# JavaScript-Events

The Document Object Model (DOM) is a programming interface for HTML and XML documents. It represents the structure of a document as a tree-like data structure, where each node in the tree corresponds to an element or content within the document.

Here are some key events in the DOM that you should know about:

**1. Node Events**

- `nodeInserted`: Fired when a new child node is inserted into a given DOM Element.
- `nodeRemoved`: Fired when a child node is removed from a given DOM Element.
- `nodesChanged`: Fired when multiple nodes are inserted or removed in a single event.

  Example:

  ```
  ```

document.addEventListener('nodeInserted', function(event) { console.log("A new element has been added:", event.target); });

```
**2. Mutation Events**

*   `DOMSubtreeModified`: Fires for any changes made to the document, such as adding
*   `DOMNodeRemoved`: Fired when a node is removed from the document.
*   `DOMNodeInserted`: Fired when a node is inserted into the document.

    Example:
    ```javascript
document.addEventListener('DOMSubtreeModified', function(event) {
  console.log("The DOM has been modified:", event.target);
});
```

**3. Form Events**
```

- `input`: Fires when the user types something in an input or textarea field.
- `focus`: Fired when a form element receives focus.
- `blur`: Fired when a form element loses focus.

  Example:

  ```
  ```

const input = document.getElementById('myInput');

input.addEventListener('input', function(event) { console.log("You typed:", event.target.value); });

input.addEventListener('focus', function() { console.log("The input field has been focused."); });

input.addEventListener('blur', function() { console.log("The input field has lost focus."); });

```
**4. Load Events**

*   `DOMContentLoaded`: Fired when the initial HTML document has been completely load
*   `load`: Fires when all resources on a page have finished loading.

    Example:
    ```javascript
document.addEventListener('DOMContentLoaded', function() {
  console.log("The DOM has been loaded.");
});

window.addEventListener('load', function() {
  console.log("The entire page, including images and subframes, has been loaded.");
});
```

**5. Mouse Events**

- `click`: Fired when a mouse button is pressed and released over an element.
- `dblclick`: Fires when a mouse button is double-clicked on an element.
- `mousedown`: Fired when the mouse button is pressed down over an element.
- `mouseup`: Fires when the mouse button is released over an element.

  Example:

  ```
  ```

const button = document.getElementById('myButton');
  ```

button.addEventListener('click', function() { console.log("The button has been clicked."); });

button.addEventListener('dblclick', function() { console.log("The button has been double-clicked."); });

```
**6. Keyboard Events**

*   `keydown`: Fires when a key is pressed down over an element.
*   `keyup`: Fired when a key is released over an element.

    Example:
    ```javascript
const input = document.getElementById('myInput');

input.addEventListener('keydown', function(event) {
  console.log("A key has been pressed:", event.key);
});

input.addEventListener('keyup', function(event) {
  console.log("A key has been released:", event.key);
});
```

**7. Scroll Events**

- `scroll`: Fires when the user scrolls a document, an element, or a region of a document.

  Example:

  ```
  ```

const scrollableDiv = document.getElementById('myScrollableDiv');

scrollableDiv.addEventListener('scroll', function() { console.log("The scrollable area has been scrolled."); });

```
These are just some examples of the events available in the DOM. Each event can be us
```

---

*Curated by Brajesh Kumar*
```