Functions

- To keep the browser from executing a script when the page loads, you can put your script into a function.
- A function contains code that will be executed by an event or by a call to the function.
- You may call a function from anywhere within a page (or even from other pages if the function is embedded in an external .js file).
- Functions can be defined both in the <head> and in the <body> section of a document. However, to assure that a function is read/loaded by the browser before it is called, it could be wise to put functions in the <head> section.

Function Definition

- function *functionname*(*var1,var2,…,varX*)
  { *some code* }
- The parameters var1, var2, etc. are variables or values passed into the function. The { and the } defines the start and end of the function.
- A function with no parameters must include the parentheses () after the function name.
- Do not forget about the importance of capitals in JavaScript! The word *function* must be written in lowercase letters, otherwise a JavaScript error occurs! Also note that you must call a function with the exact same capitals as in the function name.
- The return statement is used to specify the value that is returned from the function.

```
<html>
<head>
    <script type="text/javascript">
        function product(a,b)
        {
            return a*b;
        }
    </script>
</head>
<body>
    <script type="text/javascript">
        document.write(product(4,3));
    </script>
</body>
</html>
```

Function () Constructor

- can define your function dynamically using **Function()** constructor along with the **new** operator.
- **Syntax**

var variablename = new Function(Arg1, Arg2…, "Function Body");

Example:

```
<script type="text/javascript">
var func = new Function("x", "y", "return x*y;");
function secondFunction(){
var result;
result = func(10,20);
document.write ( result );
}
</script>
```

Popup Boxes

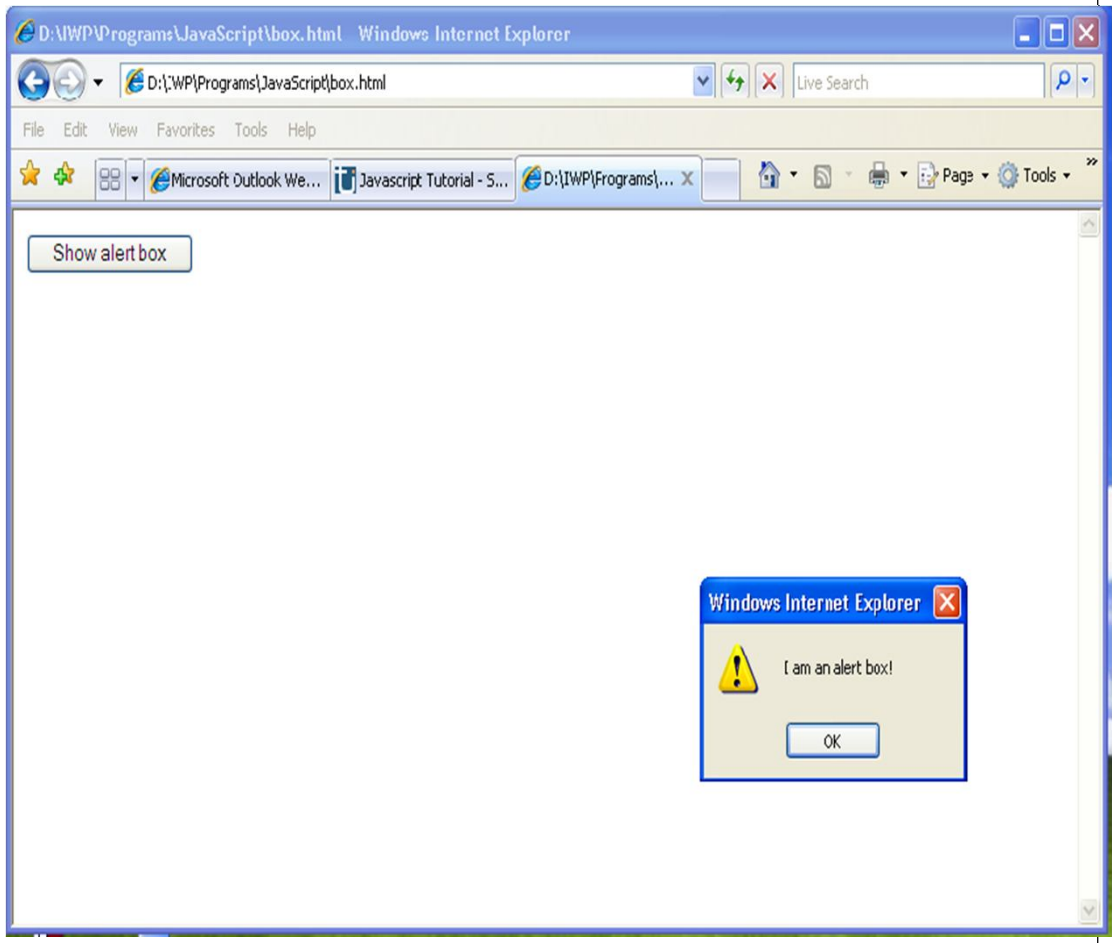
- JavaScript has three kind of popup boxes:

1. **Alert Box**
2. **Confirm Box**
3. **Prompt box**

Alert Box

An alert box is often used if you want to make sure information comes through to the user. When an alert box pops up, the user will have to click "OK" to proceed.

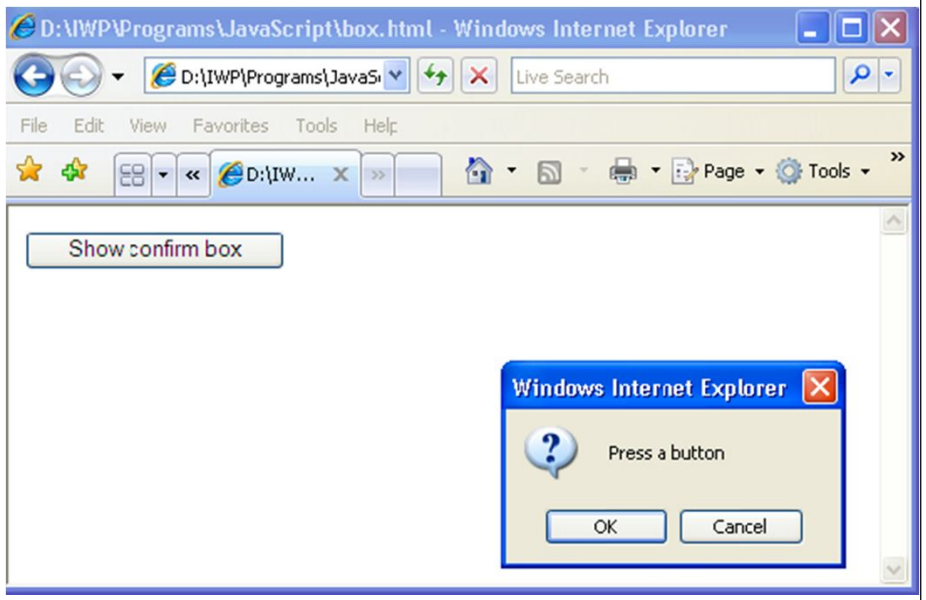
```
<html> <head>
<script type="text/javascript">
function show_alert()
{ alert("I am an alert box!"); }
</script> </head>
<body>
<input type="button"
onClick="show_alert()"
value="Show alert box" >
</body> </html>
```



Confirm Box

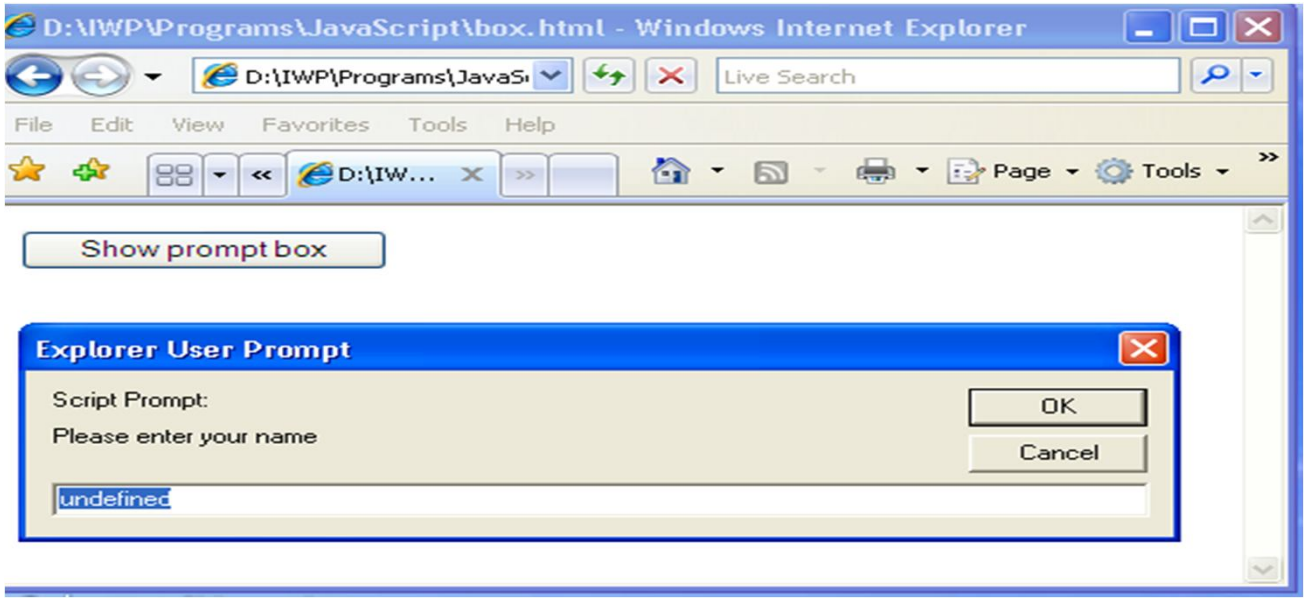
A confirm box is often used if you want the user to verify or accept something. If the user clicks "OK", the box returns true. If the user clicks "Cancel", the box returns false.

```
<html> <head>
<script type="text/javascript">
function show_confirm()
{
var r=confirm("Press a button");
if (r==true)
 {   alert("You pressed OK!");   }
else
  {   alert("You pressed Cancel!");   }
}
</script> </head>
<body>
<input type="button" onclick="show_confirm()" value="Show confirm box" >
</body> </html>
```

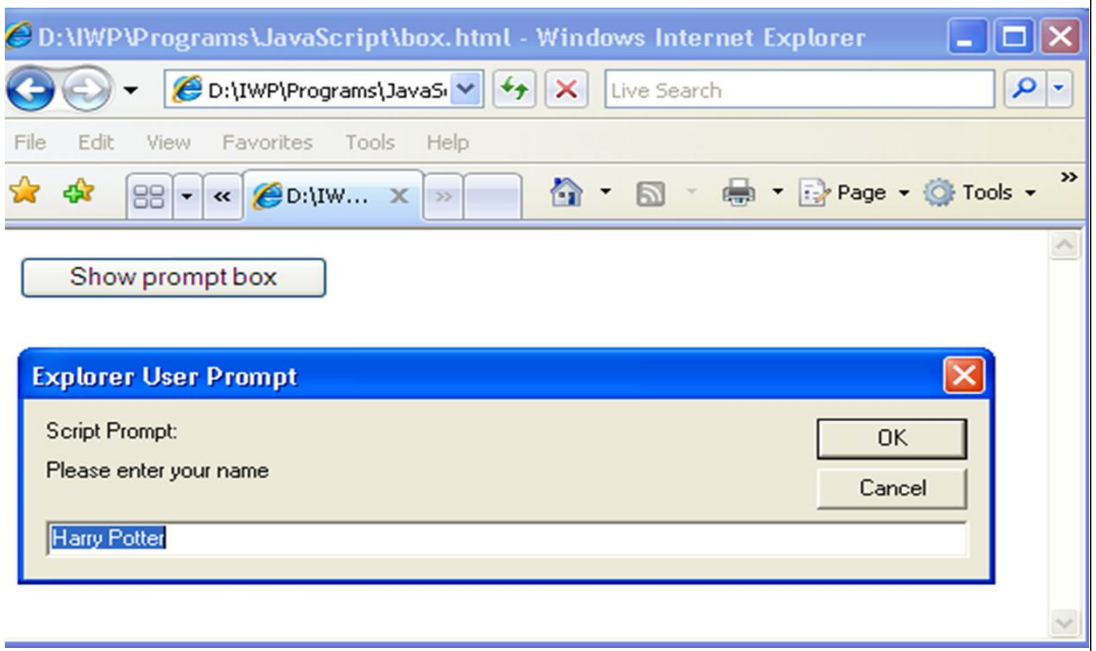


Prompt Box

- A prompt box is often used if you want the user to input a value before entering a page.
- When a prompt box pops up, the user will have to click either "OK" or "Cancel" to proceed after entering an input value.
- If the user clicks "OK" the box returns the input value. If the user clicks "Cancel" the box returns null.



```
<html> <head> <script type="text/javascript">
function show_prompt()
{
var name=prompt("Please enter your name","Harry Potter");
        if (name!=null && name!="")
        {   document.write("<p>Hello " + name + "! How are you
        today?</p>");  }
}</script> </head>
<body>
<input type="button"
onClick="show_prompt()"
value="Show prompt box" >
</body> </html>
```



Objects

- **JavaScript is an Object Based Programming language.**
- An Object Based Programming language allows you to define your own objects and make your own variable types.
- An object is just a special kind of data. An object has properties and methods.

var val = new String(string);

- **Properties** are the values associated with an object.

```
var txt="Hello World!";
document.write(txt.length);
```

- **Methods** are the actions that can be performed on objects.

```
var str="Hello world!";
document.write(str.toUpperCase());
```

String object

- The String object is used to manipulate a stored piece of text.

```
var txt="Hello world!";
var txt=new String("Hello world!“);
document.write(txt.length);  //12
document.write(txt.toUpperCase()); //HELLO WORLD!
document.write(txt.match("world") ); //world
document.write(txt.match("World") ) ;//null
document.write(txt.indexOf("world")); //6
var str="Visit Microsoft!";
document.write(str.replace("Microsoft","CTS")); //Visit CTS!
```

Example:

```
var txt = "Hello World!";
document.write("<p>Big: " + txt.big() + "</p>");
document.write("<p>Small: " + txt.small() + "</p>");
document.write("<p>Bold: " + txt.bold() + "</p>");
document.write("<p>Italic: " + txt.italics() + "</p>");
document.write("<p>Strike: " + txt.strike() + "</p>");
document.write("<p>Fontcolor: " + txt.fontcolor("green") + "</p>");
document.write("<p>Fontsize: " + txt.fontsize(6) + "</p>");
document.write("<p>Subscript: " + txt.sub() + "</p>");
document.write("<p>Superscript: " + txt.sup() + "</p>");
document.write("<p>Link: " + txt.link("http://www.w3schools.com") +
  "</p>");
document.write("<p>Blink: " + txt.blink() + " (does not work in IE, Chrome,
  or Safari)</p>");
```

- charAt(index)→Returns the character at the specified index
- concat(str1,str2…)
- lastIndexOf()
- slice(bindx,eindx) →extracts a section of a string and returns a new string
- split(separator) →a String object into an array of strings by separating the string into substrings.
- substr(start,length)
- substring(indx,indx)
- toLowerCase()
- search(regexp)
- localeCompare ()→returns a number indicating whether a reference string comes before or after or is the same as the given string in sorted order.

Date object

- new Date( )
- new Date(milliseconds)
- new Date(datestring)

```
var dt = new Date("December 25, 1995 23:15:00");
```

- new Date(year,month,date,hour,minute,second,millisecond)

Date Object

- The Date object is used to work with dates and times.
- var d = new Date();
- getDate() → Returns the day of the month (from 1-31)
- getDay() → Returns the day of the week (from 0-6)
- getYear()→Returns the year (four digits)
- getHours()→Returns the hour (from 0-23)
- getMinutes()→Returns the minutes (from 0-59)
- getMonth()→Returns the month (from 0-11)
- getSeconds()→Returns the seconds (from 0-59)
- getTime() → Returns the number of milliseconds since midnight Jan 1, 1970

Date object methods

- getFullYear() →Returns the year of the specified date according to local time.
- getMilliseconds()
- toDateString() →Returns the "date" portion of the Date as a human-readable string.
- toLocaleDateString() →Returns the "date" portion of the Date as a string, using the current locale's conventions (OS)
- toLocaleString() →Converts a date to a string, using the current locale's conventions

Math Object

- sqrt(x)→Returns the square root of x
- ceil(x)→Returns x, rounded upwards to the nearest integer
- floor(x)→Returns x, rounded downwards to the nearest integer
- exp(x)→Returns the value of E^x
- log(x)→Returns the natural logarithm (base E) of x
- max(x,y,z,…,n)→Returns the number with the highest value
- min(x,y,z,…,n)→Returns the number with the lowest value
- pow(x,y)→Returns the value of x to the power of y
- random()→Returns a random number between 0 and 1
- round(x)→Rounds x to the nearest integer
- sin(x)→Returns the sine of x (x is in radians)
- cos(x)→Returns the cosine of x (x is in radians)
- tan(x)→Returns the tangent of an angle
- **abs()**→Returns the absolute value of a number

Property	Description
E	Returns Euler's number (approx. 2.718)
LN2	Returns the natural logarithm of 2 (approx. 0.693)
LN10	Returns the natural logarithm of 10 (approx. 2.302)
LOG2E	Returns the base-2 logarithm of E (approx. 1.442)
LOG10E	Returns the base-10 logarithm of E (approx. 0.434)
PI	Returns PI (approx. 3.14)
SQRT1_2	Returns the square root of 1/2 (approx. 0.707)
SQRT2	Returns the square root of 2 (approx. 1.414)

Boolean Object

- The Boolean object represents two values: "true" or "false".
- var myBoolean=new Boolean();
- 0 →False
- 1 →True

Numbers

- The Number() function converts the object argument to a number that represents the object's value.
- If the value cannot be converted to a legal number, NaN is returned.

var val = new Number(number);

Property

- NaN →a special value representing Not-a-Number
- MAX_VALUE →The largest possible value a number in JavaScript can have 1.7976931348623157E+308
- MIN_VALUE→The smallest possible value a number in JavaScript can have 5E-324
- NEGATIVE_INFINITY→A value that is less than MIN_VALUE.
- POSITIVE_INFINITY →A value that is greater than MAX_VALUE

Methods

- toExponential(x) →Converts a number into an exponential notation
- toFixed(x) →Formats a number with a specific number of digits to the right of the decimal.
- toPrecision(x) →Defines how many total digits (including digits to the left and right of the decimal) to display of a number
- toString()→ Converts a Number object to a string
- valueOf() →Returns the primitive value of a Number object

```
var test4= 10 , test5= 20;
document.write(test4 +test5); //1020
document.write(Number(test4)+Number(test5)); //30
document.write(Number("99.66") +Number("01.34") ); //101
```

The parseInt() function parses a string and returns an integer.

```
document.write(parseInt("10.33") ) ;  //10
document.write(parseFloat("10.33") ); //10.33
```