JS JavaScript
TUTORIAL

JavaScript Focus Events

Summary: in this tutorial, you will learn about the JavaScript focus events that keep track of the elements that users focus on.

Introduction to JavaScript focus events

The `focus` events fire when an element receives or loses focus. These are the two main focus events:

- `focus` fires when an element has received focus.
- `blur` fires when an element has lost focus.

The `focusin` and `focusout` fire at the same time as `focus` and `blur` , however, they bubble while the `focus` and `blur` do not.

The following elements are focusable:

- The **window** gains focus when you bring it forward by using `Alt+Tab` or clicking on it and loses focus when you send it back.
- **Links** when you use a mouse or a keyboard.
- **Form fields** like input text when you use a keyboard or a mouse.
- Elements with **tabindex**, also when you use a keyboard or a mouse.

JavaScript focus event examples

The following example shows how to handle the `focus` and `blur` events. When you move focus to the `password` field, the background changes to `yellow` . If you move the mouse to the `username` field, the background changes to `white` .

```
<!DOCTYPE html>
<html>
<head>
    <title>JavaScript Focus Events</title>
</head>
<body>
    <p>Move focus to the password field to see the effect:</p>

    <form id="form">
        <input type="text" placeholder="username">
        <input type="password" placeholder="password">
    </form>

    <script>
        const pwd = document.querySelector('input[type="password"]');

        pwd.addEventListener('focus', (e) => {
            e.target.style.backgroundColor = 'yellow';
        });

        pwd.addEventListener('blur', (e) => {
            e.target.style.backgroundColor = '';
        });
    </script>
</body>
</html>
```

Summary

- Use the `focus` event to handle the state of element when it has or loses the focus.