



INVEST IN POMERANIA ACADEMY

Podstawy HTML, CSS, Bootstrap, JS i Ajax

Część 1



Fundusze
Europejskie
Program Regionalny



Rzeczpospolita
Polska



URZĄD MARSZAŁKOWSKI
WOJEWÓDZTWA POMORSKIEGO

Unia Europejska
Europejski Fundusz
Rozwoju Regionalnego





HELLO

Krzysztof Drzymalski

Frontend Developer





Agenda

1. Omówienie dokumentacji MDN
2. Dostępne narzędzia
3. Podstawowe znaczniki HTML + Emmet
4. CSS
5. Selektory CSS
6. Rozmieszczanie
7. Bootstrap wstęp i instalacja



Współpraca

- Zadajemy pytania w dowolnym momencie – kanał merytoryka
- Krótkie przerwy (5 min) co godzinę
- Zadania do zrobienia znajdują się na githubie =>
<https://github.com/infohareacademy/jjdzr10-materialy-HTML-CSS-Bootstrap-JS-AJAX>



01. Dokumentacja

Na przykładzie MDN

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



02. Dostępne narzędzia

Z jakich narzędzi możemy korzystać podczas pracy na frontendzie





Dostępne narzędzia

IDE (Integrated Development Environment)

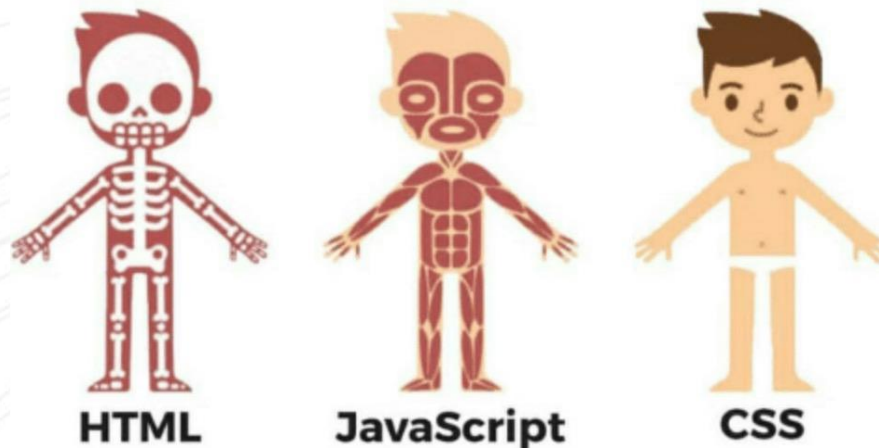
- IntelliJ Idea Ultimate
- Webstorm

Edytory tekstu

- Visual Studio Code
- Sublime text
- Brackets
- Notepad++
- Zwykły Notatnik
- I wiele wiele innych...

Kilka ważnych pojęć

- **HTML** – język znaczników, budujemy za jego pomocą strukturę naszej strony. Powstał we wczesnych latach 90
- **CSS** – język arkusza stylów, za jego pomocą „upiększamy” nasz HTML. Wizualne kwestie oraz „fajerwerki powstają w tym języku”
- **JavaScript** – kompletny język programowania, za jego pomocą wprowadzamy interaktywność do naszej aplikacji





Podstawowy szablon HTML

Podstawowe elementy szablonu strony:

- **Doctype** – znacznik określający typ dokumentu
- **Html** – główny znacznik otwierający dokument HTML
- **Head** – znacznik zawierający informacje o dokumencie
- **Body** – treść dokumentu

```
<!doctype html>
<html lang="en">
  <head>
    <title>Document</title>
  </head>
  <body>
  </body>
</html>
```



03.

Podstawowe Znaczniki HTML

+ Emmet





Znaczniki HTML

Znaczniki dzielą się na 2 grupy:

- **Kontenerowe** - należy je otworzyć i zamknąć (div, p, a, buton, h1-h6)
- **Samodzielne** - jeden tag jest zarówno otwierającym jak i zamykającym (br, input, img)

```
<!doctype html>
<html lang="en">
  <head>
    <title>Document</title>
  </head>
  <body>
    <h1>Hello world!</h1>
    <br />
  </body>
</html>
```

Znacznik otwierający → `<h1>`

Znacznik samodzielny → `
`

`</h1>` ← Znacznik zamykający



Zagnieżdżanie znaczników

Źle

```
<div>
  <p>Paragraph 1</p>
  <p>Paragraph 2</p>
  <div>
    <p>Paragraph 3</p>
    <p>Paragraph 4 <span>My span</p></span>
  </div>
</div>
```

Dobrze

```
<div>
  <p>Paragraph 1</p>
  <p>Paragraph 2</p>
  <div>
    <p>Paragraph 3</p>
    <p>Paragraph 4 <span>My span</span></p>
  </div>
</div>
```



Zadanie 1 niepoprawny HTML

1. Otwórz plik o nazwie zadanie1.html
2. Uruchom go w przeglądarce
3. Wygląda to inaczej niż byśmy chcieli, postaraj się poprawić błędy (pamiętaj że po każdej zmianie musisz odświeżyć przeglądarkę żeby zobaczyć efekt)

Czas: 10 min



Atrybuty HTML

Znaczniki poza treścią mogą zawierać także dodatkowe parametry zwane **atrybutami**

Atrybuty pozwalają modyfikować właściwości i domyślne zachowanie elementu. Niektóre atrybuty są opcjonalne a niektóre musimy podać, żeby element działał prawidłowo

Atrybuty dodawane są według poniższego schematu:

`<tag atrybut1=„wartość1” atrybut2=„wartość2”>Treść</tag>`

Przykład z życia:

```
<a id="myLink" href="https://google.com">My link</a>
```



Id vs Class

Dwa bardzo popularne atrybuty to:

- **id** – unikalna wartość w obrębie jednego dokumentu. Dzięki niemu możemy bezpośrednio odwołać się do konkretnego elementu
- **class** – atrybut przechowujący wartość powtarzającą się. Często nadaje się taką samą wartość class dla kilku elementów. Dzięki temu jesteśmy w stanie grupować elementy i np. ostylewać je tak samo

```
<ul id="myList">  
  <li class="list-element">List element 1</li>  
  <li class="list-element">List element 2</li>  
  <li class="list-element">List element 3</li>  
</ul>
```

```
<p class="simple-paragraph">Paragraph1</p>  
<p class="simple-paragraph">Paragraph2</p>  
<p class="simple-paragraph">Paragraph3</p>
```

- Emmet to zapis tekstu, który zostanie przekształcony przez IDE lub jakiś plugin na prawidłowy kod HTML
- Nie jest to podpowiadanie składni, ale metodyka szybkiego tworzenia elementów w strukturze dokumentu HTML

Istotne jest to że nie trzeba korzystać z takich narzędzi jeśli ich nie potrzebujemy. Mają one na celu przyspieszyć naszą pracę, a nie ją komplikować

Dwie przydatne ściąggi:

- <https://docs.emmet.io/cheat-sheet/>
- <https://devhints.io/emmet>

Wskazują znaczenie sekcji znajdujących się po nich. Stopniujemy ich od h1 do h6, gdzie nagłówek h1 jest ważniejszy od h6

Mają hierarchiczną strukturę, co oznacza że powinniśmy stosować je w odpowiedniej kolejności (od h1 do h6) przykład:

Źle

```
<h3>Nagłówek 3</h3>  
<h4>Nagłówek 4</h4>  
<h1>Nagłówek 1</h1>  
<h6>Nagłówek 6</h6>  
<h2>Nagłówek 2</h2>  
<h5>Nagłówek 5</h5>
```

Dobrze

```
<h1>Nagłówek 1</h1>  
<h2>Nagłówek 2</h2>  
<h3>Nagłówek 3</h3>  
<h4>Nagłówek 4</h4>  
<h5>Nagłówek 5</h5>  
<h6>Nagłówek 6</h6>
```

Tworzony przy użyciu znacznika `<p></p>`

Służy do tworzenia bloków tekstowych na stronie

```
<p>
  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque a ultrices justo. Aenean iaculis id orci ac mattis.
  Fusce tincidunt enim ut sapien egestas rhoncus ut eget mauris. Phasellus aliquam dictum urna, ac imperdiet justo aliquam vitae.
  Aliquam ut est felis. Aenean volutpat tempus luctus. Mauris consequat, lacus eu dapibus maximus, urna erat porttitor leo, at placerat nulla velit quis odio.
  Sed lacinia felis eu scelerisque convallis. Quisque et fringilla elit.
</p>
```

Przydatna strona do generowania dużych ilości sztucznego tekstu:

<https://www.lipsum.com/feed/html>

Ich celem jest nawigowanie użytkownika, tworzymy za pomocą tagu `<a>`

Ich podstawowym atrybutem jest **href** określający zasób do którego będziemy kierować (nawigować) użytkownika

```
<a href="https://google.com">Navigate to Google</a>
```


Interaktywny element tworzony używając tagu `<button></button>`

Jak sama nazwa sugeruje pozwala umieścić przycisk na stronie. Powinien zawierać atrybut **type** (np. `button`, `reset`, `submit`) zwłaszcza wewnątrz formularza

Jesteśmy w stanie dezaktywować przycisk nadając mu atrybut **disabled**

Aktywny przycisk

```
<button>Click me</button>
```

Click me

Nieaktywny przycisk

```
<button disabled="disabled">You can't click me :P</button>
```

You can't click me :P

Zasadniczo listy możemy podzielić na dwa typy:

- **Uporządkowane** – tworzymy za pomocą znacznika ``
- **Nieuporządkowane** – tworzymy za pomocą znacznika ``

Elementy wewnątrz listy dodajemy znacznikami ``

```
<ol>
  <li>List item 1</li>
  <li>List item 2</li>
</ol>
```

Kod

1. List item 1
2. List item 2

Wynik

```
<ul>
  <li>List item 1</li>
  <li>List item 2</li>
</ul>
```

Kod

- List item 1
- List item 2

Wynik

Podstawowym tagiem służącym do wstawiania obrazków jest ``

```

```

Istotne atrybuty:

- **src** – tym atrybutem określamy ścieżkę do wczytania grafiki
- **alt** – za jego pomocą nadajemy obrazkowi „tekst alternatywny”, wyświetli się ona w momencie gdy nie uda się załadować grafiki oraz czytniki strony (dla osób niewidomych) będą mogły go zobaczyć

Zarówno **src** jak i **alt** są atrybutami wymaganymi

Uniwersalny kontener **blokowy** tworzymy za pomocą znacznika `<div></div>`

Możemy go używać jako opakowanie na treść i dowolnie ostylewać

Często jego celem jest pogrupowanie elementów w HTML

Domyślnie jest niewidoczny

Uniwersalny kontener **liniowy** tworzymy za pomocą znacznika ``

Zwykle używany do stylowania fragmentów treści zawartych w innych kontenerach

Od diva odróżnia go przede wszystkim to że jest **elementem liniowym**



Elementy blokowe i liniowe

Dwa główne sposoby na wyświetlanie elementów:

- **Blokowy** – wyświetlane są zawsze w nowej linii i zajmują całą jej powierzchnię „rzucając” inne elementy na dół (np. `div`)
- **Liniowy** – zajmują tylko tyle miejsca ile potrzebują (np. `span`)

BLOCK-LEVEL ELEMENTS:



INLINE ELEMENTS:



Semantyka – definiowanie precyzyjnego znaczenia poszczególnych elementów oraz ich funkcji

Po co nam to?

- Lepsze pozycjonowanie naszej strony w wynikach wyszukiwania
- Poprawa accessibility (możliwość korzystania ze strony dla osób z dysfunkcjami)
- Zwiększa czytelność kodu
- Nadaje elementom więcej znaczenia i sugeruje ich przeznaczenie



Popularne znaczniki semantyczne

- `<header></header>` – kontener dla nagłówka
- `<footer></footer>` – kontener dla stopki
- `<main></main>` – kontener głównej części dokumentu
- `<nav></nav>` – kontener na elementy nawigacyjne
- `<section></section>` – odrębna sekcja strony
- `<article></article>` – artykuł
- `<aside></aside>` – poboczna sekcja strony

Ściągą semantyczną z MDN : <https://developer.mozilla.org/en-US/docs/Glossary/Semantics>



Zadanie semantyka

- Otwórz plik zadanie2.html
- Zamień divy na właściwe elementy semantyczne

Czas: 10 min

Jeśli nie chcemy żeby przeglądarka wyświetlała nasz element ale z jakiegoś powodu nie chcemy usuwać go z kodu możemy go „zakomentować”

Składnia komentarza w HTML to : `<!-- tutaj treść komentarza -->`

```
<!--      <div>Commented element</div>-->
```

```
<!--      Komentarze możemy też pisać bez użycia tagów-->
```

Kiedy warto użyć:

- Gdy na szybko chcemy coś przetestować
- Jeżeli dany fragment kodu jest mało czytelny i chcemy lepiej opisać co tam się dzieje

Jego głównym celem jest zbieranie danych od użytkownika

Posiada kilka charakterystycznych elementów:

- `<form>` - otacza całą sekcję formularza
- `<input>` - element do wprowadzania danych wejściowych
- `<label>` - etykieta dla elementów input. Za pomocą atrybutu for określamy przynależność do konkretnego inputa
- `<textarea>` - dla wielolinijkowych danych
- `<button>` - przycisk (domyślnie wysyła formularz)

Główne typy pól formularza to

- **text** – zwykły ciąg znaków
- **password** – również tekst, ale zamaskowany
- **number** – liczba
- **date** – kalendarz wyboru daty (zależne od przeglądarki)
- **range** – suwak (wybór liczby z przedziału)
- **file** – umożliwia upload pliku
- **radio** – umożliwia skonstruować pytanie jednokrotnego wyboru
- **checkbox** – umożliwia skonstruować pytanie wielokrotnego wyboru



Przykładowy formularz

```
<form>
  <label for="fname">Imię:</label>
  <input type="text" id="fname" name="fname"><br/>
  <label for="lname">Nazwisko:</label>
  <input type="text" id="lname" name="lname"><br/>

  <!--pytanie wielokrotnego wyboru-->
  <p>Jaki środek transportu posiadasz?</p>
  <input type="checkbox" id="vehicle1" name="vehicle1" value="Bike">
  <label for="vehicle1"> Mam rower</label><br>
  <input type="checkbox" id="vehicle2" name="vehicle2" value="Car">
  <label for="vehicle2"> Mam samochód</label><br>
  <input type="checkbox" id="vehicle3" name="vehicle3" value="Boat">
  <label for="vehicle3"> Mam łódź</label><br>
  <input type="checkbox" id="vehicle4" name="vehicle4" value="Plane">
  <label for="vehicle4"> Mam samolot</label><br>

  <!--pytanie jednokrotnego wyboru-->
  <p>Którą technologię znasz najlepiej?</p>
  <input type="radio" id="html" name="fav_language" value="HTML">
  <label for="html">HTML</label><br>
  <input type="radio" id="css" name="fav_language" value="CSS">
  <label for="css">CSS</label><br>
  <input type="radio" id="javascript" name="fav_language" value="JavaScript">
  <label for="javascript">JavaScript</label>
<br/>
<label for="about">Opowiedz coś o sobie</label><br/>
<textarea id="about" name="about"></textarea>
<br/>

  <input type="submit" value="Wyślij">
</form>
```

Imię:

Nazwisko:

Jaki środek transportu posiadasz?

- ☐ Mam rower
- ☒ Mam samochód
- ☒ Mam łódź
- ☐ Mam samolot

Którą technologię znasz najlepiej?

- ☐ HTML
- ☒ CSS
- ☐ JavaScript

Opowiedz coś o sobie

Wyślij



Zadanie – formularz osobowy

Otwórz plik zadanie3.html

Utwórz formularz, który będzie posiadał następujące pola:

1. Login
2. Hasło
3. Wzrost
4. Data urodzenia
5. Czy palisz papierosy? (tylko jedna odpowiedź jest dozwolona)
6. Zaznacz swoje zainteresowania (wiele odpowiedzi spośród odpowiedzi jest prawidłowa: Sport, Sztuka, Muzyka, Programowanie, Gotowanie, Samochody, Moda, Zdrowy tryb życia)
7. Napisz coś o sobie (pole wielolinijkowe)

Na samym dole dodaj dwa przyciski z treścią "Wyślij" oraz „Reset”

Czas 20 min



Kaskadowe arkusze stylów



- Język arkuszy stylów
- Pozwala nadawać wygląd naszym dokumentom HTML
- Dzięki niemu jesteśmy w stanie ustawiać i pozycjonować elementy

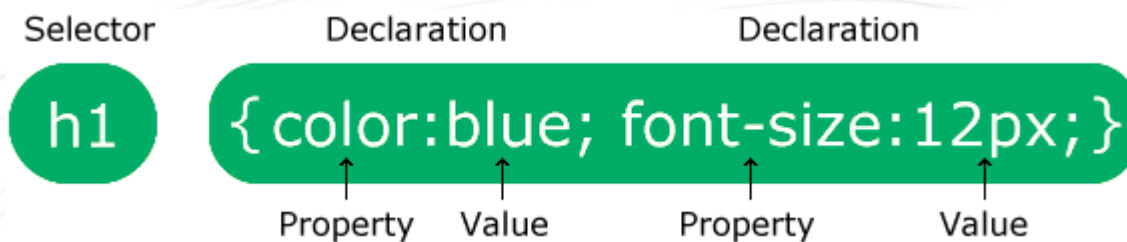




Składnia CSS

Składnia stylowania składa się z 3 elementów:

- Selektor np. h1
- Właściwości np. color
- Wartości np. niebieski (blue)





Sposoby na dodanie CSS do HTML

Inline

Dodawanie stylowania bezpośrednio do elementu używając atrybutu style

```
<p style="color:blue;border:1px solid black"></p>
```

Tag style

Umieszczamy tag `<style>` wewnątrz `<head>` i piszemy tam selektory

```
<head>
  <title>Document</title>
  <style>
    p {
      color: blue;
      border: 1px solid black;
    }
  </style>
</head>
```

Arkusz zewnętrzny

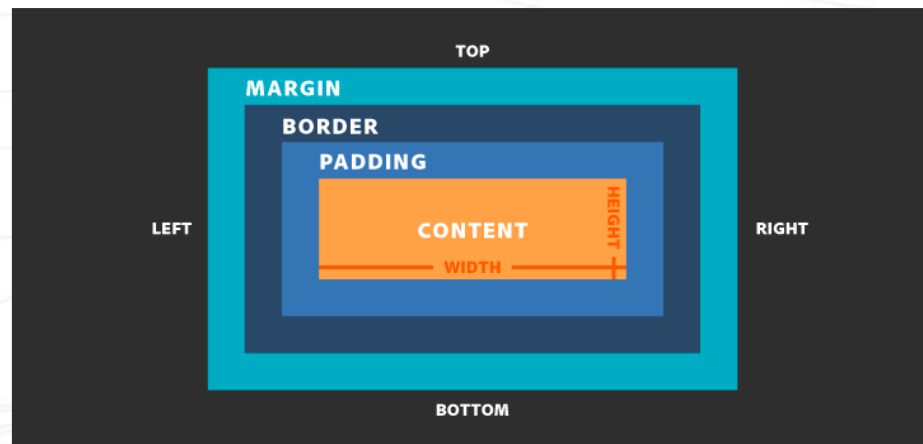
Wydzielamy selektory do osobnego pliku .css, a następnie importujemy je w `<head>` za pomocą tagu `<link>`

```
<head>
  <title>Document</title>
  <link rel="stylesheet" type="text/css" href="styles.css">
</head>
```


Box Model

Model składa się z następujących elementów:

- Margin
- Border
- Padding
- Content



Właściwość `box-sizing` określa jak właściwości `width` i `height` wpływają na rozmiary elementu. Przyjmuje dwie wartości:

- **content-box** – domyślna wartość, w przypadku której właściwości `width` i `height` zawierają jedynie content. Zmiana padding czy border **WPŁYNIE** na rozmiar elementu
- **border-box** – w tym przypadku właściwości `width` oraz `height` uwzględniają już w sobie padding i border więc zmiana paddingu czy bordera **NIE WPŁYNIE** na rozmiar elementu



05. **Selektory CSS**

Docieranie do konkretnych elementów



Unikalny identyfikator elementu, odnosimy się do niego za pomocą znaku #

Selektor

```
#myParagraph {  
  color: red;  
}
```

Element

```
<p id="myParagraph">Paragraph 1</p>
```

Za pomocą class grupujemy elementy i nadajemy im te same style. Do class odnosimy się za pomocą znaku .

Selektor

```
.listItem {  
    color: yellow;  
}
```

Elementy

```
<ul>  
    <li class="listItem">List item 1</li>  
    <li class="listItem">List item 2</li>  
    <li class="listItem">List item 3</li>  
</ul>
```



Pseudoklasy CSS

Umożliwiają stylowanie specjalnych stanów elementów. Dodajemy je do selektora za pomocą znaku : (dwukropek)

Często używane:

- **hover** – stan po najechaniu kursorem na element
- **active** – stan aktywnego „wciśnięcia”
- **visited** – link który był już odwiedzony

Przykład:

```
button:active {  
    color: yellow;  
}
```




Pseudo elementy CSS

Umożliwiają stylowanie specjalnych części elementów. Dodajemy je do selektora za pomocą znaku `::` (podwójny dwukropek)

Często używane:

- **before** – jeżeli dodamy właściwość **content** to pozwoli nam stworzyć nowy element i wstawić jako pierwsze dziecko wybranego elementu bez modyfikacji HTML
- **after** – jeżeli dodamy właściwość **content** to pozwoli nam stworzyć nowy element i wstawić jako ostatnie dziecko wybranego elementu bez modyfikacji HTML
- **first-letter** – aplikuje style do pierwszej litery danego elementu



Stopień ważności selektora

Gdy kilka selektorów wskazuje na ten sam element przeglądarka musi wybrać które style zaaplikować elementowi Ten selektor który jest silniejszy (ma większy numer) wygrywa

Od najsilniejszego:

- Użycie słowa kluczowego **!important**
- Stylowanie **inline** za pomocą tagu **style**
- Selektor id (**#myId**)
- Selektor class, atrybutu, pseudo klasy (**.my-class [href=„mylink”] :visited :hover**)
- Selektor elementu, pseudo elementu

Kalkulator siły selektora:

<https://specificity.keegan.st/>

Przykład stylowania importantem

```
button {  
  background-color: blue !important;  
}
```



Examples of table styles

Selektor	Przykład	Opis
element	div	Wszystkie divy
element.klasa	div.container	Wszystkie divy z klasą container
element element	p a	Wszystkie linki których przodkiem jest p
element > element	div > p	Wszystkie p których BEZPOŚREDNIM rodzicem jest div
element,element	div,span	Wszystkie div'y i span'y
[atrybut]	[disabled]	Wszystkie elementy posiadające atrybut disabled
element + element	a + p	Wszystkie p które mają bezpośrednio przed sobą a
element ~ element	section ~ div	Wszystkie divy które są za section
gwiazdka	*	Wszystkie elementy

Ściąga dla selectorów: https://www.w3schools.com/cssref/css_selectors.php

Kolor **tła** definiujemy właściwością **background-color**

Kolor **czcionki** definiujemy właściwością **color**

Kolor **obramowania** definiujemy właściwością **border-color**

Sposoby na zapis wartości koloru:

- **Rgb:** (0,0,255)
- **Hex:** #0000FF
- **Hsl:** (240°,100%,50%)
- **Nazwa:** blue

Kalkulator: <https://www.rapidtables.com/convert/color/hsl-to-rgb.html>

Dobieranie kolorów: <https://color.adobe.com/pl/create/color-wheel>

Pozwalają nam określać jak duże mają być nasze elementy. Dzielimy je na:

- **Absolutne** – Nie są obliczane np. piksele (px) lub milimetry (mm)
- **Relatywne** – Obliczane względem czegoś **em** (względem rozmiaru czcionki rodzica), **rem** (względem globalnego rozmiaru czcionki) **vw** (względem szerokości ekranu), **vh** (względem wysokości ekranu) **procenty** (względem rodzica)



Zadanie stylowanie

- Weź kod formularza z zadania 3
- Zastosuj następujące style:
 - Formularz musi mieć obramowanie o grubości 2 px i koloru czerwonego
 - Wewnątrz formularza musi być padding 10px z każdej strony
 - Przyciski muszą mieć minimum 80px szerokości oraz być w jednej linii. Odstęp między nimi musi wynosić co najmniej 16px
 - Po najechaniu kursorem na przycisk Wyślij jego kolor tła musi zmienić się na niebieski
 - Odstęp wertykalny (pionowy) między inputami musi wynosić co najmniej 8px
 - Kolor tekstu w całym formularzu musi być szary

Czas: 15min



06. **Rozmieszczanie**

Ustawianie elementów tak jak chcemy



Właściwość position

Za pomocą tej właściwości wskazujemy przeglądarce jak ma traktować i przemieszczać dany element. Właściwość position może przyjmować jedną z 5 wartości:

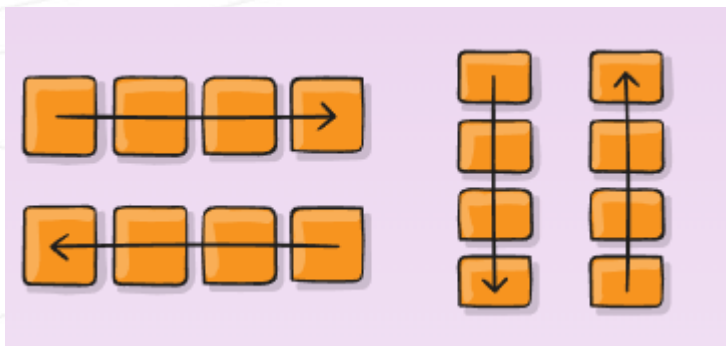
- **Static** – wartość domyślna, element pozostaje we flow documentu właściwości top/left/right/bottom nie mają wpływu na pozycjonowanie
- **Relative** – pozwala na przemieszczanie top/left/right/bottom, relatywny do swojej pozycji, pozostaje we flow dokumentu
- **Fixed** – usuwany z głównego flow, przyklejony do viewport nawet podczas scrollowania
- **Absolute** – pozycjonowany za pomocą top/left/right/bottom względem najbliższego przodka który ma position inne niż static
- **Sticky** – połączenie relative i fixed. Relative do określonego miejsca, potem zmienia się w fixed. Pozycjonowanie na przestrzeni najbliższego scrollującego przodka

Flexbox

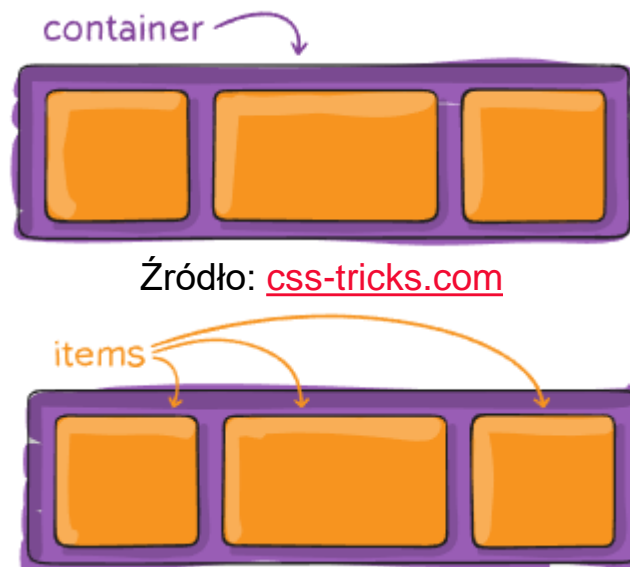
Nowoczesna właściwość wspierająca rozmieszczanie i tworzenie layoutów. Za jej pomocą możemy rozmieszczać elementy liniowo (pionowo lub poziomo)

Nadajemy ją elementowi ustawiając właściwość **display** na wartość **flex** lub **inline-flex**

Kierunek ustalamy za pomocą **flex-direction**, a następnie dzięki **justify-content** oraz **align-items** rozmieszczamy elementy

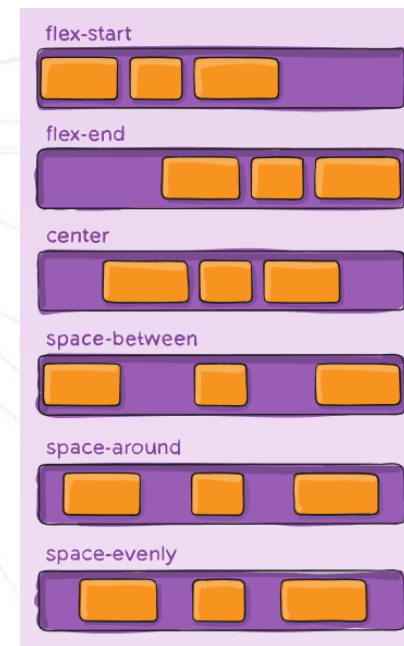


Źródło: css-tricks.com



Źródło: css-tricks.com

Źródło: css-tricks.com



Źródło: css-tricks.com

Świetna ściągą całego flexboxa: <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

Dev Tools

Możemy uruchomić narzędzia developerskie klikając prawym przyciskiem myszy na dowolnym elemencie i wybierając opcję **zbadaj**. Opcjonalnie możemy również skorzystać z przycisku **f12**



Rynek woła o programistów – nie daj się prosić!

Stare indyjskie przysłowie mówi: Jeśli do świata IT wejdiesz, życie Twoje się zmieni, a Ty na brak pracy narzekać nie będziesz. I programować umieć nie musisz, trenerzy z InfoShare Academy wszystkiego Cię nauczą.

Poznaj nasze kursy

Wypełnij test predyspozycji

Sprawdź nasze kursy

Technologia

Poziom

Tryb

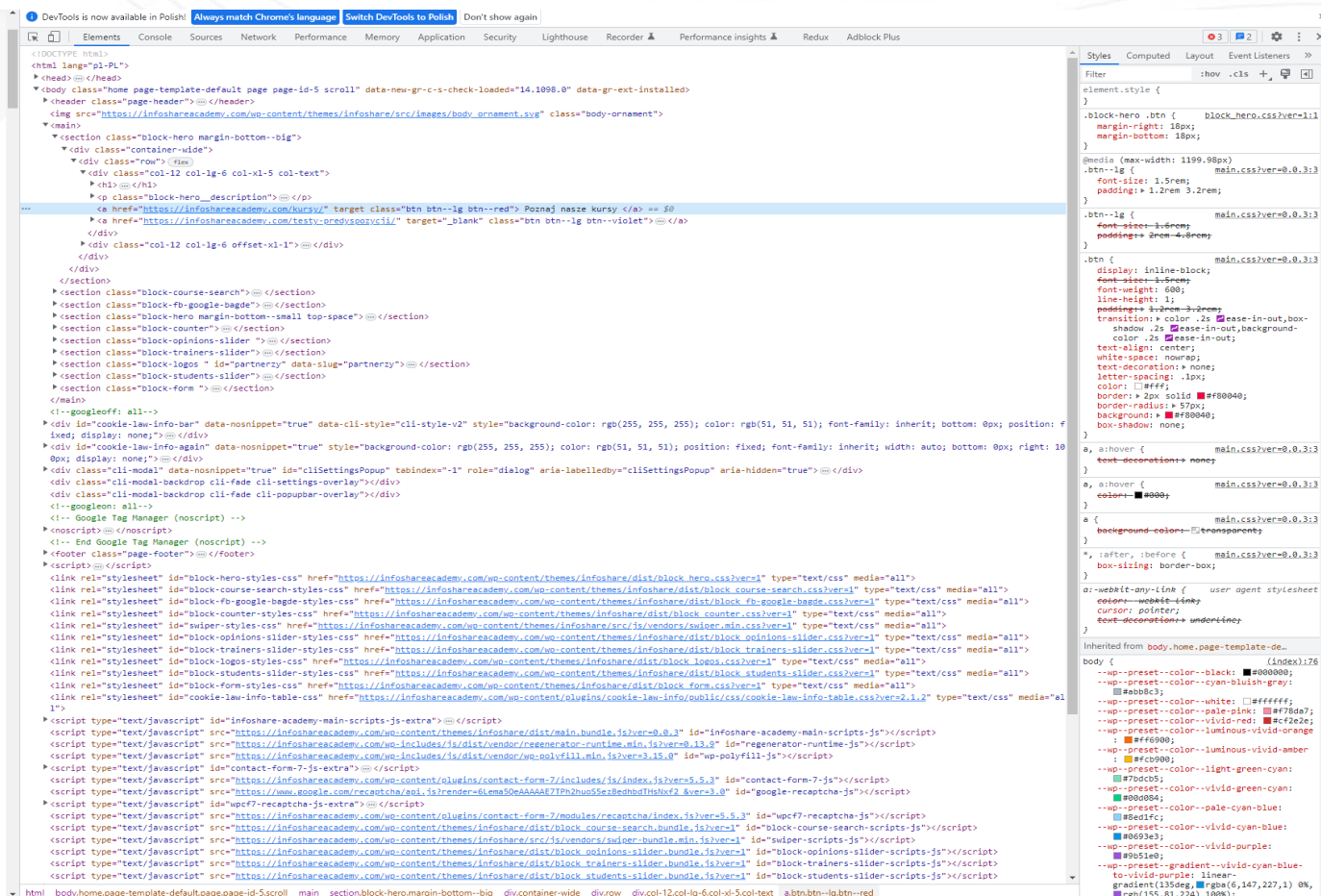
Szukaj

TYLKO DO 28.02.2023

TYLKO DO 28.02.23

PRZEBRAŃOWIENIE
OSTATNIE TYGODNIE
NISKICH CEN!

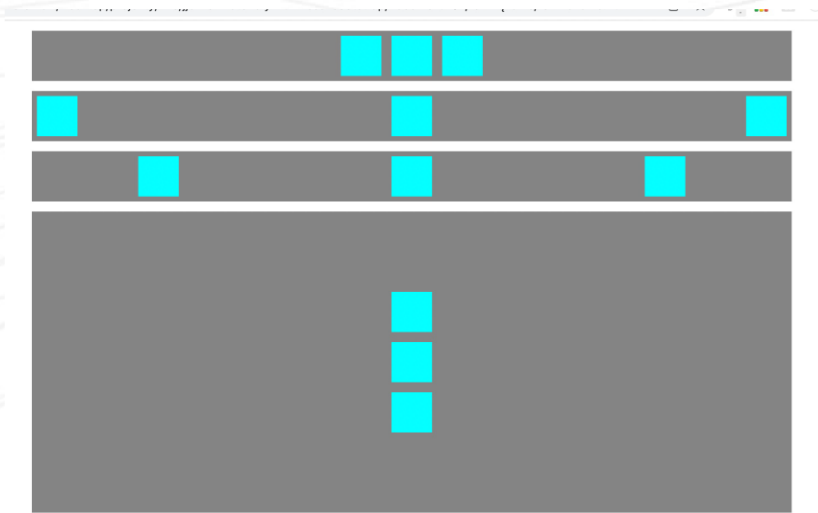
PRZEBRAŃOWIENIE
OSTATNIE TYGODNIE
NISKICH CEN!



infoShare
ACADEMY

Zadanie flexbox

- Otwórz plik zadanie5.html
- Spraw aby pasek nawigacyjny został przyklejony na górze ekranu nawet podczas scrollowania
- Poustawiaj niebieskie kwadraty w taki sposób jak na zdjęciu poniżej (za pomocą flexboxa).
- Nie zmieniaj struktury HTML pliku



Czas: 15min



Framework CSS



Czym są frameworki CSS

Są to projekty zawierające kompletne i gotowe zestawy styli możliwe do zastosowania w innych projektach.

Zalety:

- Dzięki nim szybciej stworzymy wygląd naszej aplikacji
- Z góry dbają o dobre praktyki i dostarczają nam cały wachlarz narzędzi

Wady:

- Zwiększają próg wejścia do naszej aplikacji (dla nowych programistów)
- Często narzucają nam pewne ograniczenia

Oficjalna dokumentacja Bootstrapa:

<https://getbootstrap.com/docs/5.3/getting-started/introduction/>



Instalacja

Istnieje kilka sposobów na dodanie Bootstrapa do naszej strony:

1. Podlinkowanie go za pośrednictwem serwerów CDN
2. Ściągnięcie go na dysk i dodanie do odpowiednich folderów
3. Skorzystanie z bundlerów jak webpack czy Vite (zaawansowany sposób)

Dokładna instrukcja instalacji:

<https://getbootstrap.com/docs/5.2/getting-started/introduction/>

Pamiętaj żeby dodać znacznik `<meta name="viewport" content="width=device-width, initial-scale=1">` do pliku HTML



Dostępne funkcje

Bootstrap udostępnia nam mnóstwo funkcji, z których możemy skorzystać po instalacji. Najczęściej używane to :

- Gotowe komponenty (Karty, Nawigacja, Przyciski itp..)
- Budowanie layoutu
- Klasy wspierające (helpers & utilities)
- Gotowe formularze
- Możliwości personalizacji



Do zobaczenia



Fundusze
Europejskie
Program Regionalny



Rzeczpospolita
Polska



URZĄD MARSZAŁKOWSKI
WOJEWÓDZTWA POMORSKIEGO

Unia Europejska
Europejski Fundusz
Rozwoju Regionalnego

