JS JavaScript TUTORIAL

JavaScript Keyboard Events

Summary: in this tutorial, you will learn how to work with JavaScript keyboard events including `keydown` , `keypress` , and `keyup` .

Introduction to JavaScript keyboard events

When you interact with the keyboard, the keyboard events are fired. There are three main keyboard events:

- `keydown` – fires when you press a key on the keyboard and fires repeatedly while you're holding down the key.
- `keyup` – fires when you release a key on the keyboard.
- `keypress` – fires when you press a character keyboard like `a` , `b` , or `c` , not the left arrow key, home, or end keyboard, … The `keypress` also fires repeatedly while you hold down the key on the keyboard.

The keyboard events typically fire on the text box, though all elements support them.

When you press a character key once on the keyboard, three keyboard events are fired in the following order:

1. `keydown`
2. `keypress`
3. `keyup`

Both `keydown` and `keypress` events are fired before any change is made to the text box, whereas the keyup event fires after the changes have been made to the text box. If you hold down a character key, the `keydown` and `keypress` are fired repeatedly until you release the key.

When you press a non-character key, the `keydown` event is fired first followed by the `keyup` event. If you hold down the non-character key, the `keydown` is fired repeatedly until you release the key.

Handling keyboard events

To handle a keyboard event, you follow these steps:

- First, select the element on which the keyboard event will fire. Typically, it is a text box.
- Then, use the `element.addEventListener()` to register an event handler.

Suppose that you have the following text box with the id `message` :

```html
<input type="text" id="message">
```

The following illustrates how to register keyboard event listeners:

```javascript
let msg = document.getElementById('#message');

msg.addEventListener("keydown", (event) => {
    // handle keydown
});

msg.addEventListener("keypress", (event) => {
    // handle keypress
});

msg.addEventListener("keyup", (event) => {
    // handle keyup
});
```

If you press a character key, all three event handlers will be called.

The keyboard event properties

The keyboard event has two important properties: `key` and `code` . The `key` property returns the character that has been pressed whereas the `code` property returns the physical key code.

For example, if you press the `z` character key, the `event.key` returns `z` and `event.code` returns `KeyZ` .

See the following example:

```
<!DOCTYPE html>
<html>
<head>
    <title>JavaScript Keyboard Events: Key/Code</title>
</head>
<body>
    <input type="text" id="message">

    <script>
        let textBox = document.getElementById('message');
        textBox.addEventListener('keydown', (event) => {
            console.log(`key=${event.key},code=${event.code}`);

        });
    </script>
</body>
</html>
```

If you type character `z`, you will see the following message:

```
key=z,code=KeyZ
```

How it works:

- First, select the text box with the id `message` by using the `getElementById()` method.
- Then, register a `keydown` event listener and log the key and code of the key that has been pressed.

Summary

- When you press a character key on the keyboard, the `keydown`, `keypress`, and `keyup` events are fired sequentially. However, if you press a non-character key, only the `keydown` and `keyup` events are fired.

- The keyboard `event` object has two important properties: `key` and `code` properties that allow you to detect which key has been pressed.
- The `key` property returns the value of the `key` pressed while the `code` represents a physical key on the keyboard.