

Preventing Default Behavior

Preventing default behavior

The event capturing and bubbling process can be stopped by calling the `stopPropagation()` method on the event object provided to the handler.

Once `stopPropagation()` is called, the browser stops the traversal process but still calls relevant registered handlers on the current node.

A web developer may want to prevent the browser from using a built-in handler for an event.

Example:

Whenever a user clicks a form's submit button, the web browser sends the form data to the web server. The event object's `preventDefault()` method stops the web browser from performing the built-in handler. The built-in handlers that are often prevented are clicking elements, submitting forms, and moving the mouse into or out of an element.

The example below uses two event handlers for the password textbox:

1. `preventSpaces()` is a keydown event handler that listens for key presses. If the space key is pressed, `event.preventDefault()` stops the space from appearing in the textbox.
2. `checkPassword()` is an input event handler that is called when the password input changes. `checkPassword()` displays Weak, Stronger, Moderate, or Strong in the `` tag depending on various criteria for the password.

Testing password strength.

Start typing a password. Verify the message to the right of the widget changes as the password is improved:

- abc - Weak
- abc1 - Stronger
- abc1D - Moderate
- abc1De - Strong

Copy and paste the following files and observes the output :

HTML FILE :

```
<label for="password">Password:</label>
```

```
<input type="text" id="password">
<span id="strength"></span>
```

JS FILE :

```
// Register event handlers
let passwordInput = document.querySelector("#password");
passwordInput.addEventListener("keydown", preventSpaces);
passwordInput.addEventListener("input", checkPassword);
```

```
function preventSpaces(event) {
  // Ignore space input
  if (event.key === " ") {
    event.preventDefault();
  }
}
```

```
function checkPassword() {
  let password = this.value;
  let passwordStrength = 0;
```

```
  // Password should be long enough
  if (password.length >= 6) {
    passwordStrength++;
  }
```

```
  // Password should contain at least one uppercase letter
  if (password.toLowerCase() !== password) {
    passwordStrength++;
  }
```

```
  // Password should have at least one digit
  let containsDigit = false;
  for (let i = 0; i < password.length; i++) {
    if (password[i] >= "0" && password[i] <= "9") {
      containsDigit = true;
    }
  }
```

```
  if (containsDigit) {
    passwordStrength++;
  }
```

```
}

// Convert passwordStrength into a descriptive word
let strengthWord = "Weak";
if (passwordStrength === 1) {
    strengthWord = "Stronger";
}
else if (passwordStrength === 2) {
    strengthWord = "Moderate";
}
else if (passwordStrength === 3) {
    strengthWord = "Strong";
}

// Display strengthWord
document.querySelector("#strength").innerHTML = strengthWord;
}
```