JavaScript Day - 18

Browser Default Actions

Browser Actions

Many events automatically lead to certain actions performed by the browser.

For instance:

- A click on a link – initiates navigation to its URL.
- A click on a form submit button – initiates its submission to the server.
- Pressing a mouse button over a text and moving it – selects the text.

If we handle an event in JavaScript, we may not want the corresponding browser action to happen, and want to implement another behavior instead.

Preventing Browser Actions

There are two ways to tell the browser we don’t want it to act:

- The main way is to use the event object. There’s a method event.preventDefault().
- If the handler is assigned using on<event> (not by addEventListener), then returning false also works the same.

Preventing Browser Actions

In this HTML, a click on a link doesn't lead to navigation; the browser doesn't do anything:

```
<a href="/" onclick="return false">Click here</a>
or
<a href="/" onclick="event.preventDefault()">here</a>
```

Preventing Browser Actions

Menu items are implemented as HTML-links <a>, not buttons <button>. There are several reasons to do so, for instance:

Many people like to use “right click” – “open in a new window”. If we use <button> or , that doesn’t work.

Search engines follow <a href="..."> links while indexing.

So we use <a> in the markup. But normally we intend to handle clicks in JavaScript. So we should prevent the default browser action.

Common Events

Load Events:

- onload: This event fires when the entire page, including images and other resources, has finished loading. This is commonly used to execute code that relies on the complete page structure.

Common Events

User Input Events:

- onclick: This event triggers when the user clicks on an element, typically a button or an image. It's often used to submit forms, trigger animations, or navigate to different parts of the webpage.
- onmouseover: This event fires when the user moves the mouse cursor over an element. It's commonly used to display tooltips or change the appearance of the element on hover.
- onmouseout: This event is the opposite of onmouseover and triggers when the mouse cursor moves away from an element.
- onfocus: This event fires when an element gains focus, such as when a user clicks on an input field. It's often used to highlight the field or display instructional text.
- onblur: This event happens when an element loses focus, such as when a user tabs away from an input field. It can be used for validation purposes.
- onchange: This event triggers when the value of an element changes. It's commonly used for form elements like text fields and dropdown menus to perform actions based on user input.

Common Events

Form Events:

- onsubmit: This event fires when a form is submitted, typically when the user clicks a submit button. It's often used to validate form data before submission.

Common Events

Keyboard Events:

- onkeydown: This event triggers when a key is pressed down. It's useful for capturing keyboard shortcuts or implementing custom functionality based on key presses.
- onkeyup: This event fires when a key is released. It can be used in conjunction with onkeydown for more complex interactions.

Common Events

Mouse Events:

- onclick (covered earlier): We discussed this for user clicks, but it also applies to mouse clicks.
- ondblclick: This event triggers when the user double-clicks on an element.
- onmousedown: This event fires when the user presses the mouse button down on an element.
- onmouseup: This event happens when the user releases the mouse button over an element.