

Formularze HTML

Czas: 2 × 45 min

praca indywidualna

Cel: Budowa formularza zgłoszeniowego z różnymi typami pól, poprawnymi etykietami i walidacją natywną oraz obserwacja danych wysyłanych metodą GET.

Po zajęciach potrafisz:

- dobiera typ pola input do rodzaju danych (text, email, tel, date, number, range, radio, checkbox)
- łączy każde pole z etykietą label (for + id) i grupuje pola w fieldset z legend
- stosuje atrybuty walidacji required, pattern, min, max, maxlength
- odczytuje wysłane pary nazwa=wartość w pasku adresu i w F12 → Network
- wyjaśnia, dlaczego walidacja w przeglądarce nie zastępuje sprawdzania danych na serwerze

Sprzęt i materiały

- komputer z Windows 11
- folder `r1_06_start` z Pliki_do_zadan (plik `potwierdzenie.html`)

Oprogramowanie

- **Visual Studio Code** — rozszerzenie Live Server (autor: Ritwick Dey) — instalacja z panelu Rozszerzenia (Ctrl+Shift+X) code.visualstudio.com
- **Przeglądarka Chrome, Edge lub Firefox** — narzędzia deweloperskie — klawisz F12
- **Walidator W3C (Nu Html Checker)** — sprawdzanie przez wklejenie kodu (text input) lub wgranie pliku (file upload) validator.w3.org/nu

Przygotowanie przed zajęciami

- Przypomnij sobie z wykładu 1.05 różnicę między metodami GET i POST.

Bezpieczeństwo

- Metodą GET dane trafiają do adresu URL, historii przeglądarki i logów serwera — nigdy nie wysyłaj tak haseł ani danych wrażliwych.
- Do testów wpisuj dane fikcyjne (adresy z domeny `example.com`).

Przebieg zadania

1 Szkielet formularza

Formularz zapisów na Szkolny Bieg na Orientację wyśle dane do strony `potwierdzenie.html`, która pokaże, co dotarło.

```
<form action="potwierdzenie.html" method="get">
  <fieldset>
    <legend>Dane uczestnika</legend>
    <p>
      <label for="imie">Imię *</label>
      <input type="text" id="imie" name="imie" required>
    </p>
    ...
  </fieldset>
</form>
```

1. Utwórz folder `zad06`, skopiuj do niego `potwierdzenie.html` (nie zmieniaj go) i utwórz `index.html` z nagłówkiem strony, `main` i stopką.
2. Każde pole umieść w akapicie, z etykietą `label` połączoną przez `for` i `id`. Atrybut `name` jest obowiązkowy — bez niego pole nie zostanie wysłane.

2 Pola formularza

Pola oznaczone * mają atrybut `required`.

1. Fieldset „Dane uczestnika”: imię * i nazwisko * (text, `maxlength` 30 i 40), e-mail * (email), telefon opiekuna (tel, dokładnie 9 cyfr — `pattern="[0-9]{9}"` i odpowiedź w `title`), data urodzenia * (date, od 2006-01-01 do 2013-12-31).
2. Klasa * — lista `select` z pierwszą opcją „— wybierz —” o pustej wartości oraz dwiema grupami `optgroup`: Technik informatyk (1T, 2T, 3T) i Technik programista (1P, 2P, 3P).
3. Fieldset „Udział w zawodach”: trasa * jako trzy przyciski `radio` o wspólnej nazwie (3 km, 5 km, 8 km), kod drużyny (3 wielkie litery, myślnik, 2 cyfry, np. ABC-12), liczba osób towarzyszących (number 0–3, domyślnie 0), ocena kondycji (range 1–10).
4. Trzy pola `checkbox` o wspólnej nazwie `dodatki` (posiłek, koszulka, transport) oraz `textarea` „Uwagi” (4 wiersze, maks. 300 znaków).
5. Poza fieldsetami: obowiązkowa zgoda na regulamin (checkbox) oraz przyciski `button` typu submit i reset.

3 Wysłanie i obserwacja danych

1. Uruchom stronę przez Live Server. Kliknij etykietę „Imię” — kursor powinien przejść do pola. Sprawdź to dla kilku pól (to test poprawności `for / id`).
2. Wypełnij formularz poprawnymi danymi (zaznacz 2 dodatki) i wyślij. Przepisz z paska adresu część po znaku `?`. Jak zapisano znak `@` i spacje?
3. Wróć, wyślij ponownie z otwartym F12 → Network. Kliknij żądanie `potwierdzenie.html` → Payload (ładunek) i zrób zrzut parametrów.
4. Odpowiedz na podstawie tabeli w `potwierdzenie.html`: czy wysłano niezaznaczone checkboxy? Ile razy pojawia się `dodatki`? Co wysłał pusty telefon? Czy wysłano przyciski?

4 Walidacja natywna i jej ograniczenia

1. Przetestuj i zapisz komunikaty przeglądarki dla: pustego imienia, e-maila `ola@`, telefonu z 8 cyframi, kodu `abc-12`, daty spoza zakresu, niezaznaczonej trasy.
2. W F12 → Elements usuń atrybut `required` z pola nazwisko (prawy przycisk → Edit attribute lub Delete) i wyślij formularz z pustym nazwiskiem. Co się stało? Zapisz wniosek o zaufaniu do danych z przeglądarki.
3. Dodaj tymczasowo do znacznika `form` atrybut `novalidate` i sprawdź, czy formularz z błędami zostanie wysłany. Usuń atrybut.
4. Sprawdź `index.html` walidatorem W3C i popraw błędy.

W sprawozdaniu umieść

- ☐ zrzut wypełnionego formularza i zrzut tabeli z `potwierdzenie.html`
- ☐ część adresu po `?` oraz zrzut zakładki Payload w F12
- ☐ tabela testów walidacji natywnej (pole, wpisana wartość, komunikat)
- ☐ wnioski z testów usunięcia `required` i `novalidate`
- ☐ odpowiedzi na pytania kontrolne

Pytania kontrolne

1. Czym różni się atrybut `id` od `name` w polu formularza? Który z nich trafia do serwera?
2. Dlaczego przyciski radio jednej grupy muszą mieć ten sam `name`, a różne `value`?
3. Kiedy wybierzesz metodę GET, a kiedy POST? Podaj po jednym przykładzie.
4. Dlaczego dane z formularza trzeba sprawdzić ponownie na serwerze, mimo atrybutów `required` i `pattern`?

Kryteria oceny

Kryterium	Punkty
Wszystkie wymagane typy pól (text, email, tel, date, select z optgroup, radio, number, range, checkbox, textarea, button)	5
Etykiety label for/id dla każdego pola, 2 × fieldset z legend, poprawne atrybuty name/value	3
Atrybuty walidacji: required, pattern (2), min/max, maxlength zgodnie z treścią	3
Obserwacja wysłanych danych: adres, Payload, odpowiedzi o checkboxach i pustych polach	3
Tabela testów walidacji i wnioski z testów required/novalidate; walidator bez błędów	3
Odpowiedzi na pytania kontrolne	3
Razem	20

Sprawozdanie przygotuj na wzorze dołączonym do zadania (plik .docx o tym samym numerze).