

🚩 Current Skill DOM events

Add Events

Events are actions that occur as a result of the user's action or as result of state change of the elements of a DOM tree. For example: `click`, `load`, `focusin`, and `focusout`. These are all events that help us react with JavaScript.

We can assign JS functions to listen for these events in elements and react or respond when the event occurs.

Here is a list of events. There are three ways you can assign a function to a certain event.

If `foo()` is a custom function, you can register it as a click event listener (it will be called when the button element is clicked) in three ways:

```
// first way
<button onclick=foo>Alert</button>

// the second way
<div id="main">
  <button>Alert</button>
</div>
<script>
  // Getting the button
  var btn = document.querySelector('button');
  btn.onclick=foo;
</script>

// the third way
<div id="main">
  <button >Alert</button>
</div>
<script>
  // Getting the button
  var btn = document.querySelector('button');
```

```
btn.addEventListener('click', foo);
```

```
</script>
```

Remove Events

The `removeEventListener()` method **detaches an event listener** (that was previously added with the `addEventListener()` method) **from the event it is listening to**.

```
<div id="main">

  <button onclick="foo">Alert</button>

</div>

<script>

  // Getting the button

  let btn = document.querySelector('button');

  // Remove the event listener

  btn.removeEventListener('click',foo);

</script>
```

Event Types

The browser triggers many events. A full list is available in MDN, but here are some of the most common event types and event names:



- **Mouse events** (MouseEvent): mousedown, mouseup, click, dblclick, mousemove, mouseover, mousewheel, mouseout, and contextmenu
- **Keyboard events** (KeyboardEvent): keydown, keypress, and keyup
- **Form events**: focus, blur, change, and submit
- **Window events**: scroll, resize, hashchange, load, and unload

Example



```
<!DOCTYPE html>

<html lang="">

<head>

  <meta charset="">

  <meta name="viewport" content="width=, initial-scale=">
```



```
<title></title>
```

```
</head>
```

```
<body>
```

```
  <div class="d-flex h-100">
```

```
    <p>try pressing some keys:</p>
```

```
    <p id="log"></p>
```

```
  </div>
```

```
  <script>
```

```
    document.addEventListener('keydown', logKey);
```

```
    function logKey(event) {
```

```
      log.textContent += " " + e.code;
```

```
    }
```

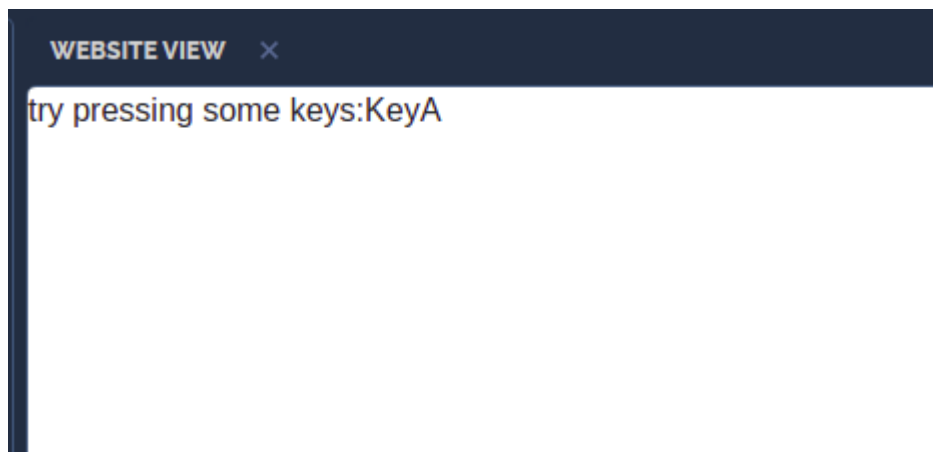
```
  </script>
```

```
</body>
```

```
</html>
```

Output

When pressing the key **a**, this function displays the following:



[< Previous](#)

[next >](#)

