JS JavaScript
TUTORIAL

JavaScript change Event

Summary: in this tutorial, you'll learn about the JavaScript `change` event of the input text, radio button, checkbox, and select elements.

The `change` event occurs when the element has completed changing.

To attach an event handler to the change event of an element, you can either call the `addEventListener()` method:

```
element.addEventListener('change', function(){
    // handle change
});
```

or use the `onchange` attribute of the element. For example:

```
<input type="text" onchange="changeHandler()">
```

However, it is a good practice to use the `addEventListener()` method.

Using JavaScript change event for input elements

The change event of an `<input>` element fires when the `<input>` element loses focus. The `change` event does not fire when you're tying.

The following example shows the value of the input text when it loses focus.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>JavaScript change Event for Input Element</title>
</head>
<body>
    <label for="message">Message:</label>
    <input type="text" class="input" id="message" name="message">
    <button>Submit</button>
    <p id="result"></p>
    <script>
        let input = document.querySelector('.input');
        let result = document.querySelector('#result');
        input.addEventListener('change', function () {
            result.textContent = this.value;
        });
    </script>
</body>
</html>
```

Output

In this example, if you type some text on the `<input>` element and move focus to the button, the `change` event fires to show the entered text.

> Note that if you want to handle every change of the value, you use the `input` event instead.

Using JavaScript change event for radio buttons

A radio button fires the `change` event after you select it.

The following example shows how to handle the `change` event of the radio buttons:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>JavaScript change Event for Radio Buttons</title>
</head>
<body>
    <span>Status:</span>
    <input type="radio" id="pending" name="status">
    <label for="pending">Pending</label>
    <input type="radio" id="resolved" name="status">
    <label for="resolved">Resolved</label>
    <input type="radio" id="rejected" name="status">
    <label for="rejected">Rejected</label>
    <p id="result"></p>

    <script>
        let result = document.querySelector('#result');
        document.body.addEventListener('change', function (e) {
            let target = e.target;
            let message;
            switch (target.id) {
                case 'pending':
                    message = 'The Pending radio button changed';
                    break;
                case 'resolved':
                    message = 'The Resolved radio button changed';
                    break;
                case 'rejected':
                    message = 'The Rejected radio button changed';
                    break;
            }
            result.textContent = message;
        });
    </script>
</body>
</html>
```

Output

Status: ○ Pending ○ Resolved ○ Rejected

How it works:

- First, register an event handler to the `change` event of the `body`. When a radio button is clicked, its `change` event is bubbled to the body. This technique is called event delegation.
- Then, show a corresponding message based on which radio button is selected.

Using JavaScript change event for checkboxes

Similar to radio buttons, checkboxes fire the `change` event after selection, whether checked or unchecked. For example:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>JavaScript change Event for Checkboxes</title>
</head>
<body>
    <label for="status">Web Technology:</label>
    <input type="checkbox" id="html"> HTML
    <input type="checkbox" id="css"> CSS
    <input type="checkbox" id="js"> JavaScript
    <p id="result"></p>

    <script>
        let result = document.querySelector('#result');

        document.body.addEventListener('change', function (e) {
            let target = e.target;
            let message;
            switch (target.id) {
                case 'html':
                    message = 'The HTML checkbox changed';
                    break;
                case 'css':
                    message = 'The CSS checkbox changed';
                    break;
                case 'js':
                    message = 'The JavaScript checkbox changed';
                    break;
            }
            result.textContent = message;
        });
    </script>
</body>
</html>
```

Output

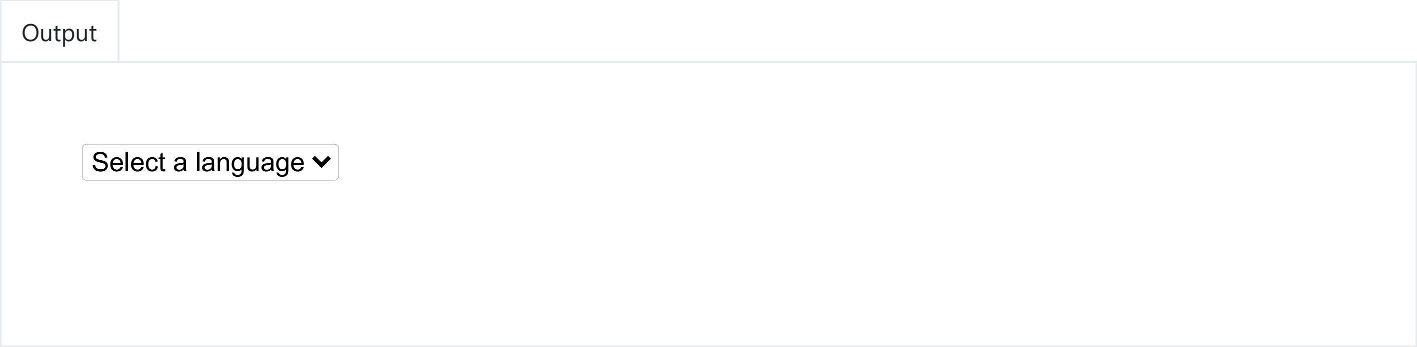
Web Technology: □ HTML □ CSS □ JavaScript

Using JavaScript change event for the select element

The `<select>` element fires the `change` event once the selection has completed.

The following example shows how to handle the `change` event of the `<select>` element. The `<p>` element with the id `result` will display the selected item:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>JavaScript change Event for Select element</title>
</head>
<body>
    <select id="lang">
        <option value="">Select a language</option>
        <option value="JavaScript">JavaScript</option>
        <option value="TypeScript">TypeScript</option>
        <option value="PHP">PHP</option>
        <option value="Python">Python</option>
        <option value="Java">Java</option>
    </select>
    <p id="result"></p>
    <script>
        let select = document.querySelector('#lang');
        let result = document.querySelector('#result');
        select.addEventListener('change', function () {
            result.textContent = this.value;
        });
    </script>
</body>
</html>
```



How it works:

- First, select the `<select>` element by its id (`lang`);
- Then, show the selected value in the `<p>` element.

Summary

- The `<input>` element fires the `change` event once it loses focus.

- The radio button, checkbox, and select elements fire the change event after they have been selected.