

Wprowadzenie do obsługi formularz

Wprowadzenie do obsługi formularz

Formularze w HTML służą do zbierania danych od użytkownika. JavaScript pozwala na dynamiczną interakcję z formularzami, takimi jak odczytywanie i walidowanie danych, kontrolowanie, kiedy formularz zostanie przesłany, oraz dynamiczne modyfikowanie formularza w odpowiedzi na interakcje użytkownika.

Przykładowy formularz HTML:

```
<form id="formularz">
  <label for="email">Email:</label>
  <input type="email" id="email" name="email">

  <label for="haslo">Hasło:</label>
  <input type="password" id="haslo" name="haslo">

  <button type="submit">Zaloguj się</button>
</form>
```

Odczytywanie danych z formularzy

Dzięki JavaScript możemy łatwo uzyskać dostęp do wartości wprowadzonych przez użytkowników w polach formularza. Możemy to zrobić, używając metody `value` dla danego pola.

Przykład odczytywania wartości z pól formularza:

```
let formularz = document.getElementById('formularz');
formularz.addEventListener('submit', function(event) {
  event.preventDefault(); // Zapobieganie domyślnemu przesyłaniu
formularza

  let email = document.getElementById('email').value;
  let haslo = document.getElementById('haslo').value;

  console.log('Email:', email);
  console.log('Hasło:', haslo);
});
```

Walidacja danych

Walidacja danych to proces sprawdzania, czy wprowadzone dane są zgodne z określonymi kryteriami. Przed przesłaniem formularza możemy upewnić się, że użytkownik wprowadził poprawne informacje, np. czy pole e-mail zawiera adres w odpowiednim formacie lub czy hasło ma wystarczającą liczbę znaków.

Przykład podstawowej walidacji:

```
let formularz = document.getElementById('formularz');
formularz.addEventListener('submit', function(event) {
    event.preventDefault();

    let email = document.getElementById('email').value;
    let haslo = document.getElementById('haslo').value;

    if (email === "" || !email.includes('@')) {
        alert('Proszę wprowadzić poprawny adres e-mail.');
```

// Można tu dodać kod, który faktycznie przesyła dane na serwer

```
    } else if (haslo.length < 6) {
        alert('Hasło musi zawierać co najmniej 6 znaków.');
```

// Można tu dodać kod, który faktycznie przesyła dane na serwer

```
    } else {
        console.log('Dane są poprawne, przesyłanie formularza...');
```

// Można tu dodać kod, który faktycznie przesyła dane na serwer

```
    }
});
```

Dynamiczna walidacja danych

Dynamiczna walidacja oznacza sprawdzanie poprawności danych w momencie ich wprowadzania, zamiast czekania na kliknięcie przycisku „Wyślij”. Takie podejście poprawia interaktywność i zapewnia lepszą jakość doświadczeń użytkownika.

Przykład dynamicznej walidacji pola e-mail:

```
let poleEmail = document.getElementById('email');
poleEmail.addEventListener('input', function() {
    let email = poleEmail.value;

    if (!email.includes('@')) {
        poleEmail.style.borderColor = 'red'; // Zmiana koloru ramki na
czzerwony
```

// Zmiana koloru ramki na zielony

```
    } else {
        poleEmail.style.borderColor = 'green'; // Zmiana koloru ramki na
zielony
```

// Zmiana koloru ramki na zielony

```
    }
});
```

Zapobieganie przesłaniu formularza

Formularze domyślnie odświeżają stronę po przesłaniu, co nie zawsze jest pożądane. Aby temu zapobiec, używamy metody `event.preventDefault()`, jak pokazano w poprzednich przykładach.

Zwykła walidacja a walidacja po stronie serwera

Walidacja w JavaScript, choć bardzo użyteczna, nie powinna być jedynym zabezpieczeniem. Nawet jeśli użytkownik pomyślnie przejdzie walidację po stronie klienta, musimy upewnić się, że dane są sprawdzone i odpowiednio zabezpieczone także na serwerze.

Zadania Powtórkowe

Zadanie 1:

Napisz funkcję, która wyświetla współrzędne kliknięcia użytkownika w dowolnym miejscu na stronie.

Zadanie 2:

Napisz skrypt, który zmienia kolor przycisku na niebieski, gdy zostanie na niego najechana myszka (mouseover), a na czerwony, gdy myszka zostanie z niego usunięta (mouseout).

Zadanie 3:

Napisz funkcję, która dodaje do pola tekstowego komunikat „Domyślna wartość”, gdy zostanie dwukrotnie kliknięte.

Zadania do samodzielnego wykonania

Zadanie 4:

Stwórz formularz z polem tekstowym „Imię” oraz przyciskiem „Wyślij”. Napisz walidację, która sprawdza, czy użytkownik wprowadził imię (pole nie może być puste).

Zadanie 5:

Napisz skrypt, który sprawdza, czy hasło wprowadzone w formularzu ma co najmniej 8 znaków oraz czy zawiera co najmniej jedną liczbę.

Zadanie 6:

Stwórz formularz, w którym użytkownik wpisuje swój wiek. Napisz walidację, która sprawdza, czy wprowadzony wiek jest liczbą i czy mieści się w przedziale 18-100.

Zadanie 7:

Napisz skrypt, który sprawdza, czy dwa pola „Hasło” i „Potwierdzenie hasła” w formularzu mają tę samą wartość. Jeśli nie, wyświetl komunikat o błędzie.

Zadanie 8:

Stwórz dynamiczną walidację pola e-mail, która sprawdza, czy wprowadzone dane zawierają „@” oraz

„.”

Zadanie 9:

Stwórz formularz z polami „Imię”, „Nazwisko” oraz „E-mail”. Napisz walidację, która sprawdza, czy pola nie są puste. Jeśli któreś z pól jest puste, podświetl je na czerwono.

Zadanie 10:

Stwórz formularz HTML, który zawiera następujące pola:

Imię – pole tekstowe (nie może być puste),

Nazwisko – pole tekstowe (nie może być puste),

Data urodzenia – pole typu date (powinno być w formacie RRRR-MM-DD),

Telefon – pole tekstowe (numer telefonu powinien mieć dokładnie 9 cyfr),

Email – pole tekstowe (adres e-mail musi zawierać znak @ oraz kropkę),

Adres zamieszkania – pole tekstowe (nie może być puste),

PESEL – pole tekstowe (numer PESEL powinien zawierać dokładnie 11 cyfr).

Walidacja formularza odbywa się dynamicznie, czyli sprawdzenie poprawności danych wprowadzonych przez użytkownika jest wykonywane natychmiast po zmianie wartości w polach, a nie dopiero po przesłaniu formularza. Gdy dane są poprawne, ramka pola powinna zmieniać kolor na zielony, a gdy błędne – na czerwony.

Dodatkowo formularz posiada przycisk Zresetuj, który usuwa wszystkie wprowadzone dane z formularza i resetuje obramowania pól do domyślnych.