JavaScript HTML DOM Events

Reacting to Events

A JavaScript can be executed when an event occurs, like when a user clicks on an HTML element.

To execute code when a user clicks on an element, add JavaScript code to an HTML event attribute:

```
onclick=JavaScript
```

Examples of HTML events:

- When a user clicks the mouse
- When a web page has loaded
- When an image has been loaded
- When the mouse moves over an element
- When an input field is changed
- When an HTML form is submitted
- When a user strokes a key

In this example, the content of the `<h1>` element is changed when a user clicks on it:

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript HTML Events</h1>
<h2>The onclick Attribute</h2>

<h2 onclick="this.innerHTML='Ooops!'">Click on this text!</h2>

</body>
</html>
```

OUTPUT:

JavaScript HTML Events
The onclick Attribute
Click on this text!
AFTER “Click on this text!”
JavaScript HTML Events
The onclick Attribute

Ooops!

In this example, a function is called from the event handler:

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript HTML Events</h1>
<h2>The onclick Attribute</h2>

<h2 onclick="changeText(this)">Click on this text!</h2>

<script>
function changeText(id) {
  id.innerHTML = "Ooops!";
}
</script>
</body>
</html>
```

OUTPUT:

JavaScript HTML Events

The onclick Attribute

Click on this text!

AFTER “Click on this text!”

JavaScript HTML Events

The onclick Attribute

Ooops!

HTML Event Attributes

To assign events to HTML elements you can use event attributes.

Example

Assign an onclick event to a button element:

```
<button onclick="displayDate()">Try it</button>
```

In the example above, a function named `displayDate` will be executed when the button is clicked.

Assign Events Using the HTML DOM

The HTML DOM allows you to assign events to HTML elements using JavaScript:

Example

Assign an onclick event to a button element:

```
<script>
document.getElementById("myBtn").onclick = displayDate;
</script>
```

```
<!DOCTYPE html>

<html>

<body>

<h1>JavaScript HTML Events</h1>

<h2>The onclick Events</h2>

<p>Click "Try it" to execute the displayDate() function.</p>

<button id="myBtn">Try it</button>


<p id="demo"></p>

<script>

document.getElementById("myBtn").onclick = displayDate;

function displayDate() {

  document.getElementById("demo").innerHTML = Date();

}

</script>

</body>

</html>
```

Output:

JavaScript HTML Events

The onclick Events

Click "Try it" to execute the displayDate() function.

Try it

Wed Feb 28 2024 10:41:23 GMT+0530 (India Standard Time)

In the example above, a function named `displayDate` is assigned to an HTML element with the `id="myBtn"`.

The function will be executed when the button is clicked.

The onload and onunload Events

The `onload` and `onunload` events are triggered when the user enters or leaves the page.

The `onload` event can be used to check the visitor's browser type and browser version, and load the proper version of the web page based on the information.

The `onload` and `onunload` events can be used to deal with cookies.

```
<!DOCTYPE html>
<html>
<body onload="checkCookies()">
<h1>JavaScript HTML Events</h1>
<h2>The onload Attribute</h2>
<p id="demo"></p>
<script>
function checkCookies() {
  let text = "";
  if (navigator.cookieEnabled == true) {
    text = "Cookies are enabled.";
  } else {
    text = "Cookies are not enabled.";
  }
  document.getElementById("demo").innerHTML = text;
}
</script>
</body>
</html>
```

OUTPUT:

JavaScript HTML Events

The onload Attribute

Cookies are enabled.

The oninput Event

The `oninput` event is often to some action while the user input data.

Below is an example of how to use the oninput to change the content of an input field.

Example

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript HTML Events</h1>
<h2>The oninput Attribute</h2>

Enter your name: <input type="text" id="fname"
oninput="upperCase()">
<p>When you write in the input field, a function is triggered to
transform the input to upper case.</p>

<script>
function upperCase() {
  const x = document.getElementById("fname");
  x.value = x.value.toUpperCase();
}
</script>

</body>
</html>
```

OUTPUT:

JavaScript HTML Events

The oninput Attribute

Enter your name:
When you write in [HAI] , a function is triggered to transform the input to upper case.

The onchange Event

The `onchange` event is often used in combination with validation of input fields.

Below is an example of how to use the onchange. The `upperCase()` function will be called when a user changes the content of an input field.

EXAMPLE:

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript HTML Events</h1>
<h2>The onchange Attribute</h2>
Enter your name: <input type="text" id="fname"
onchange="upperCase()">
<p>When you leave the input field, a function transforms the input to
upper case.</p>
<script>
function upperCase() {
  const x = document.getElementById("fname");
  x.value = x.value.toUpperCase();
}
</script>
</body>
</html>
```

OUTPUT:

JavaScript HTML Events

The onchange Attribute

Enter your name: hai

When you leave the input field, a function transforms the input to upper case.

AFTER "coming out of the textbox"

JavaScript HTML Events

The onchange Attribute

Enter your name: HAI

When you leave the input field, a function transforms the input to upper case.

The onmouseover and onmouseout Events

The `onmouseover` and `onmouseout` events can be used to trigger a function when the user mouses over, or out of, an HTML element:

EXAMPLE:

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript HTML Events</h1>
<h2>The onmouseover Attribute</h2>
<div onmouseover="mOver(this)" onmouseout="mOut(this)"
style="background-color:#D94A38;width:120px;height:20px;padding:40p
x;">
Mouse Over Me</div>
```

```
<script>
function mOver(obj) {
  obj.innerHTML = "Thank You"
}
function mOut(obj) {
  obj.innerHTML = "Mouse Over Me"
}
</script>
</body>
</html>
```

OUTPUT: