Java script

Day-3

Agenda

- To explain the necessity of Scripting
- To explain writing client side scripting using JavaScript.
- To discuss about the in-built and user defined objects.
- To explain event handling using JavaScript.

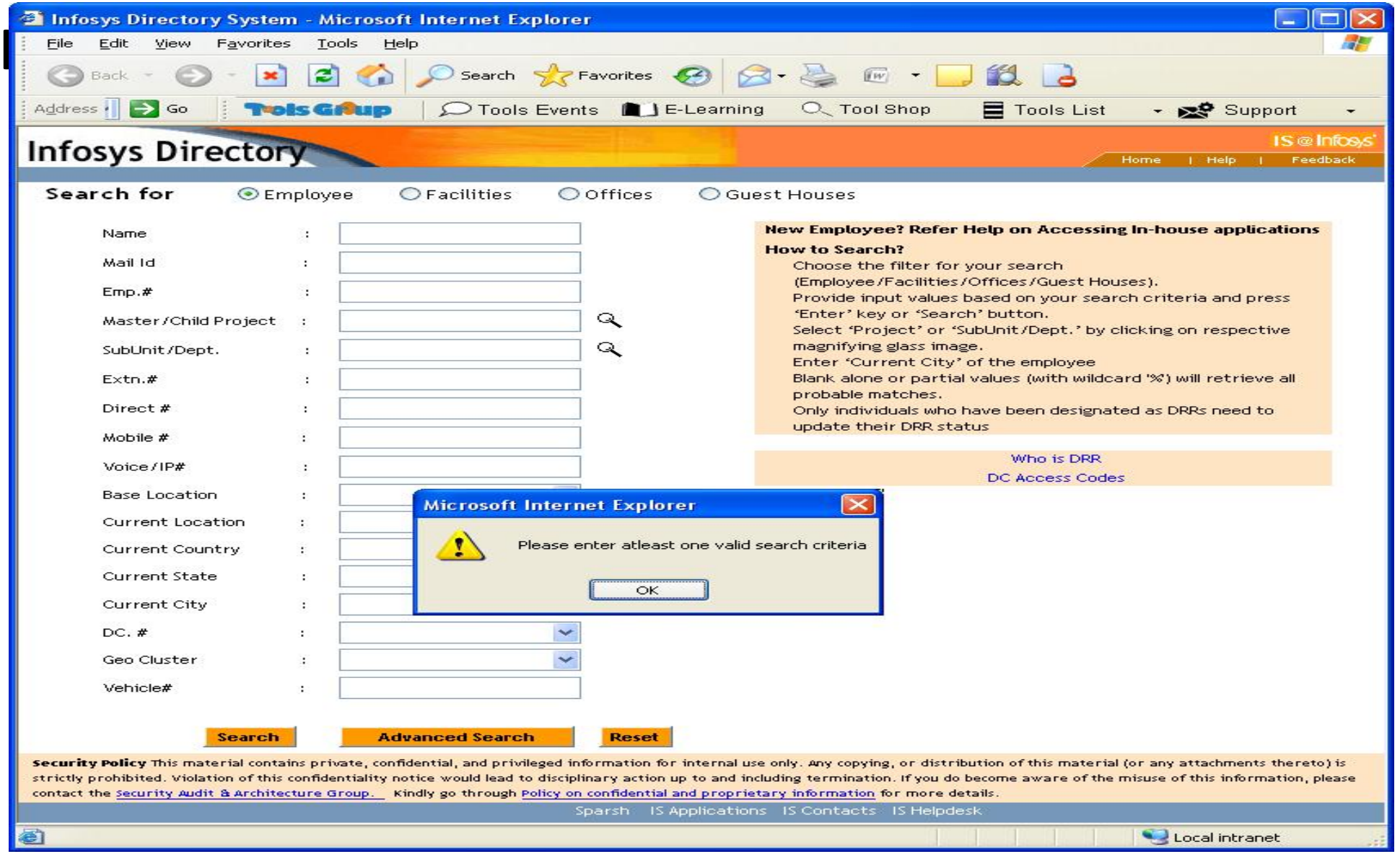
Introduction to Scripting

- Scripting Languages are mainly used to build the programming environment in HTML document
- Make Web pages dynamic and interactive.
- Some languages : VBScript, JavaScript, Jscript and ECMA Script
- Browser Includes Scripting Interpreter
- Choosing a Scripting Language
  - Browser compatibility
  - Programmer familiarity
- Scripts can be executed on client or the server (JavaScript can be used with client or server)

Client Vs. Server Scripting

Client Side Scripting	Server Side Scripting
Runs on the user's computer i.e. Browser interprets the script.	Runs on the Web server and sends the output to the browser in HTML format
Source code is visible to the user. (Source code is downloaded to the client and executed in browser)	Source code is not visible to the user. Server side source is executed on server.
Used for client side validations and functionality for the user events.	Used for business logic and data access from the database. The pages are created dynamically.
Depends on the browser and version.	Not depend on the client, any server side technology can be used.

Sa


Infosys Directory System - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address Go
Tools Events E-Learning Tool Shop Tools List Support
Infosys Directory
IS @ Infosys
Home | Help | Feedback
Search for Employee Facilities Offices Guest Houses
Name :
Mail Id :
Emp.# :
Master/Child Project :
SubUnit/Dept. :
Extn.# :
Direct # :
Mobile # :
Voice/IP# :
Base Location :
Current Location :
Current Country :
Current State :
Current City :
DC. # :
Geo Cluster :
Vehicle# :
New Employee? Refer Help on Accessing In-house applications
How to Search?
Choose the filter for your search (Employee/Facilities/Offices/Guest Houses).
Provide input values based on your search criteria and press 'Enter' key or 'Search' button.
Select 'Project' or 'SubUnit/Dept.' by clicking on respective magnifying glass image.
Enter 'Current City' of the employee
Blank alone or partial values (with wildcard '%') will retrieve all probable matches.
Only individuals who have been designated as DRRs need to update their DRR status
Who is DRR
DC Access Codes
Microsoft Internet Explorer
Please enter atleast one valid search criteria
OK
Search
Advanced Search
Reset
Security Policy This material contains private, confidential, and privileged information for internal use only. Any copying, or distribution of this material (or any attachments thereto) is strictly prohibited. Violation of this confidentiality notice would lead to disciplinary action up to and including termination. If you do become aware of the misuse of this information, please contact the Security Audit & Architecture Group. Kindly go through Policy on confidential and proprietary information for more details.
Sparsh IS Applications IS Contacts IS Helpdesk
Local intranet

History of JavaScript

- In 1995, Netscape Communications introduced ‘LiveScript’, a Web Scripting Language

- Support for LiveScript began in June, 1995, with the release of Beta Version 2.0b1 of Netscape Navigator.

- Later in 1995, after an agreement with Sun, LiveScript was re-named JavaScript, to leverage the popularity of Java.

- To this date, JavaScript continues to evolve...

Features of JavaScript

- An interpreted scripting language
- Embedded within HTML
- Minimal Syntax- Easy to learn(C syntax and java OOC)
- Mainly used for client side scripting because it is supported by all the browsers.
- Designed for programming user events
- Platform Independence/ Architecture Neutral

Embedding JavaScript into HTML page

- <SCRIPT>…..</SCRIPT> tag

- LANGUAGE - the scripting language used for writing scripts

```
<SCRIPT LANGUAGE="JavaScript">
-----
      (JavaScript code goes here)
    ----
</SCRIPT>
```

JavaScript is NOT supported by old browsers (IE 1.0).
You can enable or disable JS in new browsers

Deferred and Immediate Script

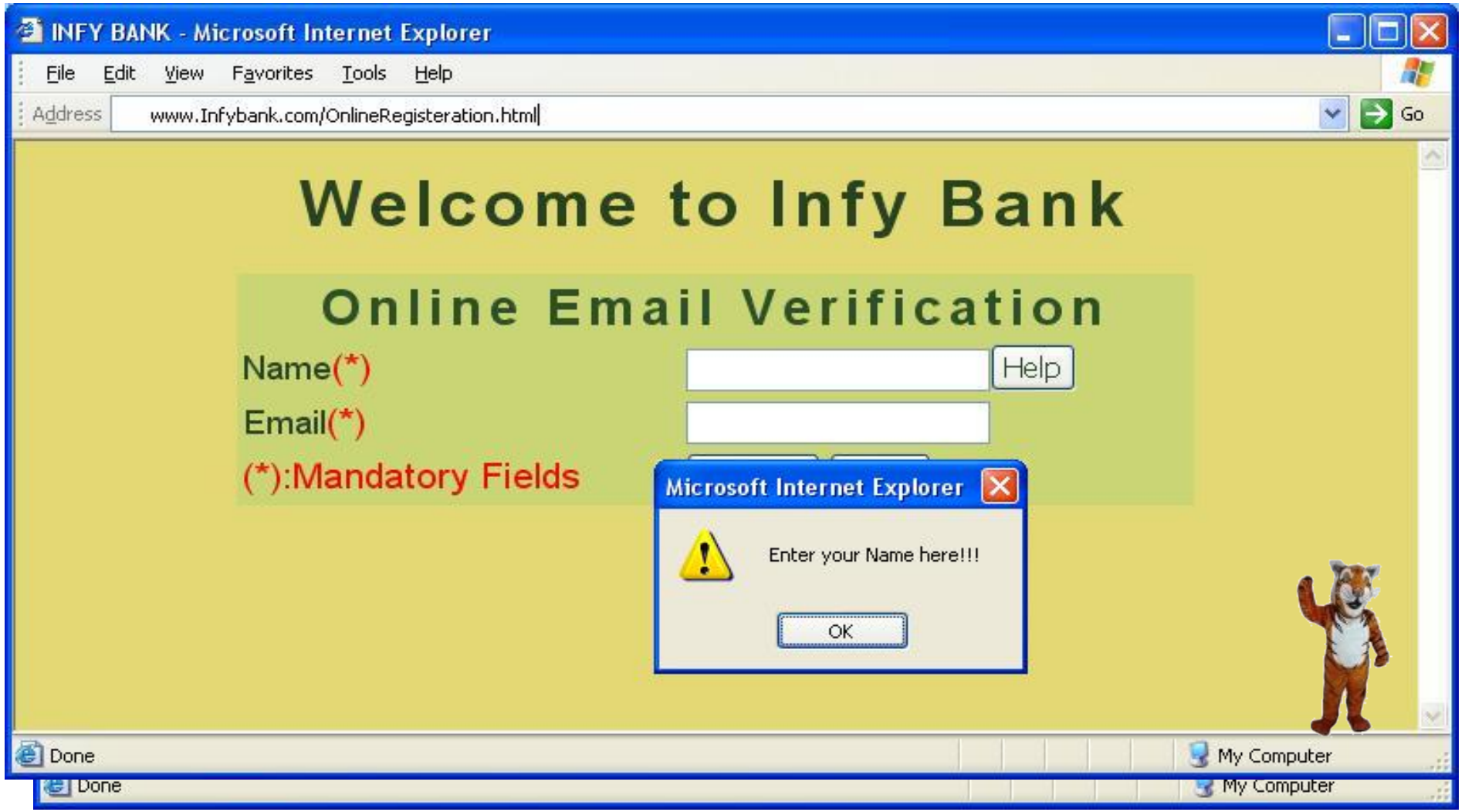
- SCRIPT tag can be placed in HEAD or BODY tag
- Placing Javascript in the HEAD tag ensures readability.
- Immediate mode
  - Scripts gets executed as the page loads.

```
<body>
<h4> Immediate Demo</h4>
<script language="JavaScript">
document.write("<h5> Using JavaScript</h5>");
</script>
</body>
```

Deferred and Immediate Script

- Deferred mode
  - Script is executed based on some user action

```
<script language="JavaScript">
<!--
/*calling function when user clicks on the button */
    function msg(){
        alert("Hi");
    }
// -->
</script>
<form name="f1">
    <input type="button" value=" ok "
onClick="msg()">
</form>
```


INFY BANK - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address www.Infybank.com/OnlineRegisteration.html
Go
Welcome to Infy Bank
Online Email Verification
Name(*)
Help
Email(*)
(*):Mandatory Fields
Microsoft Internet Explorer
Enter your Name here!!!
OK
Done
My Computer
Done
My Computer

Which of the following statements are TRUE about JS

A <SCRIPT> can be placed in HEAD or BODY tag

TRUE

B Compiled by browser

FALSE

C JavaScript is a mark up language

FALSE

D JavaScript is the ONLY client side scripting language

FALSE

JavaScript – lexical structure

- JavaScript is object based and action-oriented.
- JavaScript is case sensitive.
- A semicolon ends a JavaScript statement
- C-based language developed by Netscape
- Comments
  - Supports single line comments using //
  - and multi line comments using /*…..*/

JavaScript –Variables

- Declared using the keyword var. Declaring variables is not mandatory.
- Must start with a letter or an underscore and can have digits.
- Does not have explicit data types.
- The Data type is automatically decided by the usage.
- Scope is by default global. If a variable is prefixed by the keyword “var” within a function then it is a local variable.
- The formal parameters are local to the function.

```
function demo()
{
   var inum1 = 10; // Local to the function
   inum2 = 20;     // Global to the document.
}
demo();              // Invoking function
inum1 = inum1+1;    //Error because inum1 is local variable
inum2 = inum2+1;    // no Error
```

How to declare a local variable in JavaScript

A

Declare using var keyword

FALSE

B

Declare without var keyword

FALSE

C

Declare using var keyword inside a function

TRUE

D

Declare using var keyword outside the function

FALSE

JavaScript – Implicit data types

- JavaScript recognizes the following implicit data types
  - Number
  - String
  - Logical
  - Object
  - The special value null
- Type conversion
  - JavaScript automatically converts between data types
  - Consider

    str = “100”, num1 = 10, num2 = 20

    num3 = num1+ num2

    strsum = str + num2

    strsum = num2 + str

```
30
10020
20100
```

JavaScript – Operators

- Arithmetic Operators

  +, ++, -, --, *, /, %

- Relational Operators

  ==, !=, ===, !==, >, >=, < , <=

- Logical Operators (and , or , not)

  &&, ||, !

- Assignment Operators

  =, +=, -=, *=, /=, %=

- Strict equal (===)

  Returns true if the operands are equal and of the same type.

- Strict not equal (!==)

  Returns true if the operands are not equal and/or not of the same type.

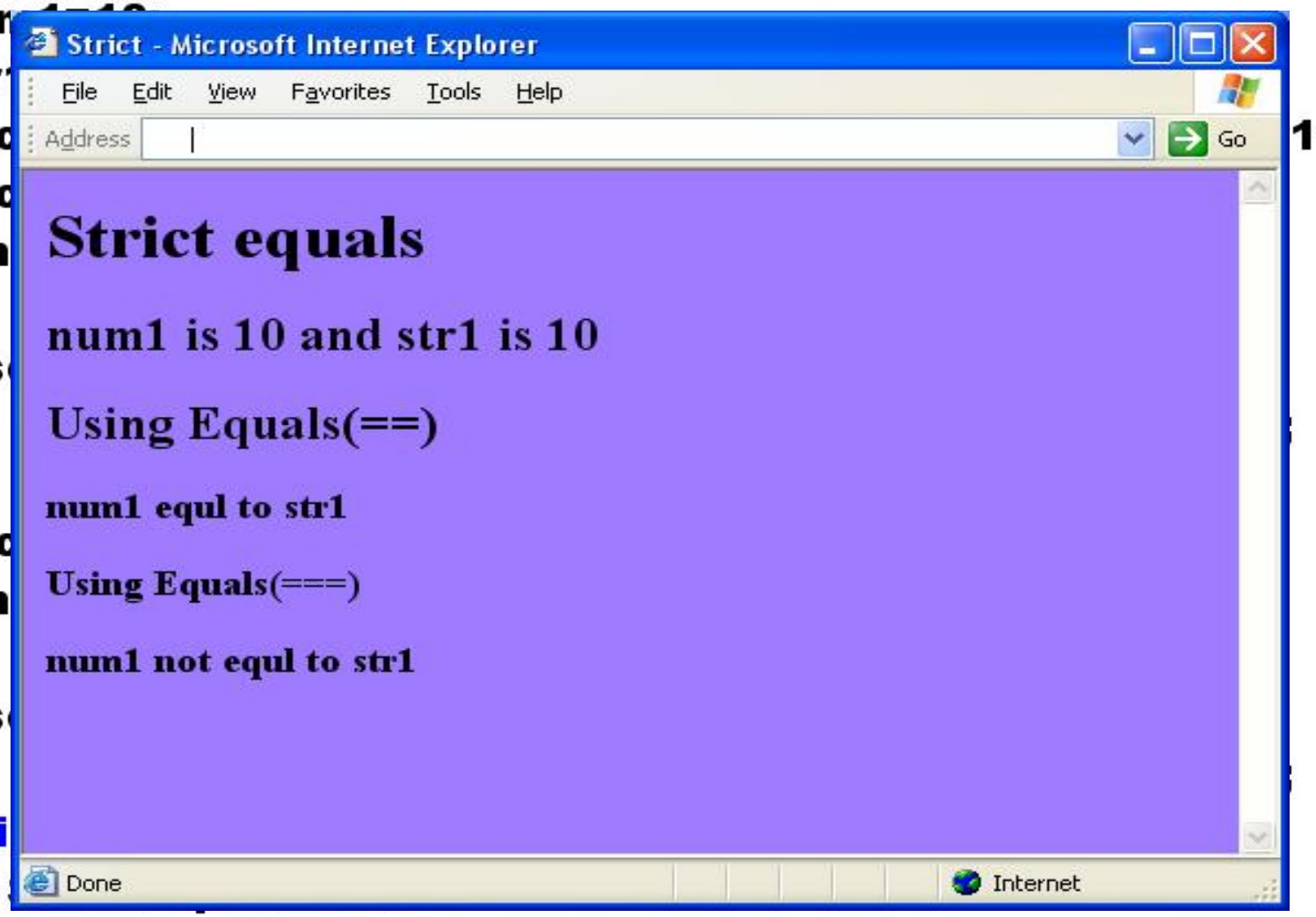
```
<script language="JavaScript">
    nu
    str
    doc                                                    1+"</h2>")
    doc
    if(n

    els

    doc
    if(n

    els

</scri
<h1>
```



Special operators

- typeof operator
  - Unary operator
  - Indicates the data type of the operand.

    Eg:

```
x=123;
   alert(typeof(x));          // Number
x="Hello"
   alert(typeof(x));          //  String
```

- new
  - Used for instantiation of objects.

    Eg: today = new Date( )

Which of the following statements are TRUE about HTML

A The formal parameters are local to the function

TRUE

B JavaScript is case sensitive

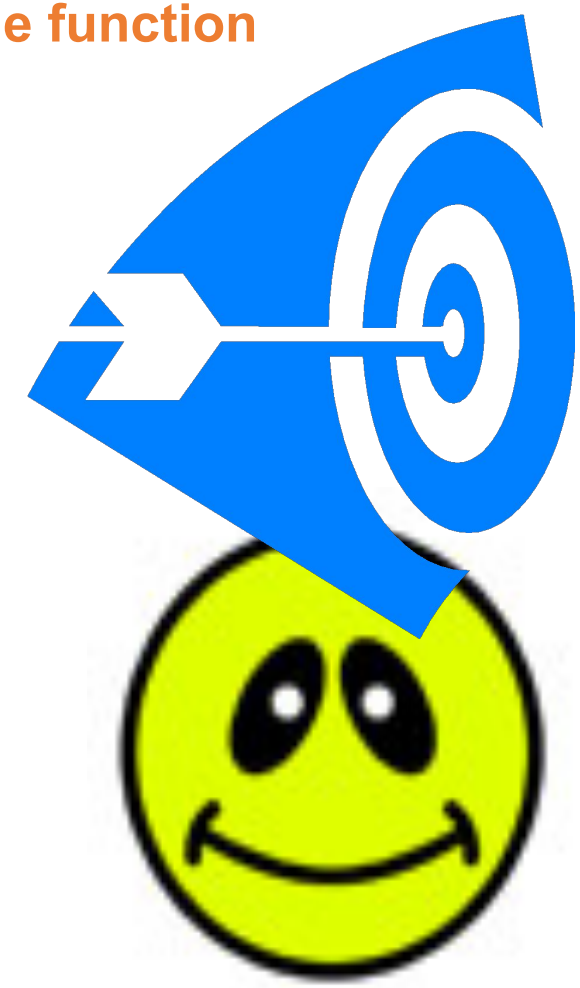
TRUE

C No explicit data type declaration

TRUE

D *new* operator is used to find the data type of the operand

FALSE



JavaScript – Control structures

- Control structure in JavaScript, as follows:
  - if
    - Is used to conditionally execute a single block of code
  - if .. else
    - a block of code is executed if the test condition evaluates to a boolean true else another block of code is executed.
  - switch …. case
    - switch statement tests an expression against a number of case options
    - executes the statements associated with the first match.

control.html

JavaScript – Loop

- while loop

The while statement is used to execute a block of code while a certain condition is true

Syntax : while (test condition)

```
{
  zero or more statements
}
```

- for loop

Iterate through a block of statements for some particular range of values

Syntax : for(initstmt; condstmt; updstmt){

```
      zero or more statements
   }
```

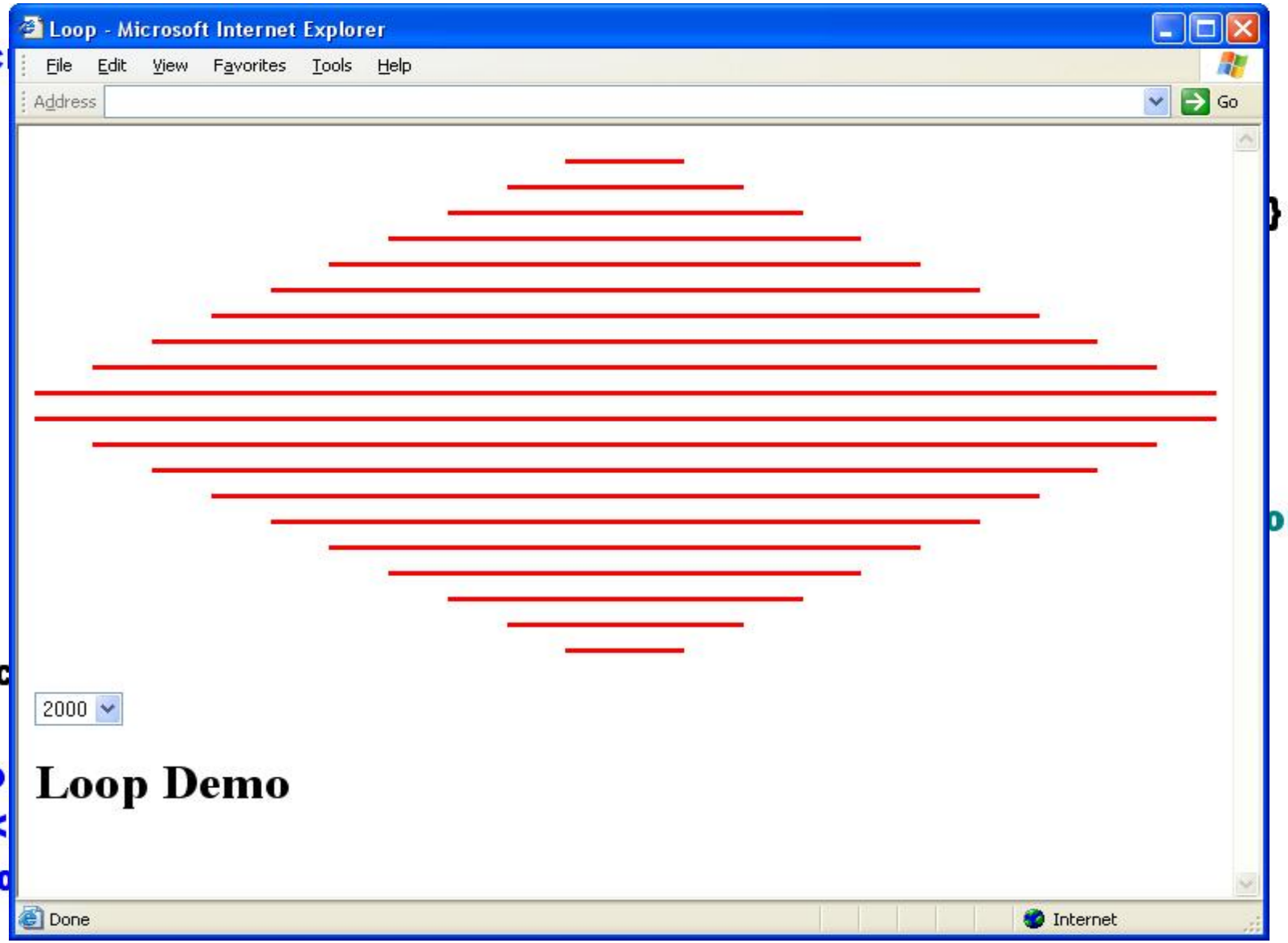
- do while loop

block of statements is executed first and then condition is checked

Syntax : do

```
{
  zero or more statements
}while ( test condition)
```

loop.html


Loop - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address
Go
2000
Loop Demo
Done
Internet

User defined Functions

- A function is a block of code that has a name.
- Way to organize your code. User can write his own functions
- JavaScript functions is to link actions on a web page with the JavaScript code.
- JavaScript has some in-built functions.

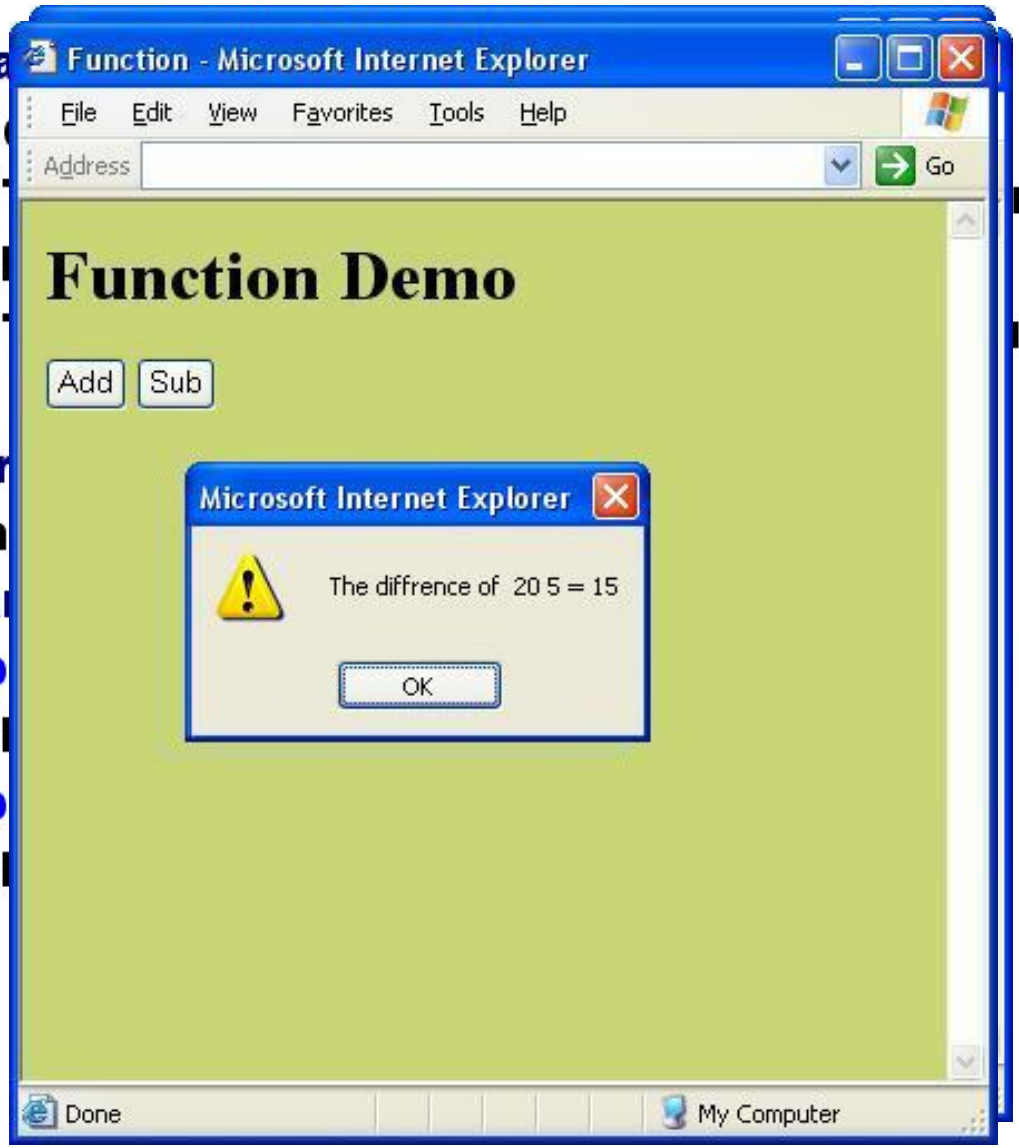
To create a function you define its name, any values ("arguments"), and some statements:

```
function myfunction(argument1,argument2,etc)   {
    some statements;
}
```

```
<script langua
   function ad
      {  alert("
   function su
      {  alert("
</script>
<body bgcolor
      <h1> Fun
      <form na
            <inp
            onCl
            <inp
            onCl
      </form>
</body>
```

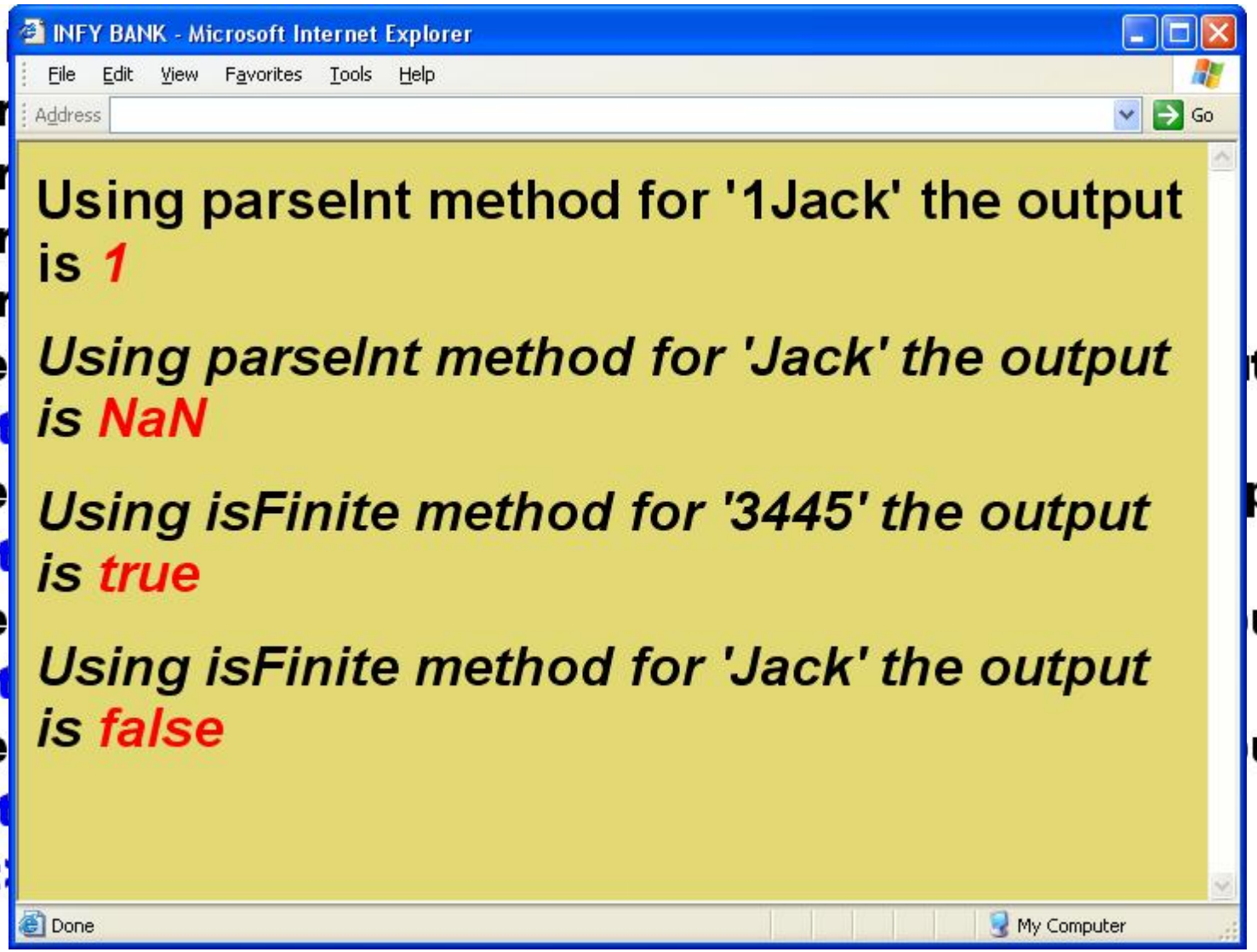
num2)); }

m1-num2)); }



Top-Level functions

- eval
  - Evaluates a string of JavaScript code without reference to a particular object.
  - Syntax eval(string)
- parseInt and parseFloat
  - Return a numeric value when given a string as an argument.
  - Syntax parseInt(string) , Syntax parseFloat(string)
- isNaN
  - Evaluates an argument to determine if it is “NaN” (not a number).
  - Syntax isNaN(testValue)
- isFinite
  - evaluates an argument to determine whether it is a finite number
  - Syntax isFinite(number)
- Number and String
  - functions let you convert an object to a number or a string.


INFY BANK - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address
Go
Using parseInt method for '1Jack' the output is 1
Using parseInt method for 'Jack' the output is NaN
Using isFinite method for '3445' the output is true
Using isFinite method for 'Jack' the output is false
Done
My Computer

In-built properties

- Infinity
  - Infinity is a numeric value representing infinity
- NaN
  - NaN is a value representing Not-ANumber.
- undefined
  - undefined is the value undefined.
- Helpful for debugging the code based on these properties.

Which of the following functions can be used to check the given data is a number or not?

A	isNaN	TRUE
B	isFinite	TRUE
C	parseInt	FALSE
D	parseFloat	FALSE

Dialog boxes (Window Object methods)

- Alert dialog box - alert(message)
  - Takes in a string argument and displays an alert box.

- Prompt dialog box - prompt(message,[inputDefault])
  - Displays a message and a data entry field

- Confirm dialog box - confirm(message)
  - Serves as a technique for confirming user actions


INFY BANK - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address www.InfyBank.com/Email.html
Go
Welcome to Infy Bank
Online Email Verification
Name(*) Jack
Email(*) Click here to enter Email.
(*):Mandatory Fields Confirm Submit
Microsoft Internet Explorer
You have successfully completed your Email Verification!!!!
OK
Done
My Computer

Summary

- Using CSS
- CSS Selectors
- Ways of using CSS
- Need of Scripting
- Client side Scripting Vs Sever side scripting.
- Introduction to JavaScript .
- JavaScript Basics
- Functions.

Thank You