

SPRAWOZDANIE Z LABORATORIUM

Technologie Webowe (HTML/CSS/JavaScript)

INFORMACJE PODSTAWOWE

Imię i nazwisko studenta: Michał Żurek

Numer albumu: 27649

Grupa: 3

Data wykonania laboratorium:

Numer laboratorium: 1

Temat laboratorium: Weryfikacja i utrwalenie wiedzy o HTML

1. CEL LABORATORIUM

[Opisz własnymi słowami, czego miałeś się nauczyć podczas tego laboratorium]

2. WYKONANE ZADANIA

Zadanie 1:

Status: Wykonane w pełni Wykonane częściowo Nie wykonane

Stworzyłem poprawny szablon dokumentu HTML5 z deklaracją DOCTYPE, elementem html z atrybutem języka oraz sekcją head zawierającą wymagane meta tagi i tytuł strony. Następnie dodałem przykładową zawartość w sekcji body i sprawdziłem poprawność struktury oraz domknięcie znaczników.

Część A:

```
<!DOCTYPE html>
<html lang="pl">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta name="description" content="Przykładowy dokument HTML5 do zadania 1">
<meta name="author" content="Michał Żurek">
<title>Zadanie 1 — Anatomia dokumentu HTML</title>
</head>
<body>
<h1>Przykładowa strona HTML5</h1>
<p>To jest przykładowa zawartość sekcji body.</p>
```

```
<p>Dokument zawiera poprawną strukturę HTML5 oraz wymagane meta tagi.</p>
</body>
</html>
```

Część B:

Dlaczego DOCTYPE jest ważny?

DOCTYPE informuje przeglądarkę, że dokument jest napisany w standardzie HTML5. Dzięki temu strona renderuje się w trybie standardowym, a nie w „quirks mode”, co zmniejsza ryzyko błędów i różnic w wyświetlaniu.

Jaką rolę pełni sekcja <head>?

Sekcja <head> przechowuje informacje o dokumencie niewidoczne bezpośrednio na stronie, np. tytuł, meta dane oraz odwołania do plików CSS i JavaScript. Przeglądarka i wyszukiwarki używają tych danych do poprawnej interpretacji strony.

Co to są meta tagi i do czego służą?

Meta tagi opisują dokument, np. kodowanie znaków, ustawienia widoku na urządzeniach mobilnych, opis strony czy autora. Pomagają w poprawnym wyświetlaniu treści, SEO oraz w przekazywaniu informacji o stronie.



Przykładowa strona HTML5

To jest przykładowa zawartość sekcji body.

Dokument zawiera poprawną strukturę HTML5 oraz wymagane meta tagi.

Zadanie 2:

Status: Wykonane w pełni Wykonane częściowo Nie wykonane

Utworzyłem stronę HTML z artykułem zawierającym poprawną hierarchię nagłówków (h1–h3) oraz co najmniej trzy paragrafy tekstu. Zastosowałem wymagane znaczniki formatowania, takie jak pogrubienie, kursywa, wyróżnienie, indeks dolny i górny, a także elementy strukturalne: linię poziomą, blok cytatu i tekst preformatowany. Dodałem również dwa opisowe linki zewnętrzne, dbając o semantyczną poprawność znaczników.

Najważniejsze fragmenty kodu:

```
<!-- Hierarchia nagłówków -->
<h1>Dlaczego warto uczyć się HTML na serio?</h1>
<h2>Podstawy struktury treści</h2>
<h3>Semantyka ma znaczenie</h3>

<!-- Formatowanie tekstu -->
<p><strong>Ważna informacja</strong> i <em>podkreślenie znaczenia</em>.</p>
<p>Wzór: H<sub>2</sub>O oraz 2<sup>10</sup>. </p>
<p><mark>Najważniejszy fragment</mark></p>

<!-- Elementy strukturalne -->
<p>Przykładowy paragraf treści.</p>
<hr>
<blockquote>„HTML to język opisu treści.”</blockquote>
<pre>
- DOCTYPE
- html
- head
- body
</pre>

<!-- Linki -->
<a href="https://developer.mozilla.org/en-US/docs/Web/HTML" target="_blank">
    Dokumentacja HTML MDN
</a>
<a href="https://validator.w3.org/" target="_blank">
    Validator W3C
</a>
```

Zrzut kranu mojej strony zadanie 2:



Dlaczego warto uczyć się HTML na serio?

HTML jest jak szkielet strony internetowej: bez niego nie ma struktury, a bez struktury nie ma porządku. Wiele osób zaczyna od kopowania gotowych fragmentów, ale szybko okazuje się, że bez zrozumienia podstaw trudno coś poprawić lub rozbudować. Gdy wiesz, czym jest nagłówek, paragraf, lista czy tabela, możesz tworzyć treści w sposób logiczny i przyjazny dla użytkownika. Co ważne, poprawny HTML pomaga też w dostępnosci i SEO. W praktyce oznacza to, że strona jest łatwiejsza do czytania zarówno dla ludzi, jak i dla robotów wyszukiwarek.

Podczas nauki warto rozróżnić znaczenie semantyczne od wyglądu. Na przykład **ważna informacja** to nie to samo co **tekst tylko pogrubiony**. Podobnie **podkreślenie znaczenia** różni się od **kursywy dla stylu**. Dobrze jest też umieć oznaczać rzeczy typu **najważniejsze fragmenty**, a w tekstu technicznych przydaje się zapis wzorów: H₂O albo 2¹⁰. Dzięki takim detaliom treść staje się czytelniejsza i bardziej profesjonalna.

Sam HTML nie jest trudny, ale wymaga konsekwencji. Największe błędy początkujących to pomijanie hierarchii nagłówków, brak opisowych linków i nadużywanie znaczników, które nie niosą znaczenia. Jeśli od początku będziesz trzymać się zasad, później łatwiej wejdzieś w CSS i JavaScript. Wtedy zauważysz, że HTML to nie „kilka tagów”, tylko język opisu dokumentu. Dobrze napisany dokument jest jak dobrze napisany tekst: ma tytuł, sekcje, cytat, przykłady i odnośniki do źródeł.

Podstawy struktury treści

Zaczynaj od jednego nagłówka **h1**, a potem dziel temat na sekcje za pomocą **h2**. Jeśli w sekcji potrzebujesz dodatkowego podziału, użyj **h3**. Taka struktura jest czytelna dla człowieka i dla technologii wspomagających, np. czytników ekranu. W praktyce oznacza to, że użytkownik może szybko „przeskakiwać” po nagłówkach i znaleźć interesujący go fragment.

Semantyka ma znaczenie

„HTML to język opisu treści. Najpierw sens, potem wygląd.”

Gdy trzemasz się semantyką, łatwiej też utrzymać stronę. Zamiast zastanawiać się, co oznacza kolejny przypadkowy znacznik, widzisz od razu: to nagłówek, to cytat, to akapit. To oszczędza czas i zmniejsza liczbę błędów. W dodatku wyszukiwarki lepiej rozumieją stronę, gdy struktura jest logiczna.

Przykład preformatowanego tekstu

Przykład prostego szablonu:
- DOCTYPE
- html lang="pl"
- head z meta
- body z treścią

Linki do źródeł

Jeśli chcesz poczytać więcej, sprawdź: [Dokumentację HTML na MDN](#) oraz [Validator HTML W3C](#).

Zadanie 3:

Status: Wykonane w pełni Wykonane częściowo Nie wykonane

W zadaniu utworzyłem różne typy list HTML: nieuporządkowaną z zaletami HTML, uporządkowaną z krokami tworzenia strony, listę definicji z podstawowymi terminami oraz listę zagnieżdzoną z podlistami technologii. Następnie przygotowałem tabelę porównującą tagi HTML z podziałem na thead, tbody i tfoot, dodałem atrybuty id i class oraz zastosowałem rowspan i colspan. Całość została zapisana z poprawną składnią i strukturą znaczników.



Listy i tabele w HTML

Zalety języka HTML

- Prosta składnia
- Szerokie wsparcie przeglądarek
- Semantyczna struktura treści
- Łatwa integracja z CSS i JS
- Dobra dostępność

Kroki tworzenia strony HTML

1. Utworzenie pliku .html
2. Dodanie deklaracji DOCTYPE
3. Zbudowanie sekcji head
4. Dodanie treści w body
5. Test w przeglądarce

Terminy HTML

HTML	Język znaczników do budowy stron WWW
Tag	Element składowy HTML w nawiasach ostrych
Atrybut	Dodatkowa właściwość znacznika
Element	Tag wraz z zawartością
Semantyka	Znaczeniowy opis struktury treści

Technologie webowe

- Frontend
 - HTML
 - CSS
 - JavaScript
- Backend
 - Node.js
 - PHP
 - Python
- Bazy danych
 - MySQL
 - PostgreSQL
 - MongoDB

Tabela tagów HTML

Nazwa tagu	Przeznaczenie	Przykład użycia	Czy wymaga zamknięcia?
<p>	Paragraf tekstu	<p>Tekst</p>	Tak
<a>	Link	Link	Tak
	Obraz		Nie
 	Złamanie linii	 	Nie
	Ważny tekst	Ważne	Tak

Najważniejsze fragmenty kodu:

<!-- Lista nieuporządkowana -->

```
<ul>
  <li>Prosta składnia</li>
  <li>Szerokie wsparcie</li>
  <li>Semantyka</li>
  <li>Dostępność</li>
  <li>Integracja z CSS/JS</li>
</ul>
```

<!-- Lista uporządkowana -->

```
<ol>
  <li>Utwórz plik HTML</li>
  <li>Dodaj DOCTYPE</li>
  <li>Zbuduj head</li>
```

```
<li>Dodaj body</li>
<li>Test w przeglądarce</li>
</ol>

<!-- Lista definicji -->
<dl>
  <dt>HTML</dt>
  <dd>Język znaczników</dd>
  <dt>Tag</dt>
  <dd>Znacznik w nawiasach</dd>
</dl>

<!-- Lista zagnieżdzona -->
<ul>
  <li>Frontend
    <ul>
      <li>HTML</li>
      <li>CSS</li>
    </ul>
  </li>
</ul>

<!-- Tabela z thead/tbody + rowspan/colspan + id/class -->





```

```
<td rowspan="2">Nie</td>
</tr>
<tr>
  <td>&lt;br&gt;</td>
  <td>Nowa linia</td>
  <td>&lt;br&gt;</td>
```

```
</tr>
</tbody>

<tfoot>
<tr>
    <td colspan="4">Tabela tagów HTML</td>
```

```
</tr>
</tfoot>
</table>
```

Zadanie 4:

Status: Wykonane w pełni Wykonane częściowo Nie wykonane

W zadaniu stworzyłem stronę HTML z galerią multimedialną, w której osadziłem różne typy mediów. Dodałem obrazy za pomocą znaczników `` oraz `<figure>` z podpisami `<figcaption>`, stosując wymagane atrybuty alt, width, height oraz lazy loading. Następnie umieściłem wideo i audio z kontrolkami oraz wieloma źródłami w różnych formatach, a także osadziłem zewnętrzną treść przez iframe z atrybutami bezpieczeństwa. Całość została opisana komentarzami HTML wyjaśniającymi strukturę.

Moja galeria multimedialna

Obrazy (img)



Obrazy (figure + figcaption)



Obraz 1 przedstawiony w elemencie figure (z podpisem).



Obraz 2 w figure — drugi przykład z figcaption.

Wideo



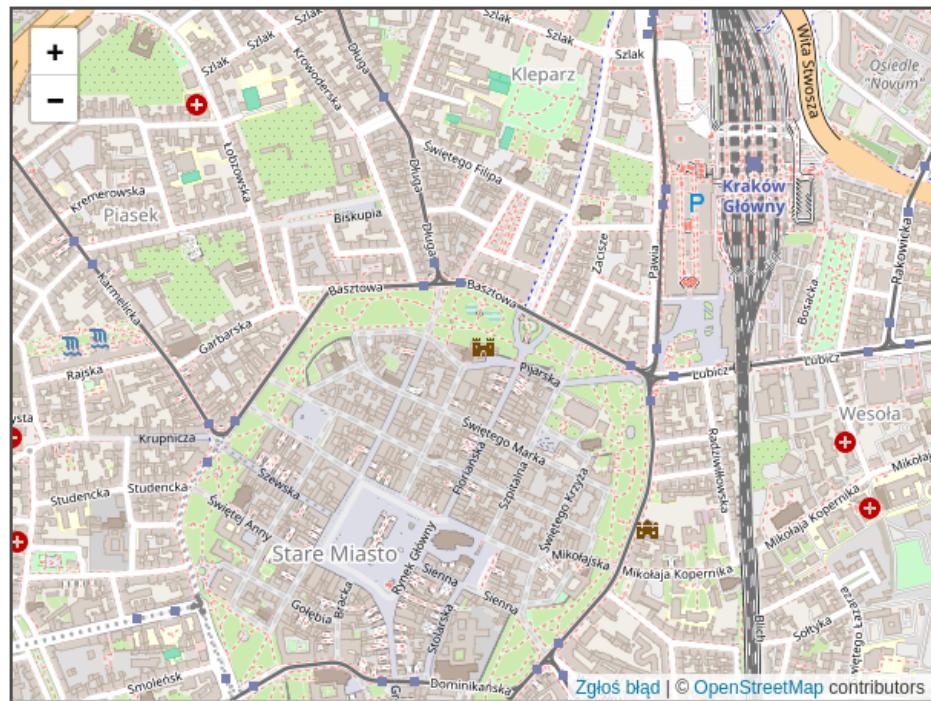
Wideo



Audio



Osadzona treść — mapa



Najważniejsze fragmenty kodu:

```
<!-- Obrazy img z wymaganymi atrybutami + lazy -->


![Widok miasta](https://www.openstreetmap.org/export/embed.html)
```

Zadanie 5:

Status: Wykonane w pełni Wykonane częściowo Nie wykonane

W zadaniu stworzyłem kompleksowy formularz kontaktowy w HTML5 z wykorzystaniem różnych typów pól wejściowych i wbudowanej validacji. Formularz podzieliłem na logiczne sekcje za pomocą znaczników **fieldset** i **legend**, a każde pole połączyłem z etykietą przez atrybuty **for**/**id** dla lepszej dostępności. Zastosowałem pola tekstowe, email, telefon, datę, hasło, radio, checkbox, select, range, file, textarea i url wraz z atrybutami validacji takimi jak **required**, **pattern**, **minlength** i **maxlength**. Dodałem także przyciski wysyłania i resetowania oraz atrybuty **data-*** do oznaczenia akcji formularza. Kod jest za długi żeby go tu wklejać to zostawiam na githubie.

Formularz kontaktowy

Podstawowe dane

Imię i nazwisko:

Email:
(np. jan.kowalski@example.com)

Telefon (opcjonalnie):
+48 123 456 789

Data urodzenia:
dd . mm . yyyy

Hasło:

Potwierdź hasło:

Wybory

Płeć:

Kobieta
 Mężczyzna
 Inna / nie chcę podawać

Zainteresowania (min. 5 opcji):

Frontend
 Backend
 UX/UI
 Aplikacje mobilne
 AI

Kraj:

Poziom doświadczenia (0-10):

Dodatkowe informacje

Prześlij CV (PDF/DOC):

Wiadomość (max 500 znaków):

Adres strony WWW (opcjonalnie):
<http://example.com>

Akceptuję regulamin (wwwmagazyn)

Zadanie 6:

Status: Wykonane w pełni Wykonane częściowo Nie wykonane

W zadaniu przeprowadziłem refaktoryzację kodu HTML do wersji semantycznej. Zamiast ogólnych kontenerów użyłem znaczników semantycznych HTML5, takich jak header, nav, main, article, section, aside oraz footer, aby jednoznacznie określić rolę poszczególnych części strony. Główną treść umieściłem w elemencie article, a powiązane treści poboczne w aside. Nawigację wydzieliłem znacznikiem nav, a nagłówek strony w header.

Dodałem poprawną hierarchię nagłówków h1–h3 oraz podzieliłem treść na logiczne sekcje przy użyciu section i paragrafów. Datę publikacji zapisałem za pomocą znacznika time z atrybutem datetime, co umożliwia maszynową interpretację daty. Dane kontaktowe umieściłem w znaczniku address. Dzięki temu struktura dokumentu stała się bardziej czytelna i znaczeniowa.

Zastosowanie semantycznego HTML poprawia dostępność strony dla czytników ekranu oraz ułatwia analizę treści przez wyszukiwarki (SEO). Kod jest też łatwiejszy w utrzymaniu i rozwoju. Znacznik div pozostaje uzasadniony w sytuacjach czysto technicznych lub stylizacyjnych, gdy element nie niesie własnego znaczenia semantycznego.

Blog technologiczny

Aktualności i porady o HTML, CSS i JavaScript

- [Artykuł](#)
- [Polecane](#)
- [Kontakt](#)

Semantyczny HTML — dlaczego ma znaczenie?

Data publikacji: 12.01.2026

Lepsza struktura dokumentu

Semantyczne znaczniki opisują rolę treści (nagłówek, nawigacja, artykuł), dzięki czemu kod jest czytelniejszy i łatwiejszy do utrzymania.

SEO i dostępność

Wyszukiwarki i czytniki ekranu lepiej rozumieją stronę, gdy używa ona elementów takich jak article, nav czy header.

Cytat i przykład

„Najpierw sens, potem wygląd.”

Przykład struktury:
- header
- nav
- main
- article/section
- footer

Powiązane materiały

- [MDN — HTML](#)
- [W3C Validator](#)

Kontakt

Michał Żurek
email: michal@example.com

Najważniejsze fragmenty kodu:

Semantyczny nagłówek strony

```
<header>
  <h1>Blog technologiczny</h1>
  <p>Aktualności i porady IT</p>
</header>
```

Nawigacja

```
<nav>
  <ul>
    <li><a href="#artykul">Artykuł</a></li>
    <li><a href="#kontakt">Kontakt</a></li>
  </ul>
</nav>
```

Główna część dokumentu

```
<main>
  <article>
    <header>
      <h2>Tytuł artykułu</h2>
      Data: <time datetime="2026-01-30">30.01.2026</time>
    </header>
```

Podział na sekcje tematyczne

```
<section>
  <h3>Sekcja treści</h3>
  <p>Treść akapitu...</p>
</section>
```

Treść poboczna

```
<aside>
  <h3>Powiązane tematy</h3>
  <ul><li>HTML5</li></ul>
</aside>
```

Dane kontaktowe

```
<footer>
  <address>
    Michał Żurek<br>
    email: michal@example.com
  </address>
</footer>
```

Zadanie 7:

Status: Wykonane w pełni Wykonane częściowo Nie wykonane

3. WNIOSKI I REFLEKSJE

Co nowego się nauczyłem?

Podczas wykonywania zadań utrwalilem pełną strukturę dokumentu HTML5 oraz poprawne użycie podstawowych i semantycznych znaczników. Nauczyłem się stosować właściwą hierarchię nagłówków, różne typy list i tabel, a także znaczniki formatowania tekstu zgodnie z ich znaczeniem, a nie tylko wyglądem. Przećwiczyłem osadzanie multimediów (obrazy, audio, wideo, iframe) z odpowiednimi atrybutami oraz tworzenie rozbudowanych formularzy z walidacją HTML5. Zrozumiałem też, jak ważny jest semantyczny HTML dla dostępności i SEO.

Co było najtrudniejsze?

Najbardziej wymagające było poprawne zbudowanie rozbudowanego formularza z wieloma typami pól i atrybutami walidacji oraz dopilnowanie wszystkich wymagań technicznych jednocześnie. Trudniejsze było także poprawne użycie znaczników semantycznych przy refaktoryzacji kodu oraz zapamiętanie różnic między znacznikami o podobnym wyglądzie, ale innym znaczeniu (np. strong vs b, em vs i, article vs section).

Co chciałbym jeszcze przećwiczyć?

Chciałbym jeszcze przećwiczyć semantyczne projektowanie całych stron, bardziej złożone formularze oraz dostępność (accessibility) w praktyce. Dodatkowo chcę utrwalić pracę z multimediami i atrybutami oraz połączyć HTML z CSS, aby lepiej rozumieć zależność między strukturą a prezentacją strony.

4. SAMOOCENA

Ocena własnego zaangażowania: Bardzo wysokie Wysokie Średnie Niskie

Procent wykonanych zadań: 86 %