JAVA SCRIPT EVENT HANDLING

EVENTS AND EVENT HANDLING

▶ When a browser reacts on user action, then it is called as an event. For example, when you click on the submit button, then if the browser displays an information box.

▶ Event-driven programming is a style of programming in which pieces of code, event handlers, are written to be activated when certain events occur

▶ Events represent activity in the environment including, especially, user actions such as moving the mouse or typing on the keyboard

▶ An event handler is a program segment designed to execute when a certain event occurs

Examples of Events

▶ An HTML web page has finished loading

▶ An HTML input field was changed

▶ An HTML button was clicked

EXAMPLE 1

```
<!DOCTYPE html>
<html>
<body>
<button onclick = "document.getElementById('demo').innerHTML=Date()"> The time is? </button>
<p id="demo"></p>
</body>
</html
```

--this.innerHTML--

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript HTML Events</h2>
<button onclick="this.innerHTML=Date()">The time is?</button>
</body>
</html>
```

Madhu Bhan

Mouse Event

Event Performed	Event Handler(Attribute)	Description
click	onclick	When mouse click on an element
mouseover	onmouseover	When the cursor of the mouse comes over the element
mouseout	onmouseout	When the cursor of the mouse leaves an element
mousedown	onmousedown	When the mouse button is pressed over the element
mouseup	onmouseup	When the mouse button is released over the element
mousemove	onmousemove	When the mouse movement takes place.

Keyboard events (KEYBOARD EVENT OCCURS WHEN A USER INTERACTS WITH THE KEYBOARD

Event Performed	Event Handler(Attribute)	Description
Keydown & Keyup	onkeydown & onkeyup	When the user press and then release the key

Form events Form event occurs when the user performs some action within the form such as submitting the form, selecting input field, etc.

Event Performed	Event Handler(Attribute)	Description
focus	onfocus	When the user focuses on an element
submit	onsubmit	When the user submits the form
blur	onblur	When the focus is away from a form element
change	onchange	When the user modifies or changes the value of a form element

Window/Document Events

Event Performed	Event Handler(attribute)	Description
load	onload	When the browser finishes the loading of the page
unload	onunload	When the visitor leaves the current webpage, the browser unloads it
resize	onresize	When the visitor resizes the window of the browser

onmouseover Event-Example

```html
<html>
<body>
<p>This example demonstrates how to assign an "onmouseover" and "onmouseout" event to a h1 element.</p>
<h1 id="demo" onmouseover="mouseOver()" onmouseout="mouseOut()">Mouse over me</h1>
<script>
function mouseOver() {
  document.getElementById("demo").style.color = "red";}
function mouseOut() {
  document.getElementById("demo").style.color = "green";}
</script>
</body>
</html>
```
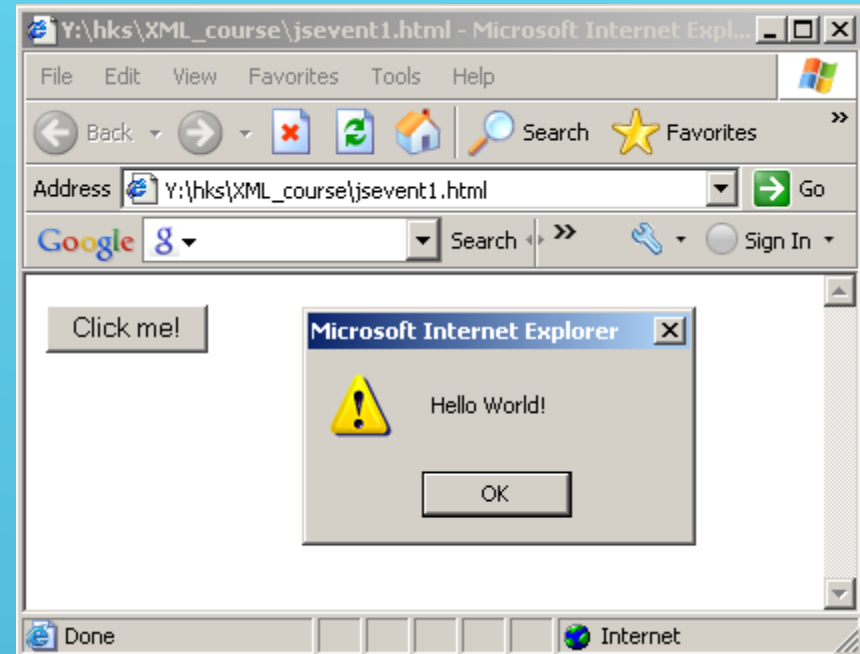
onfocus Event-Example

<html>

<head> Javascript Events**</head>**

<body>

<h2> Enter something here**</h2>**

<input type="text" id="input1" onfocus="focusevent()"**/>**

<script>

```
  function focusevent()

  {

     document.getElementById("input1").style.background=" aqua";}

  }
```

</script>

</body>

</html>

SIMPLE BUTTON EVENT EXAMPLE

```html
<<html>
<head>
<script type="text/javascript">
function displaymessage()
{
    alert("Hello World!");
}
</script>
</head>
<body>
<form>
<input type="button" value="Click me!"
    onclick="displaymessage()" >
</form>
</body>
</html>
```

Madhu Bhan

ONFOCUS() & ONBLUR() EXAMPLE

```
<html>
<head>
<title>XHTML onfocus and onblur Example</title>
</head>
<body>
<form>
<input type="text" onfocus="javascript:alert('text box is focussed')"
        onblur="javascript:alert('text box not focussed')">
</form>
</body>
</html>
```

Demo

PASSWORD FIELD IN TEXT BOX

```html
<html>
<body>

<form>
Username:
<input type="text" name="user">
<br>
Password:
<input type="password" name="password">
</form>
<p>
Note that when you type characters in a password field,
the browser displays asterisks or bullets instead of the characters.
</p>
</body>
</html>
```

VALIDATING FORM INPUT USING GETELEMENTBYID()

▶ We can get the elements in a form using getElementById().

▶ The getElementById() method returns a reference to the first object with the specified ID

▶ Syntax   : document.getElementById(param1)
  Parameters : param1   Required; the element's id value

```html
<html>
<head>
<title> GetElementID ex</title>
</head>
<body>
<script type="text/javascript">
function notEmpty()
{
var myTextField = document.getElementById('myText');
if(myTextField.value != "")
   alert("You entered: " + myTextField.value)
else
alert("Would you please enter some text?")
}
</script>
<form>
<input type='text' id='myText' />
<input type='button' onclick='notEmpty()'
  value='Form Checker' />
</form>
</body>
</html>
```

EXTERNAL JAVASCRIPT FILE

▶ External JavaScript file

▶ We can create external JavaScript file and embed it in many html page.

▶ It provides **code re-usability** because single JavaScript file can be used in several html pages.

▶ An external JavaScript file must be saved by .js extension. It is recommended to embed all JavaScript files into a single file. It increases the speed of the webpage.

▶ Let's create an external JavaScript file that prints Hello Javatpoint in a alert dialog box.

EXAMPLE CODE

message.js

```
function msg(){
 alert("Hello Javatpoint");  }
```

Example.html

```
<html>
<head>
<script type="text/javascript" src="message.js"></script>
</head>
<body>
<p>Welcome to JavaScript</p>
<form>
<input type="button" value="click" onclick="msg()"/>
</form>
</body>
</html>
```

Madhu Bhan

ADVANTAGES OF EXTERNAL JAVASCRIPT

▸ It helps in the reusability of code in more than one HTML file.

▸ It allows easy code readability.

▸ It is time-efficient as web browsers cache the external js files, which further reduces the page loading time.

▸ It enables both web designers and coders to work with html and js files parallelly and separately, i.e., without facing any code conflictions.

▸ The length of the code reduces as only we need to specify the location of the js file.

EXERCISE

▶ Develop, test, and validate an XHTML document that collects the following information from the user: last name, first name, middle initial, age (restricted to be greater than 17, and weight (restricted to the range of 45-75). You must have event handlers for the form elements that collect this information that check the input data for correctness. Messages in *alert* windows must be produced when errors are detected.