

Szkielet semantyczny strony

Czas: 2 × 45 min

praca indywidualna

Cel: Budowa strony ze znacznikami semantycznymi HTML5 i poprawną hierarchią nagłówków oraz porównanie jej z wersją zbudowaną z samych elementów div.

Po zajęciach potrafisz:

- dobiera znaczniki header, nav, main, section, article, aside i footer do roli fragmentu strony
- buduje poprawną hierarchię nagłówków h1–h6 bez przeskakiwania poziomów
- oznacza daty znacznikiem time z atrybutem datetime
- porównuje drzewo dostępności strony semantycznej i strony na div w narzędziach F12
- sprawdza stronę walidatorem W3C i poprawia błędy

Sprzęt i materiały

- komputer z Windows 11
- plik `r1_02_orion_div.html` z folderu Pliki_do_zadan
- folder `nazwisko_imie` z zadania 1

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

- Przejrzyj z wykładu 1.02 tabelę znaczników semantycznych i zasady hierarchii nagłówków.

Bezpieczeństwo

- Na stronie nie umieszczaj danych osobowych (adresu, telefonu, zdjęć innych osób) — wystarczy imię i klasa.

Przebieg zadania

1 Analiza strony na div

Strona koła astronomicznego „Orion” wygląda w przeglądarce poprawnie, ale jej kod nic nie mówi o znaczeniu poszczególnych części.

1. W folderze `nazwisko_imie` utwórz `zad02` i skopiuj do niego `r1_02_orion_div.html`. Otwórz plik przez Live Server.
2. Na wydruku lub w notatkach przypisz każdemu elementowi `div` znacznik semantyczny, który powinien go zastąpić (np. `div class="naglowek" → header`). Wyniki wpisz do tabeli w sprawozdaniu.
3. Sprawdź plik walidatorem W3C. Zapisz błędy dotyczące nagłówków i wypisz obecną kolejność poziomów nagłówków (h1, h1, h4, ...).
4. Otwórz F12 → Elements → panel Accessibility (Ułatwienia dostępu) i włącz „Enable full-page accessibility tree” (w Firefoksie: zakładka Accessibility). Zrób zrzut drzewa — zwróć uwagę, czy pojawiają się role banner, navigation, main, complementary, contentinfo.

2 Wersja semantyczna

Utwórz nowy plik `index.html` i przenieś do niego treść, zmieniając strukturę. Treść ma zostać ta sama — zmieniasz tylko znaczniki.

```
body
├─ header    (h1 + podtytuł)
├─ nav       (lista ul z 4 odnośnikami)
├─ main
│   ├─ section#o-kole      (h2)
│   ├─ section#spotkania  (h2)
│   └─ section#obserwacje (h2)
│       └─ article × 4     (h3 + time + p)
├─ aside     (h2 Ciekawostka)
└─ footer#kontakt
```

1. Zbuduj szkielet według schematu. Odnośniki menu umieść w liście `ul` wewnątrz `nav` (bez znaków `|`).
2. Ustal hierarchię: jeden `h1` w nagłówku strony, tytuły sekcji jako `h2`, tytuły wpisów jako `h3`. Podtytuł strony to zwykły akapit, nie nagłówek.
3. Każdy wpis obserwacji zamień na `article`, a datę zapisz jako `<time datetime="2026-03-12">12 marca 2026</time>`.
4. Dopisz czwarty `article` o swoim hobby (tytuł h3, data dzisiejsza w `time`, co najmniej dwa zdania). W stopce dodaj akapit z imieniem, nazwiskiem i klasą (`© 2026 ...`).
5. Sprawdź, czy odnośniki z menu nadal przewijają stronę do sekcji (atrybuty `id`).

3 Walidacja i porównanie

1. Sprawdź `index.html` walidatorem W3C i popraw wszystkie błędy. Zaznacz opcję „outline” i zapisz zarys nagłówków (Heading-level outline).
2. Ponownie otwórz drzewo dostępności w F12, tym razem dla `index.html`. Zrób zrzut i wypisz role, które pojawiły się w porównaniu z wersją na div.
3. Porównaj obie wersje w przeglądarce bez stylów — czy wyglądają inaczej? Zapisz wniosek: co daje semantyka, skoro wygląd prawie się nie zmienia?
4. W VS Code użyj Edit → Replace in Files (Ctrl+Shift+H), aby policzyć wystąpienia `<div` w obu plikach (szukaj bez zamiany). Wpisz liczby do sprawozdania.

W sprawozdaniu umieść

- ☐ tabela: element div z pliku startowego → znacznik semantyczny i uzasadnienie
- ☐ błędy walidatora dla wersji div i wynik walidacji `index.html` (0 błędów)
- ☐ zarys nagłówków (outline) przed i po zmianach
- ☐ zrzuty drzewa dostępności obu wersji i lista nowych ról
- ☐ plik `index.html` w folderze `zad02`
- ☐ odpowiedzi na pytania kontrolne

Pytania kontrolne

1. Czym różni się `section` od `article`? Podaj przykład treści dla każdego z nich.
2. Dlaczego na stronie powinien być jeden element `main` i jeden nagłówek `h1`?
3. Komu i do czego służą znaczniki semantyczne (podaj co najmniej trzy przykłady: czytnik ekranu, wyszukiwarka, ...)?
4. Kiedy nadal warto użyć zwykłego `div`?

Kryteria oceny

Kryterium	Punkty
Tabela przypisania div → znacznik semantyczny z uzasadnieniem	3
Szkielet: header, nav z listą, main, 3 section, aside, footer zgodnie ze schematem	5
Hierarchia h1–h3 bez przeskoków, artykuły z <code>time datetime</code>	3
Własny artykuł o hobby i stopka z autorem; działające kotwice menu	2
Walidacja bez błędów, outline i porównanie drzew dostępności (zrzuty, wnioski)	4
Odpowiedzi na pytania kontrolne	3
Razem	20

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