

Tworzenie nowoczesnych aplikacji backendowych w praktyce

HTML i CSS

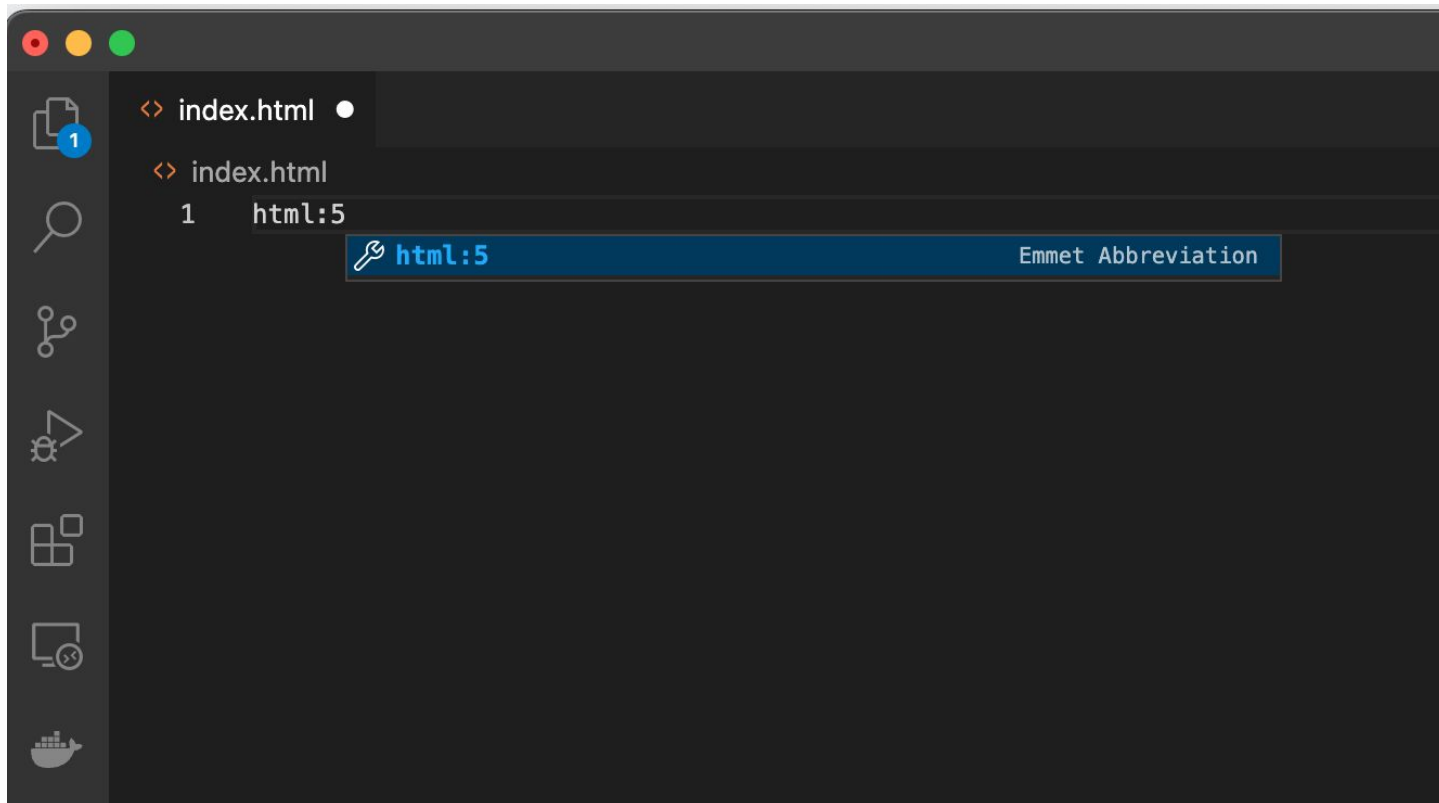
15.11.2022

allegro

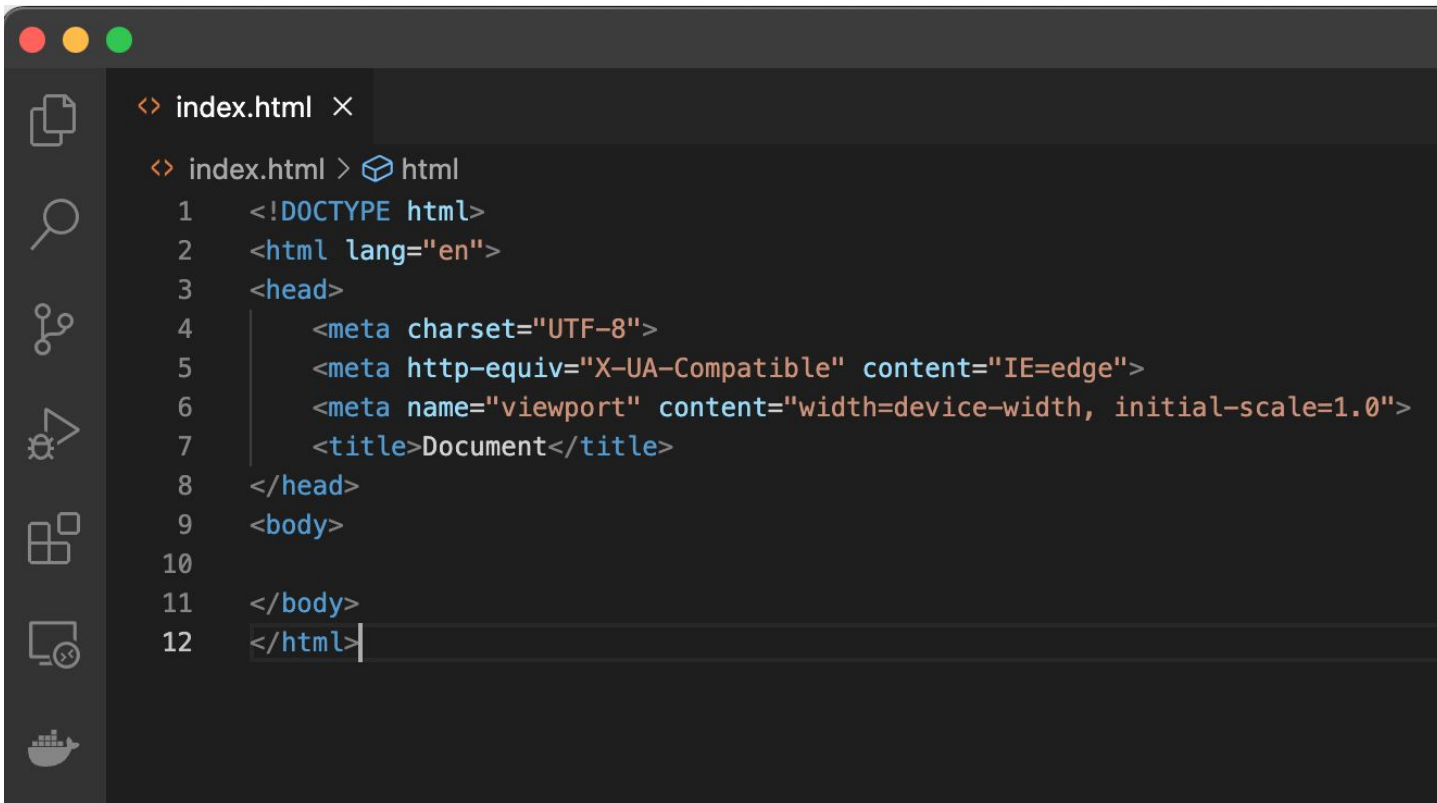
HTML



HTML5



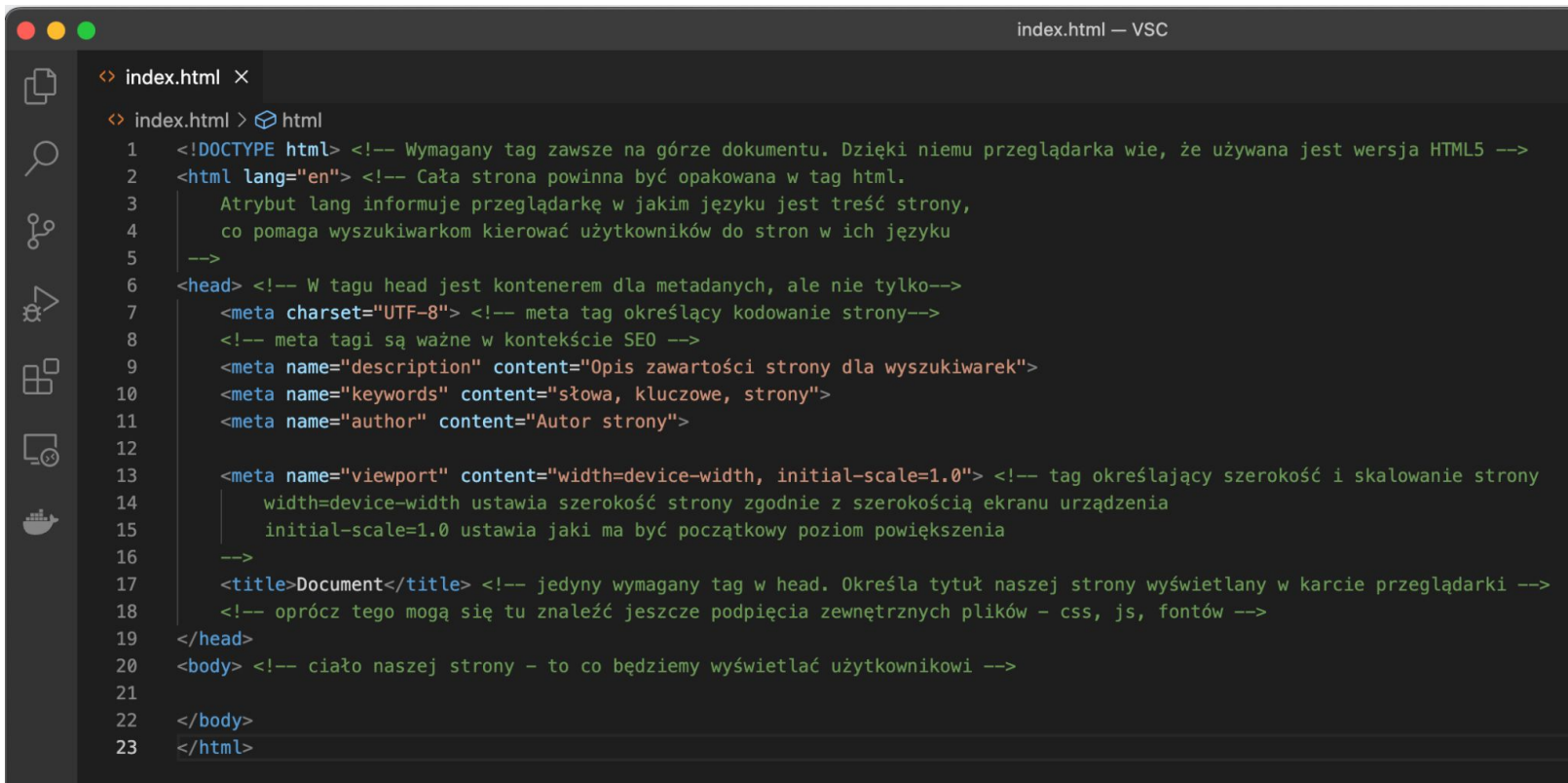
HTML5



A screenshot of a code editor window with a dark theme. The window has three colored window control buttons (red, yellow, green) in the top-left corner. On the left side, there is a vertical toolbar with icons for file management, search, and other editor functions. The main editing area shows a file named 'index.html' with a tab icon and a close button. The code is written in a light blue font on a dark background. It starts with a DOCTYPE declaration, followed by an html tag with a lang attribute. The head section contains meta tags for charset, X-UA-Compatible, and viewport, along with a title tag. The body section is currently empty. The cursor is positioned at the end of the closing html tag on line 12.

```
<> index.html ×  
  
<> index.html > html  
1  <!DOCTYPE html>  
2  <html lang="en">  
3  <head>  
4      <meta charset="UTF-8">  
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">  
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">  
7      <title>Document</title>  
8  </head>  
9  <body>  
10  
11 </body>  
12 </html>
```

HTML5



```
index.html — VSC
index.html x
index.html > html
1 <!DOCTYPE html> <!-- Wymagany tag zawsze na górze dokumentu. Dzięki niemu przeglądarka wie, że używana jest wersja HTML5 -->
2 <html lang="en"> <!-- Cała strona powinna być opakowana w tag html.
3     Atrybut lang informuje przeglądarkę w jakim języku jest treść strony,
4     co pomaga wyszukiwarkom kierować użytkowników do stron w ich języku
5     -->
6 <head> <!-- W tagu head jest kontenerem dla metadanych, ale nie tylko-->
7     <meta charset="UTF-8"> <!-- meta tag określający kodowanie strony-->
8     <!-- meta tagi są ważne w kontekście SEO -->
9     <meta name="description" content="Opis zawartości strony dla wyszukiwarek">
10    <meta name="keywords" content="słowa, kluczowe, strony">
11    <meta name="author" content="Autor strony">
12
13    <meta name="viewport" content="width=device-width, initial-scale=1.0"> <!-- tag określający szerokość i skalowanie strony
14    width=device-width ustawia szerokość strony zgodnie z szerokością ekranu urządzenia
15    initial-scale=1.0 ustawia jaki ma być początkowy poziom powiększenia
16    -->
17    <title>Document</title> <!-- jedyny wymagany tag w head. Określa tytuł naszej strony wyświetlany w karcie przeglądarki -->
18    <!-- oprócz tego mogą się tu znaleźć jeszcze podpięcia zewnętrznych plików - css, js, fontów -->
19 </head>
20 <body> <!-- ciało naszej strony - to co będziemy wyświetlać użytkownikowi -->
21
22 </body>
23 </html>
```

Podstawowe znaczniki

- **<div>** - podstawowy, ogólny kontener, którego używamy gdy inne semantyczne tagi nie pasują (o nich za chwilę).
- **<p>** - paragraf, dla treści, która ma być od nowego akapitu
- **<h1>**, **<h2>**, **<h3>**, **<h4>**, **<h5>**, **<h6>** - nagłówki strony

Wszystkie powyższe to **elementy blokowe** - oznacza to, że taki element zawsze zaczyna się w nowej linii i zajmuje całą dostępną szerokość. Więcej o elementach blokowych https://developer.mozilla.org/en-US/docs/Web/HTML/Block-level_elements

Nagłówek 1

Nagłówek 2

Nagłówek 3

Nagłówek 4

Nagłówek 5

Nagłówek 6

Podstawowe znaczniki

Istnieją też elementy **śródliniowe (inline)**.

Element śródliniowy nie zaczyna się w nowej linii i zajmuje tylko tyle szerokości, ile jest to konieczne.

Najpopularniejsze znaczniki:

- **** - generyczny inline'owy kontener, który sam w sobie niczego nie reprezentuje, używany głównie do stylizacji treści w elemencie blokowym. Używamy go wtedy, gdy żaden inny tag semantyczny nie pasuje.
- **<a>** - link do zewnętrznego zasobu
- **** - obrazki
- **<button>, <input>, <label>, <textarea>** - tagi formularzy (sam tag **<form>** jest elementem blokowym, o formularzach więcej za chwilę)

Więcej o elementach inline:

https://developer.mozilla.org/en-US/docs/Web/HTML/Inline_elements

```
<div>Treść zaznaczona na żółto <span style="background-color: yellow;">jest</span> opakowana w span</div>
```

Treść zaznaczona na żółto **jest** opakowana w span

Znaczniki semantyczne

- Określają charakter, funkcję elementu np. **<header>** **<nav>** **<section>** **<footer>**
- Czasem określają również wygląd elementów np. **** **** **<var>** ****

Semantyka strony istotna jest ze względu na:

- **SEO** - czytelniejsza dla botów wyszukiwarek - lepsze pozycjonowanie strony
- **A11y** (accessibility) czyli dostępność dla osób niepełnosprawnych. Możliwa do czytania przez syntezy mowy

Znaczniiki formatujące

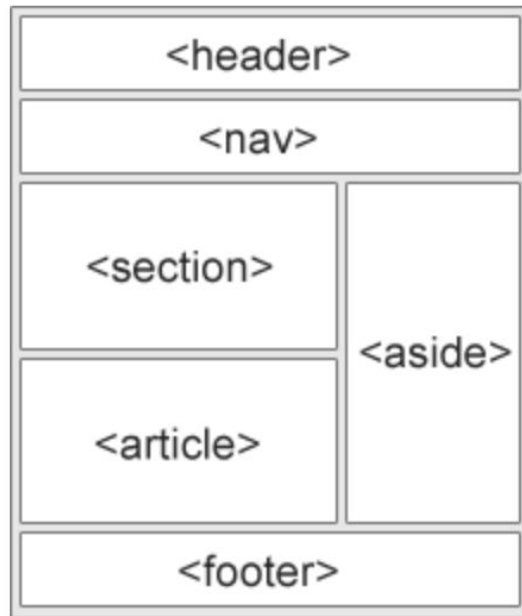
- Ich zadaniem jest tylko zmieniać wygląd elementów
- Nie odnoszą się w żaden sposób do jego znaczenia
- Przykładowe znaczniki - **<center>** ****

Czym się różni **** od ****?

Najważniejsze elementy semantyczne

- **<header>** - definiuje nagłówek dokumentu lub sekcji
- **<nav>** - definiuje elementy nawigacyjne
- **<main>** - główna treść strony
- **<section>** - sekcja dokumentu, może zawierać w sobie nagłówek
- **<article>** - niezależny fragment tekstu, może to być np. post na blogu
- **<aside>** - element, który nie jest bezpośrednio powiązany ze stroną
- **<footer>** - stopka dokumentu lub sekcji

Ogólnie w HTML5 zakładamy, że nasza strona ma układ publikacji



Inne ciekawe tagi

```
<details>  
  <summary>Wersje HTML</summary>  
  <p>HTML4</p>  
  <p>HTML5</p>  
</details>
```

Określa dodatkowe szczegóły, które widoczne zostaną po kliknięciu

► Wersje HTML

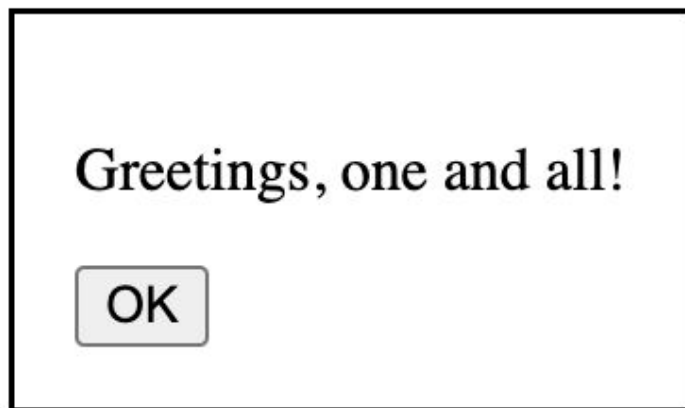
▼ Wersje HTML

HTML4

HTML5

Inne ciekawe tagi

```
<dialog open> <!-- open określa, że ma być widoczny-->
  <p>Greetings, one and all!</p>
  <form method="dialog"> <!-- przy submicie formularza dialog zostanie zamknięty-->
    <button>OK</button> <!-- nie musimy określać type="submit"
    bo to jedyny przycisk w formularzu-->
  </form>
</dialog>
```



Inne ciekawe tagi

```
<figure> <!-- określa element posiadający niezależną treść, np zdjęcie, diagram, kod-->  
    
  <figcaption>Rys.1 Randomowe zdjęcie. </figcaption> <!-- definiuje podpis pod zdjęciem -->  
</figure>
```



Rys.1 Randomowe zdjęcie.

Listy

- **Uporządkowana (*ordered list*) lista numerowana**

`` - w środku powinny znaleźć się elementy ``

```
<h3>Lista numerowana</h3>
<ol>
  <li>Owoce</li>
  <li>Warzywa</li>
</ol>
```

Lista numerowana

1. Owoce
2. Warzywa

- **Nieuporządkowana (*unordered list*) lista punktowana**

`` - w środku również używamy ``

```
<h3>Lista punktowana</h3>
<ul>
  <li>Owoce</li>
  <li>Warzywa</li>
</ul>
```

Lista punktowana

- Owoce
- Warzywa

Formularze

Tag **<form>** służy do definiowania formularzy, czyli interaktywnych sekcji do przesyłania informacji

Formularz pozwala nam na wysyłanie zapytań za pomocą metod **GET** lub **POST**. W tym celu należy zdefiniować atrybut **method**

```
<form method="GET / POST / DIALOG">
```

Atrybut **action** definiuje akcję, która zostanie wykonana przy wysłaniu formularza. Jeśli nie zadeklarujemy żadnej, zapytanie zostanie wysłane na aktualny adres, na którym się znajdujemy. Możemy w ten sposób przechodzić do innych podstron przysyłając jakieś informacje. Akcją może być również np. wysłanie maila

```
<form action="adres.html / mailto:umk@umk.pl">
```

Żeby wysłać formularze bez zmieniania strony potrzebujemy użyć JavaScript. O czym więcej później

```
<form action="adres.html" method="GET">
  <label>
    Adres
    <input name="adres" value="Torun">
  </label>
  <button>Wyślij</button>
</form>
```

Adres

Adres w przeglądarce przed wysłaniem formularza:

<http://127.0.0.1:5500/index.html>

Po wysłaniu:

<http://127.0.0.1:5500/adres.html?adres=Torun>

Elementy formularza

W formularzu mamy do wyboru wiele pól

- **<button>** - przycisk, atrybut **type** ustala jakiego jest typu. Pierwszy przycisk w formularzu będzie miał bez zdefiniowania **type=submit**
- **<input>** - generyczne pole formularza, może przyjmować wiele typów np. **text**, **number** lub zmienić się nawet w button gdy podamy mu **type=button**. Więcej tutaj <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input>
- **<label>** - podpis elementu formularza. Gdy opakujemy nim inny element formularza, np. **input**, to kliknięcie na ten tag ustawi focus na zagnieżdżony element. Innym sposobem jest nadanie mu atrybutu **for="id elementu"**
- **<select>** - pole wyboru, w środku powinny być zdefiniowane opcje do wyboru, za które odpowiada poniższy tag
- **<option>** - definiuje opcję dostępną w select
- **<textarea>** - wielowierszowe pole do wprowadzania tekstu

Więcej o elementach formularzy: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/form>

CSS



CSS - podpięcie

```
<!DOCTYPE html>
<html lang="pl">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>CSS</title>
  link
```

</head>

link

<body>

link:atom

link:css

Emmet Abbreviation

</body>

link:favicon

</html>

link:im

link:import

link:manifest

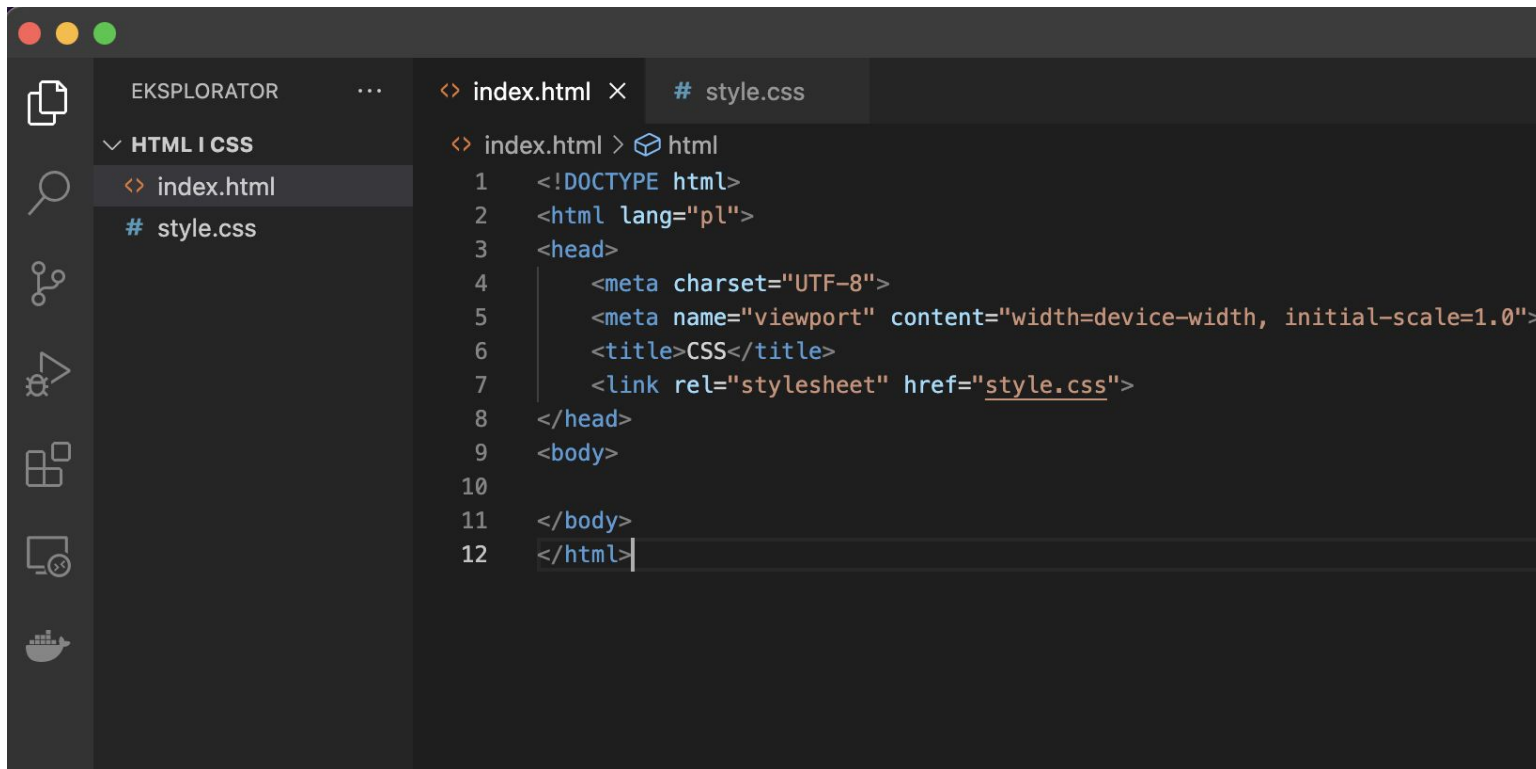
link:mf

link:print

link:rss

link:touch



CSS - podpięcie



The screenshot shows a code editor interface with a sidebar on the left and a main editing area. The sidebar, titled 'EKSPLORATOR', shows a file tree with 'HTML I CSS' expanded, containing 'index.html' and 'style.css'. The main area shows the 'index.html' file with the following code:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE html>
<html lang="pl">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>CSS</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
</body>
</html>
```

CSS - alternatywnie w pliku html

```
<> index.html >  html
1  <!DOCTYPE html>
2  <html lang="pl">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>CSS</title>
7      <style>
8          body {
9              background-color: bisque;
10         }
11     </style>
12 </head>
13 <body>
14
15 </body>
16 </html>
```

CSS - inline elementu

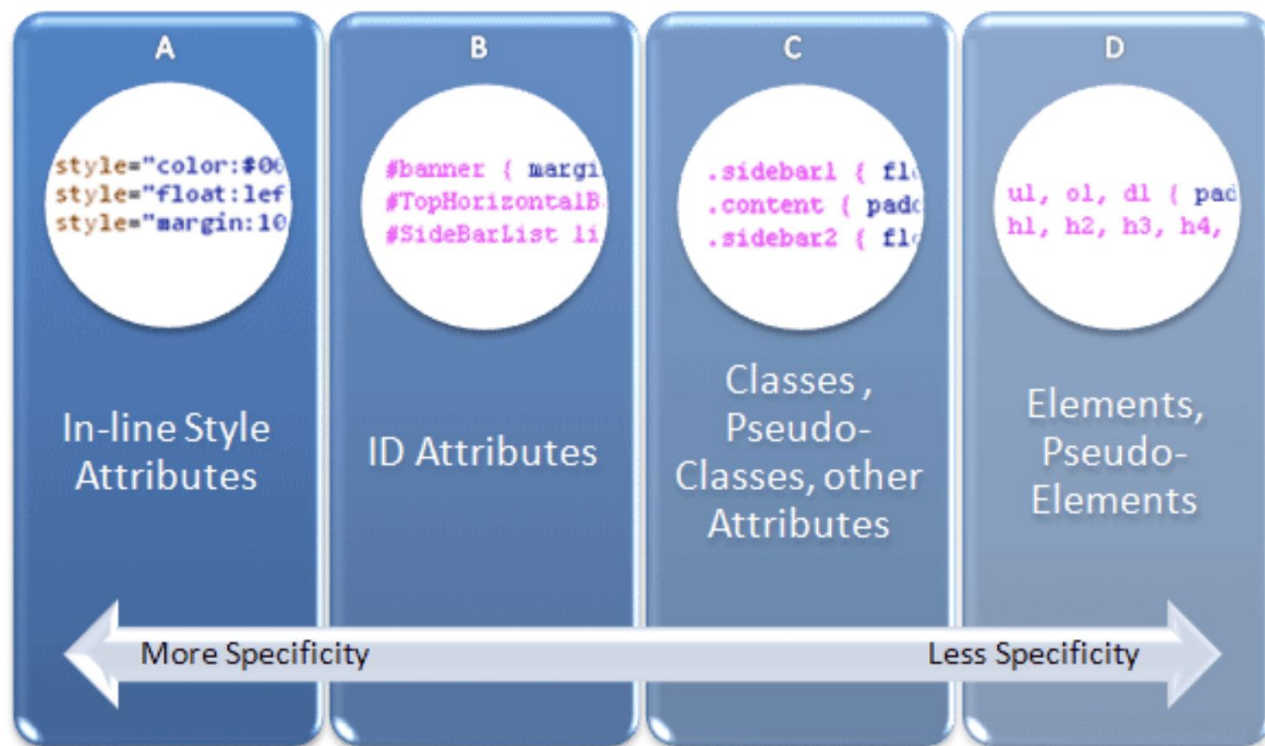
```
<> index.html >  html
1  <!DOCTYPE html>
2  <html lang="pl">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1.0">
6    <title>CSS</title>
7  </head>
8  <body style="background-color:  bisque">
9
10 </body>
11 </html>
```

CSS - selektory

<code>.intro</code>	Wszystkie elementy z <code>class="intro"</code>
<code>.name1.name2</code>	Wszystkie elementy, które mają obydwie klasy: <code>name1</code> i <code>name2</code>
<code>.name1 .name2</code>	Wszystkie elementy z klasą <code>name2</code> , które są potomkiem elementu z klasą <code>name1</code>
<code>#firstname</code>	Wszystkie elementy z <code>id="firstname"</code>
<code>*</code>	Wszystkie elementy
<code>p</code>	Wszystkie elementy <code><p></code>
<code>p.intro</code>	Wszystkie elementy <code><p></code> , które mają klasę <code>class="intro"</code>
<code>div, p</code>	Wszystkie elementy <code><div></code> i wszystkie elementy <code><p></code>
<code>div p</code>	Wszystkie elementy <code><p></code> , które są wewnątrz elementów <code><div></code>
<code>div > p</code>	Wszystkie elementy <code><p></code> , których rodzicem jest element <code><div></code>
<code>div + p</code>	Każdy pierwszy element <code><p></code> , który jest umieszczony zaraz po elemencie <code><div></code>
<code>p ~ ul</code>	Każdy element <code></code> , który jest poprzedzony elementem <code><p></code>

https://www.w3schools.com/csSref/css_selectors.php

CSS - siła selektorów

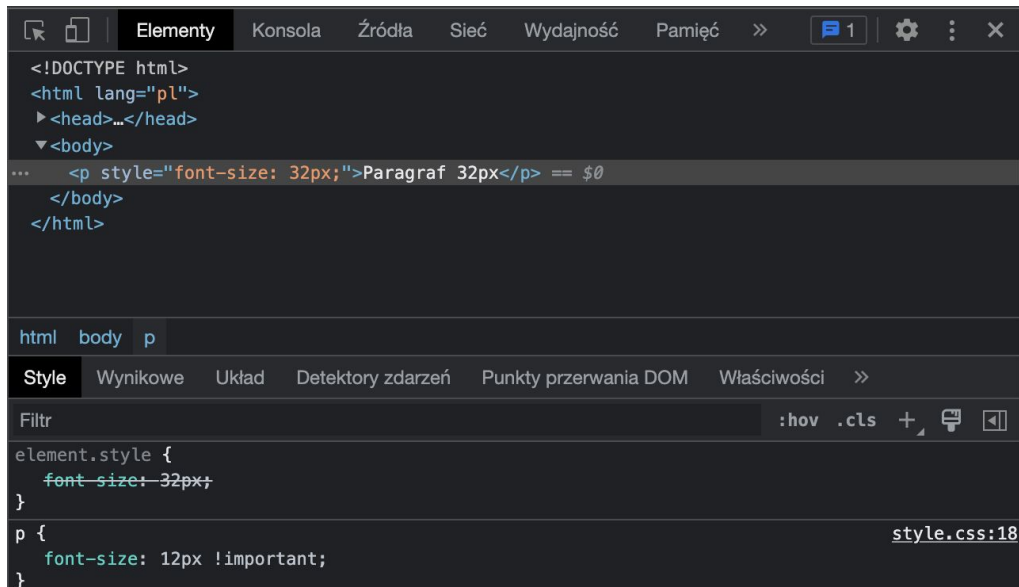


CSS - najsilniejszy selektor

Dodanie **!important** do dyrektywy CSS ma największą moc, nadpisuje wszystko. Dlatego też nie jest zalecaną praktyką.

```
<p style="font-size: 32px;">Paragraf 32px</p>
```

```
p {  
  font-size: 12px !important;  
}
```



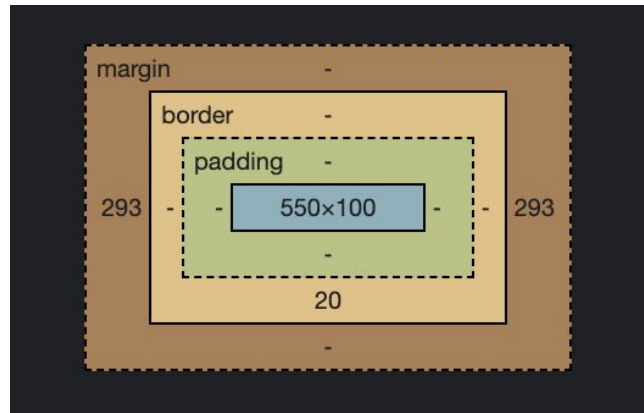
CSS - box model

W skład box model wchodzi

- **Content**, czyli wysokość i szerokość wyliczona dynamicznie lub ustalona przez własności **width** i **height**
- **Padding**, czyli wewnętrzny margines. Ustawia się go za pomocą własności o tej samej nazwie
- **Border**, czyli ramka, ustawić go można za pomocą własności o tej samej nazwie
- **Margin**, czyli margines zewnętrzny, analogicznie ustawia się go za pomocą własności o tej samej nazwie

Więcej informacji:

https://developer.mozilla.org/en-US/docs/Learn/CSS/Building_blocks/The_box_model



CSS - własność box-sizing

W css możemy ustawić box-sizing elementów. Najważniejsze wartości tej własności to:

- content-box** - ustawiony domyślnie. W przypadku tego ustawienia szerokość i wysokość liczone są następująco:
 Szerokość: $width + padding + border$
 Wysokość: $height + padding + border$
- border-box** - najczęściej wybierane ustawienie globalne dla wszystkich elementów. W tym przypadku padding i border są wliczone już w ustawiony **width**.

```
div {
  width: 160px;
  height: 80px;
  padding: 20px;
  border: 8px solid red;
  background: yellow;
}

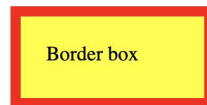
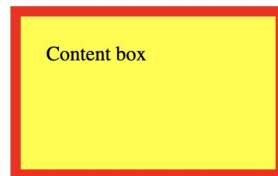
.content-box {
  box-sizing: content-box;
  /* Total width: 160px + (2 * 20px) + (2 * 8px) = 216px
   Total height: 80px + (2 * 20px) + (2 * 8px) = 136px
   Content box width: 160px
   Content box height: 80px */
}

.border-box {
  box-sizing: border-box;
  /* Total width: 160px
   Total height: 80px
   Content box width: 160px - (2 * 20px) - (2 * 8px) = 104px
   Content box height: 80px - (2 * 20px) - (2 * 8px) = 24px */
}
```

```
* {
  box-sizing: border-box;
}
```

Więcej informacji:

<https://developer.mozilla.org/en-US/docs/Web/CSS/box-sizing>



CSS - display

Właściwość ta określa zewnętrzny lub wewnętrzny typ wyświetlania elementu.

- **Zewnętrzny** to właśnie **inline** lub **block**, który określa udział elementu w layoucie strony,
- **Wewnętrzny** określa formatowanie elementów podrzędnych, które chcemy odpowiednio wypozycjonować. Np. **flex**, **grid**

Więcej informacji:

<https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

<https://css-tricks.com/snippets/css/complete-guide-grid/>

Ściągi:

<https://flexbox.malven.co>

<https://grid.malven.co>

Zadanie 1 - hamburger

Za pomocą HTML i CSS odtworzyć stronę zgodnie z makietą widoczną po prawej stronie.

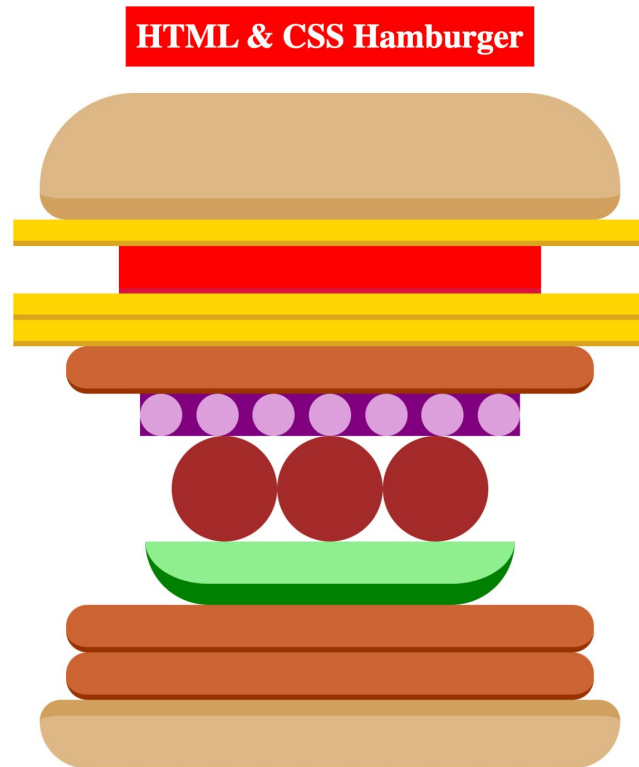
Kryteria akceptacji:

- Nowe repozytorium na github (może być jedno na cały blok zajęć, każde kolejne zadanie w oddzielnym folderze)
- HTML i CSS są w oddzielnych plikach

Przydatne linki:

- <https://developer.mozilla.org/en-US/docs/Web/CSS/display>
- <https://developer.mozilla.org/en-US/docs/Web/CSS/border-radius>

Link do repozytorium z rozwiązaniem przesłać w prywatnej wiadomości na Slack



Zadanie 2 - formularz

Za pomocą HTML i CSS przygotować formularz

- Zadanie wrzucone na github
- Kliknięcie na label każdego pola ustawia focus na pole i obramówka robi się również pomarańczowa
- Wszystkie pola poza językami programowania są wymagane i powinny być walidowane

Przydatne linki:

- <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input>
- https://developer.mozilla.org/en-US/docs/Learn/Forms/Form_validation

Link do repozytorium z rozwiązaniem przesłać w prywatnej wiadomości na Slack

Employment Form

First name

Last name

Jan

Kowalski


E-mail

Phone

Date of birth

jk@gamil.com

505 505 505

dd.mm.rrrr 

Gender

What are you looking for

☐ Software Engineer

☐ UX Designer

☐ Analytics Specialist

☐ Project Manager

Which programming languages do you know?

☐ Java

☐ Javascript

☐ C#

☐ Kotlin

☐ Python

☐ Scala

Message

Type your message

Reset

Submit

Zadanie 2 - walidacja

Walidacja

- Pola tekstowe powinny wymagać wpisanego tekstu
- Pole e-mail powinno wymagać podania prawidłowego adresu e-mail
- Pole phone powinno przyjmować numer telefonu w formacie 505 505 505
- Tylko jedno wybrane stanowisko powinno móc być wybrane
- W polu daty nie powinniśmy móc wybrać daty późniejszej niż dziś

Przydatne linki:

- <https://developer.mozilla.org/en-US/docs/Web/HTML/Attributes/pattern>
- <https://developer.mozilla.org/en-US/docs/Web/HTML/Attributes/required>

First name

Jan

E-mail



Wypełnij to pole.

Phone

What are you looking for

☒ Software Engineer

☐ UX Designer



Wybierz jedną z opcji.

Which languages do you know?

Zadanie 2 - mobile web

Formularz powinien być responsywny i wyświetlać się prawidłowo również na telefonach.

Przydatne rzeczy:

- <https://developer.mozilla.org/en-US/docs/Web/CSS/flex-wrap>
- https://developer.mozilla.org/en-US/docs/Web/CSS/Media_Queries/Using_media_queries

Employment Form

First name

Jan

Last name

Kowalski

E-mail

jk@gamil.com

Phone

505 505 505

Date of birth

dd.mm.rrrr



Gender

What are you looking for

- ☐ Software Engineer ☒ UX Designer
☐ Analytics Specialist ☐ Project Manager

Which programming languages do you know?

- ☐ Java ☐ Javascript
☐ C# ☐ Kotlin
☐ Python ☐ Scala

Message

Type your message

Reset

Submit

Inne przydatne linki

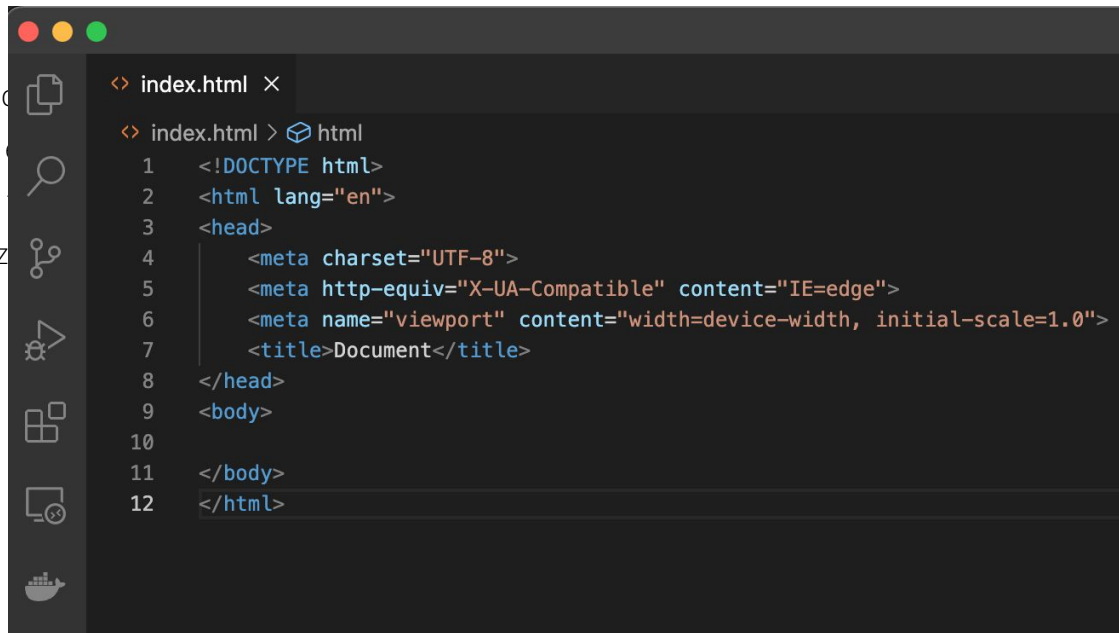
Ściągi

- <https://htmlcheatsheet.com/css/>
- <https://htmlcheatsheet.com>
- <https://htmlreference.io>
- <https://cssreference.io>

HTML5

Więcej niż jeden cel:

- Przekazanie praktycznej, aktualnej wiedzy
- Pokazanie jak wygląda praca frontend
- Zaprezentowanie najpopularniejszych
- Przygotowanie do rozpoczęcia pierwsz
- ?



```
<> index.html X
<> index.html > html
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Document</title>
8  </head>
9  <body>
10
11 </body>
12 </html>
```