

Podstawy informatyki

HTML5 / CSS3

Laboratorium 13

Projekt końcowy: synteza HTML/CSS

Czas trwania: 2 × 45 min**Tryb:** studia stacjonarne**Jednostka:** Katedra Elektrotechniki i Technologii Nadprzewodnikowych**Prowadzący:** dr inż. Michał Łanczont

Cel ćwiczenia

Celem jest samodzielne zaprojektowanie i wykonanie małej witryny łączącej materiał z laboratoriów 1–12: semantyczny HTML, formularze lub tabele, hiperłącza, grafika, model pudełkowy, Flexbox, pozycjonowanie, responsywność i semantyczne znaczniki tekstowe.

Założenia projektu

Projekt końcowy podsumowuje materiał z laboratoriów L1–L12. Student wykonuje jednostronicową albo wielostronicową witrynę typu landing page, mini-serwis lub portfolio. Projekt pozostaje projektem **HTML/CSS bez obowiązkowego JavaScriptu**; jego celem jest integracja struktury, semantyki, formularzy lub tabel, grafiki, modelu pudełkowego, Flexboxa, pozycjonowania i responsywności.

Zasada pracy

Nie chodzi o mechaniczne “odhaczenie” znaczników. Każdy element powinien mieć uzasadnienie: tabela ma prezentować dane tabelaryczne, formularz zbierać dane, **article** zawierać treść samodzielną, a element absolutny realizować konkretny efekt nakładki lub oznaczenia.

Wymagania HTML – zachowany zakres instrukcji źródłowej

1. Poprawny szkielet HTML z UTF-8, viewportem i sensownym **title**.
2. Semantyczna struktura obejmująca co najmniej **header**, **nav**, jeden widoczny **main**, **section** lub **article** oraz **footer**.
3. Nagłówki, akapity, lista oraz co najmniej jeden obraz z poprawnym **alt** (dla obrazu dekoracyjnego dopuszczalne **alt=""**).
4. Menu z co najmniej trzema linkami: minimum jedna kotwica w bieżącej stronie i co najmniej jeden link do podstrony projektu.
5. Formularz *lub* tabela danych. Jeżeli wybrano tabelę, powinna mieć **caption**, **thead/tbody** i sensowne użycie **colspan** lub **rowspan**. Jeżeli wybrano formularz, powinien mieć etykiety, **name**, poprawne typy pól i podstawową walidację HTML.
6. Osadzony **iframe** jest opcjonalny, jeżeli ma sens w temacie. Jeżeli jest użyty, musi mieć **title** i rzeczywisty, działający zasób lokalny, **srcdoc** lub usługę pozwalającą na osadzanie – nie fikcyjny URL.

7. W treści należy użyć znaczników semantycznych tekstu: co najmniej jeden z grupy **strong/em**, jeden z grupy **code/var/kbd/samp** oraz jeden z grupy **abbr/q/cite/dfn**, jeżeli pasują do tematu.

Wymagania CSS

1. Zewnętrzny arkusz CSS i globalny **box-sizing: border-box** dla elementów i pseudoelementów.
2. Spójna typografia, kolory, odstępy i stany linków/przycisków.
3. Flexbox jako co najmniej jeden rzeczywisty układ (np. menu, karty, kolumny). Można dodatkowo użyć Grid.
4. Co najmniej jeden uzasadniony wzorzec **relative + absolute**, np. etykieta w rogu karty.
5. Mobile-first i co najmniej jeden breakpoint wynikający z zachowania treści. W praktyce projekt powinien być przetestowany w kilku szerokościach.
6. Widoczny fokus klawiatury i brak interakcji dostępnych wyłącznie przez hover.
7. Media nie mogą powodować poziomego przewijania całej strony.

Struktura katalogów

```
projekt/
|-- index.html
|-- o-mnie.html           (jeżeli projekt jest wielostronicowy)
|-- css/
|   '-- style.css
|-- images/
|   |-- hero.jpg
|   '-- avatar.jpg
|-- pages/                (opcjonalnie)
|-- README.txt
'-- raport.pdf            (jeżeli wymagany przez prowadzącego)
```

Nazwy plików powinny być krótkie, jednoznaczne i pozbawione przypadkowych spacji. Należy stosować ścieżki względne, aby projekt działał po przeniesieniu całego katalogu na inny komputer.

Etapy realizacji

Etap 1. Plan

Przed kodowaniem przygotuj prostą makietę: nazwa strony, menu, kolejność sekcji, lista potrzebnych podstron i zasobów. Określ, gdzie potrzebna jest tabela lub formularz.

Etap 2. HTML bez dekoracji

Zbuduj najpierw strukturę i treść. Strona bez CSS ma pozostawać logiczna, czytelna i możliwa do przejścia klawiaturą.

Etap 3. CSS i layout

Dodaj typografię, box model, układ Flexbox/Grid, stany interakcji i element pozycjonowany. Unikaj stylów liniowych w finalnym projekcie.

Etap 4. Responsywność

Zacznij od wąskiego widoku. Rozszerz układ przez `min-width` wtedy, gdy zawartość uzasadnia zmianę. Sprawdź obrazy, tabelę/formularz i menu.

Etap 5. Walidacja i testy

Sprawdź HTML, CSS, odsyłacze, ścieżki obrazów, formularz/tabelę, fokus Tab, kontrast i co najmniej trzy szerokości viewportu. Usuń ostrzeżenia wynikające z rzeczywistych błędów składniowych.

Przykładowa minimalna strona projektu

Poniższy szkielek nie jest gotowym projektem do skopiowania, lecz pokazuje minimalne powiązanie elementów omawianych na laboratoriach.

```
<body>
  <a class="skip-link" href="#main-content">Przejdź do treści</a>
  <header>
    <h1>Projekt automatyki domowej</h1>
  </header>
  <nav aria-label="Nawigacja główna">
    <a href="index.html">Start</a>
    <a href="#funkcje">Funkcje</a>
    <a href="o-projekcie.html">O projekcie</a>
  </nav>
  <main id="main-content">
    <section id="funkcje">
      <h2>Funkcje systemu</h2>
      <article class="card">
        <h3>Pomiar temperatury</h3>
        <p>Układ wykorzystuje czujnik <abbr title="...">...</abbr>.</p>
      </article>
    </section>
  </main>
  <footer>...</footer>
</body>
```

Najczęstsze przyczyny utraty punktów

- brak spójnej hierarchii nagłówków lub używanie elementów semantycznych przypadkowo,
- niedziałające ścieżki względne po przeniesieniu katalogu,
- obraz bez sensownego alt albo fikcyjny plik, którego nie ma w archiwum,
- formularz bez label/name lub tabela bez nagłówków i caption,
- brak widocznego fokusu klawiatury,
- layout poprawny tylko przy jednej szerokości ekranu,
- zastosowanie relative/absolute tylko “dla punktów”, bez sensownej funkcji,
- nieużywany, powtarzający się lub sprzeczny CSS pozostawiony w projekcie.

Weryfikacja końcowa projektu

Lista kontrolna

1. Otwórz każdą podstronę z katalogu projektu bez serwera i sprawdź ścieżki względne.
2. Przejdź menu myszą i klawiaturą; fokus ma być widoczny.
3. Sprawdź kotwicę wewnętrzną i link do podstrony.
4. Zweryfikuj alt obrazów i brak złamanych zasobów.
5. Jeżeli jest formularz, sprawdź etykiety i walidację; jeżeli tabela, sprawdź caption i nagłówki.
6. Sprawdź działanie Flexboxa przy zwięźaniu okna.
7. Sprawdź element relative/absolute i upewnij się, że nie zasłania treści na małym ekranie.
8. Zweryfikuj semantyczne znaczniki tekstu z L12.
9. Uruchom walidację HTML i przejrzyj CSS w DevTools.
10. Usuń fikcyjne linki, puste sekcje i nieużywane reguły CSS.

Kryteria oceny

Obszar	Punkty
Struktura i semantyka HTML, hierarchia nagłówków, nawigacja	20
Tabela lub formularz: poprawność struktury i dostępność	10
CSS: kaskada, typografia, box model, spójność wizualna	15
Układ Flexbox/Grid i poprawne pozycjonowanie	15
Responsywność mobile-first i zachowanie mediów	15
Dostępność: alt, label, fokus, sensowne elementy semantyczne	10
Poprawność techniczna: walidacja, ścieżki, brak błędów krytycznych	10
Organizacja plików, czytelność kodu i krótka dokumentacja	5
Razem	100

Warunek podstawowy

Projekt, który nie uruchamia się, ma brakującą stronę główną albo zawiera skopiowany kod, którego student nie potrafi wyjaśnić, wymaga poprawy przed ostatecznym odbiorem. Szczegółowy próg oceny ustala prowadzący zgodnie z zasadami przedmiotu.

Pliki do oddania

Należy przekazać cały katalog projektu, nie pojedynczy plik HTML. Jeżeli prowadzący wymaga raportu, powinien on krótko opisywać temat, strukturę strony, zastosowane mechanizmy HTML/CSS, testy responsywności oraz najważniejsze problemy i sposób ich rozwiązania.

Materiały pomocnicze

- HTML Living Standard: <https://html.spec.whatwg.org/>
- MDN Web Docs – HTML: <https://developer.mozilla.org/docs/Web/HTML>
- MDN Web Docs – CSS: <https://developer.mozilla.org/docs/Web/CSS>
- Walidator HTML: <https://validator.w3.org/>