



Programowanie 2022 / 2023

Laboratorium 7-8

Tworzenie formularzy w HTML.

Wykorzystanie JavaScript do walidacji



Wstęp

Tworzenie formularzy w HTMLu polega na umieszczeniu na stronie odpowiednich elementów, takich jak pola tekstowe, listy wyboru, pola wyboru (checkbox), przyciski, itp. Wszystkie te elementy są opakowane w tag `<form>`.

```
<form>
  <label for="name">Name:</label>
  <input type="text" id="name" name="name">
  <button type="submit">Submit</button>
</form>
```

W powyższym kodzie mamy pole tekstowe `<input>` o typie `text`, którego wartość będzie przekazywana pod nazwą `name` wraz z przyciskiem `<button>` typu `submit`, który wysyła formularz.

Tag `label` jest opcjonalny, ale warto go stosować, ponieważ ułatwia on dostępność dla osób korzystających z czytników ekranowych. Atrybut `for` jest powiązany z `id` pola tekstowego i wskazuje, którego elementu etykieta dotyczy.

Przykłady typów `input`:

- `text` - pojedyncza linia tekstu
- `password` - pole tekstowe, gdzie wpisywany tekst jest maskowany (zamieniany na gwiazdki lub kropki)
- `checkbox` - pole wyboru, w którym użytkownik może zaznaczyć lub odznaczyć opcję
- `radio` - przyciski wyboru, gdzie użytkownik może wybrać jedną z kilku opcji
- `submit` - przycisk wysyłający dane formularza do serwera
- `reset` - przycisk resetujący wartości formularza do stanu pierwotnego
- `button` - przycisk, który może być używany do uruchomienia skryptów lub innych działań w JavaScript

Walidacja formularza w HTML jest możliwa dzięki wykorzystaniu atrybutów formularza, takich jak `required`, `min`, `max`, `pattern`, `email` itp.

Przykładowo, atrybut `required` sprawia, że pole formularza staje się wymagane, a użytkownik nie może wysłać formularza, jeśli pole jest puste. Atrybuty `min` i `max` pozwalają określić minimalną i maksymalną wartość, jaką użytkownik może wprowadzić w polu. Atrybut `pattern` pozwala na wprowadzenie wzorca, który musi zostać dopasowany, na przykład do formatu kodu pocztowego lub numeru telefonu. Atrybut `email` sprawdza, czy wprowadzony tekst jest poprawnym adresem e-mail.

```
<form>
  <label for="name">Imię:</label>
  <input type="text" id="name" name="name" required>

  <label for="email">E-mail:</label>
  <input type="email" id="email" name="email" required>
```



```
<label for="phone">Telefon:</label>
<input type="tel" id="phone" name="phone" pattern="[0-9]{9}" required>

<input type="submit" value="Wyślij">
</form>
```

Walidacja formularza za pomocą JavaScript pozwala na bardziej zaawansowaną i spersonalizowaną walidację niż domyślne mechanizmy HTML. Pola formularza można sprawdzać przed wysłaniem ich lub w trakcie ich wprowadzania.

W celu walidacji pola w trakcie wprowadzania danych można użyć eventu 'input' dla pól formularzy.

```
nameInput.addEventListener('input', validateName);
```

Przed wysłaniem formularza można wykorzystać event 'submit'.

```
form.addEventListener('submit', (event) => {
  event.preventDefault();
  validateName();
});
```

Wywołanie event.preventDefault() w funkcji obsługi zdarzenia spowoduje zatrzymanie domyślnej akcji, dzięki czemu programista może wykonać własne działania w odpowiedzi na zdarzenie.

setCustomValidity to metoda elementu, która pozwala na ustawienie niestandardowej wiadomości błędu walidacji dla pola formularza.

```
const emailField = document.getElementById('name');
emailField.addEventListener('input', () => {
  const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
  if (!emailRegex.test(field.value)) {
    field.setCustomValidity("Proszę podać poprawny adres e-mail.");
  }
});
```

Jeśli jest potrzeba przekonwertowania wartości z formularza do obiektu można wykorzystać np. metodę reduce:

```
const form = document.querySelector('form');
const data = new FormData(form);
const formDataObj = [...data.entries()].reduce((acc, [key, value]) => {
  acc[key] = value;
  return acc;
}, {});
console.log(formDataObj);
```



Zadania

Zadanie 1. Utwórz formularz zawierający następujące pola odpowiedniego typu:

- Imię
- Nazwisko
- Email
- Hasło
- Płeć
- Telefon
- Data urodzenia
- Kraj
- Przycisk zapisz

Każde pole powinno być walidowane przy użyciu wbudowanej walidacji formularzy. Np. hasło powinno zawierać co najmniej 8 znaków.

Formularz powinien być odpowiednio wystylowany.

Zadanie 2. Z wykorzystaniem JavaScript stwórz funkcje które będą wykorzystane do walidacji pól formularza:

- funkcja sprawdzająca czy pole posiada wartość
- funkcja sprawdzająca długość znaków
- funkcja sprawdzająca poprawność adresu email
- funkcja sprawdzająca trudność hasła

Usuń walidację z zadania 1.

Aby wyłączyć walidację HTML5 dla formularza, należy dodać atrybut novalidate do tagu formularza.

Przy wprowadzaniu wartości do formularza lub po kliknięciu zapisz powinna nastąpić walidacja formularza. Do wyświetlenia błędów w formularzu wykorzystaj metodę setCustomValidity. Gdy formularz jest poprawnie wypełniony wyświetl alert.

Zadanie 3. Zmodyfikuj formularz tak aby wiadomość walidacji pojawiała się w elemencie pod polem.

Aby uzyskać dostęp następnego elementu od pola możesz wykorzystać querySelector np.

`document.querySelector(„[name='firstName'] + span.error”)`

querySelector zwróci element span z klasą error po polu z nazwą firstName.

Zadanie 4. Stwórz funkcję walidującą datę urodzenia. Osoba wypełniająca formularz powinna być pełnoletnia.

Zadanie 5. Dodaj do formularza pole „Powtórz hasło”. Dodaj do niego walidację sprawdzającą czy wartość jest taka sama jak wartość pola „Hasło”.



Zadanie 6. Dodaj do formularza kolejne pola:

- Województwo - nieaktywne
- Adres zamieszkania - nieaktywne
- Adres korespondencyjny jest taki sam jak adres zamieszkania – Pole typu checkbox domyślnie zaznaczone.
- Adres korespondencyjny – Pole ma być domyślnie ukryte oraz nieaktywne

Gdy checkbox zostanie odznaczony ma pojawić się pole „Adres korespondencyjny”.

Pole „województwo” ma stać się aktywne po wybraniu kraju. „Adres zamieszkania” oraz „Adres korespondencyjny” gdy pole „województwo” będzie miało wartość.

Zadanie 7. Zmodyfikuj formularz tak aby pole „Województwo” było listą do wyboru gdy wartość pola „Kraj” jest równa Polsce. W przeciwnym wypadku powinno być to pole tekstowe.

Zadanie 8. Zmień typ pola „Telefon” na tekst. Za pomocą JS ogranicz wpisywaną wartość tylko do cyfr.