



SNS COLLEGE OF TECHNOLOGY



Coimbatore-37.

An Autonomous Institution

COURSE NAME : 19CSE301-INTRODUCTION TO DATA SCIENCE

III YEAR/ VI SEMESTER

UNIT –CSS and Java script

Topic: Java script Event

Ms.G.Swathi

Assistant Professor

Department of Computer Science and Engineering



Introduction

- The change in the state of an object is known as an **Event**.
- In html, there are various events which represents that some activity is performed by the user or by the browser.
- When javascript code is included in HTML, js react over these events and allow the execution.
- This process of reacting over the events is called **Event Handling**.
- Thus, js handles the HTML events via **Event Handlers**.
- **For example**, when a user clicks over the browser, add js code, which will execute the task to be performed on the event.



Types

- Mouse events
- Keyboard events
- Form events
- Window/Document events
- Click Event
- MouseOver Event
- Focus Event
- Focus Event
- Load event



JavaScript On Click Events

- The **onclick** event generally occurs when the user clicks on an element.
- It allows the programmer to execute a JavaScript's function when an element gets clicked.
- This event can be used for validating a form, warning messages and many more.
- Using JavaScript, this event can be dynamically added to any element.
- It supports all HTML elements except <html>, <head>, <title>, <style>, <script>, <base>, <iframe>, <bdo>,
, <meta>, and <param>.
- It means we cannot apply the **onclick** event on the given tags.
- In HTML, we can use the **onclick** attribute and assign a JavaScript function to it.
- We can also use the JavaScript's **addEventListener()** method and pass a **click** event to it for greater flexibility.



Syntax

Syntax

- Now, we see the syntax of using the **onclick** event in [HTML](#) and in [javascript](#) (without **addEventListener()** method or by using the **addEventListener()** method).

- **In HTML**

```
<element onclick = "fun()">
```

- **In JavaScript**

```
object.onclick = function() { myScript };
```

- **In JavaScript by using the `addEventListener()` method**

```
object.addEventListener("click", myScript);
```



JavaScript Double Click Events

- The **dblclick** event generates an event on double click the element. The event fires when an element is clicked twice in a very short span of time. We can also use the JavaScript's **addEventListener()** method to fire the double click event.
- In [HTML](#), we can use the **ondblclick** attribute to create a double click event.

Syntax :

- Now, we see the syntax of creating double click event in HTML and in [javascript](#) (without using **addEventListener()** method or by using the **addEventListener()** method).

- **In HTML**

```
<element ondblclick = "fun()">
```

- **In JavaScript**

```
object.ondblclick = function() { myScript };
```

- **In JavaScript by using the `addEventListener()` method**

```
Object.addEventListener("dblclick", myScript);
```



JavaScript On load Events

- In JavaScript, this event can apply to launch a particular function when the page is fully displayed.
- It can also be used to verify the type and version of the visitor's browser.
- We can check what cookies a page uses by using the **onload** attribute.
- In HTML, the onload attribute fires when an object has been loaded.
- The purpose of this attribute is to execute a script when the associated element loads.
- In [HTML](#), the **onload** attribute is generally used with the **<body>** element to execute a script once the content (including CSS files, images, scripts, etc.) of the webpage is completely loaded.
- It is not necessary to use it only with [<body> tag](#), as it can be used with other HTML elements.
- The difference between the **document.onload** and **window.onload** is: **document.onload** triggers before the loading of images and other external content. It is fired before the **window.onload**.
- While the **window.onload** triggers when the entire page loads, including [CSS](#) files, script files, images, etc.

Syntax

```
window.onload = fun()
```



JavaScript On resize Events



- The **onresize** event in JavaScript generally occurs when the window has been resized. To get the size of the window, we can use the JavaScript's ***window.outerWidth*** and ***window.outerHeight*** events.
- We can also use the JavaScript's properties such as ***innerWidth***, ***innerHeight***, ***clientWidth***, ***ClientHeight***, ***offsetWidth***, ***offsetHeight*** to get the size of an element.
- In HTML, we can use the **onresize** attribute and assign a JavaScript function to it.
- We can also use the [JavaScript's addEventListener\(\)](#) method and pass a **resize** event to it for greater flexibility.
- **Syntax**
- Now, we see the syntax of using the **onresize** event in [HTML](#) and in [javascript](#) (without **addEventListener()** method or by using the **addEventListener()** method).
- **In HTML**

```
<element onresize = "fun()">
```
- **In JavaScript**

```
object.onresize = function() { myScript };
```
- **In JavaScript by using the addEventListener() method**

```
object.addEventListener("resize", myScript);
```




Thank
You