UNIT 3

Web Development II

WEB

- Web is networked information system that contains huge collection of files written in HTML or other programming language.
- HTML is only used for static webpages. To make it dynamic various server side scripting (e.g. php, asp) and Client side scripting (Javascript, vbscript) must be written
- Server side script runs on server whereas client side script runs on client computer's browser

WHAT IS SCRIPT?

- A script is generally a series of program or instruction, which has to be executed on other program or application.
- The scripts can be written in two forms:
  - Server side script (php, asp, jsp, vb etc)
  - Client side script (javascript, vbscript, jquery etc)

WHAT IS SCRIPT?

- The client script executes the code to the client side which is visible to the users. The client side scripting requires browsers to run the script on client machine but does not interact with server while processing the client side scripts.
- The server side script is executed in the server end which is hidden from the users. The server side scripting involves server for its processing.

SERVER SIDE SCRIPT

- any scripts or programs that run on the web server is known as server side scripts.
- operations like customization of website, dynamic change in the website, content, response generation to the user's queries, accessing the database, and so on are performed at the server end.
- Earlier, the server side scripting is implemented by the CGI (Common Gateway Interface) scripts.
- When a browser sends a request to the server for a webpage consisting of server side scripting, the web server processes the script before serving it to the browser.
- The web server abstracts the scripts from the end user until serving the content which makes the data and source code more secure.

SERVER SIDE SCRIPT

- **PHP:** It is the most popular server side language used on the web which was designed to extract and manipulate information in the database using SQL queries. It is used in facebook, wordpress and Wikipedia etc.
- **Python:** It is fast and contains shorter code. It is good for beginners as it concentrates on the readability and simplicity of the code. Python funtions well in the object oriented environment and used in famous sites like youtube, google, etc.

SERVER SIDE SCRIPT

- **Ruby:** It contains complex logic which packages the back-end with database utility which can also be provided by PHP and SQL.
- **ASP.NET:** It is a web development platform, which provides a programming model, a comprehensive software infrastructure and various services required to build up robust web applications for PC, as well as mobile devices.

SERVER SIDE SCRIPT: ADVANTAGES

- User can create one template for the entire website
- The site can use a content management system which makes editing simpler.
- Generally quicker to load than client-side scripting
- User is able to include external files to save coding.
- Scripts are hidden from view so it is more secure. Users only see the HTML output.
- User does not need to download plugins like Java or Flash.

SERVER SIDE SCRIPT: DISADVANTAGES

- Many scripts and content management systems tools require databases in order to store dynamic data.
- It requires the scripting software to be installed on the server.
- The nature of dynamic scripts creates new security concerns, in some cases making it easier for hackers to gain access to servers exploiting code flaws.

CLIENT SIDE SCRIPT

- Client-side scripting is performed to generate a code that can run on the client end (browser) without needing the server side processing.
- The client-side scripting can be used to examine the user's form for the errors before submitting it and for changing the content according to the user input.
- The effective client-side scripting can significantly reduce the server load.
- For example, when a user makes a request via browser for a webpage to the server, it just sent the HTML and CSS as plain text, and the browser interprets and renders the web content in the client end.

CLIENT SIDE SCRIPT

- **HTML:** It is the fundamental building blocks of web programming which provides the frame to the website. It describes the arrangement of the content.
- **CSS:** CSS provides the way to design the graphic elements which help in making the appearance of the web application more attractive.

CLIENT SIDE SCRIPT

- **JavaScript:** It is also a client-side scripting language which essentially devised for the specific purpose, but currently there are various JavaScript frameworks used as server-side scripting technology.
- **VBScript:** Microsoft VBScript (Visual Basic Script) is a general-purpose, lightweight and active scripting language developed by Microsoft that is modeled on Visual Basic.

CLIENT SIDE SCRIPT: ADVANTAGES

- Allow for more interactivity by immediately responding to users' actions.
- Execute quickly because they do not require a trip to the server.
- May improve the usability of Web sites for users whose browsers support scripts.
- Can give developers more control over the look and behaviour of their Web widgets.
- Can be substituted with alternatives (for example, HTML) if users' browsers do not support scripts
- Are reusable and obtainable from many free resources.

CLIENT SIDE SCRIPT: DISADVANTAGES

- Not all browsers support scripts, therefore, users might experience errors if no alternatives have been provided.
- Different browsers and browser versions support scripts differently, thus more quality assurance testing is required.
- More development time and effort might be required (if the scripts are not already available through other resources).
- Developers have more control over the look and behaviour of their Web widgets; however, usability problems can arise if a Web widget looks like a standard control but behaves differently or vice-versa.

CLIENT SIDE SCRIPT: ADVANTAGES

- Allow for more interactivity by immediately responding to users' actions.
- Execute quickly because they do not require a trip to the server.
- May improve the usability of Web sites for users whose browsers support scripts.
- Can give developers more control over the look and behaviour of their Web widgets.
- Can be substituted with alternatives (for example, HTML) if users' browsers do not support scripts
- Are reusable and obtainable from many free resources.

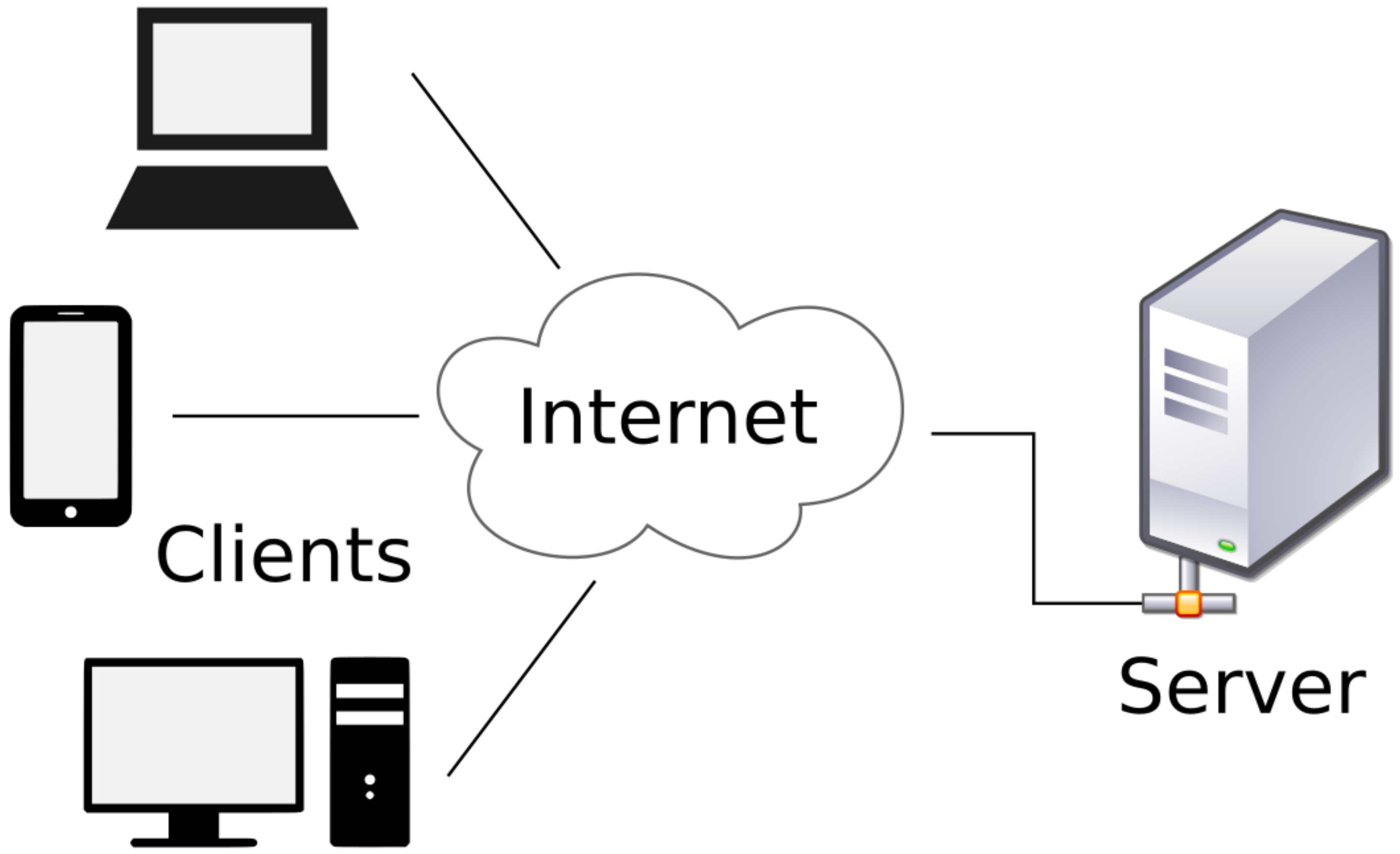
INTERNET TECHNOLOGY

- It is the technology that allows users to access information over the World Wide Web (WWW)
- WWW is internet based service.
- Internet is an infrastructure and WWW is a service
- WWW is collection of HTML pages or websites spread and managed by various web hosting servers
- Client Users request for the HTML page through client browser software like google chrome, Mozilla Firefox, opera etc. and web server replies them with requested HTML page
- Web pages can be static and dynamic

STATIC WEBPAGE

- Website that contains stable contents that are displayed using HTML and CSS only are known as static webpage
- Static webpage provide same information to all the visitors
- Static webpage loads faster and there is no use of database
- Less interactive
- Suitable for small and fixed contents which do not change over time
- Static webpages are less popular than dynamic webpage

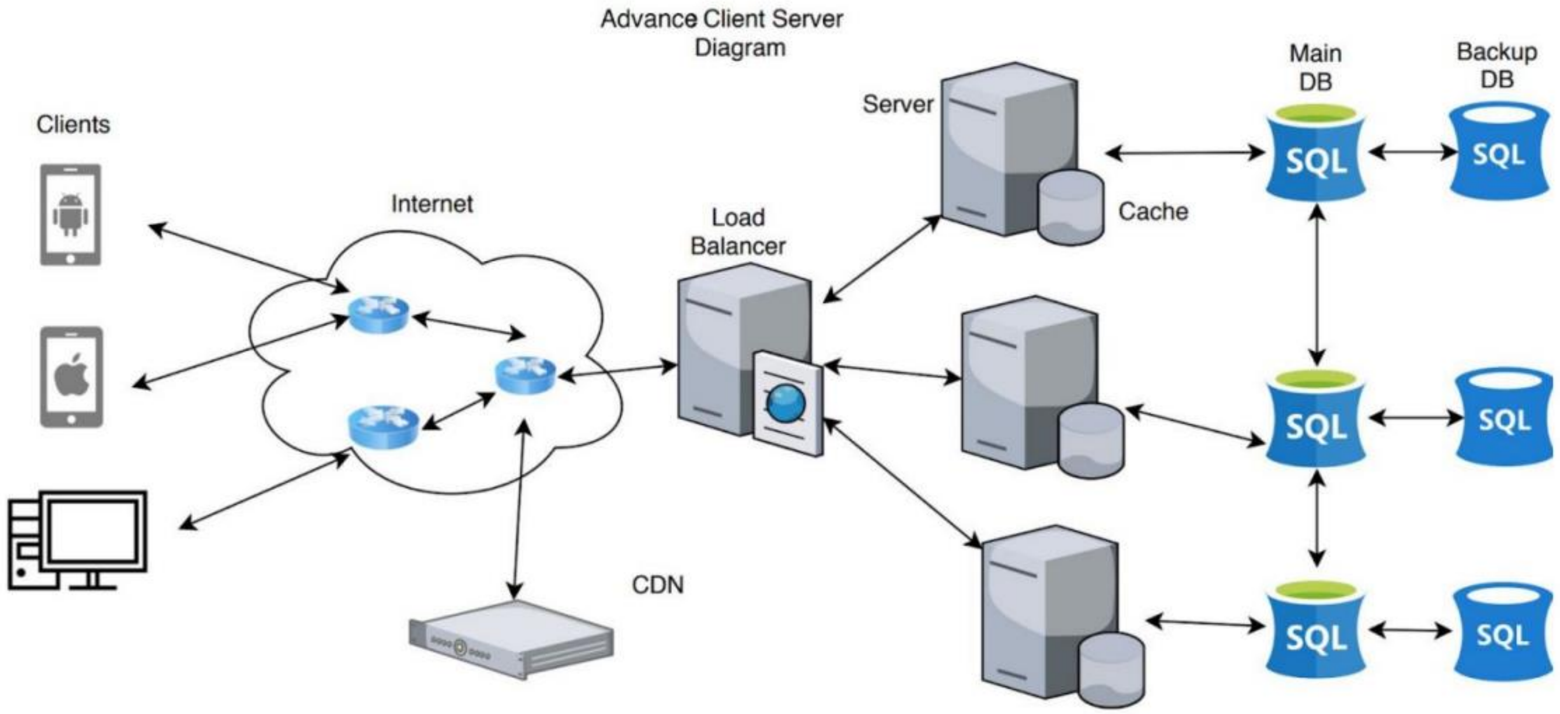
STATIC WEBPAGE



DYNAMIC WEBPAGE

- Dynamic webpages are the one whose content changes dynamically
- It accesses contents from database and the Content Management System (CMS)
- The content of website is also updated when we alter or update the database
- Both client side scripting or server side scripting languages are used to create dynamic websites
- Dynamic websites are more popular such as Facebook, news portals etc.
- the content of dynamic web page is generated each time it is accessed.

CLIENT SERVER ARCHITECTURE



JAVASCRIPT FUNDAMENTAL

- JavaScript is a dynamic scripting language
- It is light weight and most commonly used as part of web pages
- JavaScript code can be inserted into HTML pages and executed by all types of browsers
- It is used to make web pages more dynamic and interactive
- It runs on client side's computers and does not require constant communication to server

ADVANTAGES OF JAVASCRIPT

- Less server interaction
  - validate input before sending, saves server traffic & load
- Easy to learn
  - commands and syntax are easy to learn
- Immediate feedback to visitors
  - fast feedback, don't need to wait for server reply
- Increased interactivity
  - more event handling function such as onclick, onmousehover
- Quick development
  - can be developed in short period of time

ADVANTAGES OF JAVASCRIPT

- Richer interfaces
  - drag-and-drop components, plugins, libraries
- Transmitting information
  - user's frequent searches, web analytics, tracking etc
- Easy debugging and testing
  - Interpreted line-by-line, easy to find errors and update
- Interactive content
  - browser games, playing audio video etc
- Validating input values
  - validate user's form before submitting to server

ADDING JAVASCRIPT TO HTML PAGE

- JavaScript is embedded between <Script> .... </Script> HTML tags.
- JavaScript codes can be written between these tags
- Code can be placed anywhere but normally recommended to keep within <head> tags.

```
<html>
<head><title>JavaScript Code Example</title>
<script type="text/JavaScript">
// JavaScript code goes here
</script>
```

JS CODE - 1

```
<html>
<head><title>JavaScript Code Example</title>
<script type="text/JavaScript">
document.write("Hello World");
</script>
</head>
<body>
</body>
</html>
```

DATA TYPES IN JAVASCRIPT

- Data types specify what type of data can be stored and manipulated
- let n = 7;   //decimal number
- n = 15.5;   //floating point numbers
- var age = "twenty five"; //string
- var colors = ["Red","Green", "Blue"]; //array

DATA TYPES IN JAVASCRIPT

Primitive Data Types

- Numbers
- String
- Boolean
- NULL

Non-Primitive Data Types

- Objects
- Array
- Functions

DATA TYPES IN JAVASCRIPT

- Number
  - number data type can be an integer, floating point value, exponential value, NaN (Not a Number) or infinity
  - Integer number can be represented in decimal, hexadecimal and octal form
  - Floating point numbers are used to represent numbers that requires the use of decimal points
  - Floating point numbers can represent exponential notations also

e.g.
```
var x = 20
var y = 15.5;
var z = 10e2;
```

DATA TYPES IN JAVASCRIPT

- String
  - String data type can be any group of characters enclosed in a single (' ') or double (" ") quotes
  - String is a sequence of one or more characters that are enclosed by quotes
  - If a string begins with single quote, then it must end with single quote
  - If string begins with double quote, then it must end with double quote

e.g.
```
var name = "Devendra Chapagain";      //double quote
var university = "Tribhuban University";
var qualification = 'ME Computer Engineering';
```

DATA TYPES IN JAVASCRIPT

- Boolean
  - Boolean data type is mostly used to check a logical condition
  - It has only two values either TRUE or FALSE
  - It is mostly used in logics and comparison to check condition
  - We can also use 1 for TRUE and 0 for FALSE in JavaScript

e.g.
```
var a = 3;
var b = 6;
if(a<b){
        alert("a is the smaller than b");
}
```

DATA TYPES IN JAVASCRIPT

- NULL
  - Null value means no value, which identifies an empty.
  - Null value is represented by NULL only
  - It is special value which represents "nothing", "empty" or "value unknown"
  - It can be used to initialize variables

  e.g. var a = NULL;

DATA TYPES IN JAVASCRIPT

- Array
  - Array is the collection of similar types of objects or data types
  - It can hold more than one value at a time under single variable name
  - Array contains more than one value with an integer index, where the index starts from 0
  - Index of array is written in square brackets and items are separated by commas.

e.g.

```
var array_name = [item1, item2, item3, ......];
var array_name = new array(item1, item2, item3,.....);
```

DATA TYPES IN JAVASCRIPT

- Function
  - JavaScript function is a block of code designed to perform a particular task
  - Function is a block of reusable code which can be called anywhere in program
  - It helps programmer in writing modular codes
  - It allows programmer to divide or break down the big problem into number of small and manageable functions

```
<script type = "text/JavaScript">
function function_name (parameters-list){
        //statements
}
</script>
```

VARIABLES IN JAVASCRIPT

- Variables
  - Variables are names given to computer memory locations which are used to store values in program
  - Values of variables can be changed during the program execution
  - JavaScript is un-typed or loosely typed language so no need to define data types
  - Variable can hold value of any data type once declared
  - JavaScript variables have two scopes
  - 1. Global Variables:

    Variables with global scope, it can be declared anywhere

    2. Local Variables:

    Variable will be only visible within function where it is defined

RULES FOR JAVASCRIPT VARIABLES

1. Variable names are case sensitive (a & A are not same)

   "num" and "Num" are not same

1. Name must start with an alphabet (a-z) or underscore (_)

   Valid: num, _num

2. Name can contain letters, digits, underscore

   Valid names: num5, odd55num, odd_num

3. Names can not start with digits or numbers

   Invalid names: 6num, 56price

4. Reserved keywords can not be used as names

   Invalid names: var if, var switch, var for

5. Variables are declared using "var" or "let" keywords

   var n; var percent;

6. Variables name can be short (just single letter) or longer

   var a;   var total_obtained_marks;

7. Variable names can not contain space

   Invalid: total num

JAVASCRIPT OPERATORS

- Operators are symbols that tell the compiler to perform some specific tasks
- Operators operates on some operands
- In A+B operation, A & B are operands and + is operator
- Operators can be Unary (i++), Binary (a+b) and Tertiary (a<b ? a : b)
- Various types of operators in JavaScript are:
  - Arithmetic Operators (+,-,*,/,%)
  - Relational Operators (<,>, <=,>=,==,!=)
  - Logical Operators (&&, | |, !)
  - Assignment Operators (=, +=, -=, *=, /=)
  - Conditional Operators (? :)
  - Increment/Decrement Operators (++, --)
  - Bitwise Operators (&, |, ~, ^, <<, >>)

ARITHMETIC OPERATORS

- Operators which are used to perform arithmetic operations such as addition, subtraction, multiplication, division, modulo division are called arithmetic operators
- They are binary operators i.e. it operates on two operands

Operator	Operation	Description	Example	Result (let a= 5, b=3)
+	Addition	Adds two operands	a+b	8
-	Subtraction	Subtracts second from first operand	a-b	3
*	Multiplication	Multiplies two operands	a*b	15
/	Division	Divide first by second operand	a/b	1.6
%	Modulus	Gives remainder of division	a%b	2

RELATIONAL OPERATORS

- Relational operators are used for comparison
- Result of relational operators are always Boolean i.e. Either True(1) or False(0) based on comparison

Operator	**Operation**	**Description**	**Example**	**Result (let a= 1, b=2)**
==	Equal to	checks if the operands are equal	a==b	F
!=	Not Equal to	checks if the operands are not equal	a!=b	T
<	Less than	checks if first is less than second	a<b	T
>	Greater than	checks if first is greater than second	a>b	F
<=	Less than or equal to	checks if the first is less than or equal to second	a<=b	T
>=	Greater than or equal to	checks if the first is greater than or equal to second	a>=b	F

LOGICAL OPERATOR

- Logical Operators are used to evaluate the expressions which may be True or False
- Logical expression can have either True or False
- We can combine multiple expressions using logical operators

Operator	Operation	Description	Example	Result (let a= 5, b=3)
&&	AND	Returns True if both expressions are True otherwise false	(a>b)&&(a>8)	False
||	OR	Returns True if any one or both expression is True	(a>b)||(a>8)	True
!	NOT	Equivalent to NOT Operation	a!=b	True

CONDITIONAL (TERNARY) OPERATOR

- Conditional operator is also called ternary or tertiary operator because it takes 3 operands
- syntax:(conditional expression) ? (statement if true) : (statement if false)

Operator	Operation	Description	Example	Result (let a= 5, b=3)
? :	Conditional Operator	Takes 3 operands. Checks if the conditional expression is True or False. Executes Statement right after question mark if condition is True otherwise statement after colon is executed	(a<b)? a : b	Returns a if a is smaller else b is returned

INCREMENT/DECREMENT OPERATORS

- Increment/Decrement operators are used to increase or decrease values of variables or expressions by single unit (1)
- Each has two types: Pre and Post

Operator	Operation	Description	Example	Result (let a= 5)
++a	Pre Increment	Increases value of a by 1 before other operations are performed	++a	6
a++	Post Increment	Increases value of a by 1 after other operations are performed	a++	6
--a	Pre Decrement	Decreases value of a by 1 before other operations are performed	--a	4
a--	Post Decrement	Decreases value of a by 1 after other operations are performed	a--	4

BITWISE OPERATORS

- Any operators which are used to operate at bits level of any value are called bitwise operators
- It operates on one or more bit patterns or binary numerals at the level of their individual bits

Operator	Operation	Description	Example	Result (let a= 1, b=2)
&	Bitwise AND	Binary equivalents are performed AND operation	c=a&b	00
|	Bitwise OR	Binary equivalents are performed OR operation	c=a|b	11
~	Bitwise NOT	Binary equivalents are inverted	c=~a	10
^	Bitwise XOR	Both binary equivalents are XORed	c=a^b	11
<<	Left Shift	Binary equivalents are shifted left	c=a<<2	
>>	Right Shift	Binary equivalents are shifted right	c=a>>3	

ASSIGNMENT OPERATORS

- Assignment operators assigns value of right operand or expression to the left operand

Operator	Operation	Description	Example	Equivalent
=	simple assignment	Assigns value of right operand to left	a=b	
+=	Addition assignment	Adds left operand value with right and assigns to left operand	a+=b	a = a+b
-=	Subtraction Assignment	Subtracts left operand value with right and assigns to left operand	a-=b	a=a-b
*=	Multiplication Assignment	Multiplies left operand value with right and assigns to left operand	a*=b	a=a*b
/=	Division Assignment	Divides left operand value by right and assigns to left operand	a/=b	a=a/b
%=	Modulo Assignment	Divide left operand value by right and assigns remainder to left operand	a%=b	a=a%b

JAVASCRIPT FUNCTIONS

- Function is a block of reusable code designed to perform a particular task.
- Function block can be called from anywhere in program
- It helps modular programming and divide large problem into multiple small & manageable codes
- Advantages:
  - JavaScript functions are must.
  - Increases code reusability
  - Help to structure modular coding and reduce complexity
  - Make code readable and easily extendable
  - Can be called from anywhere once defined
  - Divide big problem into number of small manageable functions

FUNCTION DEFINITION

- Functions are defined by using function keyword followed by unique function name, and list of parameters and statement of block surrounded by curly braces.
- Syntax:

```
<script type="text/JavaScript">
function function_name(parameter-list){

    //block of codes


    }
    </script>
```

FUNCTION CALLING

- To call the function defined we write name of function followed by parenthesis and semicolon
- Function will start executing only after calling

```
<script type="text/javaScript">
function sayHello(){
        document.write("Hello Everyone");
}

sayHello();
</script>
```

FUNCTION PARAMETERS

- A function can take multiple parameters separated by comma
- Function parameter can have value of any data type

```
<script type="text/javaScript">
function add(var a, var b){
var sum = a+b;
document.write("Sum="+sum);
}
add(3,4);
</script>
```

JAVASCRIPT CONTROL STRUCTURES

- A control structure refers to the flow of execution of the program
- Control structure enables us to specify the order in which the various instructions in a program are to be executed

1. Selection Control Structure
   1. If Statement
   2. If … else statement
   3. If … else if … else statement
   4. Switch statement
2. Looping Control Structure
   1. For loop
   2. While loop
   3. Do … while loop

IF STATEMENT

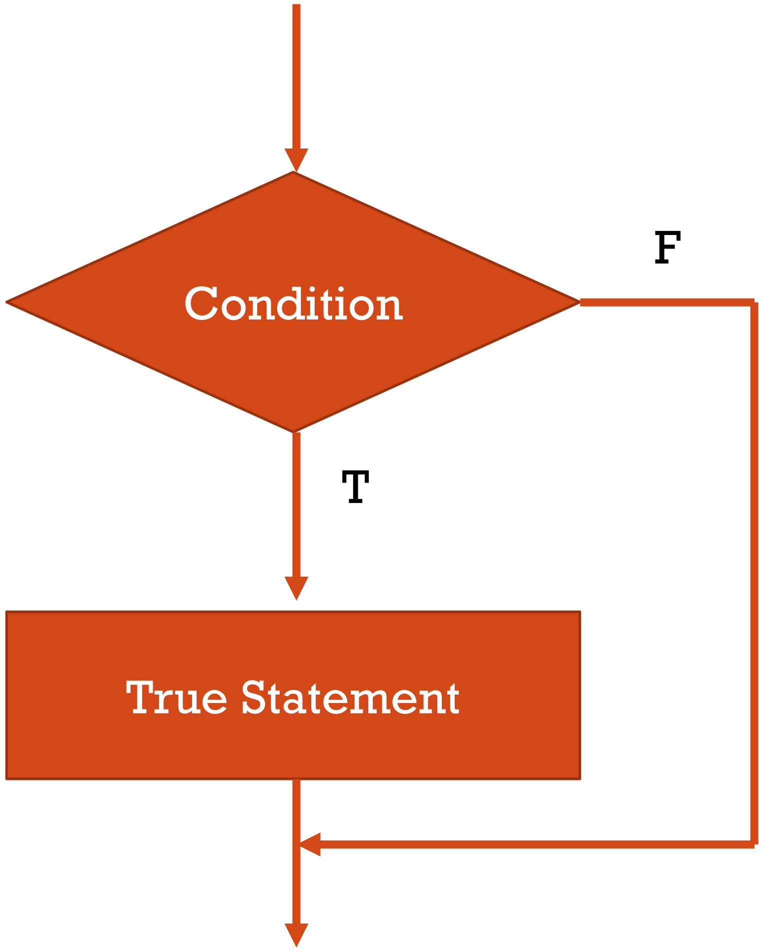
- If statement is the fundamental control statement that allows JavaScript to make decisions and execute statements conditionally
- Syntax

```
if(condition){
        //statements to be executed if condition is true
}
```

E.g.

```
var percent;
if(percent>=40){
        document.write("Exam Passed");
}
```

IF STATEMENT



IF … ELSE STATEMENT

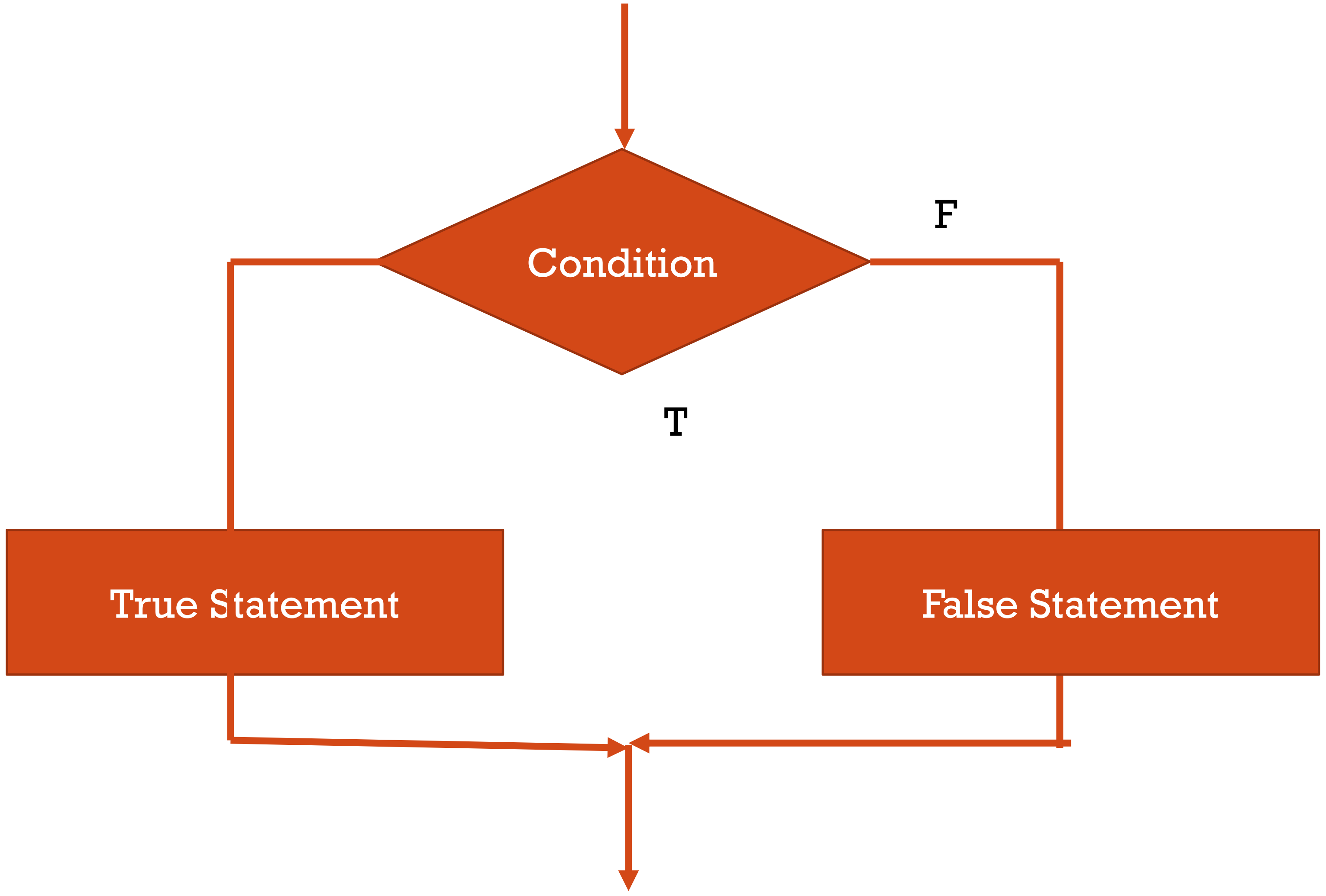
- It offers two way to execute statements based on conditions
- Syntax

```
if(condition){
        //statements to be executed if condition is true
}else{
        //statements to be executed if condition is false
}
```

E.g.

```
var percent;
if(percent>=40){
        document.write("Exam Passed");
}else{
        document.write("Exam Failed");
}
```

IF … ELSE STATEMENT



IF … ELSE IF … ELSE STATEMENT

- It offers multiple way to execute statements based on conditions
- Syntax

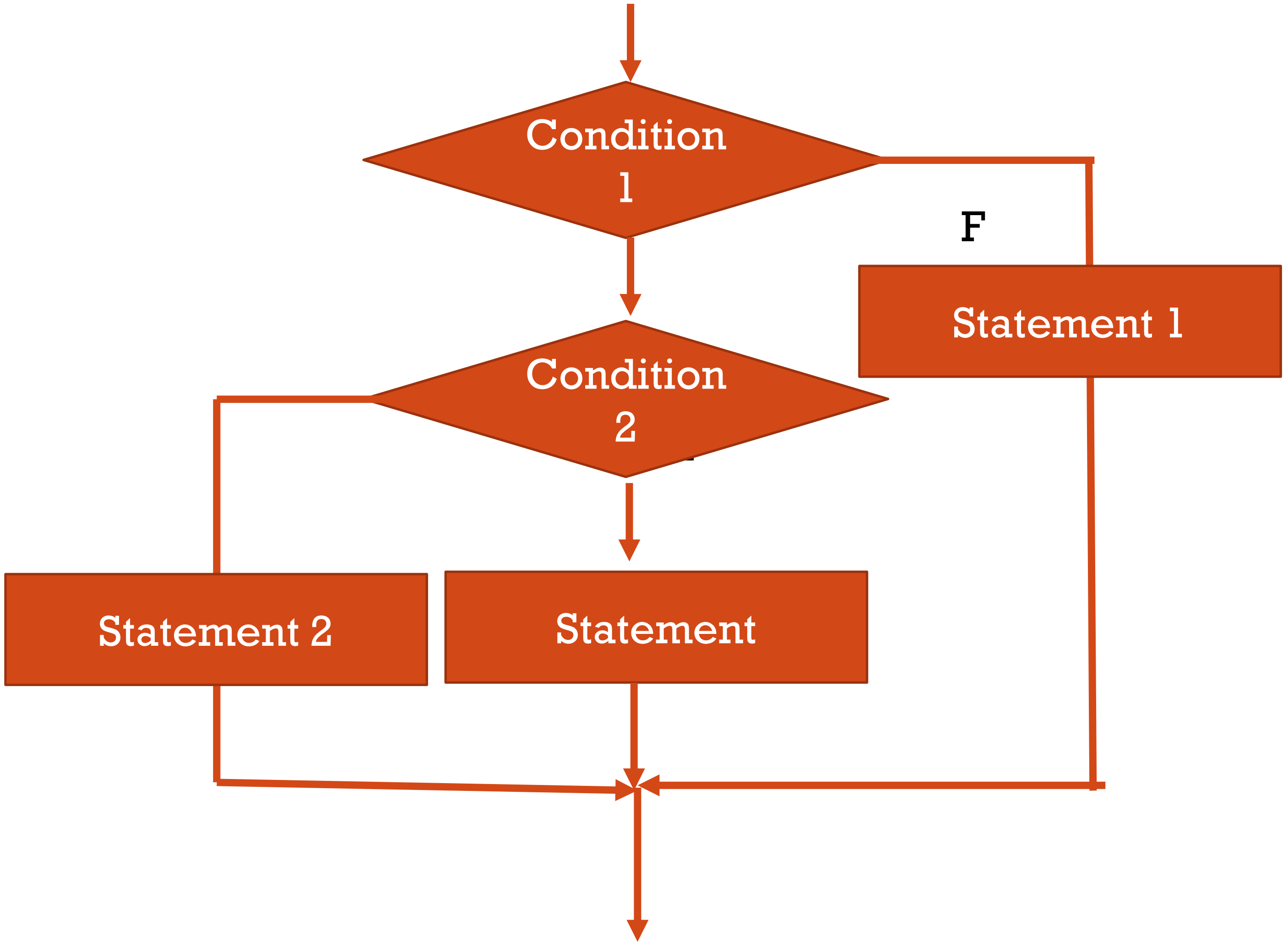
```
if(condition 1){
        //statements to be executed if condition 1 is true
}else if(condition 2){
        //statements to be executed if condition 2 is true
} else{
        //statement if no condition is true
}
```

IF … ELSE IF … ELSE STATEMENT

E.g.

```
var grade;
if(grade == 'A'){
        document.write("Excellent");
}else if(grade == 'B'){
        document.write("Good");
} else if(grade == 'C'){
        document.write("Fine");
}else if(grade =='F'){
        document.write("Bad");
}else{
        document.write("Not valid grade");
}
```

IF … ELSE … IF STATEMENT



SWITCH STATEMENT

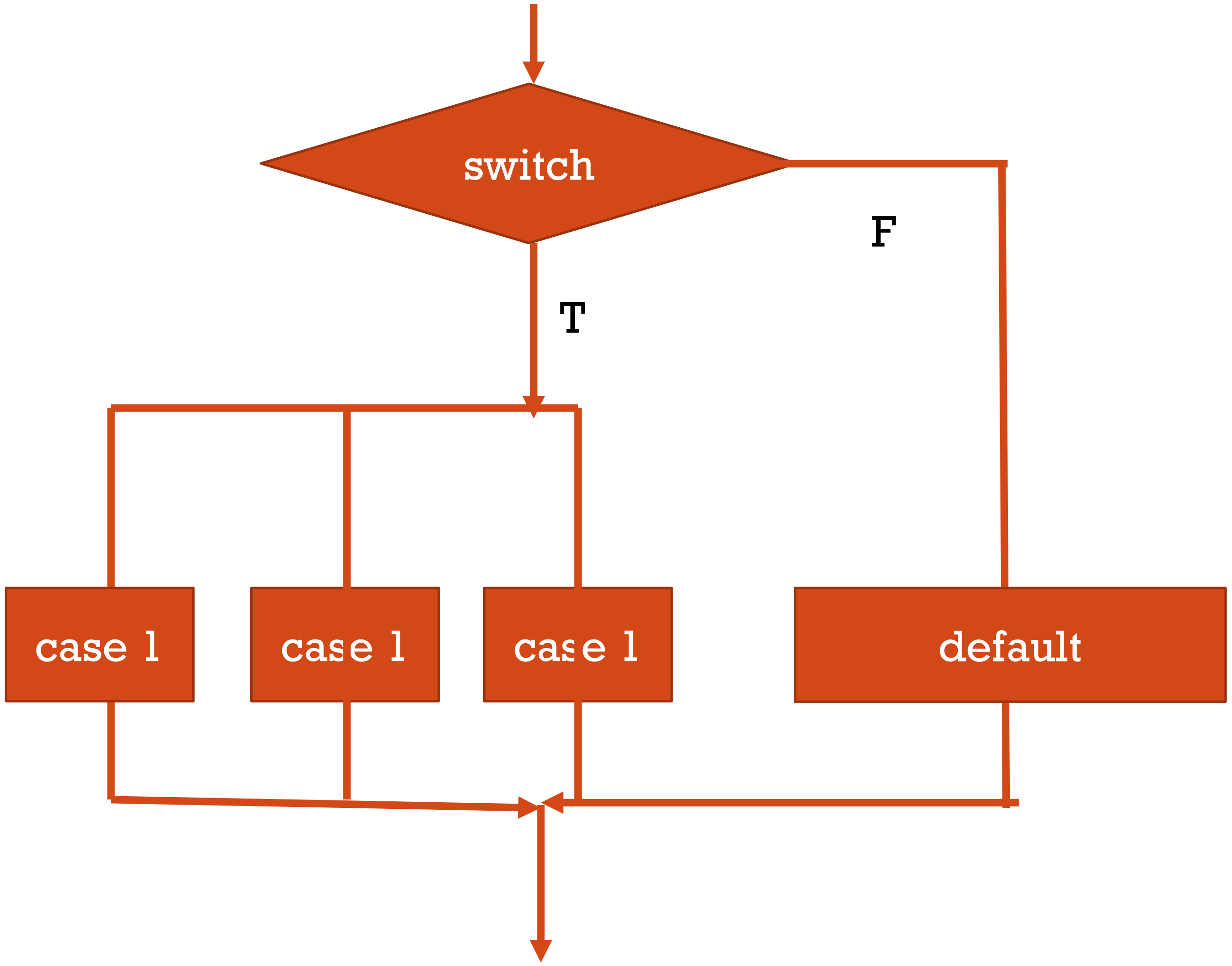
- Switch statement is similar to if-else if ladder statement
- Instead of checking conditions one by one it directly jumps to the condition matched
- Syntax

```
switch(expression){
        case value1:
        //statements for value 1
        break;
        case value2:
        //statements for value 2
        break;
        default:
        //default statements
}
```

SWITCH STATEMENT

```
var key;
switch(key){
        case 'A':
        document.write("Apple");
        break;
        case 'B':
        document.write("Ball");
        break;
        case 'C':
        document.write("Cat");
        break;
        default:
        document.write("Invalid selection");
}
```

SWITCH STATEMENT



SOME EXERCISE

1. Write a program that checks any variable n is odd number or even.
2. Write a program that displays "outstanding result" if the value of percent variable is 100%
3. Write a JS code to show whether a number is negative, zero or positive.
4. Write a program to show the day name for equivalent number using switch.

LOOPING CONTROL STRUCTURES

- Looping control structures are used when we want to execute a block of statement repeatedly
- loop executes sequence statements many times until some condition becomes false or under some criteria

1. Looping Control Structure
    1. For loop
    2. While loop
    3. Do … while loop

FOR LOOP

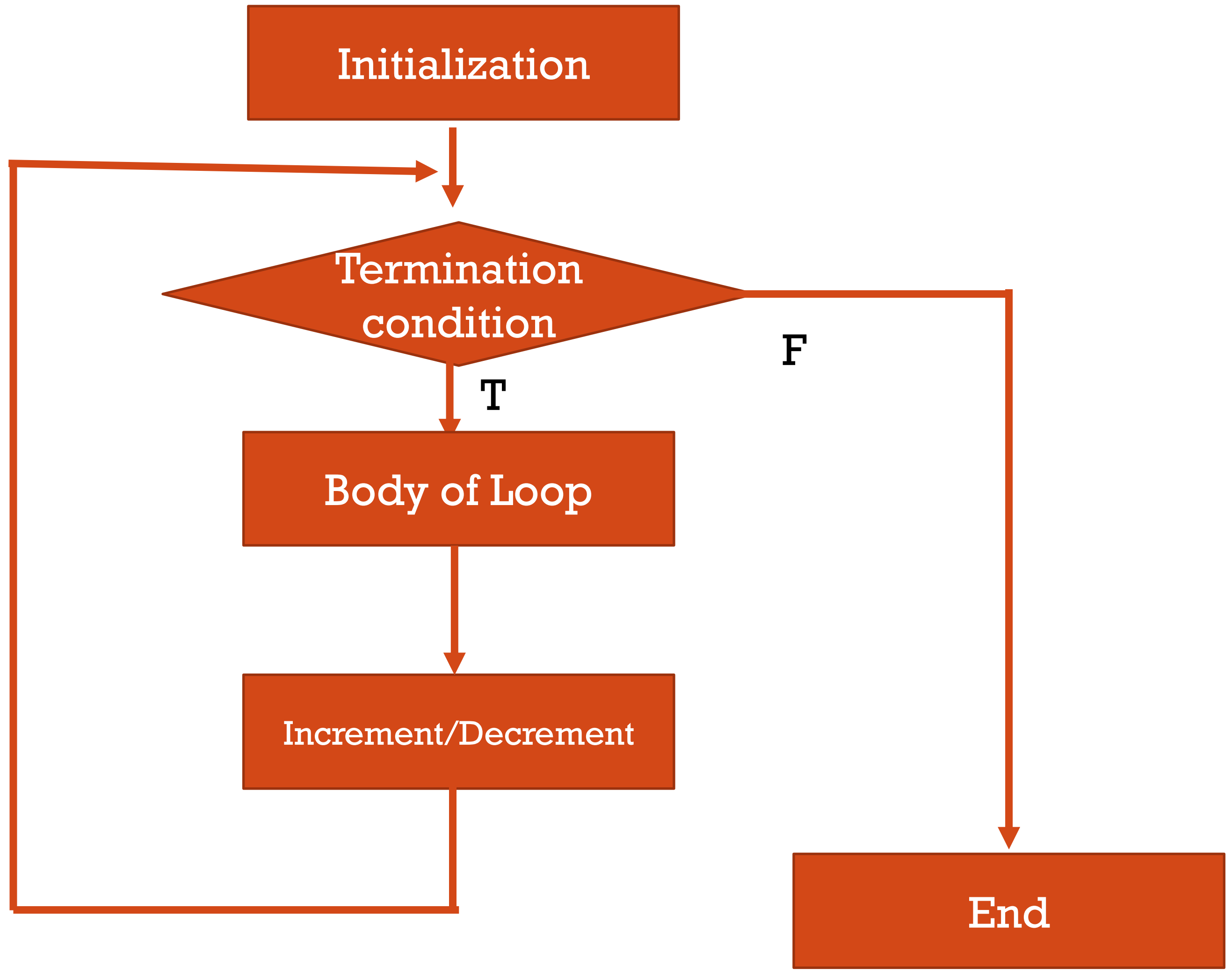
- For loop is the most common loop used
- It is entry control loop because it checks condition at entry point
- It does not execute even single statement if the termination condition is false
- Syntax

```
    for(initialization; condition; steps){
            //body of loop
    }
```

FOR LOOP

```
var i;
for(i=1; i<5; i++){
        document.write("The number is ");
        document.write(i);
}
```

FOR LOOP


Initialization
Termination condition
F
T
Body of Loop
Increment/Decrement
End

WHILE LOOP

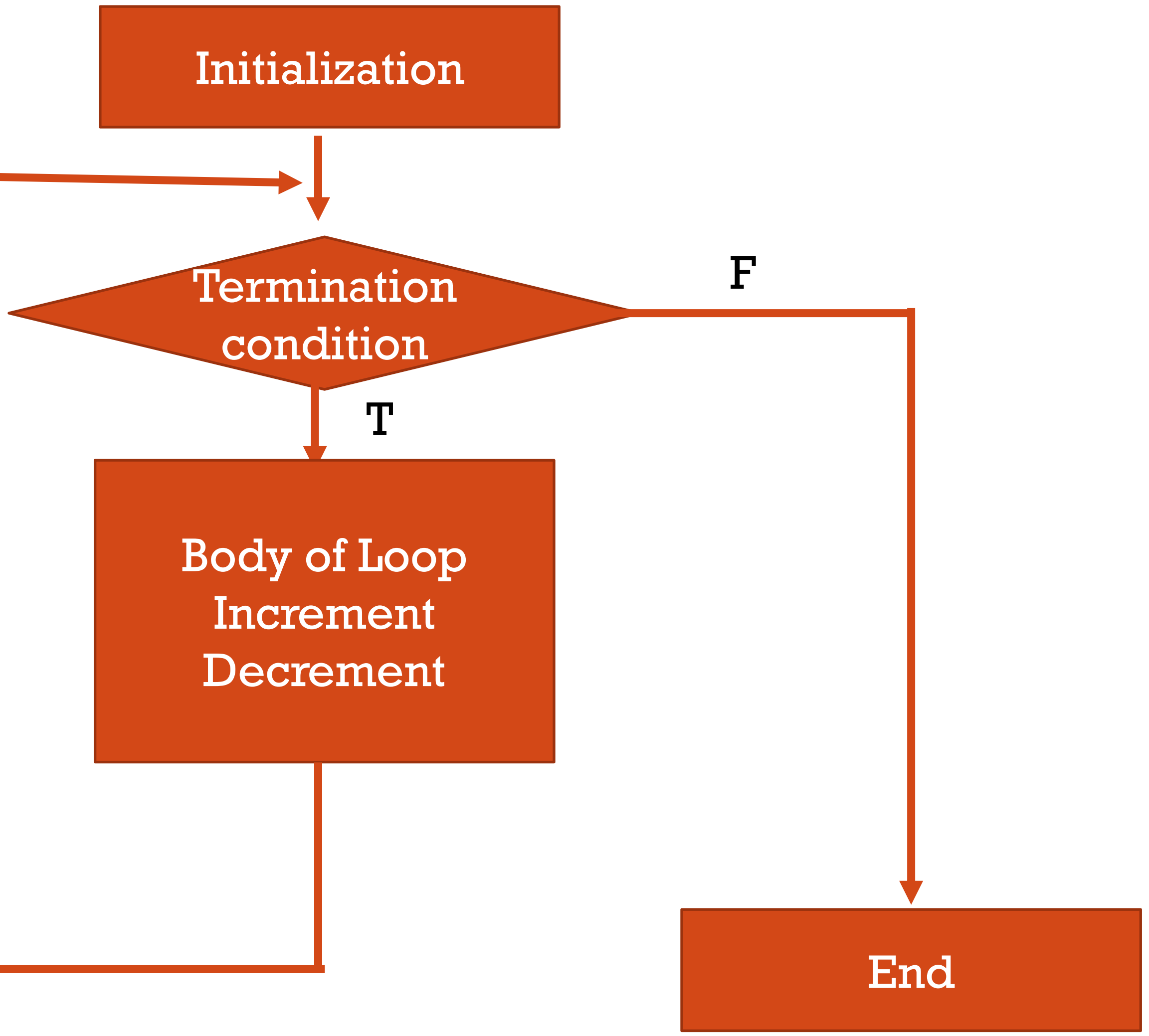
- While loop is an entry controlled loop. i.e. condition is checked first before executing block of statements
- We need to use increment/decrement statement inside while loop which gets changed on each iteration and at some point condition returns false
- Loop can be executed for uncertain number of times until termination condition is met
- Syntax

```
while(condition){
        //block of statements
        //increment or decrement statements
}
```

WHILE LOOP

```
var i=1;
while(i<10){
        document.write("The number is ");
        document.write(i);
        i++;
}
```

WHILE LOOP


Initialization
Termination condition
F
T
Body of Loop
Increment
Decrement
End

DO-WHILE LOOP

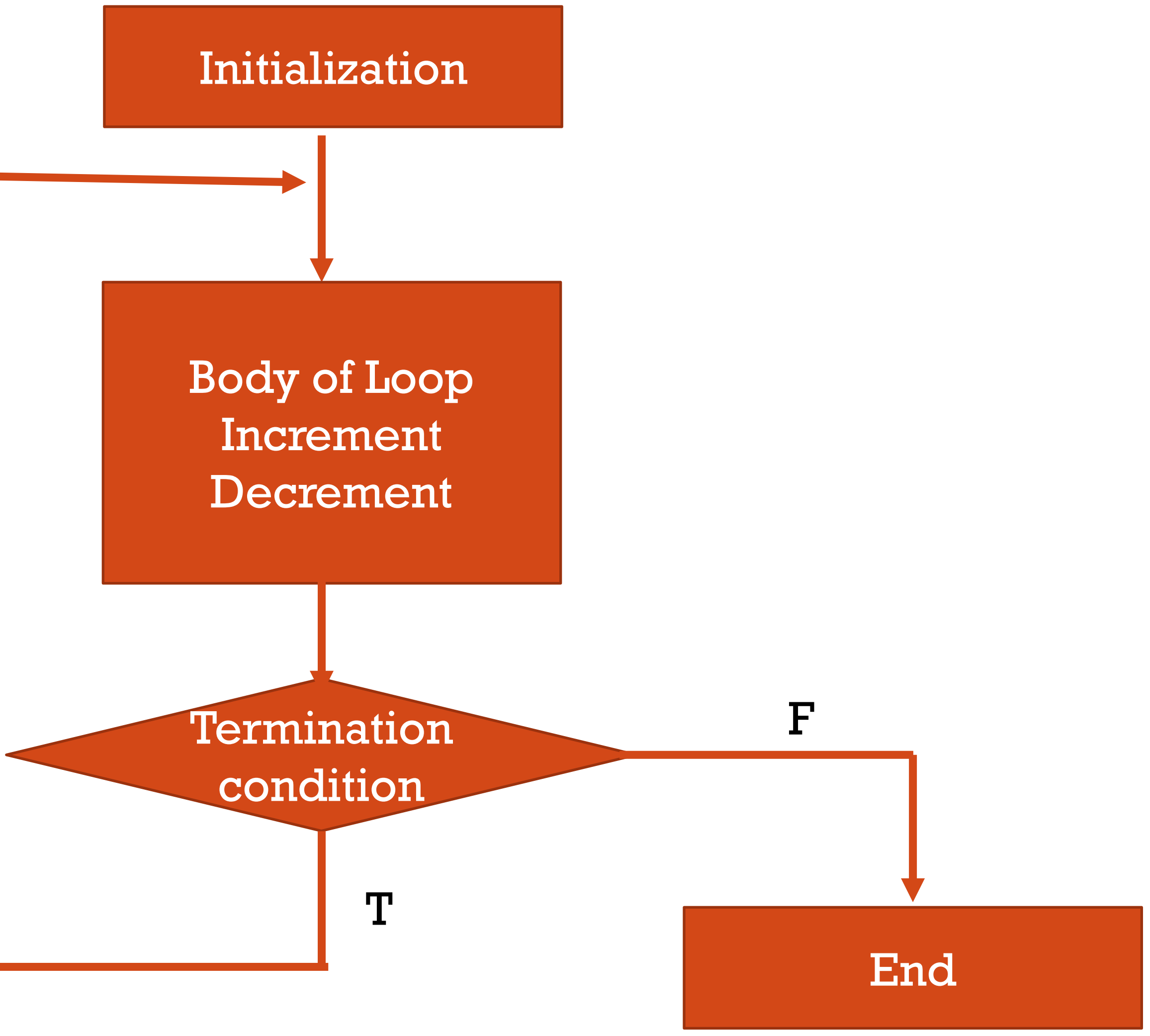
- Do-While loop is an exit controlled loop. i.e. condition is checked at last after executing block of statements
- We need to use increment/decrement statement inside while loop which gets changed on each iteration and at some point condition returns false
- Loop can be executed for uncertain number of times until termination condition is met
- Syntax

```
do{
        //statement
        //increment or decrement
}while(condition);
```

DO-WHILE LOOP

```
var i=10;
do{
        document.write("The number is ");
        document.write(i);
        i++;
}while(i<10);
```

DO-WHILE LOOP



SOME EXERCISE

1. Write a program that displays Fibonacci number less than 20
2. Write a program to display odd numbers between 50 and 100
3. Write a program to generate random number between 1-6. Continue looping until 1 or 6 is obtained. (Use random function)

JAVASCRIPT POPUP BOXES

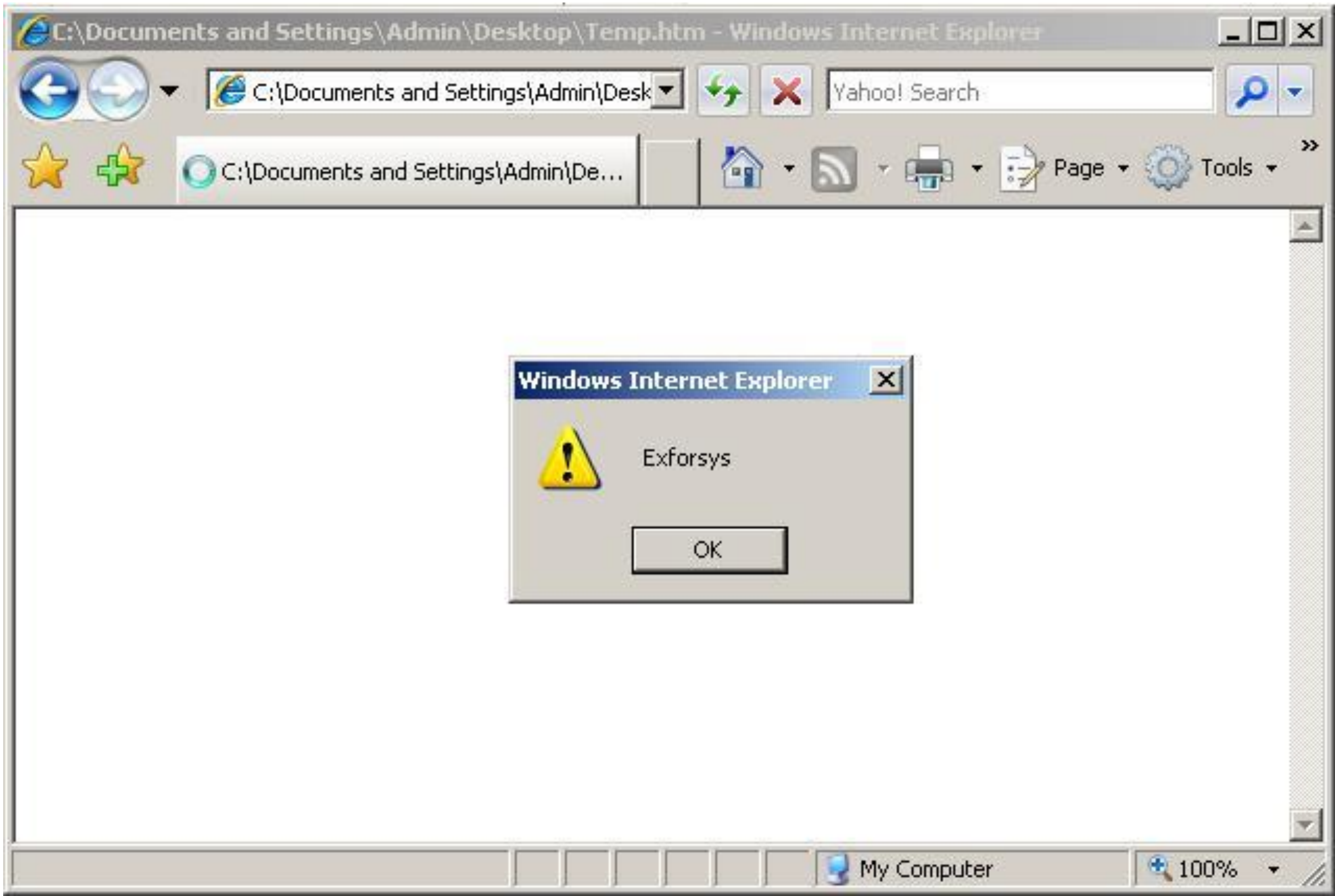
- JavaScript supports 3 types of popup boxes
- They are also called dialog boxes
- Dialog boxes increases user interaction
  1. Alert Dialog box
     - mostly used for warning message or information
  2. Confirmation Dialog Box
     - mostly used for confirmation or users permission
  3. Prompt Dialog Box
     - mostly used for accepting text input from user

ALERT DIALOG BOX

- Alert dialog box is mostly used to give a warning message or information to the users
- alert box has only one "OK" button to select and proceed
- Alert box does not return any value
- Syntax:

```
alert("This is alert message");
```

ALERT DIALOG BOX


C:\Documents and Settings\Admin\Desktop\Temp.htm - Windows Internet Explorer
C:\Documents and Settings\Admin\Desk
Yahoo! Search
C:\Documents and Settings\Admin\De...
Page
Tools
My Computer
100%

Windows Internet Explorer

Exforsys

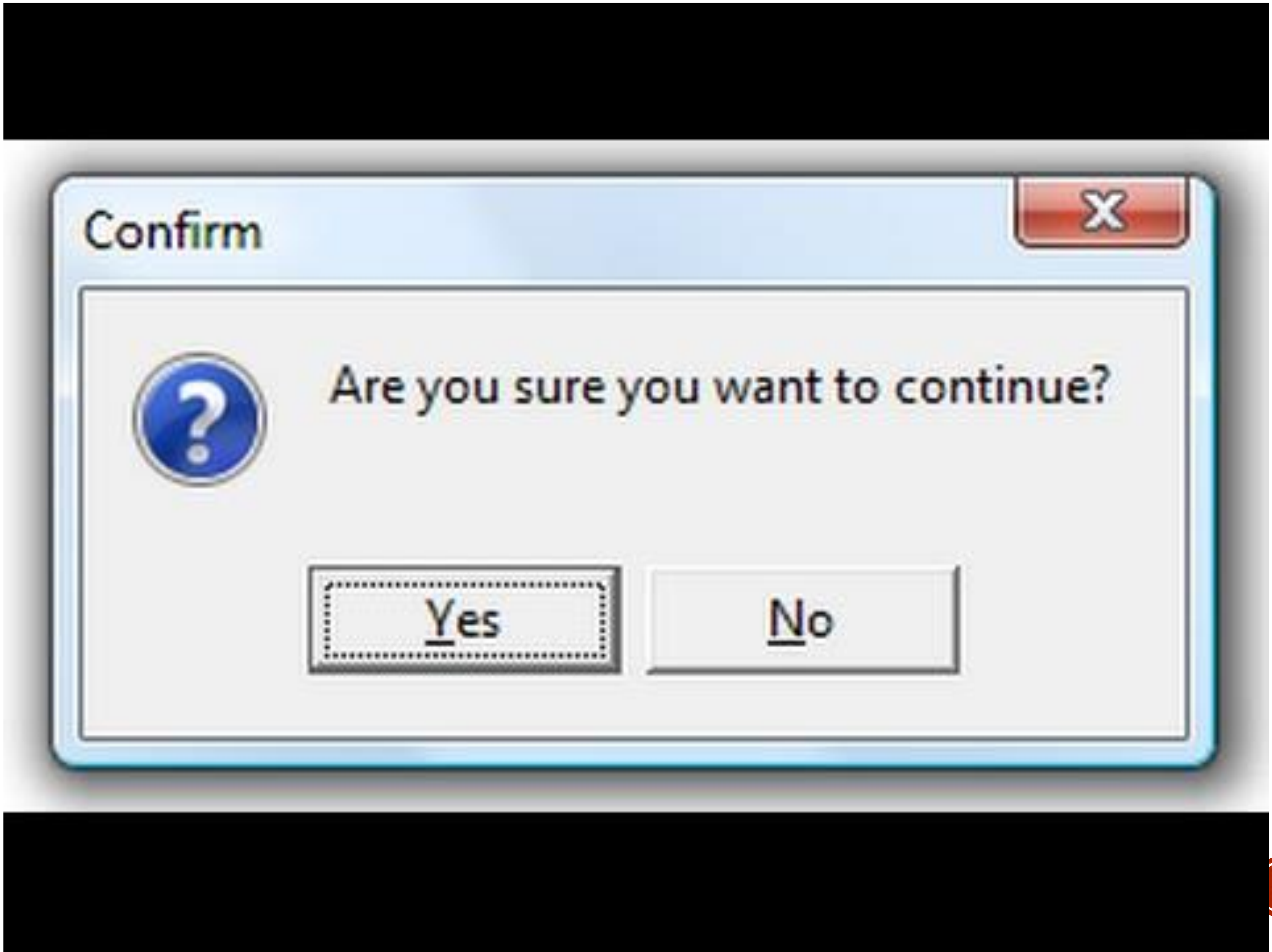
OK

CONFIRMATION DIALOG BOX

- Confirmation box is mostly used to take user's opinion or confirmation on any option
- It displays dialog box with two buttons: "OK" and "Cancel"
- If the user clicks "OK" then it returns TRUE value. If the user clicks "Cancel" button, then it returns FALSE value
- Syntax:

```
        var retVal = Confirm("Do you want to continue?");
```

CONFIRMATION DIALOG BOX


Confirm

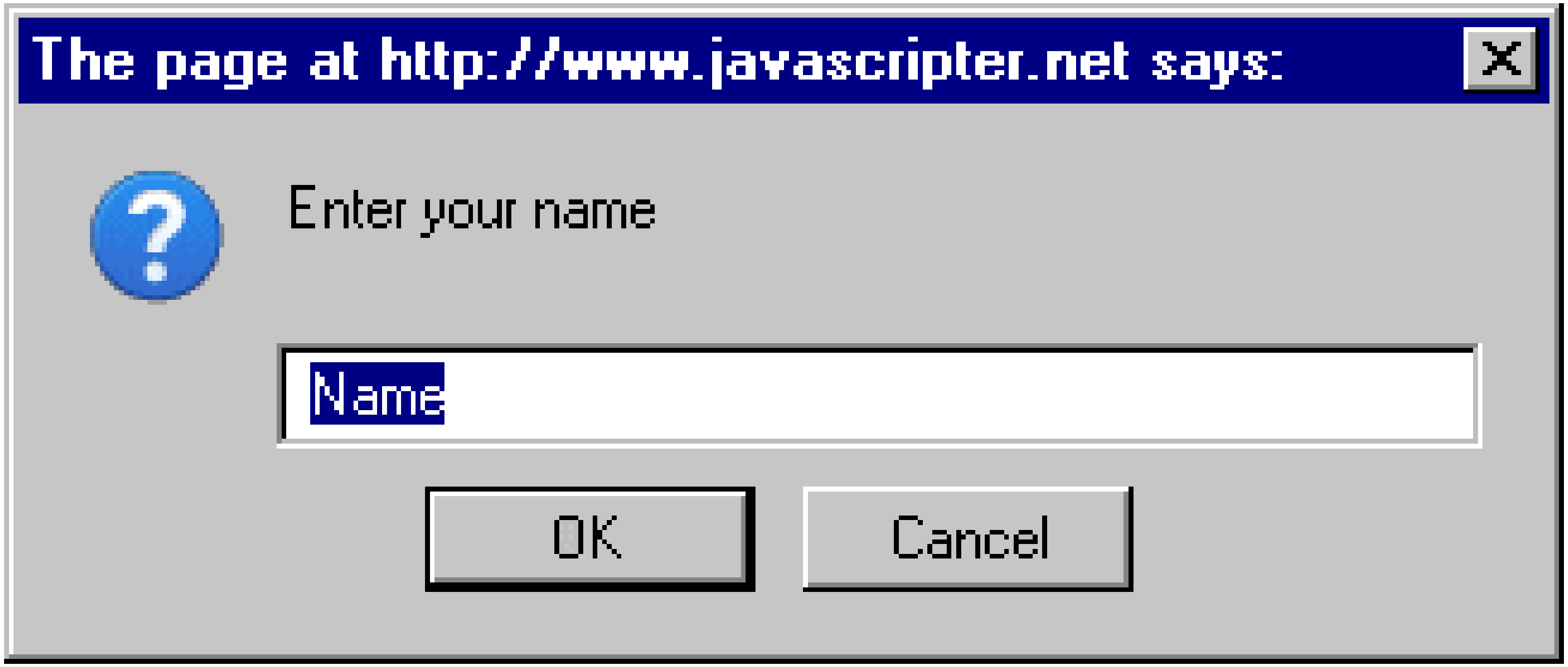
Are you sure you want to continue?

PROMPT DIALOG BOX

- Prompt dialog box is mostly used to accept some input from user.
- It prompts dialog box with text box and "OK" and "Cancel" button
- Prompt box returns TEXT value entered by user
- We can save value returned by prompt box to any variable we define
- Syntax

```
var result = prompt("Enter Your Name:");
```

PROMPT DIALOG BOX


The page at http://www.javascripter.net says:

Enter your name

Name