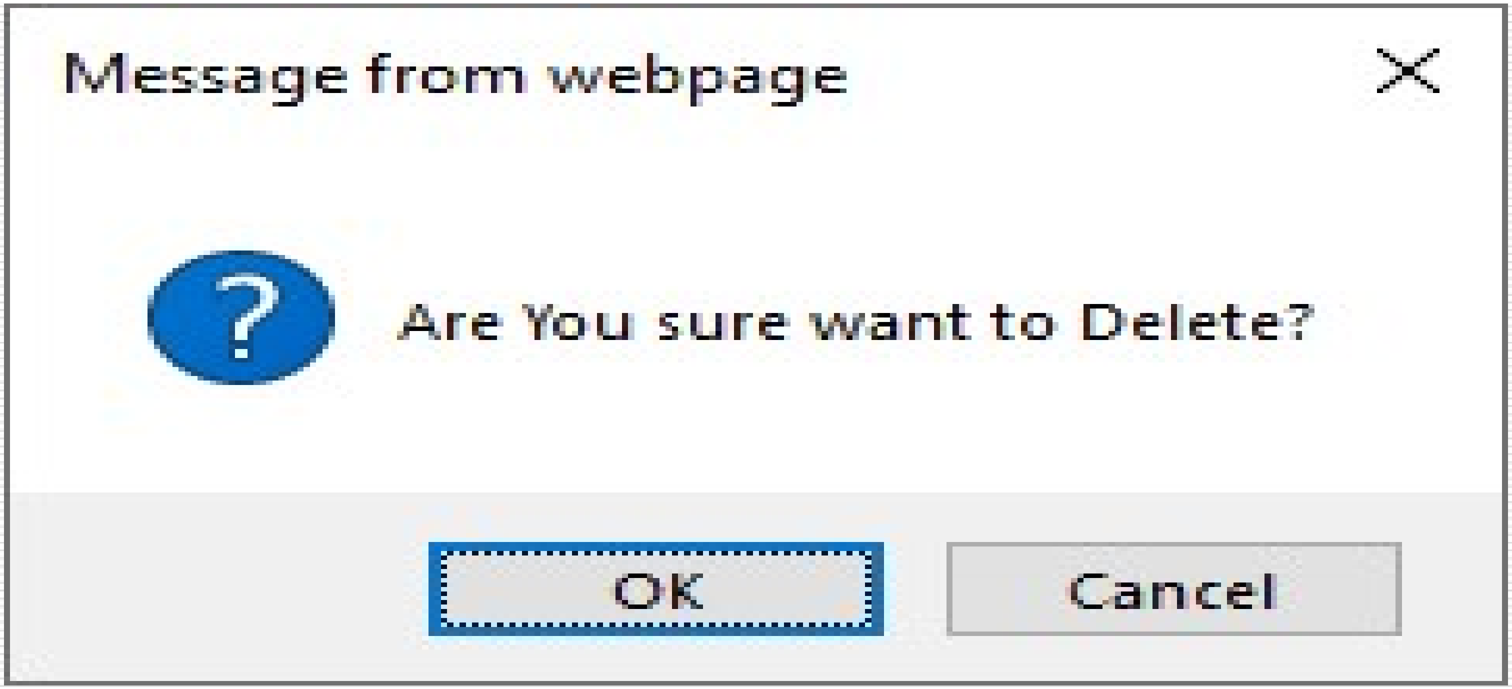
Syntax

- confirm([Message]);

Example

```
<HTML>
        <HEAD>
                <SCRIPT LANGUAGE="JAVASCRIPT">
                        confirm("Are You sure want to Delete?");
                </SCRIPT>
        </HEAD>
        <BODY>

        </BODY>
</HTML>
```

Output



Message from webpage

Are You sure want to Delete?

3. Prompt Dialogue Box

- The prompt dialog box is used to prompt the user to enter information. A prompt dialog box includes a text input field, an OK and a Cancel button.
- The prompt dialog box is very useful when you want to pop-up a text box to get user input. Thus, it enables you to interact with the user. The user needs to fill in the field and then click OK.

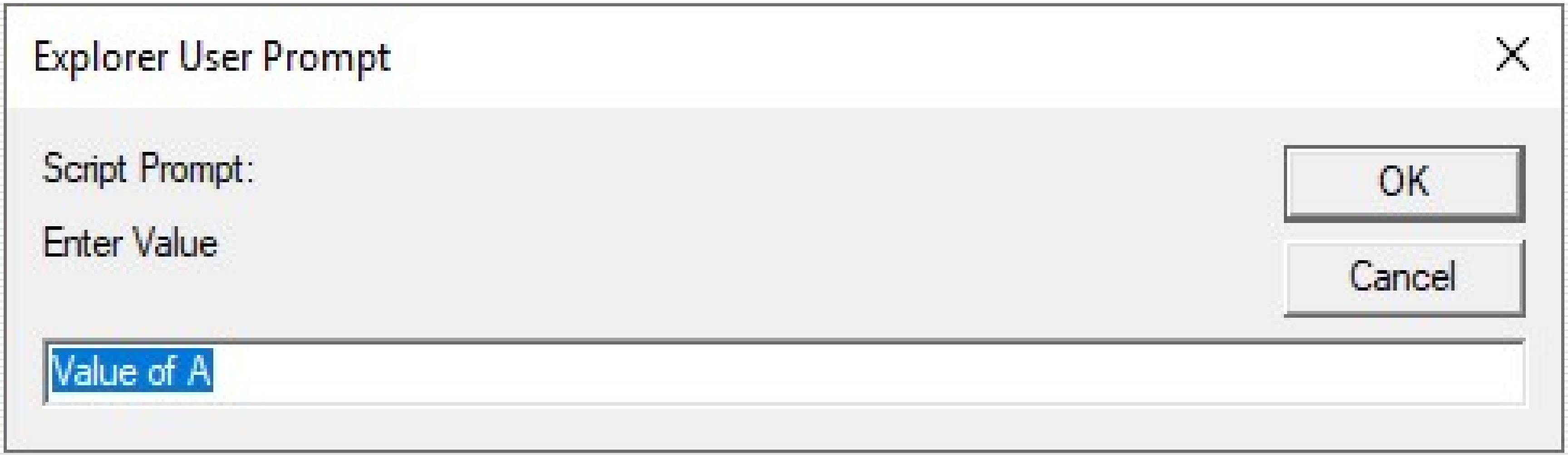
Syntax

- prompt([Message],Input Message);

Example

```
<HTML>
        <HEAD>
                <SCRIPT LANGUAGE="JAVASCRIPT">
                        var A=prompt("Enter Value","Value of A");
                        alert(A);
                </SCRIPT>
        </HEAD>
        <BODY>

        </BODY>
</HTML>
```

Output



Explorer User Prompt

Script Prompt:

Enter Value

Value of A

User Defined Functions

- A JavaScript function is a block of code designed to perform a particular task.
- A JavaScript function is executed when "something" invokes it (calls it).

- 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).

```
function name(parameter1, parameter2, parameter3)
{
  // code to be executed
}
```

Categories of UDF

1. Without argument without return value
2. With argument without return value
3. Without argument with return value
4. With argument with return value
5. As a function Parameters

1. Without argument without return value

```
<HTML>
        <HEAD>
                <SCRIPT LANGUAGE="JAVASCRIPT">
                        function add()
                        {
                                var a=10,b=20;
                                var c=a+b;
                                alert(c);
                        }
                        add();
                </SCRIPT>
        </HEAD>
        <BODY>

        </BODY>
</HTML>
```

OUTPUT:

```
30
```

2. With argument without return value

```
<HTML>
        <HEAD>
                <SCRIPT LANGUAGE="JAVASCRIPT">
                        function add(a, b)
                        {
                                var c=a+b;
                                alert(c);
                        }
                        add(10,20);
                </SCRIPT>
        </HEAD>
        <BODY>

        </BODY>
</HTML>
```

OUTPUT:

30

3. Without argument with return value

```
<HTML>
        <HEAD>
                <SCRIPT LANGUAGE="JAVASCRIPT">
                        function add()
                        {
                                var a=10,b=20;
                                var c=a+b;
                                return c;
                        }
                        alert(add());
                </SCRIPT>
        </HEAD>
        <BODY>

        </BODY>
</HTML>
```

OUTPUT:

30

4. With argument with return value

```
<HTML>
    <HEAD>
        <SCRIPT LANGUAGE="JAVASCRIPT">
            function add(a,b)
            {
                var c=a+b;
                return c;
            }
            alert(add(10,20));
        </SCRIPT>
    </HEAD>
    <BODY>

    </BODY>
</HTML>
```

OUTPUT:

30

5. With Function Parameters