

Event Object

Accessing Event Object

- Event objects are important so that we can access specific about the event like find out which key the user has pressed, where on the screen the mouse button was clicked etc.
- Both Netscape and Microsoft invented event accessing model.
- W3C borrowed the event accessing model from Netscape.
- Two models are completely incompatible.

Event type

```
<HTML>
<body>
<a href="#" onclick="f()"> Click Me</a>
<script>
function f(e) {
    if (!e)
        e = window.event;
    alert(e.type);
}
</script>
</body>
</HTML>
```

Netscape/ W3C passes event object

MSIE

type property of event object indicates which event occurred. Displays "click"

Source of the event

- Microsoft specs: **srcElement** property
- Netscape/W3C specs: **target** property

```
<html>
<body><a href="#" onclick="f()"> Click Me</a>
<script>
function f(e) {
    var targ;
    if (!e) var e = window.event;

    if (e.target) targ = e.target; Netscape/
    else if (e.srcElement) W3C
    targ = e.srcElement; MSIE
    alert(targ);
}
</script></body></html>
```

Detecting key pressed

```
<html>
<body onKeyPress="f()">
<div id="write"></div>
<script>
var char = "";
function f(e) {
    var code;
    if (!e) var e = window.event;
    if (e.keyCode) code = e.keyCode;
    else if (e.which) code = e.which;
    char = char+String.fromCharCode(code);
var writearea=document.getElementById("write");
writearea.innerHTML=char;
}
</script></body></html>
```

MSIE

Netscape/
W3C

Mouse buttons

- Netscape specs :
 - **which** (Left button – 1, Middle button – 2, Right button – 3)
- W3C specs:
 - **button** (Left button – 0, Middle button – 1, Right button – 2)
- Microsoft specs:
 - **button** (Left button – 1, Middle button – 4, Right button – 2)
- The properties work for the **mouseUp** and **mouseDown**.

```
<html><body onMouseUp="m()">Click
<script>
function m(e) {
var rightclick;
if (!e) var e = window.event;
browsername=navigator.appName;
if (browsername.indexOf("Netscape") !=-1) {
    browsername="NS";
}
else{
if (browsername.indexOf("Microsoft") !=-1)
{browsername="MSIE";}
else {browsername="N/A";}
}
if (e.which){leftclick = (e.which == 1);}
else {
    if(browsername=="MSIE"){
        leftclick = (e.button == 1);}
    else{
        leftclick = (e.button == 0);}
    }
alert('Left click: ' + leftclick);
}
</script></body></html>
```

Mouse coordinates



- Multiple set properties:
 - **clientX, clientY**: position of the mouse pointer relative to the top-left corner of the browser window's client area. **MSIE**
 - **offsetX, offsetY**: relative to a positioned (absolute or relative) element. If the event occurred outside a positioned element, then the top-left corner of the document is used instead. Works in **MSIE** and Chrome. Firefox and Netscape uses **layerX, layerY**
 - **pageX, pageY**: the position relative to the top-left corner of the document. **Netscape**
 - **screenX, screenY** : position relative to the entire computer screen. Compatible for all browser. Rarely useful.

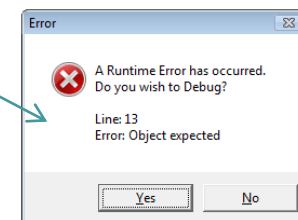
Best bet to get mouse coordinates that works on all browsers

```
<html>
<body onClick="m()">
<script>
function m(e) {
    var posx = 0;
    var posy = 0;
    if (!e) var e = window.event;
    if (e.pageX || e.pageY) {
        posx = e.pageX;
        posy = e.pageY;    }
    else if (e.clientX || e.clientY) {
        posx = e.clientX;
        posy = e.clientY ;    }
    alert("Position clicked (" + posx + ", " + posy + ")");
</script></body></html>
```

Allow only alphabets in a textbox

```
<html><body><form>NAME :  
<input type="text" name="nm" onKeyPress="alpha()" >  
</form>  
<script>  
function alpha(e) {  
    var code;  
    if (!e) var e = window.event;  
    if (e.keyCode) code = e.keyCode;  
    else if (e.which) code = e.which;  
    var character = String.fromCharCode(code);  
    if((character <'a' || character  
>'z') && (character <'A' || character >'Z')) {  
        str=document.forms[0].nm.value;  
        //document.forms[0].nm.value=substring(0,str.length-1);  
    }  
}  
</script>  
</body></html>
```

NAME: s11



Override the method

```
<html><body>
<form>
<input type="text" id="alpha" size="25">
</form>
<script type="text/javascript">
document.getElementById("alpha").onkeypress=function(e) {
var e=window.event || e;
var code=e.charCode || e.keyCode;
var character = String.fromCharCode(code);
//Allow alphabetical keys, plus BACKSPACE and SPACE
if((character <'a' || character >'z')&&(character <'A' ||
character >'Z')) return false;
else return true;
}
</script>
</body></html>
```