Image Object

- For each <img> tag in an HTML document, an Image object is created.

Properties:

- **align** → Sets or returns the value of the align attribute of an image
- **alt** → Sets or returns the value of the alt attribute of an image
- **border** → Sets or returns the value of the border attribute of an image
- **complete**→ Returns whether or not the browser is finished loading an image
- **height**→ Sets or returns the value of the height attribute of an image
- **hspace**→ Sets or returns the value of the hspace (left, right) attribute of an image
- **longDesc**→ Sets or returns the value of the longdesc attribute of an image
- **lowsrc**→ Sets or returns a URL to a low-resolution version of an image
- **name**→Sets or returns the name of an image
- **src**→ Sets or returns the value of the src attribute of an image
- **usemap**→ Sets or returns the value of the usemap attribute of an image
- **vspace**→Sets or returns the value of the vspace (top, bottom) attribute of an image
- **width**→ Sets or returns the value of the width attribute of an image

Image Object- Event

- onabort → Loading of an image is interrupted
- onerror → An error occurs when loading an image
- onload→ An image is finished loading

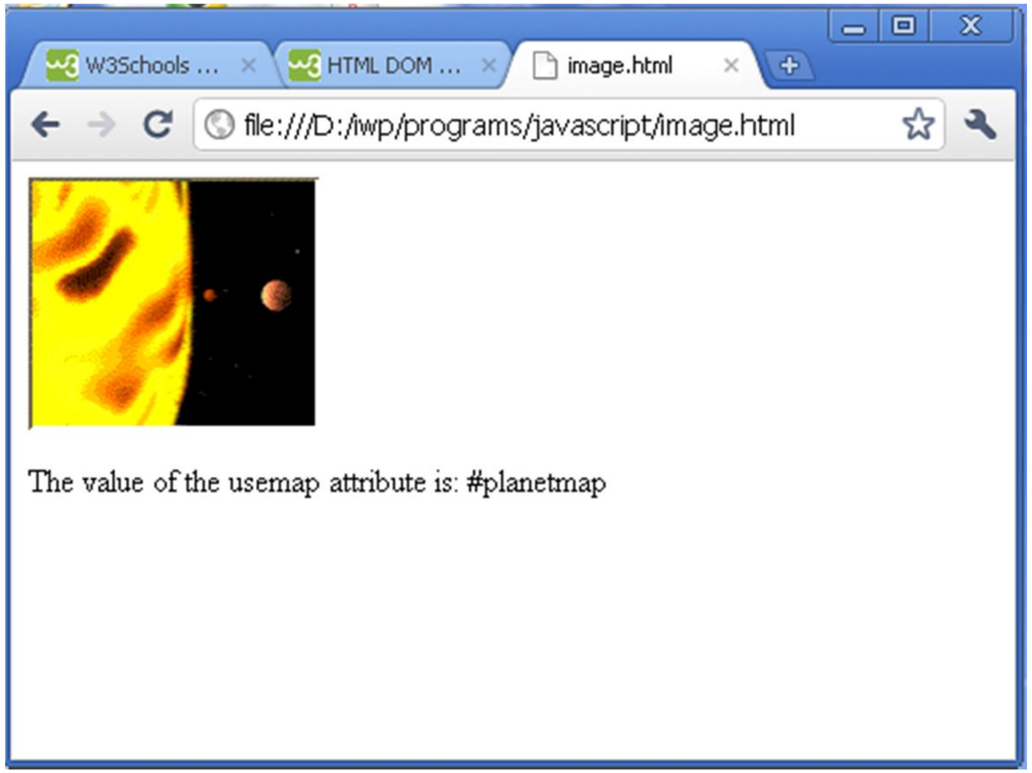
Image - useMap Property

- The useMap property sets or returns the value of the usemap attribute of an image.
- The usemap attribute specifies an image as a client-side image-map (an image-map is an image with clickable areas).
- The usemap attribute is associated with a map element's name attribute, and creates a relationship between the image and the map.

Example (image.html)

```
<html> <body>
<img id="planets" src="planets.gif" width="145" height="126"
   useMap="#planetmap">
<map name="planetmap">
<area id="venus" shape="circle" coords="124,58,8" alt="The planet Venus"
   href="venus.html">
<area id="earth" shape="square" coords="0,0,100,100" alt="The planet
   Earth" href="earth.html">
</map>
<p>The value of the usemap attribute is:
<script type="text/javascript">
document.write(document.getElementById("planets").useMap);
</script> </p> </body> </html>
```

Output



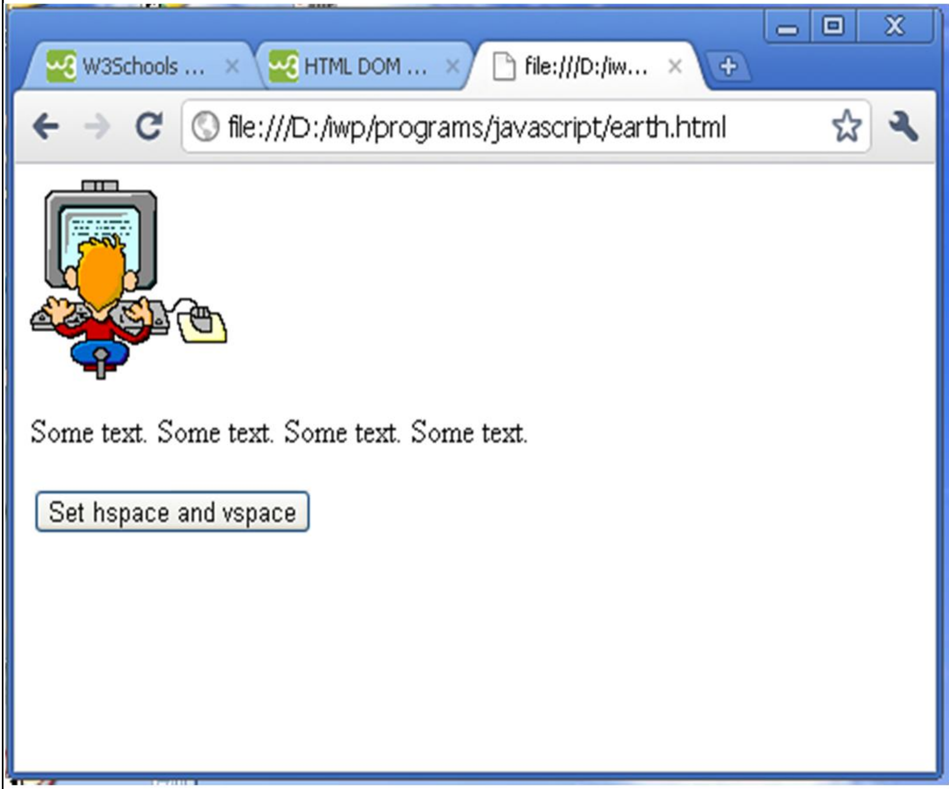
The value of the usemap attribute is: #planetmap

Example (earth.html)

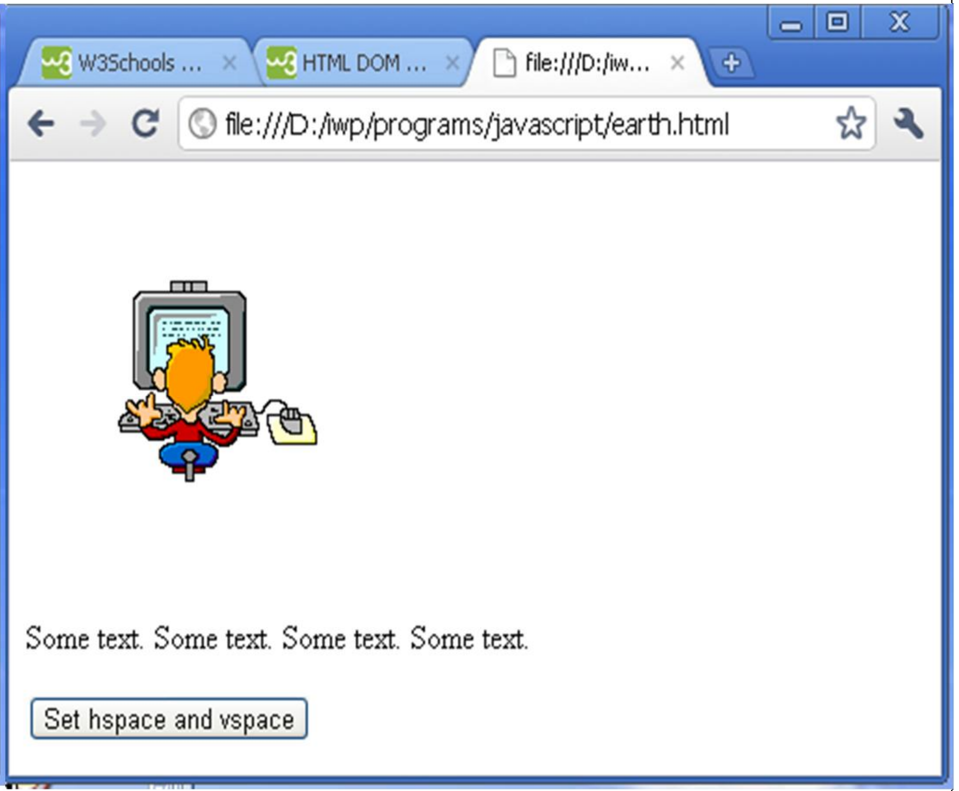
```
<html> <head>
<script type="text/javascript">
function setSpace()
{
document.getElementById("compman").hspace="50";
document.getElementById("compman").vspace="50";
}
</script> </head> <body>
 <img id="compman" src="compman.gif" alt="Computerman" width=107
   height=98>
<p>Some text. Some text. Some text. Some text.</p>
<input type="button" onclick="setSpace()" value="Set hspace and vspace">
</body> </html>
```

Output

onLoad **onClick**



Some text. Some text. Some text. Some text.



Some text. Some text. Some text. Some text.

Image Object - Complete

```
<html> <head>
<script type="text/javascript">
function alertComplete()
{
alert("Image loaded: " +
document.getElementById("compman").complete); //true or false
}
</script> </head>
<body onload="alertComplete()">
<img id="compman" src="compman.gif" alt="Computerman" />
</body> </html>
```

Events

- By using JavaScript, we have the ability to create dynamic web pages. Events are actions that can be detected by JavaScript.
- Every element on a web page has certain events which can trigger a JavaScript. For example, we can use the onClick event of a button element to indicate that a function will run when a user clicks on the button. We define the events in the HTML tags.

Examples of events:

- A mouse click
- A web page or an image loading
- Mousing over a hot spot on the web page
- Selecting an input field in an HTML form
- Submitting an HTML form
- A keystroke

Note: Events are normally used in combination with functions, and the function will not be executed before the event occurs!

Input Events

- onblur - When a user leaves an input field
- onchange –
  - When a user changes the content of an input field
  - When a user selects a dropdown value
- onfocus - When an input field gets focus
- onselect - When input text is selected
- onsubmit - When a user clicks the submit button
- onreset - When a user clicks the reset button
- onkeydown - When a user is pressing/holding down a key
- onkeypress - When a user is pressing/holding down a key
- onkeyup - When the user releases a key
- onkeyup - When the user releases a key
- onkeydown vs onkeyup - Both

Mouse and Click Events

- Mouse Events
  - onmouseover/onmouseout - When the mouse passes over an element
  - onmousedown/onmouseup - When pressing/releasing a mouse button
  - onmousedown - When mouse is clicked: Alert which element
  - onmousedown - When mouse is clicked: Alert which button
  - onmousemove/onmouseout - When moving the mouse pointer over/out of an image
  - onmouseover/onmouseout - When moving the mouse over/out of an image
  - onmouseover an image map
- Click Events- Acting to the onclick event
  - onclick - When button is clicked
  - ondblclick - When a text is double-clicked

Load Events

- onload - When the page has been loaded
- onload - When an image has been loaded
- onerror - When an error occurs when loading an image
- onunload - When the browser closes the document
- onresize - When the browser window is resized

onload and onunload

- The onLoad and onUnload events are triggered when the user enters or leaves the page.
- The onLoad event is often used to check the visitor's browser type and browser version, and load the proper version of the web page based on the information.
- Both the onLoad and onUnload events are also often used to deal with cookies that should be set when a user enters or leaves a page. For example, you could have a popup asking for the user's name upon his first arrival to your page. The name is then stored in a cookie. Next time the visitor arrives at your page, you could have another popup saying something like: "Welcome John Doe!".

onfocus, onblur and onChange

- The onFocus, onBlur and onChange events are often used in combination with validation of form fields.
- The checkEmail() function will be called whenever the user changes the content of the field:
- `<input type="text" size="30" id="email" onchange="checkEmail()" />`

onsubmit

- The onSubmit event is used to validate ALL form fields before submitting it.
- The checkForm() function will be called when the user clicks the submit button in the form. If the field values are not accepted, the submit should be cancelled. The function checkForm() returns either true or false. If it returns true the form will be submitted, otherwise the submit will be cancelled:

```
<form method="post" action="xxx.htm" onsubmit="return
   checkForm()">
```

onMouseOver

- The onmouseover event can be used to trigger a function when the user mouses over an HTML element.

- onresize → The event occurs when the size of an element has changed
- onselect →The event occurs after some text has been selected in an element
- onclick→The event occurs when the user clicks on an element
- ondblclick→The event occurs when the user double-clicks on an element
- onkeypress→The event occurs when the user is pressing a key or holding down a key
- onkeydown,() onkeyup() , onchange()

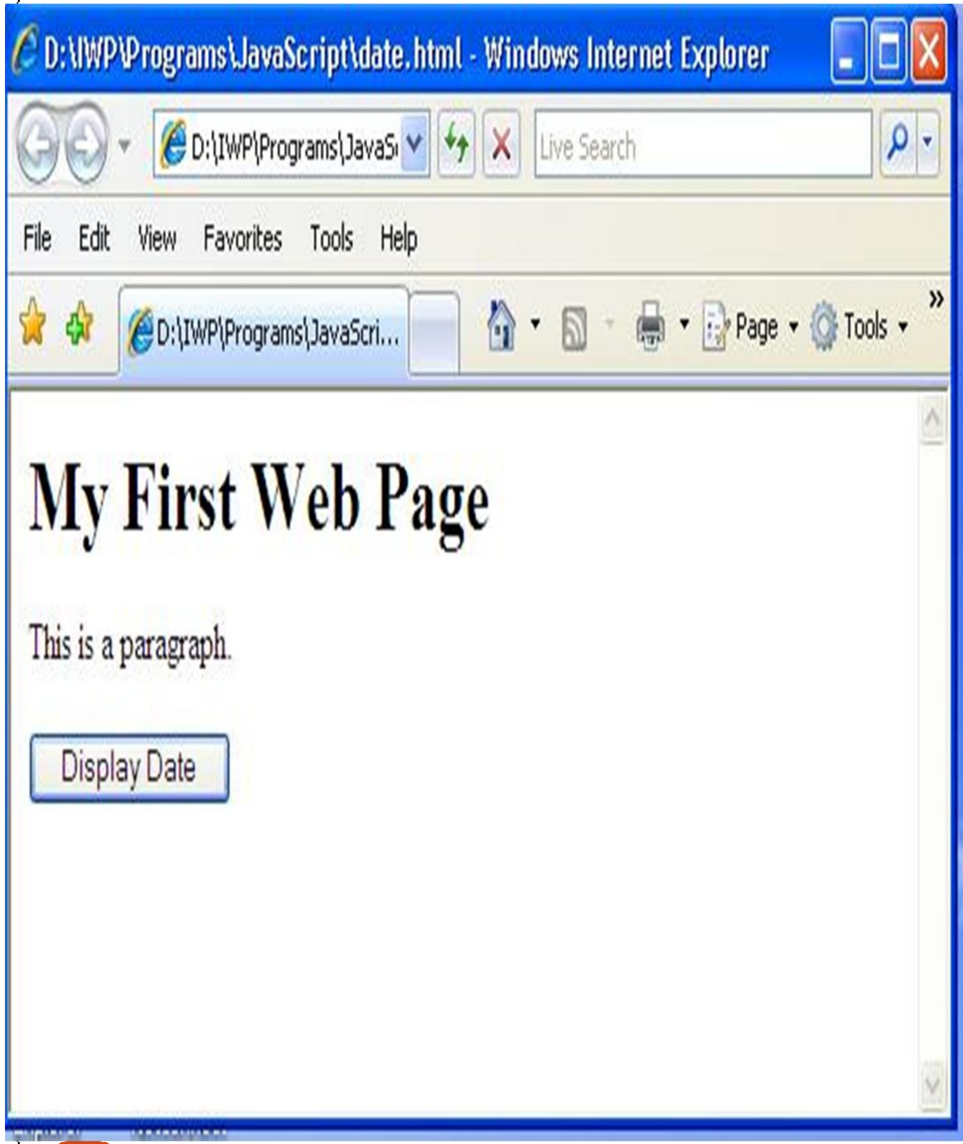
Event Properties:

- screenX → Returns the horizontal coordinate of the mouse pointer, relative to the screen, when an event was triggered
- screenY → Returns the vertical coordinate of the mouse pointer, relative to the screen, when an event was triggered

```
<html> <head>
<script type="text/javascript">
function displayDate()
{
document.getElementById("demo").innerHTML=Date();
} </script> </head>
<body>
<h1>My First Web Page</h1>
<p id="demo">This is a paragraph.</p>
<input type= "button" onClick="displayDate()" value="Display Date">
</body>
</html>
```

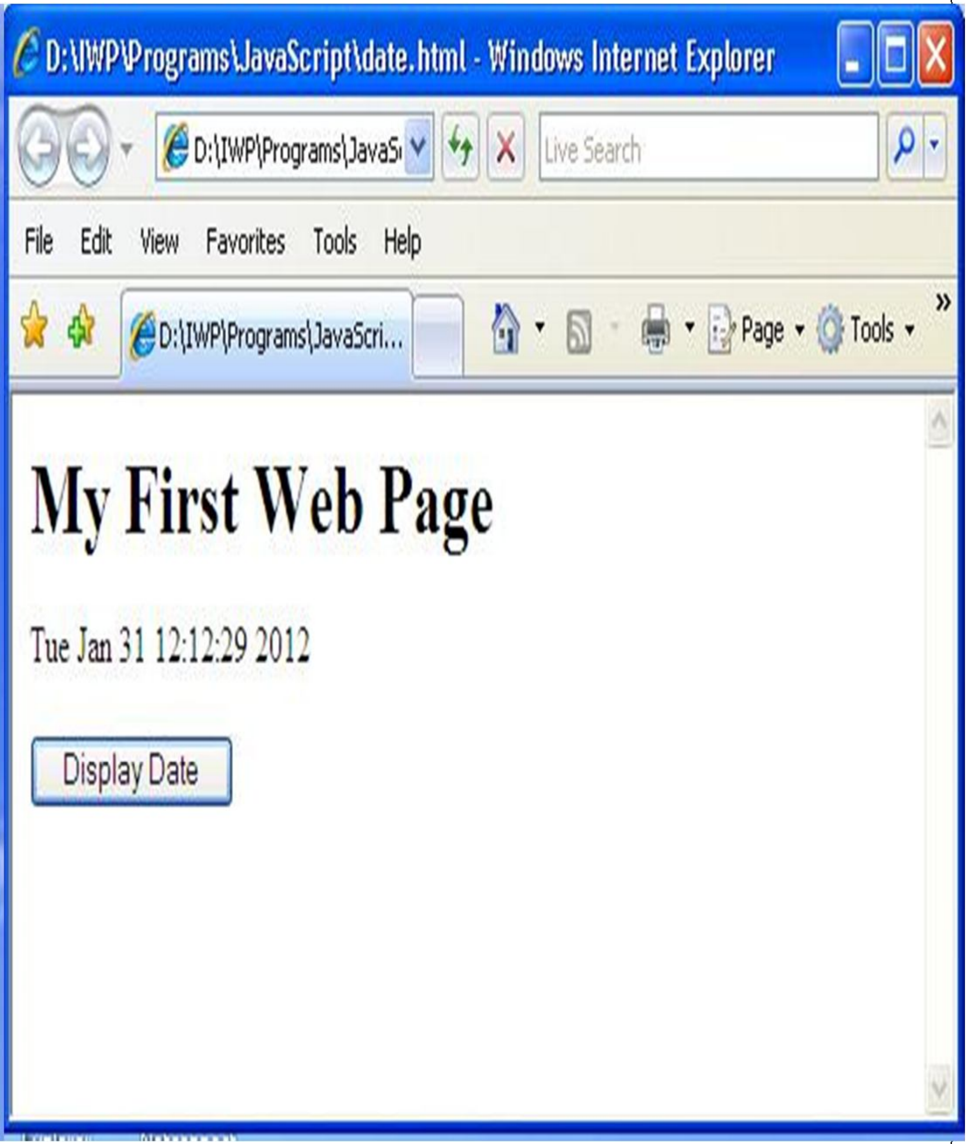
innerHTML → Sets or returns the HTML contents (+text) of an element

Output


D:\IWP\Programs\JavaScript\date.html - Windows Internet Explorer

My First Web Page

This is a paragraph.


D:\IWP\Programs\JavaScript\date.html - Windows Internet Explorer

My First Web Page

Tue Jan 31 12:12:29 2012

Creating Your Own Objects

1. Create a direct instance of an object

```
personObj=new Object();
personObj.firstname="John";
personObj.lastname="Doe";
personObj.age=50;
personObj.eyecolor="blue";
```

Alternative syntax (using object literals):

```
personObj={firstname:"John",lastname:"Doe",age:50,eyecolor:"blue"};
```

Adding a method to the personObj is also simple.

The following code adds a method called eat() to the personObj:

```
personObj.eat=eat;
```

2. Create an object constructor

- Create a function that construct objects:

```
function person(firstname,lastname,age,eyecolor)
{
this.firstname=firstname;
this.lastname=lastname;
this.age=age;            //"this" is: the instance of the object at hand.
this.eyecolor=eyecolor;
this.newlastname=newlastname; //method
}
```

- Once you have the object constructor, you can create new instances of the object, like this:

```
var myFather=new person("John","Doe",50,"blue");
var myMother=new person("Sally","Rally",48,"green");
```

```
<html><body><script type="text/javascript">
function mycircle(x,y,r)                    //Constructor
{
  this.xcoord = x;        // Adding Properties
  this.ycoord = y;
  this.radius = r;
  this.retArea = retArea; // Adding Methods
}
function retArea() {               // Method Definition
  return ( Math.PI * this.radius * this.radius );
}
var testcircle = new mycircle(3,4,5);          // Object Creation
alert( 'The area of the circle is ' + testcircle.retArea() ); // Method Call
</script></body></html>
```

JavaScript Timing Events

- With JavaScript, it is possible to execute some code after a specified time-interval. This is called timing events.

The setTimeout() Method - executes a code some time in the future

var t=setTimeout("*javascript statement*",*milliseconds*);

- The setTimeout() method returns a value. In the syntax defined above, the value is stored in a variable called t. If you want to cancel the setTimeout() function, you can refer to it using the variable name.
- The first parameter of setTimeout() can be a string of executable code, or a call to a function. The second parameter indicates how many milliseconds from now you want to execute the first parameter.
- **Note:** There are 1000 milliseconds in one second.

The clearTimeout() Method - cancels the setTimeout()

clearTimeout(*setTimeout_variable*)

- **Note:** The setTimeout() and clearTimeout() are both methods of the HTML DOM Window object.

location object

- The window.location object can be used to get the current page address (URL) and to redirect the browser to a new page.
- Properties:
  - window.location.href returns the href (URL) of the current page
  - window.location.hostname returns the domain name of the web host
  - window.location.pathname returns the path and filename of the current page
  - window.location.protocol returns the web protocol used (http: or https:)
  - window.location.assign loads a new document

```
<html> <head> <script type="text/javascript">
function Redirect()
{
   //window.location="http://www.vit.ac.in";
   document.location.href="http://www.vit.ac.in";
}
document.write("You will be redirected to main page in 10 sec.");
setTimeout("Redirect()", 10000);
</script>  </head>  </html>
```

Navigator Object

- However, there are some things that just don't work on certain browsers - especially on older browsers.
- Sometimes it can be useful to detect the visitor's browser, and then serve the appropriate information.
- The Navigator object contains information about the visitor's browser name, version, and more.

Properties:

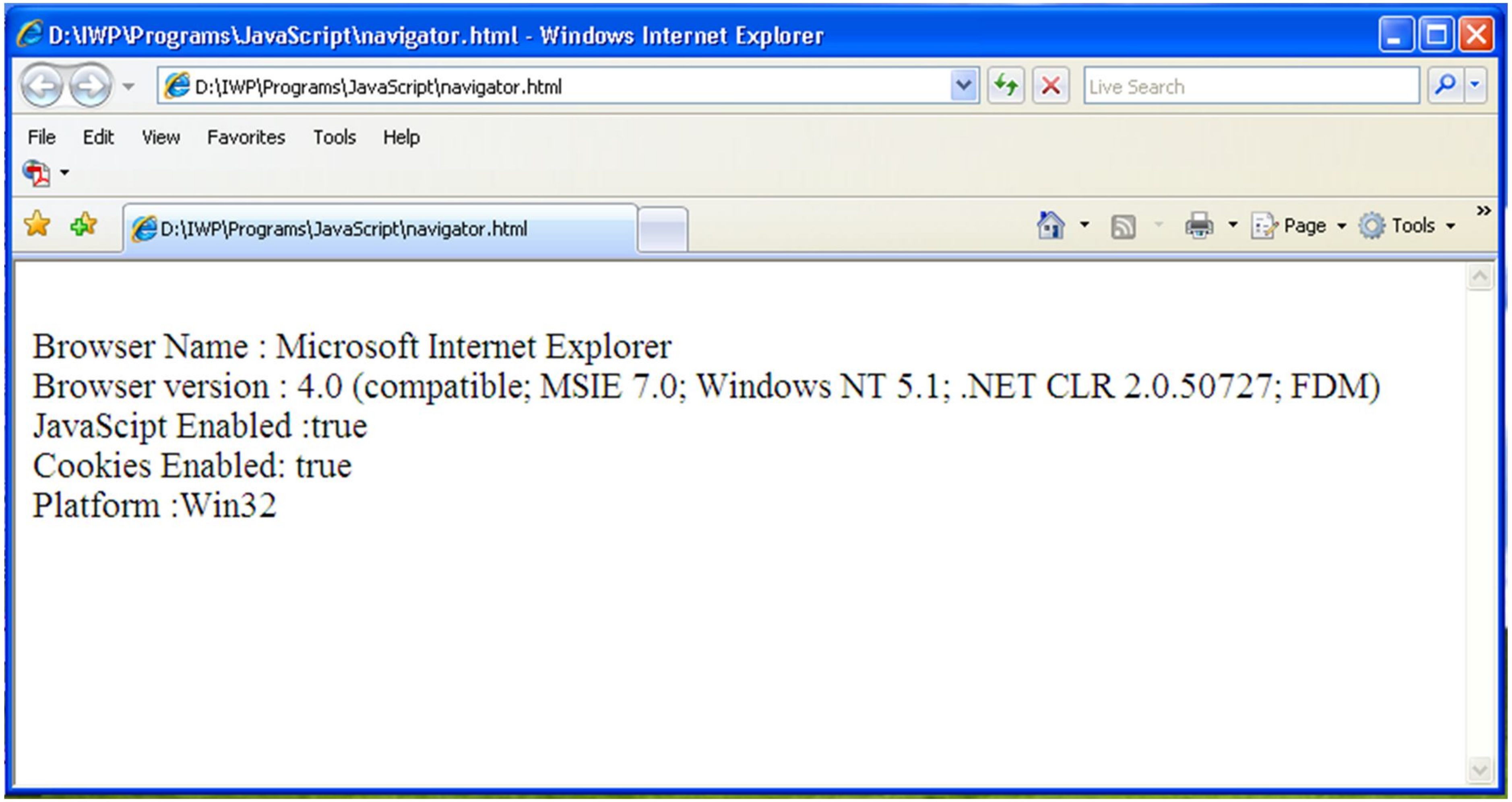
- appVersion→ This property is a string that contains the version of the browser as well as other useful information such as its language and compatibility.
- platform →This property is a string that contains the platform for which the browser was compiled."Win32" for 32-bit Windows operating systems

Navigator object properties

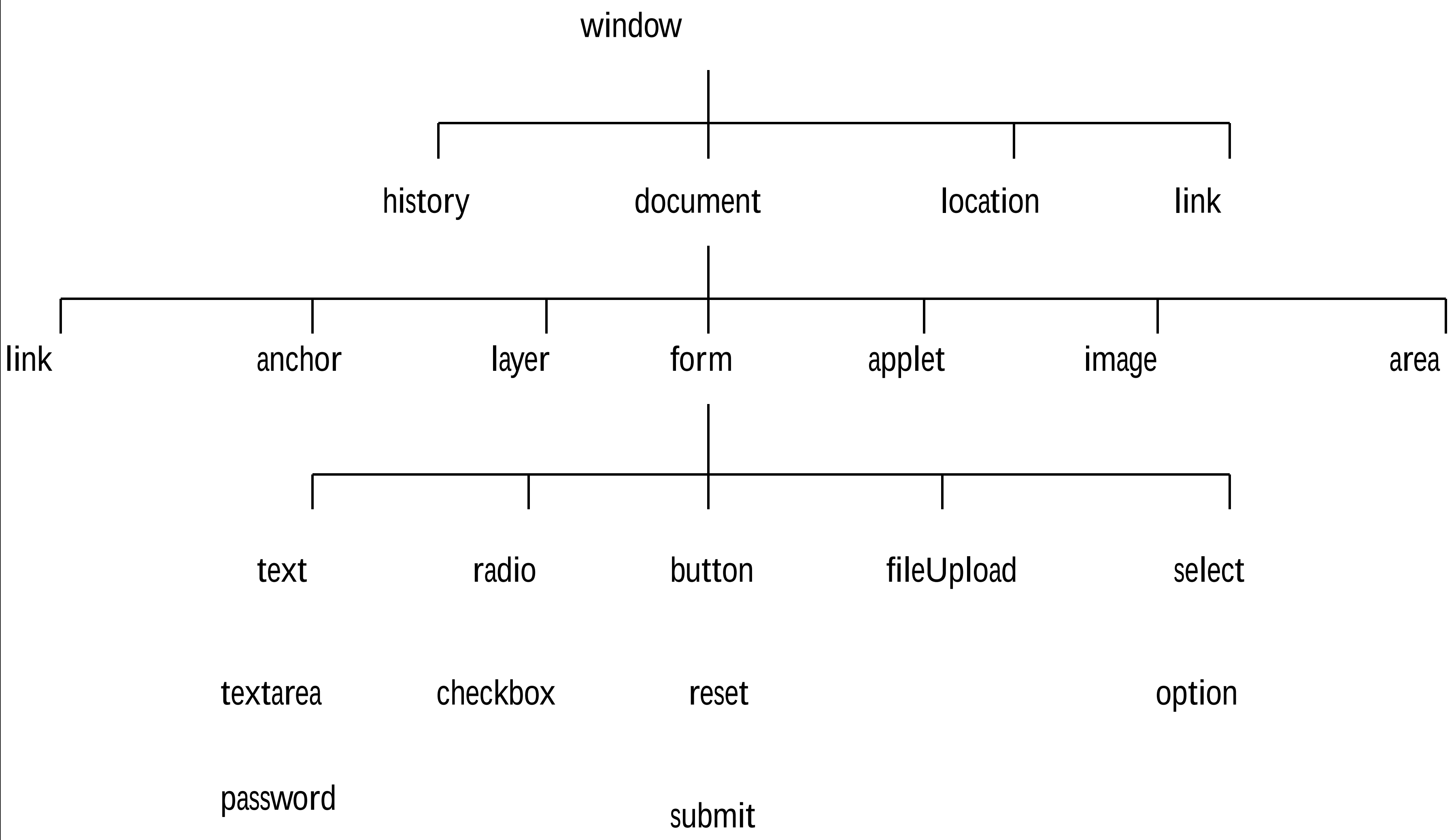
- appCodeName - The name of the browser's code such as "Mozilla".
- appMinorVersion - The minor version number of the browser.
- appName - The name of the browser such as "Microsoft Internet Explorer" or "Netscape Navigator".
- appVersion - The version of the browser which may include a compatability value and operating system name.
- cookieEnabled - A boolean value of true or false depending on whether cookies are enabled in the browser.
- cpuClass - The type of CPU which may be "x86"
- mimeTypes - An array of MIME type descriptive strings that are supported by the browser.
- onLine - A boolean value of true or false.
- opsProfile
- platform - A description of the operating system platform. In my case it is "Win32" for Windows 95.
- plugins - An array of plug-ins supported by the browser and installed on the browser.
- systemLanguage - The language being used such as "en-us".
- userAgent - In my case it is "Mozilla/4.0 (compatible; MSIE 4.01; Windows 95)" which describes the browser associated user agent header.
- userLanguage - The languge the user is using such as "en-us". userProfile

- **Methods**
  - javaEnabled() - Returns a boolean telling if the browser has JavaScript enabled.
  - taintEnabled() - Returns a boolean telling if the browser has tainting enabled. Tainting is a security protection mechanism for data.

```
<html>
<head>
<script type="text/javascript">
document.write("<br> Browser Name : " + navigator.appName);
document.write("<br> Browser version  : " + navigator.appVersion);
document.write("<br> JavaScipt Enabled :"+navigator.javaEnabled());
document.write("<br> Cookies Enabled: " + navigator.cookieEnabled)
document.write("<br> Platform  :"+ navigator.platform);
</script> </head>
</html>
```


D:\IWP\Programs\JavaScript\navigator.html - Windows Internet Explorer
D:\IWP\Programs\JavaScript\navigator.html
Live Search
File Edit View Favorites Tools Help
D:\IWP\Programs\JavaScript\navigator.html
Page
Tools

```
Browser Name : Microsoft Internet Explorer
Browser version : 4.0 (compatible; MSIE 7.0; Windows NT 5.1; .NET CLR 2.0.50727; FDM)
JavaScipt Enabled :true
Cookies Enabled: true
Platform :Win32
```


window
history
document
location
link
link
anchor
layer
form
applet
image
area
text
radio
button
fileUpload
select
textarea
checkbox
reset
option
password
submit

Frame Object

- The Frame object represents an HTML frame.
- The <frame> tag defines one particular window (frame) within a frameset.
- For each <frame> tag in an HTML document, a Frame object is created.

Frame Object Events:

- **onload** → Script to be run immediately after a frame is loaded

Frame Properties

- **contentDocument**→Returns the document object generated by a frame
- **contentWindow**→Returns the window object generated by a frame
- **frameBorder**→Sets or returns the value of the frameborder attribute in a frame.
- **longDesc**→Sets or returns the value of the longdesc attribute in a frame
- **marginHeight**→Sets or returns the value of the marginheight attribute in a frame
- **marginWidth**→Sets or returns the value of the marginwidth attribute in a frame
- **name**→Sets or returns the value of the name attribute in a frame
- **noResize**→Sets or returns the value of the noresize attribute in a frame
- **scrolling**→Sets or returns the value of the scrolling attribute in a frame
- **src**→Sets or returns the value of the src attribute in a frame

frameset.html

```
<html>
<frameset cols="50%,50%">
  <frame id="leftFrame" src="LeftFrameCode.html">
  <frame id="rightFrame" src="RightFrameCode.html">
</frameset>
</html>
```

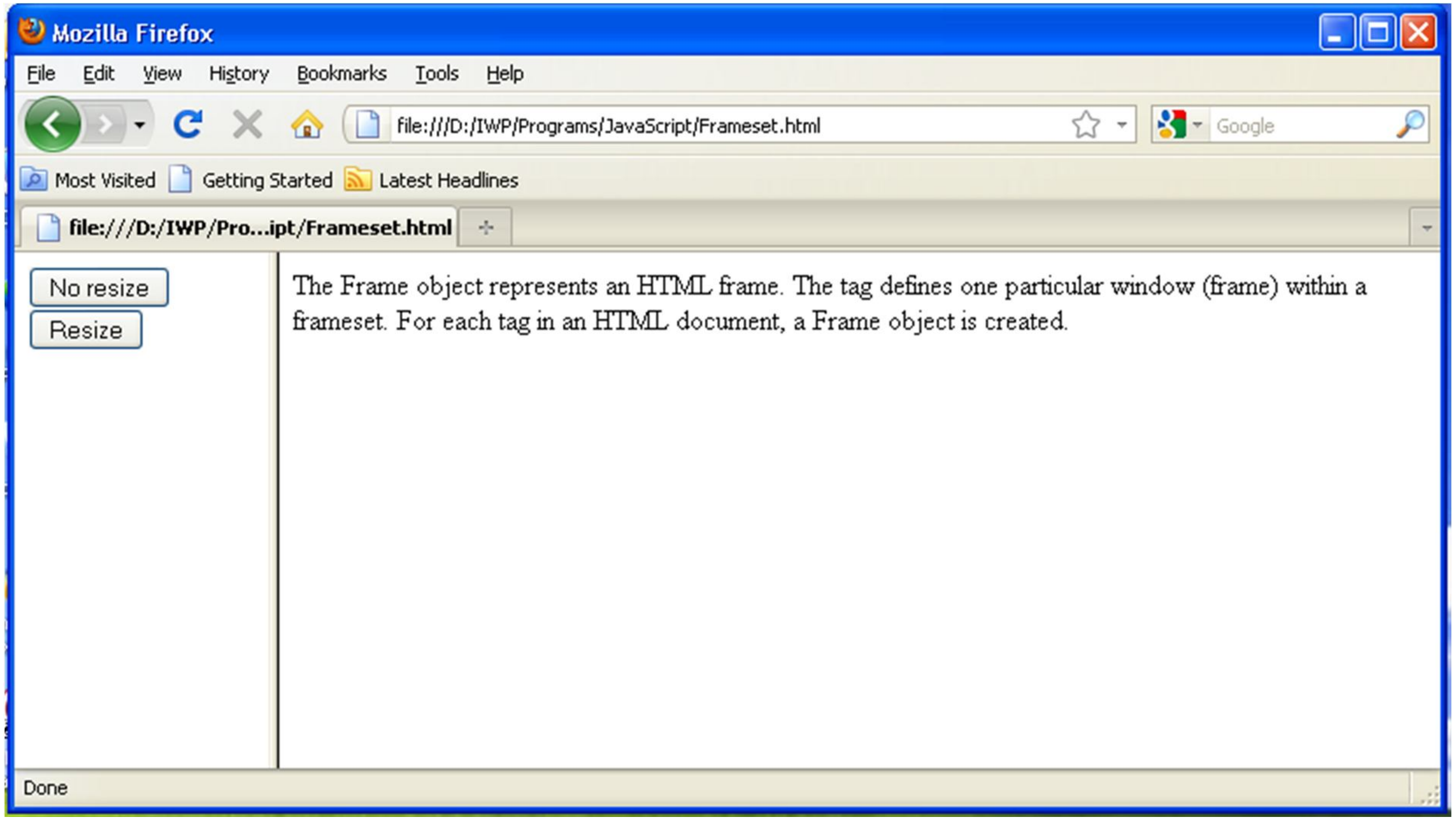
LeftFrameCode.html

```
<html><head><script type="text/javascript">
function disableResize()
  {
    parent.document.getElementById("rightFrame").noResize=true;
  }
function enableResize()
  {
    parent.document.getElementById("rightFrame").noResize=false;
  }
</script></head><body>
<input type="button" onclick="disableResize()" value="No resize“ >
<input type="button" onclick="enableResize()" value="Resize" >
</body></html>
```

RightFrameCode.html

```
<html>
<body>
The Frame object represents an HTML frame.
The <frame> tag defines one particular window (frame) within a
   frameset.
For each <frame> tag in an HTML document, a Frame object is created.
</body>
</html>
```

Output


Mozilla Firefox
File Edit View History Bookmarks Tools Help
file:///D:/IWP/Programs/JavaScript/Frameset.html
Google
Most Visited Getting Started Latest Headlines
file:///D:/IWP/Pro...ipt/Frameset.html
No resize
Resize
Done

The Frame object represents an HTML frame. The tag defines one particular window (frame) within a frameset. For each tag in an HTML document, a Frame object is created.

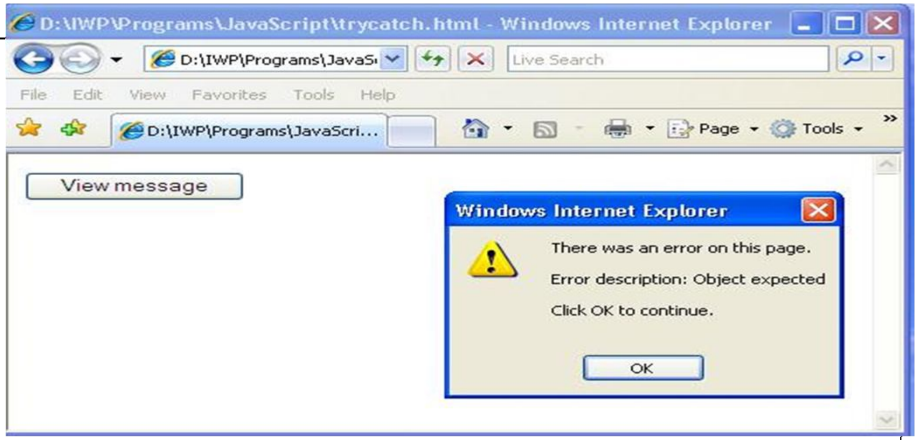
- Window.document.layer - a named document layer

Try...Catch Statement

- When browsing Web pages on the internet, we all have seen a JavaScript alert box telling us there is a runtime error and asking "Do you wish to debug?". Error message like this may be useful for developers but not for users. When users see errors, they often leave the Web page.

```
try
  {
  //Run some code here
  }
catch(err)
  {
  //Handle errors here
  }
```

```
<html> <head>
<script type="text/javascript">
var txt="";
function message()
{
try
   {        adddlert("Welcome guest!");    }
catch(err)
   {      txt="There was an error on this page.\n\n";
          txt+="Error description: " + err.message + "\n\n";
          txt+="Click OK to continue.\n\n";
           alert(txt);                              }         }
</script> </head>      <body>
<input type="button" value="View message" onclick="message()" >
</body> </html>
```


D:\IWP\Programs\JavaScript\trycatch.html - Windows Internet Explorer

Windows Internet Explorer

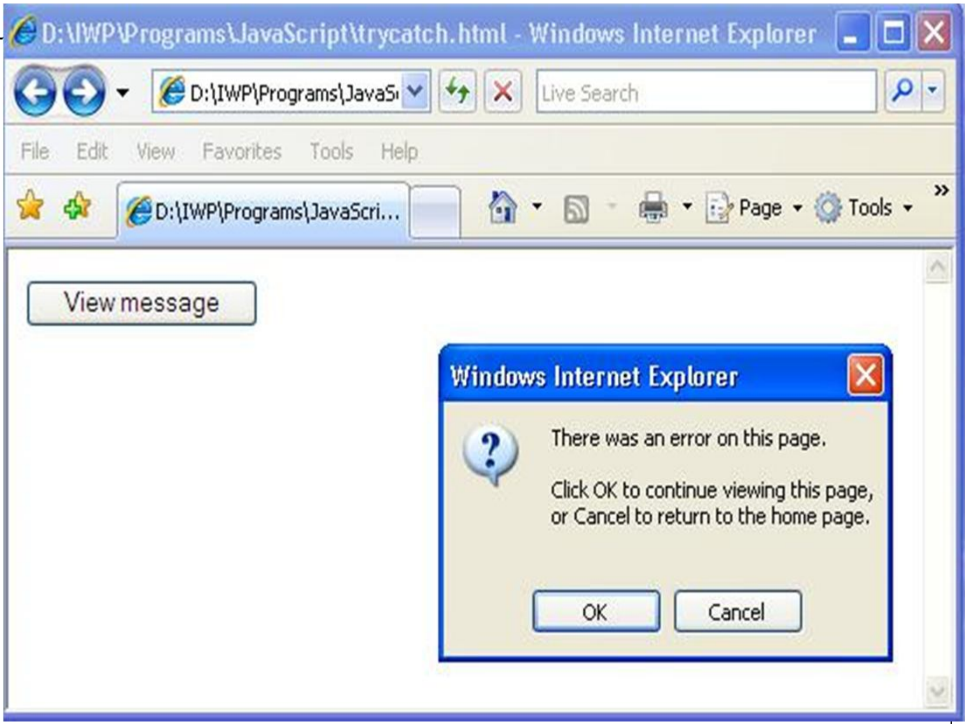
There was an error on this page.

Error description: Object expected

Click OK to continue.

D:\IWP\Programs\JavaS
Live Search
File Edit View Favorites Tools Help
D:\IWP\Programs\JavaScri...
Page
Tools
View message
OK

```
   <html>  <head>
 <script type="text/javascript">
   var txt="";
   function message()
   {
   try
{adddlert("Welcome guest!"); }
catch(err)
{
  txt="There was an error on this page.\n\n";
  txt+="Click OK to continue viewing this page,\n";
  txt+="or Cancel to return to the home page.\n\n";
  if(!confirm(txt))
  {     document.location.href="http://www.w3schools.com/";     }   }  }
</script> </head>  <body>
<input type="button" value="View message" onclick="message()" />
</body> </html>
```


D:\IWP\Programs\JavaScript\trycatch.html - Windows Internet Explorer
D:\IWP\Programs\JavaS
Live Search
File Edit View Favorites Tools Help
D:\IWP\Programs\JavaScri...
Page
Tools
View message
Windows Internet Explorer
OK
Cancel

There was an error on this page.

Click OK to continue viewing this page,
or Cancel to return to the home page.

```
<html>  <body> <script type="text/javascript">
var x=prompt("Enter a number between 5 and 10:","");
try
 {        if(x>10)
           {    throw "Err1";    }
          else if(x<5)
           {    throw "Err2";    }
          else if(isNaN(x))
           {    throw "Err3";    }  }
catch(err)
 {        if(err=="Err1")
           {    document.write("Error! The value is too high.");    }
          if(err=="Err2")
           {    document.write("Error! The value is too low.");    }
          if(err=="Err3")
           {    document.write("Error! The value is not a number.");    }   }
</script> </body> </html>
```

The throw statement allows you to create an exception. The exception can be a string, integer, Boolean or an object.

throw *exception;*