

Tworzenie Stron Internetowych

odcinek 1

Tworzenie Stron Internetowych – organizacja zajęć

Termin: poniedziałki 14:30 (90 min., 15 tyg.)

Konsultacje:

- poniedziałki:...
- dowolny termin (email, czat na Teams),
- termin po umówieniu (wideokonsultacja na Teams)

Plan: HTML, CSS, JS (podstawy)

Materiały do zajęć:

- http://www.astro.uni.wroc.pl/ludzie/kolomanski/pracownia_tsi/pracownia_tsi.html

Warunki zaliczenia:

- praca zaliczeniowa: serwis internetowy (*)
- obecność: 2 nieusprawiedliwione nieobecności na cały semestr



technologie webowe (plan na te zajęcia)

- **(X)HTML** – (*eXtensible*) *HyperText Markup Language*

(Rozszerzalny) Hipertekstowy Język Znaczników. Obecny standard w tworzeniu stron www. Opisuje strukturę dokumentu.

- **CSS** (*Cascading Style Sheets*)

Kaskadowe Arkusze Stylów. Reguły opisujące wygląd dokumentu opisanego za pomocą jakiegoś języka znaczników, np. HTML.

- **JS** (*JavaScript*)

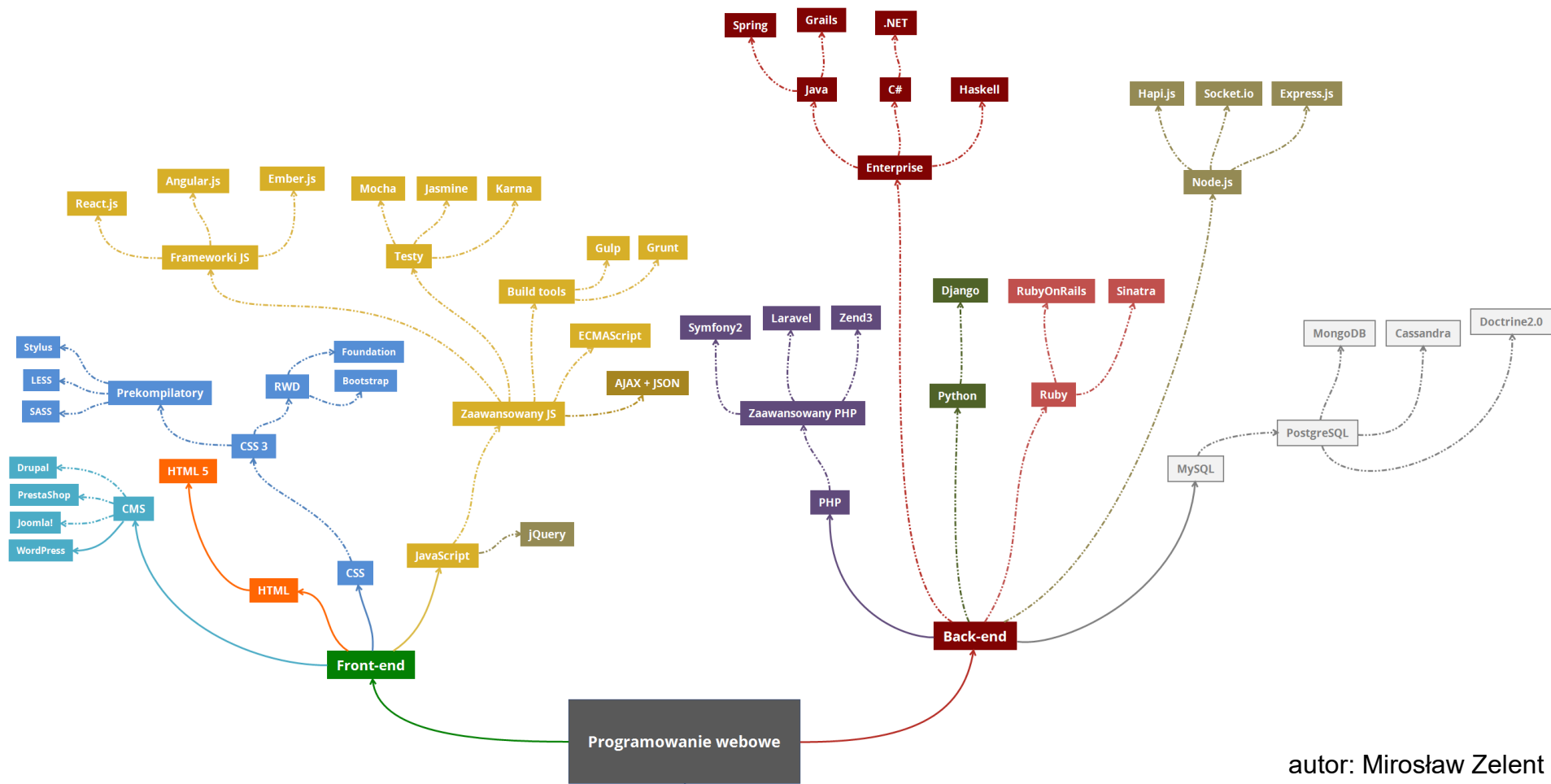
Skryptowy język programowania powszechnie używany w Internecie. Skrypty JS dodają do stron www interaktywność i funkcjonalności.



Podstawowe technologie wykorzystywane dla tworzenia stron WWW

World Wide Web

technologie webowe



autor: Mirosław Zelent

drzewo technologii webowych

ścieżka ewolucyjna html

- **odnośniki** pojawiają się w roku 1980 (dzięki fizykom z CERN) – możliwość przeglądania dokumentów z różnych miejsc
- w 1990 powstaje **WorldWideWeb** – system informacyjny oparty na hipertekście (usługa internetowa)
- pierwsza specyfikacja języka **HTML Tags** pojawia się w 1991 roku
- w 1995 pojawia się **HTML 2.0**
- ostatnia specyfikacja zawierająca elementy prezentacyjne, **HTML 4.01** pojawia się 1999 roku
- w 2000 roku HTML na bazie wersji 4.01 strict staje się **standardem międzynarodowym ISO**
- **xHTML 1.0** pojawia się w 2000 roku; dostępne są trzy wersje: **strict**, **transitional**, **frameset** („x” pochodzi od XML - języka do reprezentowania danych w strukturalizowany sposób);
- w 2001 roku udostępniona zostaje wersja **xHTML1.1** – obowiązujący, ale nie powszechny standard
- do 2009 rozwijany jest **xHTML2.0** – specyfikacja niekompatybilna wstecz; projekt zostaje zarzucony
- W październiku 2014 specyfikacja **(x)HTML5** uzyskała status rekomendacji W3C
- w listopadzie 2016 specyfikacja **(x)HTML5.1** uzyskała status rekomendacji W3C
- w grudniu 2017 specyfikacja **(x)HTML5.2** uzyskała status rekomendacji W3C
- obecnie obowiązującą wersją jest **HTML Living Standard** aktualizowany w sposób ciągły

*Sir Timothy John
"Tim" Berners-Lee*



Za tworzenie standardów pisania i przesyłu stron WWW odpowiadało *World Wide Web Consortium* (W3C, www.w3.org). Od 2019 r. tym zadaniem zajmuje się konsorcjum *Web Hypertext Application Technology Working Group* (WHATWG).

edytory html

Dokument HTML to zwykły plik tekstowy, ale ...

Podział edytorów HTML:

- edytory plików źródłowych
- edytory graficzne, tzw. WYSIWYG (ang. What You See Is What You Get, czyli To Co Widzisz Jest Tym Co Otrzymasz)
- CMS (Content Management System)

Cechy dobrego edytora (kodu):

- podświetlanie składni (unikanie błędów w poleceniach), automatyczne zamykanie znaczników, generatory znaczników
- zaawansowana edycja (wielopoziomowe cofanie, znajdź-zastąp,...)
- obsługa kodowania ISO-8859-2 lub Unicode (poprawnie wyświetlanie polskich znaków)

Przykładowe edytory HTML:

*Pajęczek, Adobe Brackets, CoffeeCup, HTML-Kit Tools, Sublime Text, **RJ TextEd**, Notepad++, Atom, Geany, Text Editor Pro, ...*

tworzenie strony internetowej

Edytor kodu źródłowego?, edytor graficzny?, CMS?

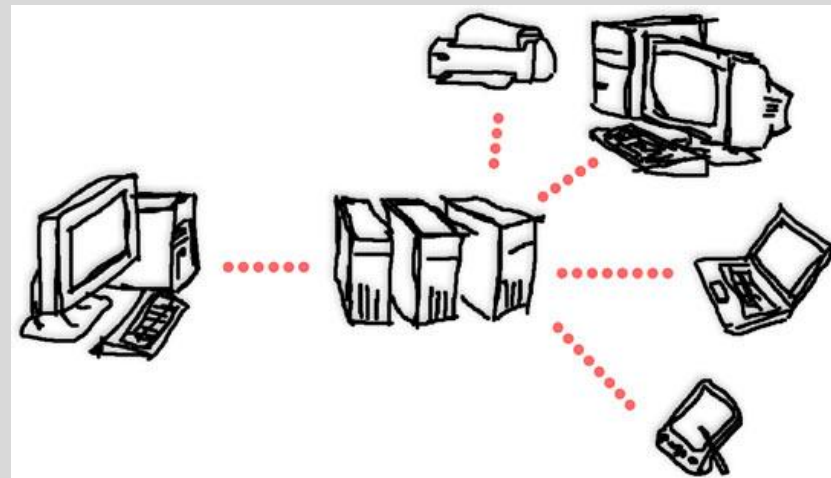
Zależy od potrzeb.

- potrzebujesz tylko prostego serwisu internetowego do realizacji jakiegoś celu – edytor graficzny
- chcesz zostać web-masterem – kod źródłowy
- serwis WWW nie będzie często wzbogacany o nowe strony – kod źródłowy/edytor graficzny
- chcesz pisać blog – CMS
- chcesz administrować swój serwis www z różnych komputerów – CMS
- chcesz mieć pełną kontrolę na wyglądem serwisu – kod źródłowy
- twój serwis ma wielu autorów – CMS
- serwis ma zawierać forum – nie konieczne CMS

tworzenie strony internetowej

O czym należy pamiętać:

- strona WWW ma działać i wyglądać poprawnie **u innych użytkowników** sieci (różne przeglądarki, sprzęt)
- odpowiedni dobór kolorów tła i tekstu (**czytelność**)
- odpowiednia (nie za duża) ilość **grafiki o dobrej jakości** i nie zbyt dużym rozmiarze (**szybkość ładowania, mniejszy transfer danych**)
- strona powinna mieć **ciekawy temat** i czytelny, estetyczny **wygląd** (warto korzystać z przykładów dobrych witryn)
- wygodny, praktyczny, logiczny i łatwo dostępny sposób **nawigacji**
- **poprawność** językową tekstu
- **aktualizacja** treści strony i usuwanie błędów; **udoskonalanie** strony



tworzenie strony internetowej

Jeden z pierwszych kroków przy tworzeniu strony internetowej to zaprojektowanie makiety (szkicu / schematu / wireframe / mock-up / blue print / page schematics).

Makieta to instrukcja budowy strony. Zawiera rozmieszczenie najważniejszych elementów strony: przyciski CTA (call to action) / nawigacja, zdjęcia, akapity z tekstem, itd. oraz wyjaśnia ich funkcje.

Każdą akcję, którą użytkownik może przeprowadzić na danej stronie musi być zawarta w makiecie.



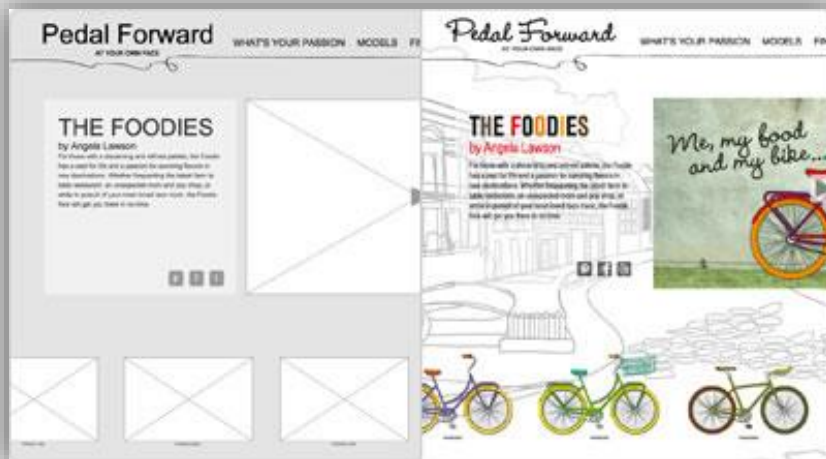
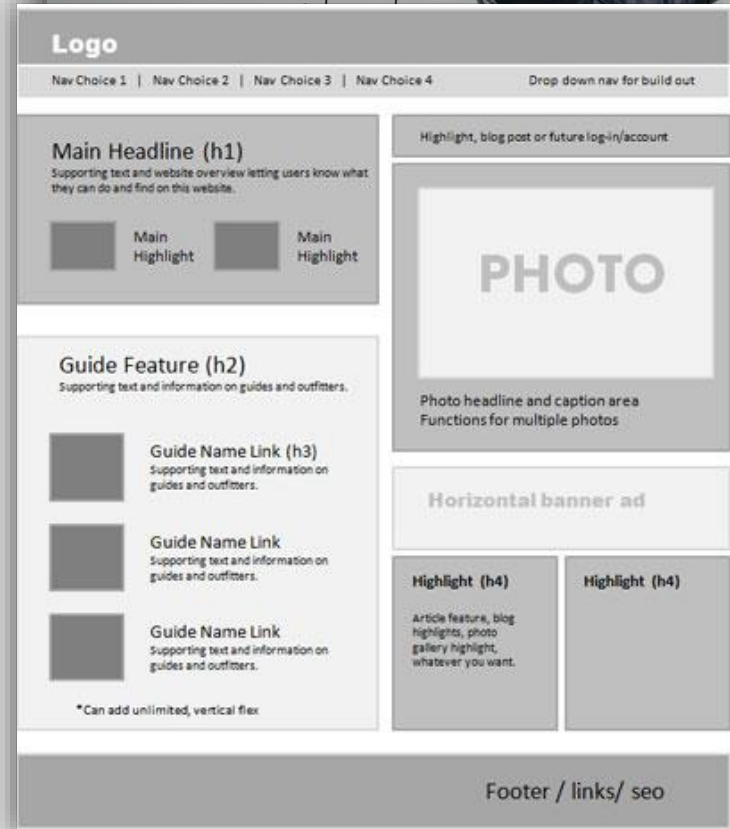
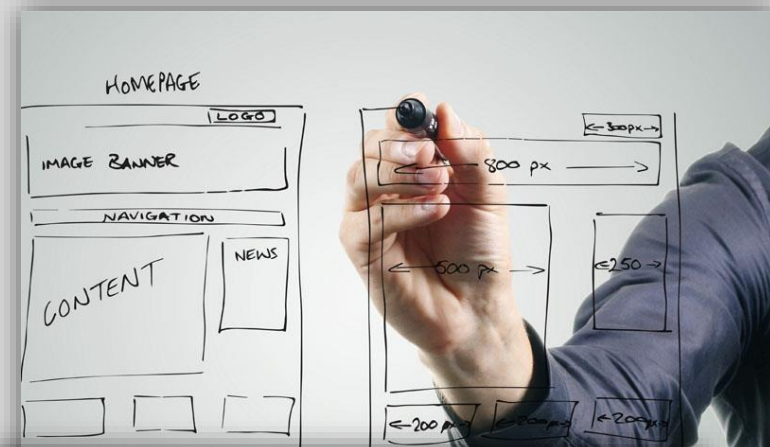
tworzenie strony internetowej

Typy makiet:

- **Makiety lo-fi** (oraz Low-fidelity mockups) – zawiera podstawowe funkcje. Może być w formie schematu blokowego lub makiety cieniowanej (gray-box)
- **Makiety hi-fi** (High-fidelity mockups) – makiety zbliżone do końcowego designu. Pojawiają się tu kolory, zdjęcia, filmy, działające menu, typografia, itd.

Makiety można tworzyć różnymi narzędziami, od ołówka i papieru po dedykowane oprogramowanie.

Przed makietowaniem należy określić zawartość / funkcjonalności strony, możliwość wykonania technologicznego i wygląd. Makietowanie wykonuje zwykle projektant – specjalista do spraw użyteczności (user experience / usability).



tworzenie strony internetowej

Dodatkowe źródła:

- Kilka porad i uwag odnośnie tworzenia makiet:
<https://grafmag.pl/artykuly/tworzenie-makiet-stron-internetowych-praktyczne-porady>
- Materiał wideo pokazujący przygotowanie makiety w programie Adobe XD:
<https://www.youtube.com/watch?v=ScQkQyisVQw>
- Siatki do ręcznego rysowania makiet:
<https://sneakpeekit.com/>
- Lista wybranych programów do tworzenia makiet:
<http://studioleopard.pl/makieta-strony-internetowej/>
- „Strona www: wersja desktopowa i mobilna (najważniejsze różnice)”
<https://advist.pl/strona-www-wersja-desktopowa-i-mobilna-najwazniejsze-roznice/>



co jest teraz modne w projektowaniu?

Ważne cechy nowoczesnej strony internetowej:

- **nowoczesne technologie** – tworzenie stron w najnowszych standardach (HTML5, CSS3) i poprawnym kodem
- **responsywność (Responsive Web Design)** – automatyczne dostosowanie układu strony do urządzenia wyświetlającego
- **długa strona (longpage)** – ograniczenie/zastąpienie podstron na rzecz jednej długiej strony (przewijanie zamiast klikania); długa strona główna przedstawia zawartość przeglądową, szczegóły mogą być prezentowane na podstronach.
- **pływająca nawigacja** – menu nawigacyjne przyklejone do krawędzi ekranu, stale dostępne przy przewijaniu strony



co jest teraz modne w projektowaniu?

Ważne cechy nowoczesnej strony internetowej, cd:

- **duże zdjęcia (hero image)** – bardzo dobrej jakości duże zdjęcia dla przykucia uwagi użytkownika i minimalistyczny wygląd elementów towarzyszących (menu), szczegółowa zawartość przeniesiona do dalszej części strony
- **typografia** – krój czcionki jako element graficzny oraz duże wyróżniające się nagłówki
- **minimalizm, płaski deseń (flat design)** – tylko kluczowe elementy w układzie strony, proste kształty i mała liczba kolorów; istotna jest czytelność a nie barokowe ozdoby, cieniowania, pseudo 3D, itd.
- **przyciski widma** – uproszczone przyciski nawigacji/odnośników: tekst, ramka, przezroczyste tło
- **ikony** – zastąpienie niektórych tekstów i zdjęć ikonami dla zwiększenia czytelności i ułatwienia nawigacji
- **ruch** – animacje elementów strony (z umiarem)

Przykładowe źródło czcionek
fonts.google.com

hero
image



składnia

Znaczniki i elementy

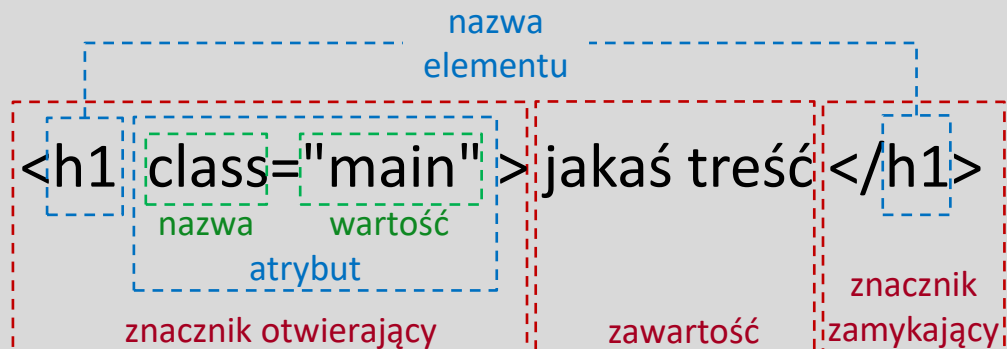
Pliki HTML są plikami tekstowymi. Podstawowymi „cegiełkami”, z których składa się HTML są **elementy**.

Podstawową częścią każdego elementu jest **znacznik**.

Znaczniki są poleceniami dla przeglądarki, które wskazują jak ma być wyświetlona **zawartość**.

Zawartość jest tym, co tworzy daną stronę internetową.

Przykład elementu (nagłówek):



Znacznik otwierający zawiera nazwę elementu oraz opcjonalne **atrybuty**

Znacznik zamykający zawiera tylko nazwę elementu poprzedzoną `/`

Znacznik otwierający i zamykający ograniczone są znakami `<` `>`

Każdy atrybut składa się z pary **nazwa**=**wartość**, wartość musi być ujęta w znaki `" "`

Atrybuty zawierają dodatkowe dane na temat elementu, np. związane z stylami CSS

Jeden element może posiadać wiele atrybutów, ale o różnych nazwach.

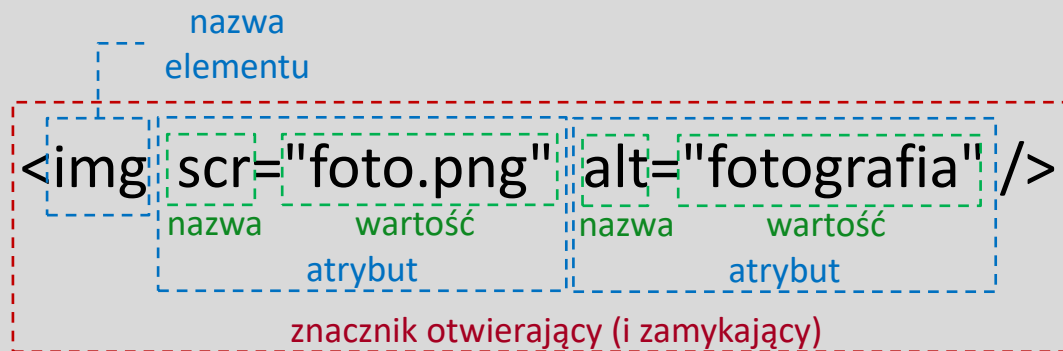
składnia

Znaczniki i elementy

Niektóre elementy nie mają zawartości (elementy puste).

Elementy puste mogą posiadać atrybuty.

Przykład elementu pustego (wstawienie grafiki):



Elementy mogą być zagnieżdżone. Przykład:

`<p>Lorem ipsum dolor sit amet, <strong>consectetur adipiscing</strong> elit, sed...</p>`

Przeplatanie znaczników jest **niezalecane/niedozwolone** – najpóźniej otwarty element jest zamykany najwcześniej.

Uwaga: istnieją pewne ograniczenia w zagnieżdżaniu (temat na później)

Białe znaki

Można wstawiać białe znaki (*spacje, znaki tabulacji, przejścia do nowej linii*) w następujących miejscach:

```
<znacznik  
  atrybut  
  =  
  "przykładowa  
  wartość"  
>  
<inny-znacznik  
>  
</znacznik>
```

oznacza to samo co:

```
<znacznik atrybut="przykładowa wartość"> <inny-znacznik/> </znacznik>
```


składnia

Encje

Encje pozwalają zapisać alternatywnie znaki takie jak np.: "<" i "&" co umożliwia ich interpretację jako zwykłych znaków, a nie jako istotnych elementów składni języka.

Przykłady:

Znak	Encja	Nazwa
<	<	Less than (znak mniejszości)
>	>	Greater than (znak większości)
"	"	Quote (cudzysłów)
'	'	Apostrophe (apostrof)
&	&	Ampersand (znak &)

Komentarze

Komentarz (tekst ignorowany przez przeglądarkę) zaczyna się od **<!--** , a kończy na **-->**

Kod przyjazny do pracy

Komentarze i białe znaki pozwalają zapisać kod HTML w bardziej przejrzysty sposób, ułatwiają jego czytanie i edycję.

struktura dokumentu HTML

Stałe elementy w strukturze dokumentu HTML to:

deklaracja, nagłówek i ciało dokumentu

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<title> tytuł strony </title>
```

```
</head>
```

```
<body>
```

```
<!-- tutaj wstaw tresc strony -->
```

```
</body>
```

```
</html>
```

struktura dokumentu HTML

Deklaracja typu dokumentu **!DOCTYPE**

Informuje przeglądarkę, z jakiego typu plikiem ma do czynienia. !DOCTYPE nie jest znacznikiem HTML.

Nagłówek: **<head>**

Nagłówek zawiera informacje o dokumencie i plikach używanych do jego wyświetlenia. W przykładzie użyty został tylko znacznik **<title>** definiujący tytuł dokumentu i kodowanie znaków:

```
<meta charset="UTF-8">  
<title> tytuł strony </title>
```

Ciało dokumentu: **<body>**

W **<body>** zawarta jest cała widoczna część dokumentu. Każdy dokument ma tylko jedno **<body>**.

struktura dokumentu HTML

```
<html>
```

```
<head>
```

```
<title>Page title</title>
```

```
</head>
```

```
<body>
```

```
<h1>This is a heading</h1>
```

```
<p>This is a paragraph.</p>
```

```
<p>This is another paragraph.</p>
```

```
</body>
```

```
</html>
```

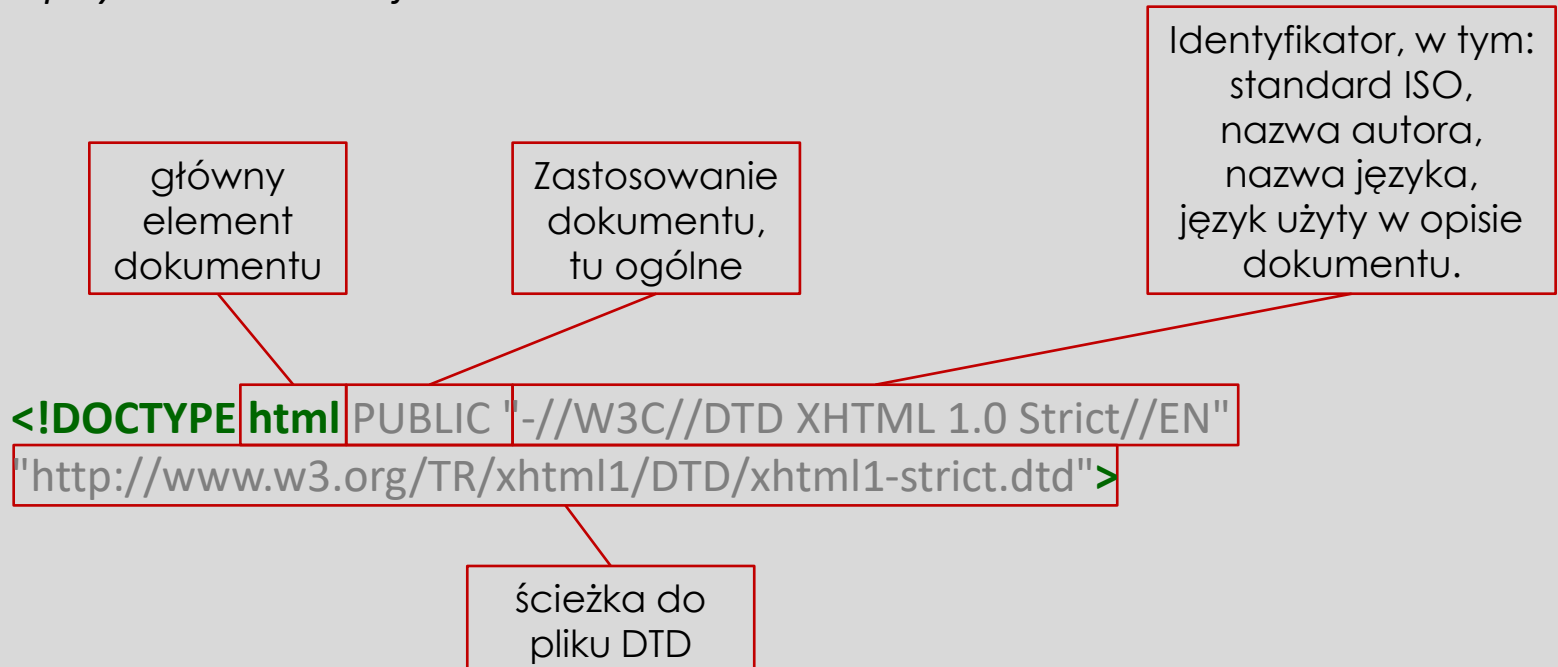
Przykładowa struktura prostego dokumentu HTML.
Tylko jasny fragment jest wyświetlany przez przeglądarkę.

zawartość DOCTYPE

Deklaracja **<!DOCTYPE>**

- Deklaracja ta musi znaleźć się na samym początku dokumentu HTML, przed znacznikiem <html>.
- Nie jest to znacznik języka HTML, ale informacją dla przeglądarki o tym w jakim języku napisany jest dokument
- **xHTML** (i HTML4) wymaga podania DTD – Document Type Definition, który określa zasady języka HTML. Dokument ten jest niezbędny dla poprawnej interpretacji kodu źródłowego przez przeglądarkę.
- **HTML5** nie wymaga podania DTD

szczegóły na przykładzie deklaracji dla *xHTML1.0 Strict* i **HTML5**:



zaczynamy pracę z kodem

Jak pracować z kodem html?

- utwórz/zmień kod w edytorze
- zapisz zmiany
- otwórz plik html w przeglądarce lub przeładuj stronę

Uwagi i przypomnienia:

- unikaj zbyt długich linii kodu (>80 znaków)
- używaj wcięć i pustych linii dla większej przejrzystości kodu
- dodawaj komentarze
- używaj małych liter dla nazw plików
- pliki HTML muszą mieć rozszerzenie .html lub .htm
- główny plik serwisu internetowego powinien mieć nazwę index.html, index.htm, default.html lub default.htm. Szczegóły zależą od ustawienia serwera.

treść – struktura – prezentacja – zachowanie

Końcowy wygląd strony internetowej to efekt współdziałania kilku technologii, pełniących różne zadania. Stronę tę tworzą:

- *treść*, czyli to co chcemy opublikować w sieci
- *struktura*, czyli podział tekstu na części o różnym znaczeniu (tzw. *semantyka dokumentu*); za strukturę odpowiada **HTML**
nadanie struktury stronie internetowej za pomocą znaczników określających znaczenie poszczególnych elementów treści
- *prezentacja/układ*, czyli definiowanie wyglądu dokumentu (jego elementów strukturalnych) za pomocą **CSS**
podanie reguł ogólne opisujących wygląd dokumentu i reguł szczegółowo opisujących wygląd poszczególnych jego elementów
- *zachowanie*, czyli kontrola zachowania zawartości i interakcja z użytkownikiem przy wykorzystaniu **Javascript**.

Dawniej

W starszych wersjach HTML opisywał zarówno strukturę, jak i prezentację.

poprawność HTML

Poziomy poprawności HTML:

1. **Poprawność składni** – poprawny zapis znaczników jest niezbędny do wyświetlenia dokumentu w przeglądarce.
2. **Poprawność struktury** – odpowiednia budowa dokumentu oznaczająca poprawne zagnieżdżanie elementów. Brak tej poprawności może powodować błędne zachowanie przeglądarki.
3. **Semantyczność kodu** – wykorzystanie elementów HTML zgodnie z ich znaczeniem.

Przeglądarka nie służy do sprawdzania poprawności kodu HTML. Przeglądarka jest litościwa. Sprawdzenie poprawności kodu (pierwsze dwa poziomy) można zrobić za pomocą walidatorów:

- walidator W3C (W3C Markup Validation Service, validator.w3.org)
- walidatory dostępne w przeglądarkach, np. Firefox z wtyczką Web Developer
- Tidy – program (opensource) Dave'a Raggetta walidujący kod HTML. Może współdziałać z edytorami HTML (<http://www.html-tidy.org/>).
- CSS HTML Validator – program dla środowiska Windows

Walidator wskazuje błędy (*errors*) i ostrzeżenia (*warnings*) oraz sugeruje jak je poprawić.

poprawność HTML

Dlaczego elementy HTML należy używać zgodnie z ich znaczeniem (kod semantyczny)?

- *(autor)* prostsza edycja, mniejsze pliki, łatwiejsze zastosowanie arkuszy CSS, łatwiejszy dostęp programów czytających text-to-speech
- *(przeglądarka)* lepsze przedstawienie strony użytkownikowi
- *(wyszukiwarka)* lepsze rozpoznawanie tematu dokumentu i jego wyszukiwanie
- *(użytkownicy)* łatwiejsze rozpowszechnianie informacji w WWW i jej ponowne użycie

W kodzie semantycznym nie używamy znaczników prezentacyjnych a dostępne znaczniki używamy do budowy struktury dokumentu.

Co znaczy „zgodnie z ich znaczeniem”?

- *kursywa*: podkreślenie ważności ****, czy np. podanie źródła cytatu **<cite>**?
- **pogrubienie**: nagłówek **<h1>** czy bardzo ważny fragment tekstu ****?

Pamiętaj o globalnej strukturze dokumentu – elementy powinny mieć odpowiedni, logiczny układ:

- najważniejsze nagłówki zamykamy w znaczniku **<h1>**
- te niższego poziomu oznaczamy **<h2>**, itd.,
- dla akapitów używamy **<p>**

poprawność HTML

Dziedzictwo xHTML:

- xHTML wymusił na twórcach pisanie kodu poprawnego, uporządkowanego, czystego i...
- stanowczo rozdzielił warstwę struktury od warstwy prezentacji
- HTML5 jest trochę mniej restrykcyjny
- W3C zaleca jednak pisanie kodu konsekwentnie i zgodnie z regułami takimi jak w xHTML

Uporządkowany kod...

- jest łatwiejszy do czytania dla twórcy i innych osób
- będzie dostępny dla programów czytających kod XML
- pozwala uniknąć problemów z uruchamianiem JavaScript lub użyciem DOM (Document Object Model).

poprawność HTML

O tym należy pamiętać (HTML5 vs. XHTML5):

- dokument HTML / XHTML **musi** zawierać deklarację **DOCTYPE** oraz **powinien/musi** zawierać elementy **html, head, body i title**
- wszystkie elementy XHTML **muszą** zawierać się w **jednym elemencie <html>**
- elementy HTML **powinny/muszą** być zawsze **zamknięte**:

`<p>jakiś akapit</p>`

`<meta charset="utf-8" />`

- elementy HTML **powinny/muszą** być **poprawnie zagnieżdżone**:

`<b><i>tekst</i></b>`

- elementy HTML **powinny/muszą** być pisane **małymi literami**

`<img scr="foto.png" alt="fotografia" />`

- nazwy atrybutów **powinny/muszą** być pisane **małymi literami**:

`<abbr title="et cetera">etc.</abbr>`

- wartości atrybutów **powinny/muszą** być w " ", a na pewno muszą jeśli zawierają spację.

- znakom równości **nie powinny/nie mogą** towarzyszyć spacje

`<link rel="stylesheet" href="styles.css">`