



CSS - Wstawianie Stylów



Wstawianie stylów lokalnie tzw “inline”

Jest to wstawianie stylów które działa jedynie wewnątrz danego elementu html

```
<element style="cecha:wartosc ; cecha2:wartosc2"> ... </element >
```

Przykłady elementów: p , header , footer ,body

Potem do tego znacznika dodajemy atrybut **style** któremu nadajemy pożądane wartości

Przykład :

```
<p style= " width:50px ; height:50px ; color:#000FFF;">  
  <h2>COKOLWIEK</h2>  
</p>
```



Wstawianie stylów bloki

Jest to wstawianie stylów które działa jedynie wewnątrz danego znacznika html

```
<div style="cecha:wartosc ; cecha2:wartosc2"> ... </div >
```

Potem do tego znacznika (div'a) dodajemy atrybut **style** któremu nadajemy pożądane wartości

Przykład :

```
<div style= " width:50px ; height:50px ; color:#000FFF;">
  <h2>COKOLWIEK</h2>
</div>
```




Wewnętrzny arkusz stylów

Umieszczamy go w "głowie" (head) dokumentu html

```
<head> <style >
    { cecha: wartosc; cecha2: wartosc2; (...) }
</style></head>
```

Przykład

```
<head>
  <style >
    p { width: 15px; font-size: large; color: blue; }
    h1 {color: aqua;}
  </style>
</head>
```



Zewnętrzny arkusz stylów

Umieszczamy go w "głowie" (head) dokumentu html

```
<head>  
<link rel="stylesheet" href="style.css"> href - ścieżka do pliku  
</head>
```

W pliku style.css

```
element{  
    cecha: wartosc;  
    (...)  
}
```

Przykład

```
header{  
    margin: auto;  
    padding-left: 0%;  
}
```



Personalizacja Styli

Strony internetowe gromadzą dane na temat użytkowników ,oraz umożliwiają **personalizację** styli poprzez zmiany w ustawieniach.Chodzi o to że w danym konkretnym przypadku na stronie jednego użytkownika dany obiekt będzie wyglądał inaczej niż na stronie wyświetlonej przez innego użytkownika



Bibliografia

<https://adrianwii.pl/files/5e0fb77e66fef294651075.pdf>

https://www.kurshtml.edu.pl/css/styl_lokalny,wstawianie.html


https://www.kurshtml.edu.pl/css/wydzielone_bloki,wstawianie.html

https://www.kurshtml.edu.pl/css/zewnetrzny_arkusz_stylow,wstawianie.html



Dziękuję za uwagę

Wykonał :Wiktor Koczwar



CSS - arkusze stylów



Alternatywny arkusz Stylów

Strona nie musi korzystać tylko z jednego arkusza stylów, jest możliwe aby użytkownik wybrał z wielu stylów dostępnych na stronie

```
<head>  
<link rel="stylesheet" href="style.css" title="default">  
<link rel="stylesheet"  
href="http://www.w3.org/StyleSheets/Core/Alternate"  
title="alternate">  
</head>
```



Import arkusza stylów

```
@import "style.css"
```

albo

```
@import url("style.css") //w miejsce style.css podajemy adres do pliku bądź strony  
internetowej zawierającej arkusz stylów
```



Kaskadowość stylów

Kaskadowość stylów oznacza że wygląd jest wynikową **działania kilku stylów**.

Jak wcześniej zostało przedstawione style mogą być **zewnętrzne** , **wewnętrzne** lub wpisane inlinowo poprzez atrybut **style=""** , ponadto każda z przeglądarek ma własne wbudowane arkusze stylów , a użytkownik może dopisać własne style

Hierarchia arkuszy stylów CSS

1. CSS ustawiony przez użytkownika
2. Style wstawiane lokalnie tzw "Inline" także w bloku
3. Wewnętrzny arkusz stylów w headzie
4. Zewnętrzny arkusz stylów



Łamanie Kaskadowości

Kaskadowość można złamać poleceniem !important

Przykład

W zewnętrznym arkuszu stylów

```
p{ color: blue !important;  
  text-align: right;  
}
```

! to będzie priorytetowo traktowane mimo że ustawimy będącymi wyżej w hierarchi stylami wprowadzanymi lokalnie .Czyli NAPIS będzie koloru niebieskiego

```
<p style="color: black;">NAPIS</p>
```



Bibliografia

<https://kamakaczmarek.net/css-omowienie-i-wstawianie-2/>

<https://www.wpdesk.pl/blog/custom-css/>

<https://www.w3.org/Style/Examples/007/alternatives.pl.html>

https://www.kurshtml.edu.pl/css/kaskadowosc_stylow,wstawianie.html

<https://kodologia.pl/blog/co-oznacza-kaskadowosc-stylow-css>



Dziękuję za uwagę

Wykonał : Wiktor Koczwar

CSS – SELEKTORY, REGUŁY CSS, PROSTE SELEKTORY

Klaudia Sulowska, ETI, gr 2

SELEKTOR

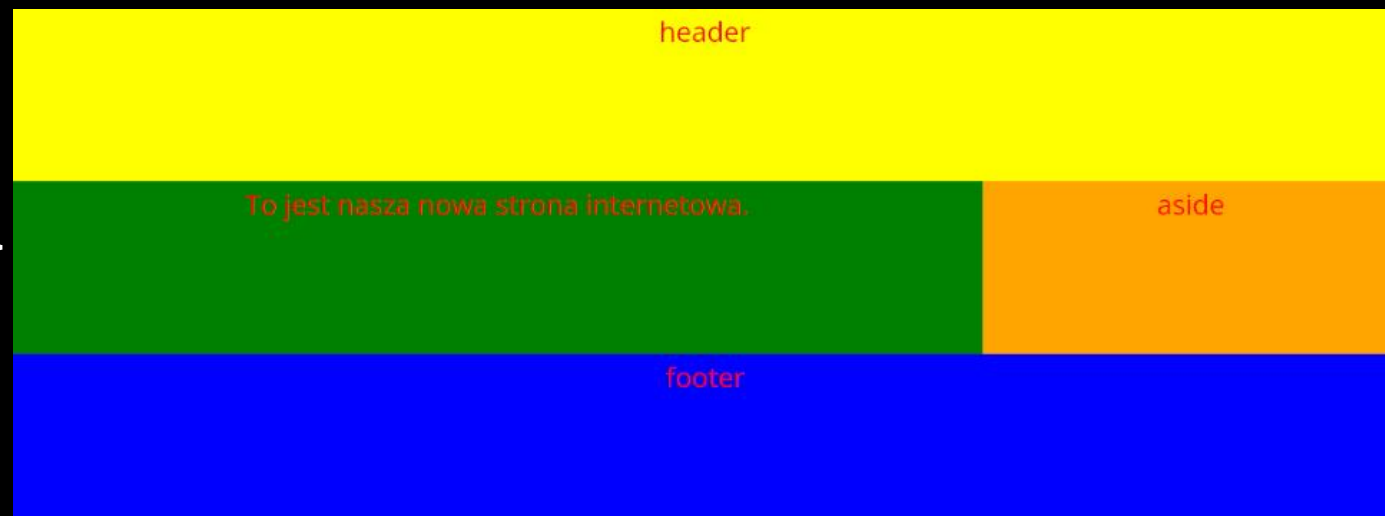
```
selector  
{  
  property: value;  
  property: value;  
  property: value;  
}
```

**wybrany element strony,
którego cechy wpisujemy
pomiędzy klamrami**

REGUŁY

- **Reguły** to instrukcje określające, jakie style i formatowanie powinny zostać zastosowane do określonych elementów na stronie internetowej. Dzięki regułom możemy stworzyć unikalny **styl wyglądu** dla danego elementu HTML bądź też grupy takich elementów HTML.

```
header { background: yellow;  
width:100%; height: 100px; }  
section { background: green;  
width:70%; height: 100px; float: left; }  
aside { background: orange;  
width:30%; height: 100px; float: right; }  
footer { background: blue;  
width:100%; height: 100px; clear:  
both; }  
p { font: 14px Arial, sans-serif; color:  
red; text-align: center; }
```



PRZYKŁAD

```
# style.css > ...
```

```
1  h1
2  {
3      color: ■olive;
4      font-size: 36px;
5      font-weight: bold;
6  }
```

Najlepsze potrawy!

- W wyniku połączenia selektora i deklaracji otrzymujemy pełną regułę CSS. Reguły CSS pozwalają na definiowanie wyglądu i formatowania elementów na stronie internetowej, co jest kluczowe w procesie projektowania stron internetowych.


SELEKTOR TYPU

- Selektor typu jest najprostszym rodzajem selektora i odnosi się do wszystkich elementów o określonym typie. Na przykład, selektor p odnosi się do wszystkich elementów <p> na stronie.

```
1  p {  
2    font-family: Roboto, Arial;  
3    margin-top: 0;  
4    margin-bottom: 0;  
5  }
```

SELEKTOR KLASOWY

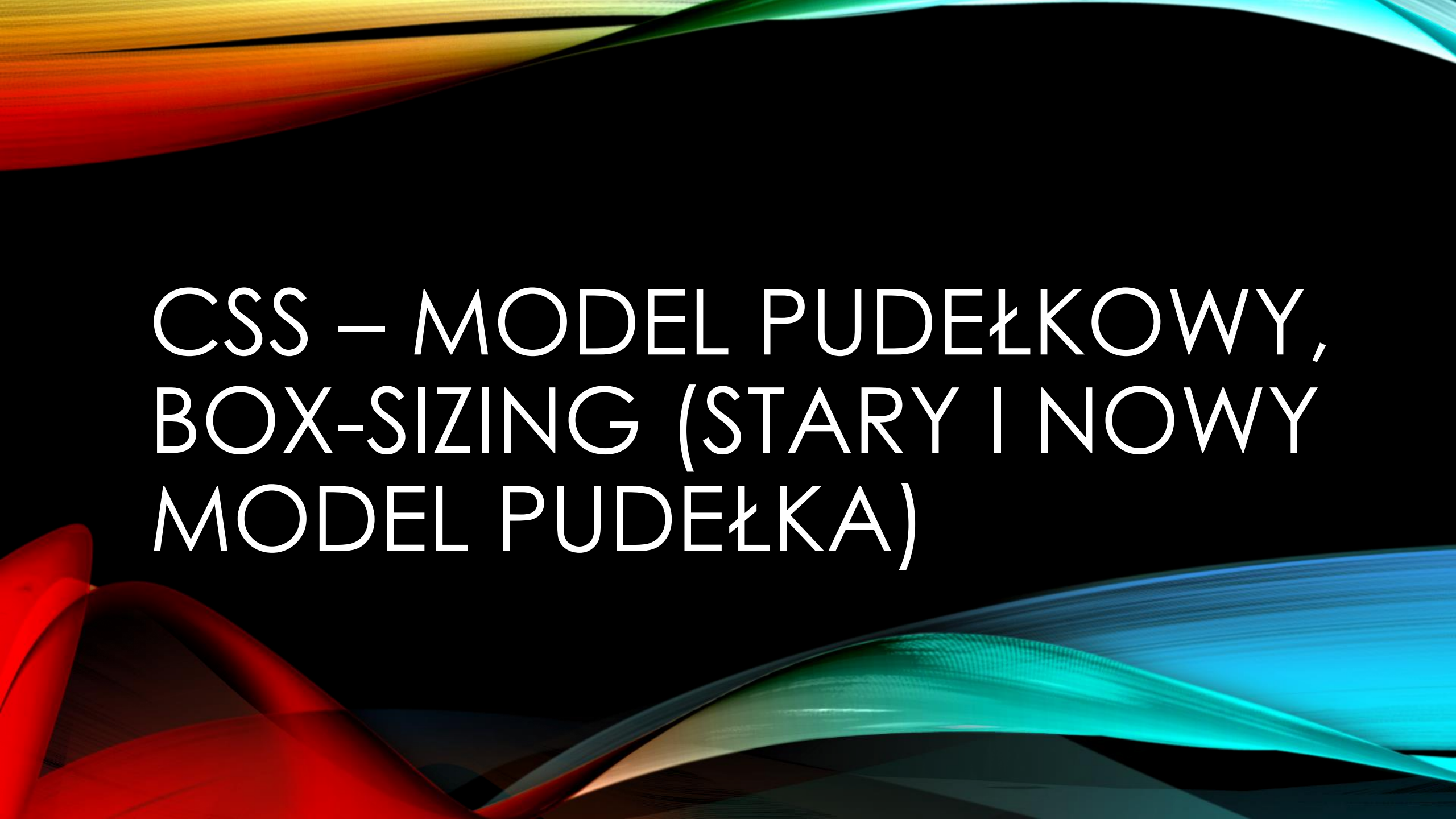
- Selektor klasowy umożliwia wybór elementów na podstawie ich klasy. Można przypisać jednemu lub wielu elementom tę samą klasę i stylizować je w ten sam sposób.

```
1  .logo {
2      font-family: "Acme", sans-serif;
3      font-size: 60px;
4      color:  red;
5      margin-top: 60px;
6      margin-left: 50px;
7      cursor: pointer;
8  }
```

SELEKTOR IDENTYFIKATORA

- Selektor identyfikatora jest używany, aby wybrać pojedynczy element na stronie na podstawie jego identyfikatora. Jest to unikalny identyfikator przypisany tylko jednemu elementowi.

```
2  #content {  
3    font-size: 50px;  
4  }
```

CSS – MODEL PUDEŁKOWY, BOX-SIZING (STARY I NOWY MODEL PUDEŁKA)

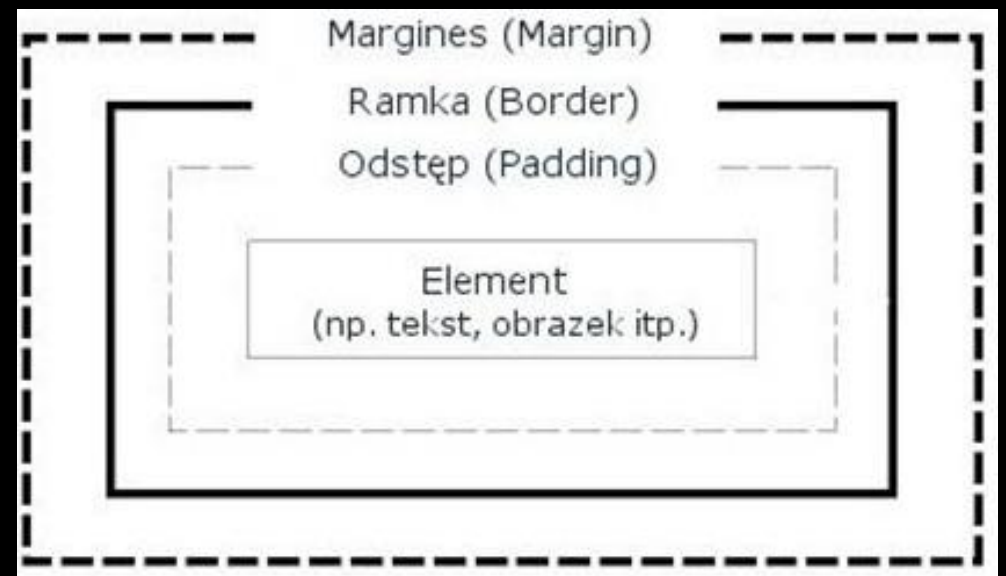
MODEL PUDEŁKOWY

Content (czyli tekst, obrazek itd.)

Padding definiuje odległość pomiędzy zawartością elementu a jego obramowaniem.

Border (jeśli w ogóle jest widoczne) jest wyświetlane wewnątrz odstępu, tuż przy jego zewnętrznych krawędziach.

Margin określa odległość pomiędzy obramowaniem elementu a innymi, otaczającymi go elementami.

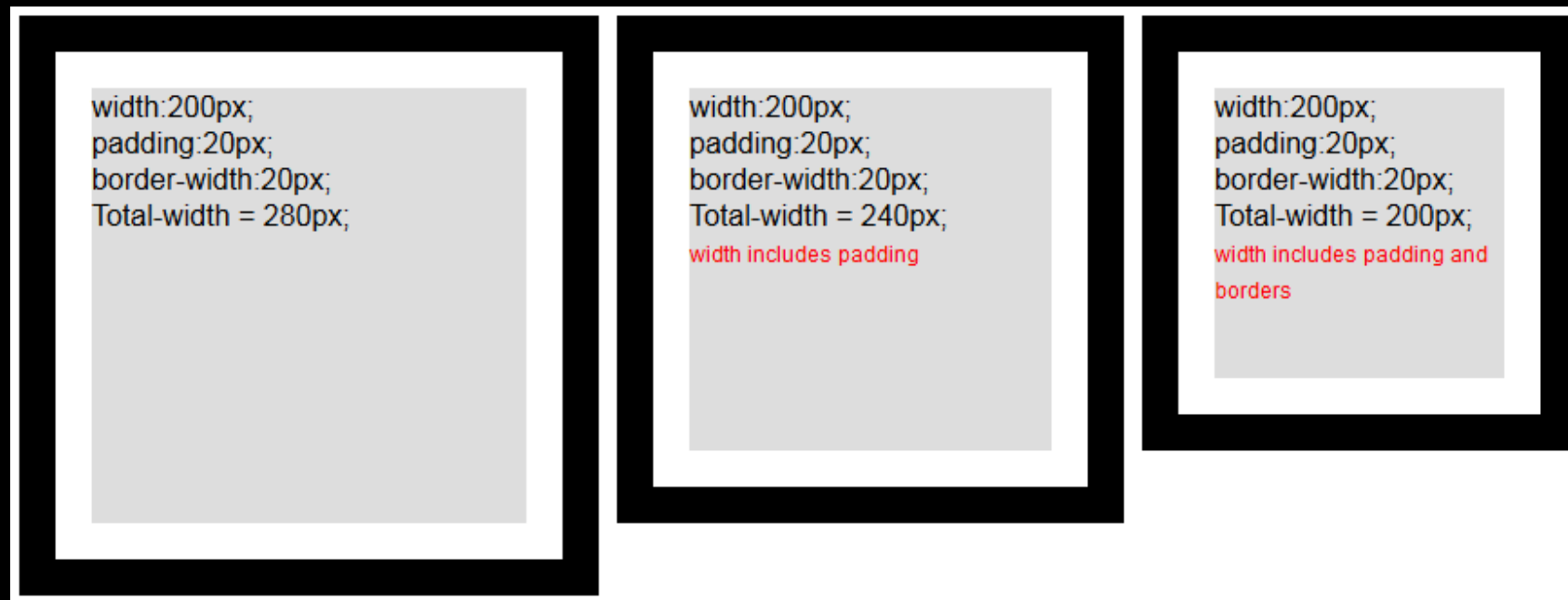


To jest tekst napisany stylem w nagłówku
poziomu pierwszgo z ustalonym
obramowaniem, odstępem od
obramowania i marginesem od
pozostałych elementów stony

To jest tekst napisany w akapice również z ustalonym
obramowaniem, odstępem od obramowania i marginesem od
pozostałych elementów stony

Box-sizing

Pozwala uwzględnić dopełnienie i obramowanie w całkowitej szerokości i wysokości elementu



STARY MODEL PUDEŁKA

```
box-sizing: content-box;
```

szerokość i wysokość elementu obejmuje tylko jego zawartość, a padding, obramowanie i margines są dodawane na zewnątrz tych wymiarów.

W tym trybie modelu pudełka CSS zachowuje się w sposób bardziej zgodny z przeglądarkami sprzed przyjęcia nowszych standardów CSS. Co może prowadzić do różnych nieoczekiwanych rezultatów w wyświetlaniu stron internetowych, zwłaszcza w przypadku starszych stron, które nie są zaktualizowane do nowszych standardów.

NOWY MODEL PUDEŁKA

```
box-sizing: border-box;
```

szerokość i wysokość elementu obejmuje zarówno jego zawartość, jak i padding i obramowanie. Marginesy są dodawane na zewnątrz tych wymiarów.

Nowy model jest bardziej logiczny i bardziej zgodny z intuicją. Ten tryb jest często używany w projektowaniu stron internetowych, ponieważ ułatwia kontrolę nad rozmiarem elementów i eliminuje nieoczekiwane zmiany w układzie elementów spowodowane dodawaniem paddingu i obramowania.



PRZYKŁADY ZASTOSOWAŃ MODELU PUDEŁKOWEGO

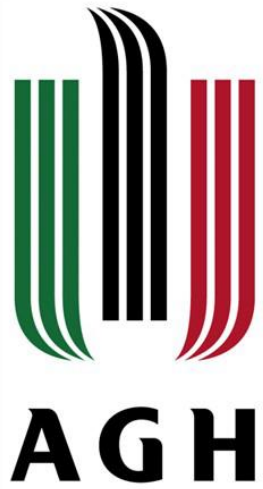
- **Układ strony**- rozmieszczanie elementów HTML na stronie;
- **Tworzenie przycisków**- wyraźne, spójne;
- **Obramowanie zdjęć**- wyróżnienie obrazów na stronie;
- **Kontrola odstępów**- równomierne odstępy między sekcjami strony;
- **Kontrola rozmiaru tekstu**- czytelny układ tekstu;
- **Kontrola elementów interaktywnych**- dotyczy: przycisków, pól formularza, pól wyboru itp.;
- **Kontrola marginesów**- ogólny układ i estetyka strony;
- **Responsive Web Design (RWD)**- dostosowanie się do różnych rozmiarów ekranów urządzeń.

BIBLIOGRAFIA

- <https://webkod.pl/kurs-css/lekcje/dzial-1/idea-stylow-css> (24.10.2023)
- https://www.w3schools.com/cssref/css_selectors.php (24.10.2023)
- https://zs7.walbrzych.pl/HTML-CSS-JS/css_selektory.html (24.10.2023)
- https://webroad.pl/html5-css3/3713-jak-pisac-deklaracje-css?fbclid=IwAR0bXGX0og1JJkXZO7XBHWP7RDU2dJLHuXI8A0n_m1aO4vXEi5-LmVgvacQ (24.10.2023)
- <https://developer.mozilla.org/en-US/docs/Web/CSS/box-sizing?retiredLocale=pl> (25.10.2023)
- https://www.w3schools.com/css/css3_box-sizing.asp (25.10.2023)
- <https://webkod.pl/kurs-css/wlasciwosci/rozmiary/box-sizing> (25.10.2023)

gra - selektory:

<https://flukeout.github.io/?fbclid=IwAR3nqIKm87fNmPNUNxTz9afbFnua6qnpOW3JttpoKqVL2DKI1Ntpl8mQWUY#>



Dziedziczenie stylów, selektory specjalne

Radosław Heide ETI

Dziedziczenie



W css, dziedziczenie kontroluje co się stanie z elementami które nie mają ustalonych własnych ustawień.

```
1 <body>
2
3   <div>
4     <p>pierwszy element - p <span>to jest element - span</span></p>
5   </div>
6
7   <p>drugi element - p</p>
8
9 </body>
```

```
1 body {
2   color:red;
3   border:1px solid black;
4 }
```

pierwszy element - p to jest element - span

drugi element - p

Właściwości dziedziczone



Właściwości dziedziczone, które domyślnie są ustawiane na wartość z elementu rodzica:

- color,
- font-family,
- font-size,
- font-style,
- font-weight,
- text-align,

Właściwości niedziedziczone



Właściwości niedziedziczone, które domyślnie posiadają wartość początkową (domyślną):

- background,
- border,
- margin,
- padding,
- width,
- height

Dodatkowe ustawienia



- inherit – wymusza dziedziczenie wartości domyślnych elementu rodzica,
- unset – ustawia wartość na dziedziczną, jeśli taka nie istnieje to wartość domyślną elementu,
- revert – przyjmuje wartość z elementu rodzica niezależnie czy jest to wartość domyślna czy ustawiona ręcznie

Selektory specjalne



- Klasy to wartości atrybutów `class=""` znajdującego się wewnątrz danego znacznika. Dopuszczalne jest stosowanie dowolnej ilości tych samych klas w całym dokumencie html. Różne znaczniki mogą mieć te same klasy. Dzięki czemu możemy wszystkim elementom posiadającym dany atrybut nadać te same właściwości (`.klasa={}`),
- Identyfikatory to wartości atrybutów `id=""` znajdującego się wewnątrz danego znacznika. W całym dokumencie html dany identyfikator może wystąpić tylko raz. Każdy identyfikator na stronie musi być unikalny (`#ajdi={}`)

Selektory pseudoklas



- a:active -link po kliknięciu i przytrzymaniu lewego klawisza myszki
- a:hover -link po najechaniu myszką

a:visited - odwiedzony link

- a:link - normalny link

to jest przykładowy link

to jest przykładowy link

to jest przykładowy link

```
a.pseudoklasy:link {
    color:blue;
}

a.pseudoklasy:visited {
    color:red;
}

a.pseudoklasy:hover {
    color:green;
}

a.pseudoklasy:active {
    font-weight:bold;
    background:black;
    color:white;
    padding:3px;
}
```

Selektory pseudoelementów




- Pierwsza litera i pierwsza linia:

```
p.psClass:first-letter{  
    color:#b900b9;  
    font-size:20px;  
}  
  
p.psClass:first-line {  
    color: violet;  
    font-weight:bold;  
}
```

Lorem ipsum dolor sit amet enim.
felis, malesuada ultricies. Curabitur et

- Przed (:before) i po (:after)

```
b.exBefore:before{  
    content: "PRZED " url(../images/effects_images/ex_info.png) " ";  
    color:orange;  
}  
  
b.exAfter:after {  
    content: " atrybut Title to: " attr(title) ". ";  
    color: red;  
    font-size:11px;  
    font-weight:normal;  
}
```

PRZED  **Tekst pierwszy**
Tekst drugi atrybut Title to: Lorem ipsum.



- <https://webkod.pl/kurs-css/lekcje/dzial-1/dziedziczenie-wlasciwosci-css> (dostęp 25.10.2023r),
- https://developer.mozilla.org/en-US/docs/Web/CSS/Inheritance#non-inherited_properties (dostęp 25.10.2023r),
- <https://www.codecademy.com/resources/docs/css/inheritance> (dostęp 25.10.2023r),
- <https://michallukasik.pl/tutorialcss/index.php?k=selektory,wstep> (dostęp 25.10.2023r),



Dziękuję za uwagę



CSS

selektory elementów, grupowanie selektorów, selektory atrybutów i
własności czcionki, jednostki, własności tekstu, własności tła

Sebastian Chojnowski ETI WIMIP 3 rok 1 grupa

Selektory elementów

- W CSS elementy wybierać ich nazwy, w stylu body, button, table itp. Po zrobieniu tak wszystkie elementy wybranego typu będą miały określone cechy np jeżeli napiszemy:
-
- button{
- Background-color: red;
- }
-
- to wszystkie elementy typu button będą miały w tle kolor czerwony

Grupowanie selektorów

- Możemy wymieniać elementy po przecinku:
-
- `h1, h2, p {`
- `color: red;`
- `}`
-
- to sprawi że wszystkie elementy `h1` i `h2` i `p` będą miały swój tekst pokolorowany na czerwono

Selektory atrybutów

- Do elementów w html możemy dodawać atrybuty, np do <a> możemy dodać "target" który określa gdzie przeniesie nas link i gdy zrobimy:
-
- `a[target] {`
- `background-color: yellow;`
- `}`
- to wszystkie elementy <a> z atrybutem target będą miały tło w kolorze żółtym. Można zrobić dać też `[attribute="value"]` co będzie dotyczyło tylko elementów których parametr target wynosi _blank. `[attribute~="value"]` oznacza że warość musi w sobie zawierać słowo value czyli w to się łąpie np `atribute="more values,,`. `[attribute^="value"]` zaczyna się na value i `[attribute$="value"]` kończy się na value

Własności czcionki

- font: *italic* **small-caps** **bold** 12px/30px Georgia, serif;
- font-style - czcionka
- font-variant – w jakich literach ma być tekst
- font-weight – pogrubienie, pochylenie itp
- font-size/line-height – rozmiar czcionki/odległość między linijkami
- font-family – rodzina czcionki, jeżeli czcionka (ta wymieniona w pierwszym punkcie) jest niezainstalowana to strona będzie brała z podanej rodziny czcionek

Jednostki

- Px - pixele
- % - procent wielkości zbiornika w którym jest tekst
- Em – relatywna jednostka, nadaje wielkość czcionki zależną od pierwotnego rozmiaru (2em oznacza 2-krotność rozmiaru bieżącej czcionki)
- Vh – Visible Height, czyli czcionka będzie zależna od widocznej wysokości elementu
- Vw – Visible Width, czyli czcionka będzie zależna od widocznej szerokości elementu
- cm,mm – normalnie jednostki si

Własności tekstu

- Color zmienia kolor czcionki
- Background tło czcionki
- text-align: ułożenie tekstu
- vertical-align: ułożenie pionowe tekstu
- text-decoration-line: linia która może przekreślać tekst albo go podkreślać albo iść nad nim
- text-decoration-color: kolor wcześniej wspomnianej linii
- text-decoration-style: wtył ww lini

text-decoration-thickness: grubość ww lini

text-transform: sprawia że tekst może być cały duży albo mały, zmusza tekst do bycia capslocowany

text-indent: wielkość wcięcia akapitu

letter-spacing: odległość między literami

line-height: odległość między linijkami

word-spacing: odległość między słowami

text-shadow: tworzy cień

Własności tła

- background-color – kolor tła
- background-image – zdjęcie w tle
- background-repeat – czy zdjęcie w tle ma się powtarzać
- background-attachment – czy ma się zdjęcie przesuwać wraz z przesuwaniem strony?
- background-position – pozycja zdjęcia w tle
- background-size – rozmiar zdjęcia
- background-origin – gdzie ma się zaczynać?

Literatura:

https://www.w3schools.com/css/css_font.asp (dostęp 26.10.2023)

<https://www.youtube.com/watch?v=QgxkYbGr2II> (dostęp 26.10.2023)

https://www.w3schools.com/css/css_background.asp (dostęp 26.10.2023)

<https://www.geeksforgeeks.org/css-fonts/> (dostęp 26.10.2023)

https://www.w3schools.com/cssref/css_units.php (dostęp 26.10.2023)

Koniec



CSS



CSS - szczegółowość i łączenie selektorów

Jakub Niemczyk

Identyfikatory i klasy

Identyfikatory zazwyczaj dodajemy do znaczników, które są unikalne(chcemy ostylewać tylko jeden dany tag), podczas gdy jedną klasę stosujemy w kilku znacznikach, które mają mieć te same style

```
<body>
  <nav id="menu">
    <ul class="lista">
      <li class="element-listy">
        <a href="#">
          Opcja 1
        </a>
      </li>
      <li class="element-listy">
        <a href="#">
          Opcja 2
        </a>
      </li>
    </ul>
  </nav>
  <ul class="lista">
    <li class="element-listy">
      1 element listy
    </li>
    <li class="element-listy">
      2 element listy
    </li>
  </ul>
</body>
```

Sposoby i priorytet stylów

CSS

```
9 <nav style="color: white">  
10
```

```
8 <style>  
9   #menu{  
10    color: white;  
11  }  
12 </style>  
13 </head>  
14 <nav id="menu">  
15  
16 </nav>
```

```
8 <link rel="stylesheet" href="style.css">  
9 </head>  
10 <nav id="menu">  
11  
12 </nav>  
13 </body>  
14 </html>
```

```
style.css  
C: > Users > kubik > Desktop > style.css > #menu  
1  #menu{  
2    color: white;  
3  }
```

Priorytety

Największy priorytet mają style wpisane w znacznik, następnie, te w tagu `<style>` a na końcu te z zewnętrznego pliku `.css`



Jak łączyć selektory?

Do łączenia selektorów służą specjalne znaki:

→ **A B**

Spacja - wszystkie selektory B, zawarte w A będą miały dany styl

→ **A > B**

Wszystkie "dzieci" "B", selektora "A" będą miały dany styl

→ **A + B**

Selektory "B" sąsiadujące z "A" będą miały dany styl

→ **A ~ B**

Wszystkie selektory "B" będące "rodzeństwem" z "A", będą miały dany styl

```
1  ✓ div p{
2    |   background-color: ■ white;
3    | }
4
5  ✓ div > p{
6    |   background-color: ■ white;
7    | }
8
9  ✓ div + p{
10   |   background-color: ■ white;
11   | }
12
13 ✓ div ~ p{
14   |   background-color: ■ white;
15   | }
```



Przykład


1. p zawarte w div
2. p jest dzieckiem diva
3. p sąsiadujące z divem
4. p i div są rodzeństwem



Pseudoklasy

Pseudoklasy dodajemy po selektorze, poprzedzając dwukropkiem:

- **::link**
Nieodwiedzony nigdy link
- **::visited**
Już odwiedzony wcześniej link
- **::hover**
Gdy najedziemy myszką
- **::active**
Wybrany link



**Kolejność, jeśli
chcemy użyć więcej
niż jednej pseudoklasy**

1. **::link i ::visited musi być przed ::hover**
2. **::hover musi być przed ::active**



Dziękuję za uwagę!

Własności Elementów w CSS

(obramowanie, obrys, wykaz, tabela, pozycjonowanie, kursor, drukowanie)

Edukacja Techniczno-informatyczna rok III
Piotr Krypel

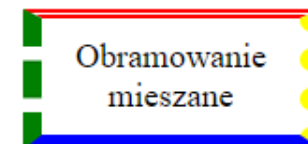
Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie
AGH University of Science and Technology

Własności – Obramowanie (border)

Własność	Przykładowy zapis w CSS
Styl obramowania	<code>border-style: (solid / double / dashed / groove / dotted / ridge / inset / outset);</code>
Szerokość	<code>border-width: 5px;</code>
Kolor	<code>border-color: rgb(255, 0, 0);</code>
Zaokrąglenie	<code>border-radius: 10px;</code>
Zdjęcie jako ramka (zastępuje kolor)	<code>border-image-source: url('obraz.png');</code>

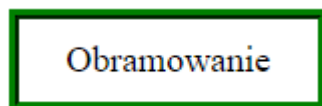
- Własności obramowania można kontrolować dla każdej krawędzi osobno.

```
div{
  border-top: 5px double red;
  border-bottom: 8px solid blue;
  border-left: 10px dashed green;
  border-right: 12px dotted yellow;
}
```

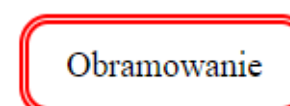


- „border:” pozwala jednocześnie zdefiniować grubość, styl i kolor obrysu.

```
div{
  border: 5px ridge green;
}
```



```
div{
  border: 5px double red;
  border-radius: 15px;
}
```

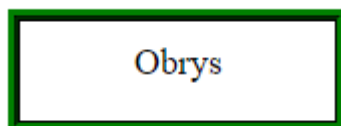


Własności – Obrys (outline)

Własność	Przykładowy zapis w CSS
Styl obrysu	<code>outline-style: (solid / double / dashed / groove / dotted / ridge / inset / outset);</code>
Szerokość	<code>outline-width: 5px;</code>
Kolor	<code>outline-color: rgb(255, 0, 0);</code>
Odstęp obrysu	<code>outline-offset: 10px;</code>

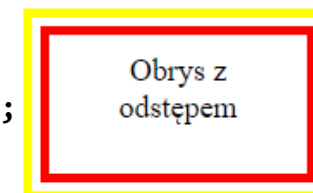
- „**outline:**” pozwala jednocześnie zdefiniować grubość, styl i kolor obrysu.

```
div{
  outline: 5px ridge green;
}
```



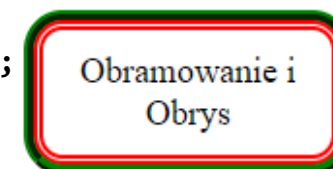
- Przykład `outline-offset`

```
div{
  border: 5px solid red;
  outline: 5px solid yellow;
  outline-offset: 5px;
}
```



- Obrys dopasowuje się do kształtu elementu.

```
div{
  outline: 5px ridge green;
  border: 5px double red;
  border-radius: 15px;
}
```



Obramowanie a obrys

Różnice pomiędzy obramowaniem a obrysem:

- Obrys nie zajmuje miejsca w modelu pudełkowym i dlatego nie wpływa na rozmiar ani układ wizualny elementów.

border:

eniam, quis nostrud exercitation:
a commodo consequat. Duis au
rit in **Przykład** volu
giat nulla pariatur. Excepteur s
ut in culpa qui officia deserunt

outline:

dolor sit amet, consectetur adi
didunt ut labore et dolore mag
qui **Przykład** nosti
liquip ex ea commodo conseq
erit in voluptate velit esse cilli

- Obrys nie musi być prostokątem.
- Obrys jest tworzony zawsze na wierzchu elementu.
- Można ustawiać atrybuty obrysu tylko w stosunku do wszystkich boków jednocześnie (nie da się tego robić oddzielnie dla każdego).

Własności – Wykaz (list)

Własność	Przykładowy zapis w CSS
Rodzaj (styl) wykazu	<code>list-style-type: (disc / circle / decimal / lower-alpha / upper-alpha / upper-roman / lower-roman);</code>
Zwijanie tekstu	<code>list-style-position: (inside / outside);</code>
Wstawienie grafiki jako punkt wykazu	<code>list-style-image: url(sciezka dostępu)</code>

- „list-style:” pozwala jednocześnie zdefiniować typ stylu oraz zawijanie tekstu.

```
ul{
  list-style: circle inside;
}
```

◦ Pierwszy element listy...Pierwszy element listy...Pierwszy element listy...

◦ Drugi element listy...Drugi element listy...Drugi element listy...

◦ Trzeci element listy...Trzeci element listy...Trzeci element listy...

Przykłady

list-style-type: square

- Punkt pierwszy
- Punkt drugi
- Punkt trzeci

list-style-type: decimal

1. Punkt pierwszy
2. Punkt drugi
3. Punkt trzeci

list-style-type: circle

- Punkt pierwszy
- Punkt drugi
- Punkt trzeci

list-style-type: upper-roman

- I. Punkt pierwszy
- II. Punkt drugi
- III. Punkt trzeci

list-style-type: lower-alpha

- a. Punkt pierwszy
- b. Punkt drugi
- c. Punkt trzeci

list-style-type: upper-alpha

- A. Punkt pierwszy
- B. Punkt drugi
- C. Punkt trzeci

list-style-type: none

- Punkt pierwszy
- Punkt drugi
- Punkt trzeci

list-style-image: url(minus.gif)

- ⊖ Punkt pierwszy
- ⊖ Punkt drugi
- ⊖ Punkt trzeci

list-style-position: outside

- Punkt pierwszy... punkt pierwszy... punkt pierwszy... punkt pierwszy... punkt pierwszy...
- Punkt drugi... punkt drugi... punkt drugi... punkt drugi... punkt drugi...
- Punkt trzeci... punkt trzeci... punkt trzeci... punkt trzeci... punkt trzeci...

list-style-position: inside

- Punkt pierwszy... punkt pierwszy... punkt pierwszy... punkt pierwszy... punkt pierwszy...
- Punkt drugi... punkt drugi... punkt drugi... punkt drugi... punkt drugi...
- Punkt trzeci... punkt trzeci... punkt trzeci... punkt trzeci... punkt trzeci...

Własności – Tabela

Własność	Przykładowy zapis w CSS
Rozmiar tabeli	width: 500px height: 300px
Podpis tabeli	caption-side: (top / bottom / left / right);
Obramowanie tabeli i komórek	border-collapse: (collapse / separate);
Rozplanowanie tabeli	table-layout: (auto / fixed);
Odstęp między komórkami	border-spacing: (5mm / 5mm 8mm);
Puste komórki tabeli	empty-cells: (show / hide);

table-layout: fixed; width: 100%

komórka1	Wszystkie kolumny tej tabeli mają jednakową szerokość	komórka3
komórka4	komórka5	komórka6
komórka7	komórka8	komórka9

table-layout: auto; width: 100%

komórka1	Szerokość komórek zależy od zawartości	komórka3
komórka4	komórka5	komórka6
komórka7	komórka8	komórka9

- Dla obramowania tabeli można zastosować te same własności co w przypadku obramowania elementu <div>

```
table{
  border: 5px dashed red;
  border-collapse: separate;
}
td{
  border: 3px solid green;
}
```

Rząd 1, Komórka 1	Rząd 1, Komórka 2
Rząd 2, Komórka 1	Rząd 2, Komórka 2
Rząd 3, Komórka 1	Rząd 3, Komórka 2

```
table{
  border: 5px dashed red;
  border-spacing: 2px 8px;
}
td{
  border: 3px solid green;
}
```

Rząd 1, Komórka 1	Rząd 1, Komórka 2
Rząd 2, Komórka 1	Rząd 2, Komórka 2
Rząd 3, Komórka 1	Rząd 3, Komórka 2

empty-cells: show;

Rząd 1, Komórka 1	Rząd 1, Komórka 2
Rząd 2, Komórka 1	
Rząd 3, Komórka 1	Rząd 3, Komórka 2

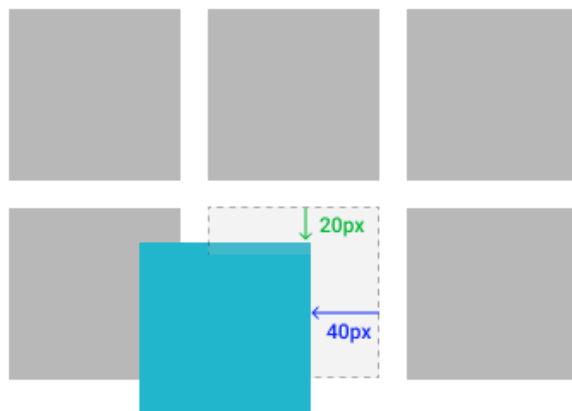
empty-cells: hide;

Rząd 1, Komórka 1	Rząd 1, Komórka 2
Rząd 2, Komórka 1	
Rząd 3, Komórka 1	Rząd 3, Komórka 2

Pozycjonowanie

- Pozycja relatywna (position: relative)

```
div{
  position: relative;
  right: 40px;
  top: 20px;
}
```

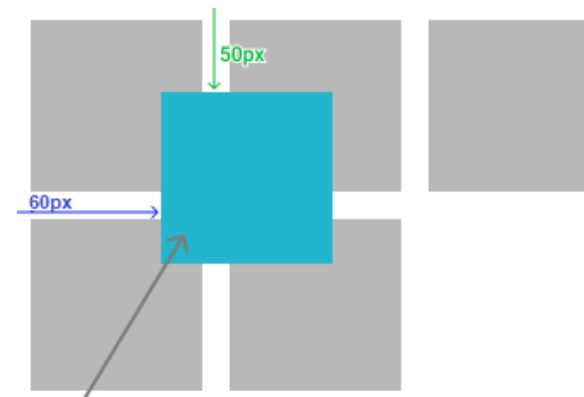


Pozycjonowanie relatywne (względne) pozwala przesunąć wybrany element w dowolną stronę względem jego położenia spoczynkowego, czyli pozycji gdy nie używamy żadnego pozycjonowania. (position: static; – wartość domyślna).

Sąsiadujące obiekty HTML zachowują się tak, gdyby obiekt nadal znajdował się w oryginalnej pozycji.

- Pozycja absolutna (position: absolute)

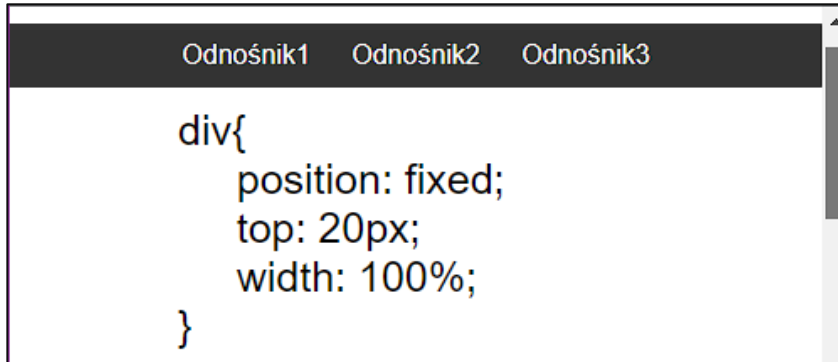
```
div{
  position: absolute;
  left: 60px;
  top: 50px;
}
```



Pozycjonowanie absolutne pozwala przesunąć wybrany element w dowolną stronę względem najbliższego elementu z pozycją inną niż „static”. Jeżeli taki element nie istnieje to przesuwany się względem granic strony <html>.

Element o pozycji "absolute" jest niezależny od położenia innych elementów HTML, oraz inne elementy go nie uwzględniają.

- Pozycja ustalona (position: fixed)



Pozycja ta działa podobnie jak pozycjonowanie absolutne, z tą różnicą, że pozycja elementu jest zawsze obliczana względem krawędzi okna przeglądarki.

Element z pozycją „fixed” jest nieruchomy podczas przewijania strony i widoczny jest na ekranie ciągle w tym samym miejscu.

- Pozycja statyczna (position: static)

```

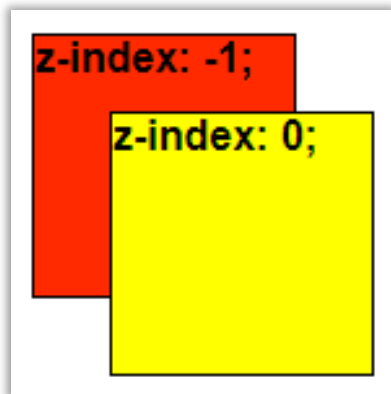
div{
  position: static;
}
```



Przywraca oryginalne pozycjonowanie elementu. Jest to przydatne np. w celu usunięcia ogólnej deklaracji pozycjonowania, umieszczonej w arkuszu stylów.

- Nakładanie (z-index)

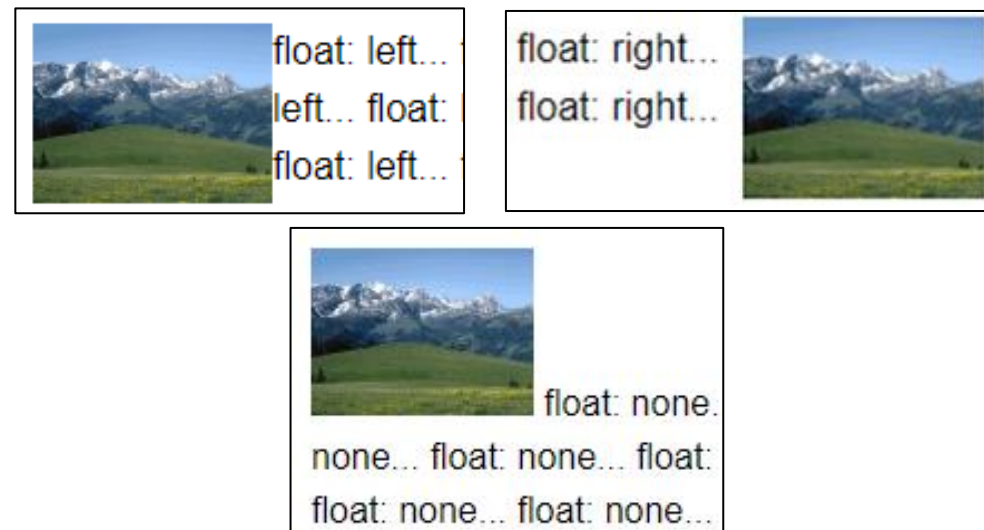
```
.redSquare{
  position: absolute;
  top: 10px;
  left: 10px;
  z-index: -1;
}
.yellowSquare{
  position: absolute;
  top: 40px;
  left: 40px;
  z-index: 0;
}
```



Polecenie „z-index” pozwala określić, kolejność pozycjonowania elementów. Normalnie elementy które zostały wpisane na końcu, zakrywają te poprzednie. Dzięki z-index możemy to dowolnie zmieniać.

„z-index” możemy zastosować, tylko jeśli element jest poddany pozycjonowaniu.

- Ustawienie (float)

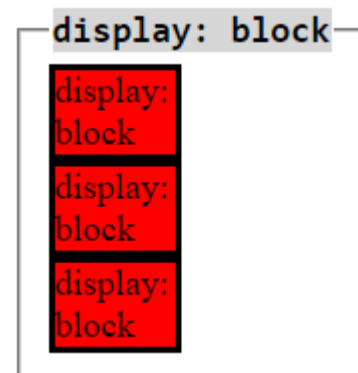
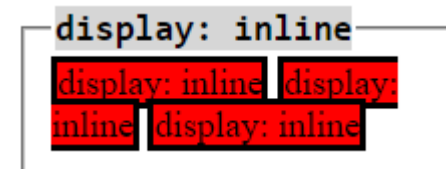
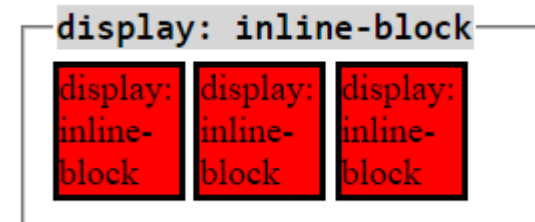


Polecenie „float” pozwala zdefiniować ustawienie danego elementu względem elementów, które z nim sąsiadują. Dzięki temu możemy ustalić sposób "oblewania" tekstem elementu pływającego (np. obrazka).

- Wyświetlanie (display)

Polecenie to pozwala określić sposób, w jaki będzie wyświetlony na ekranie dany element. Jest to bardzo przydatna funkcja, jeśli chcemy zrezygnować z domyślnego sposobu wyświetlania.

























Wartość	Opis
block	Element jest wyświetlany w bloku z odstępem na górze i na dole.
inline	Element jest wyświetlany w linii, sąsiadując z innymi elementami.
inline-block	Element zachowuje się jak element zastępowany, w linii.
none	Element nie jest wyświetlany na stronie.
table-cell	Element jako komórka tabeli, np. <code><td>...</td></code> , <code><th>...</th></code> .



Kursor

Własność	Przykładowy zapis w CSS
Kształt (styl) kursora	<code>cursor: pointer;</code>
Kursor obrazkowy	<code>cursor: url(obrazek.png);</code>
Kolor wskaźnika tekstowego	<code>caret-color: red;</code>

Rodzaje kursorów

 auto	 move	 no-drop	 col-resize
 all-scroll	 pointer	 not-allowed	 row-resize
 crosshair	 progress	 e-resize	 ne-resize
 default	 text	 n-resize	 nw-resize
 help	 vertical-text	 s-resize	 se-resize
 inherit	 wait	 w-resize	 sw-resize

Możliwe jest również podanie listy kursorów rozdzielonych przecinkami. W takim przypadku wczytany zostanie pierwszy z listy plik, który przeglądarka potrafi obsłużyć.

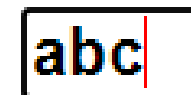
```
div{
  cursor: url(example.svg), url(hyper.cur), pointer;
}
```

Kursor myszy ma punkt docelowy zwany hotspotem. Można go zmienić jego pozycję w kursorze, określając współrzędne x i y obrazku kursora.

```
div{
  cursor: url(moj-kursor.cur) 6 6, auto;
}
```

Przykład zmiany koloru wskaźnika tekstowego

```
input{
  caret-color: red;
}
```




Drukowanie

- Drukowanie w CSS polega na dostosowaniu sposobu, w jaki zawartość strony internetowej jest prezentowana w wersji wydrukowanej.


Wartość	Opis
page-break-before	Określa, czy przed danym elementem ma wystąpić podział strony przed wydrukiem.
page-break-after	Określa, czy po danym elemencie ma wystąpić podział strony po wydruku.
page-break-inside	Określa, czy dany element może być podzielony między stronami w czasie drukowania.
page-break-before: always;	Powoduje, że podział strony zawsze nastąpi przed danym elementem.
page-break-after: avoid;	Stara się uniknąć podziału strony po danym elemencie.
@page	Pozwala na dostosowanie stylów dla konkretnych stron, takich jak strona pierwsza, ostatnia lub konkretne sekcje dokumentu.

Przykład

```
section {
  page-break-inside: always;
}
```



```
section {
  page-break-after: always;
}
```



Bibliografia

- <https://www.kurshtml.edu.pl/css> (dostęp 23.10.2023)
- <https://www.w3schools.com/css/default.asp> (dostęp 23.10.2023)
- <https://webdevschool.pl/post/pozycja-static-relative-absolute/> (dostęp 24.10.2023)
- <https://www.webcodegeeks.com/css/css-cursor-pointer-example/> (dostęp 25.10.2023)

Frameworki

Frameworki to narzędzia lub szkielety programistyczne, które są używane do ułatwienia procesu tworzenia oprogramowania.

Frameworki dostarczają gotowych struktur, bibliotek, komponentów oraz reguł, które pozwalają programistom rozwijać aplikacje lub projekty oprogramowania bardziej efektywnie, ponieważ eliminują konieczność pisanie od zera wielu podstawowych funkcji.

Wybrane Frameworki

Tailwind

Blazor

Tailwind

Tailwind jest frameworkiem służącym do łatwiejszego, projektowania stron zarówno pod względem struktury jak i wyglądu, umożliwia on zdefiniowanie właściwości klasy bez użycia CSS, lecz poprzez wpisanie ich w samą nazwę klasy diva

Wybrane funkcje Tailwinda

- **text-color-size**
- **Font-bold**
- **Dark:text-white**

Blazor

- Pozwala na ułatwienie zarządzania i wyglądu aplikacji webowych, przy użyciu języka C# z wykorzystaniem platformy .net, co może współgrać z Javascriptem lub całkowicie go wyeliminować, ponadto komponenty utworzone w Blazorze działają na każdej innej platformie czy systemie operacyjnym

GRID

- Grid polega na podzieleniu diva na siatkę którą możemy łączyć w osobne bloczki i manipulować nimi

strona.html X

style.css

strona.html > html > head > div.container > div.item-4

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Document</title>
7      <link rel="stylesheet" href="style.css">
8
9      <div class = "container">
10         <div class = "item-1">1</div>
11         <div class = "item-2">2</div>
12         <div class = "item-3">3</div>
13
14     </div>
15
16
17 </head>
18 <body>
19
20 </body>
21 </html>
```

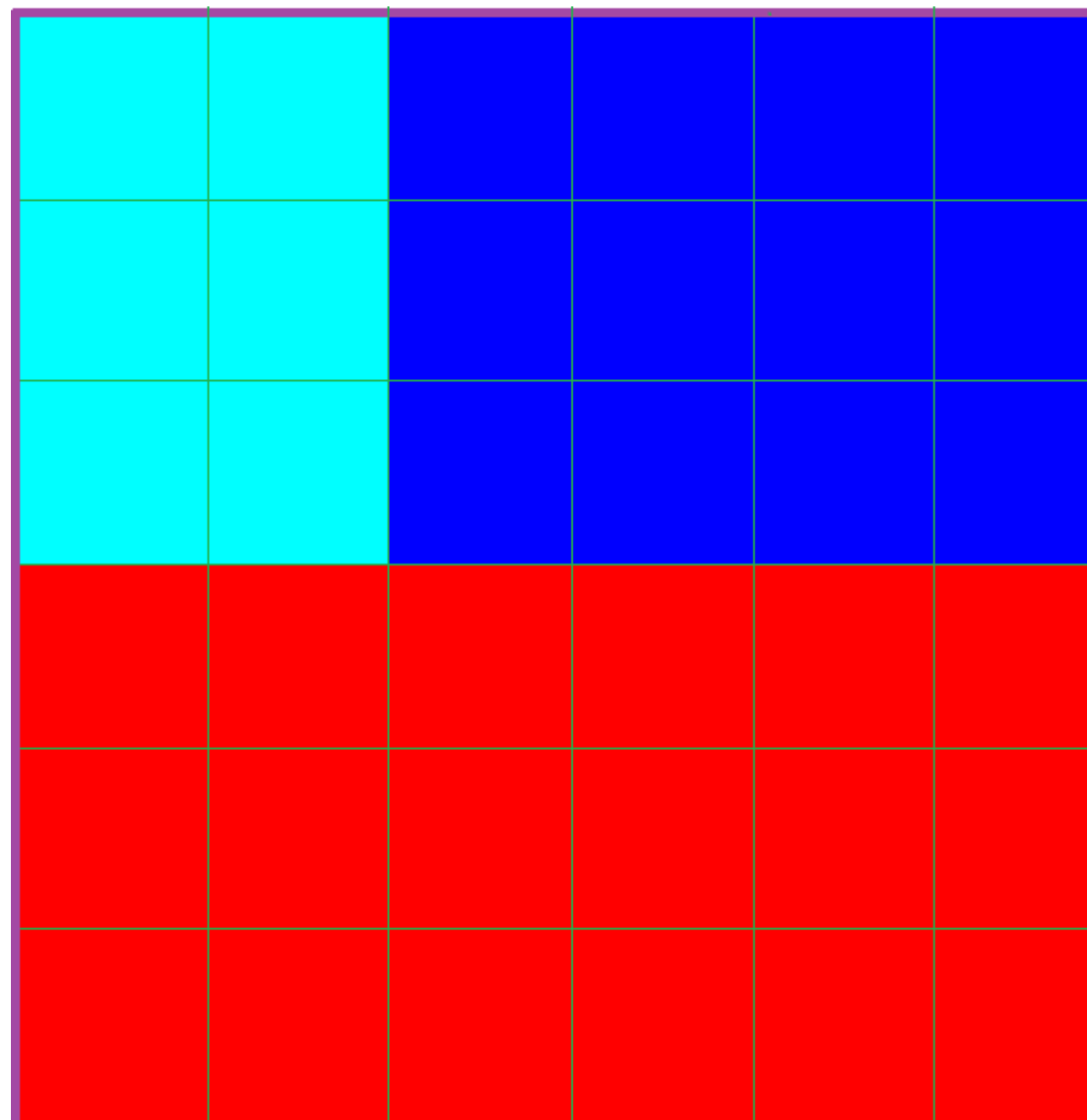
Display: grid;

```
< strona.html  # style.css  X
# style.css > .container
1
2  .container{
3    display: grid;
4    grid-template-rows: 100px 100px 100px 100px 100px 100px;
5    grid-template-columns: 100px 100px 100px 100px 100px 100px;
6
7  }
8
9  .item-1{
10   background-color: aqua;
11 }
12
13 .item-2{
14   background-color: blue;
15 }
16
17 .item-3{
18   background-color: red;
19 }
20 }
```

1 1	2 2	3 3			6
2					
3					
6					

Grid-area

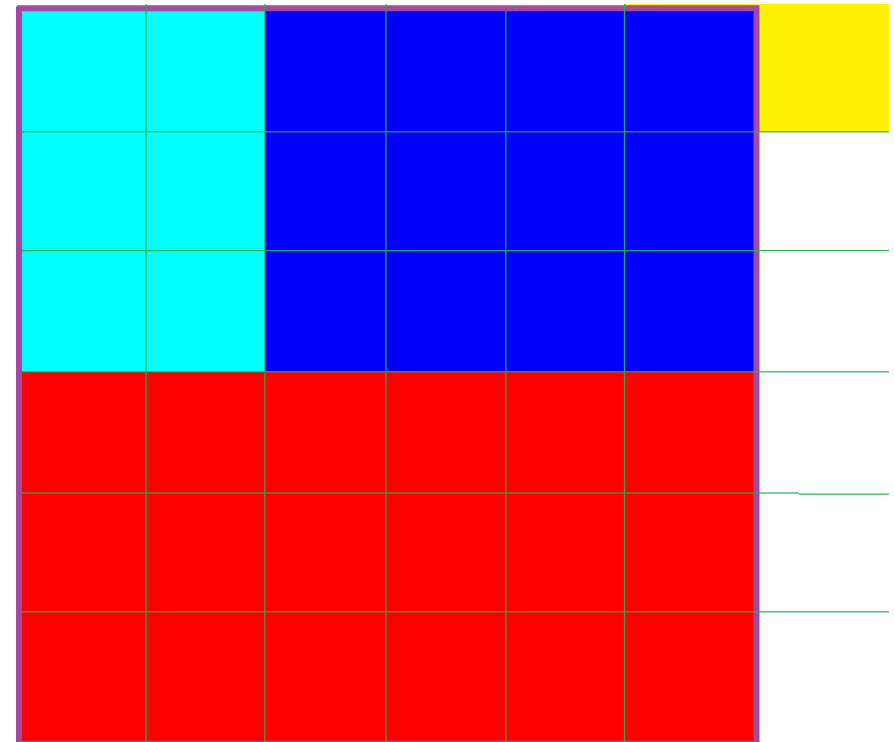
```
strona.html # style.css X
# style.css > .item-3
1
2 .container{
3   display: grid;
4   grid-template-rows: 100px 100px 100px 100px 100px 100px;
5   grid-template-columns: 100px 100px 100px 100px 100px 100px;
6 }
7
8 .item-1{
9   background-color: aqua;
10  grid-area: 1/ 1/ 4 /3; /* wiersz początkowy, kolumna początkowa , wiersz końcowy , kolumna końcowa*/
11 }
12
13 .item-2{
14   background-color: blue;
15   grid-area: 1/ 3 /4 /6
16 }
17
18 .item-3{
19   background-color: red;
20   grid-area: 4/1/6/6
21 }
```



Grid-auto-flow, grid-auto-columns

```
strona.html X # style.css
strona.html > html
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7   <link rel="stylesheet" href="style.css">
8
9 <div class = "container">
10   <div class = "item-1">1</div>
11   <div class = "item-2">2</div>
12   <div class = "item-3">3</div>
13   <div class = "item-4">3</div>
14
15 </div>
16
17 </head>
18 <body>
19
20 </body>
21 </html>
```

```
strona.html # style.css X
# style.css > .container
1
2 .container{
3   display: grid;
4   grid-template-rows: 100px 100px 100px 100px 100px 100px;
5   grid-template-columns: 100px 100px 100px 100px 100px 100px;
6   grid-auto-columns: 100px;
7   grid-auto-flow: column;
8 }
9
10 .item-1{
11   background-color: aqua;
12   grid-area: 1/ 1/ 4 /3;
13 }
14
15 .item-2{
16   background-color: blue;
17   grid-area: 1/ 3 /4 /6
18 }
19
20 .item-3{
21   background-color: red;
22   grid-area: 4/1/6/6
23 }
24
25 .item-4{
26   background-color: yellow;
```

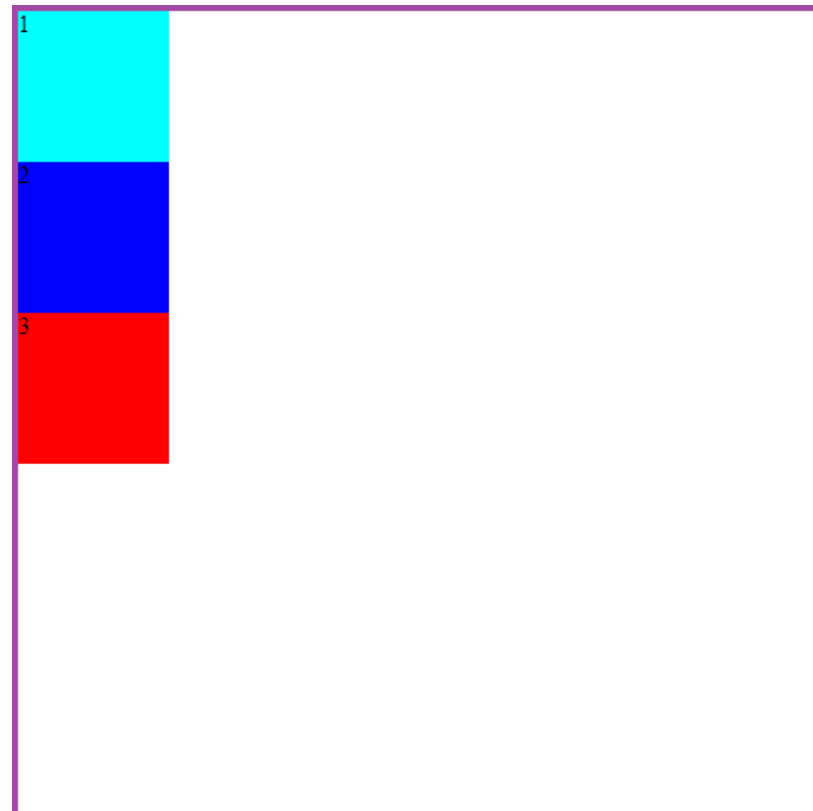


Flexbox

- Flexbox jest narzędziem pozwalającym na ułatwienie rozmieszczenia elementów na stronie

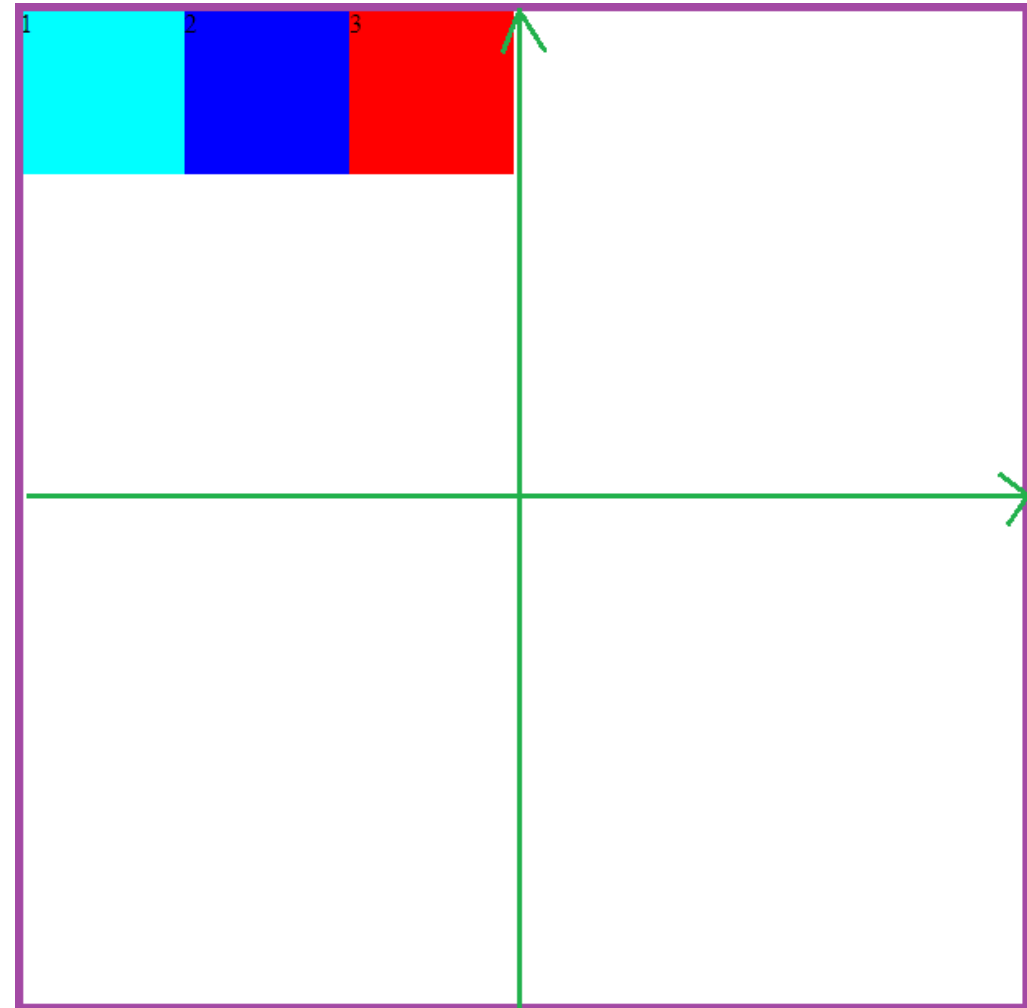
Flexbox funkcje

```
5
6 .item-1{
7     background-color: aqua;
8     height: 100px;
9     width: 100px;
10 }
11
12 .item-2{
13     background-color: blue;
14     height: 100px;
15     width: 100px;
16 }
17
18 .item-3{
19     background-color: red;
20     height: 100px;
21     width: 100px;
22 }
23
```



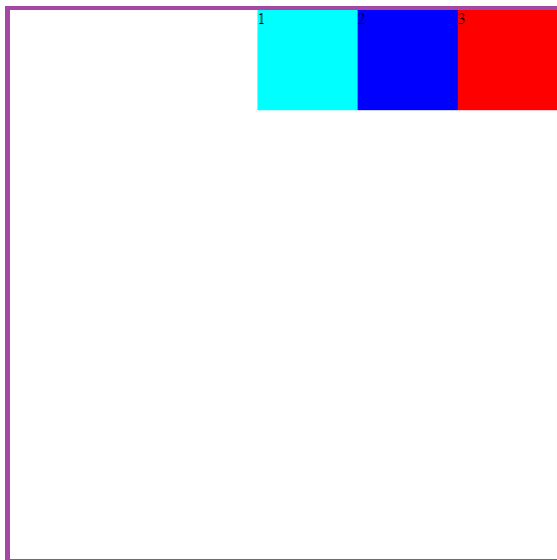
Display: flex

```
strona.html  # style.css x
# style.css > .container
1
2 .container{
3   display: flex;
4
5 }
6
7 .item-1{
8   background-color: aqua;
9   height: 100px;
10  width: 100px;
11 }
12
13 .item-2{
14   background-color: blue;
15   height: 100px;
16   width: 100px;
17 }
18
19 .item-3{
20   background-color: red;
21   height: 100px;
22   width: 100px;
23 }
24
```

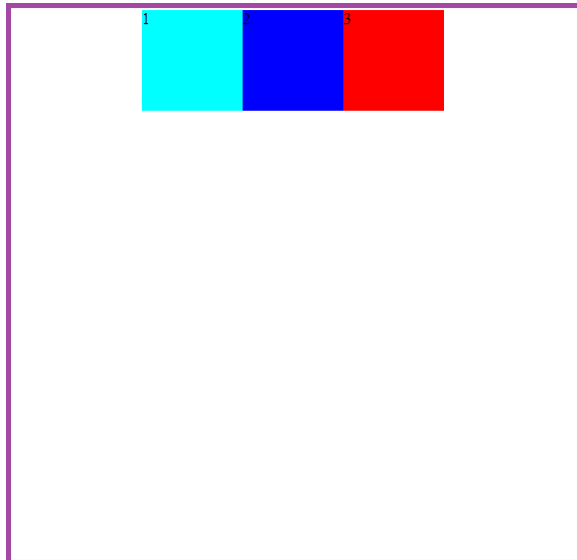


Justify-content

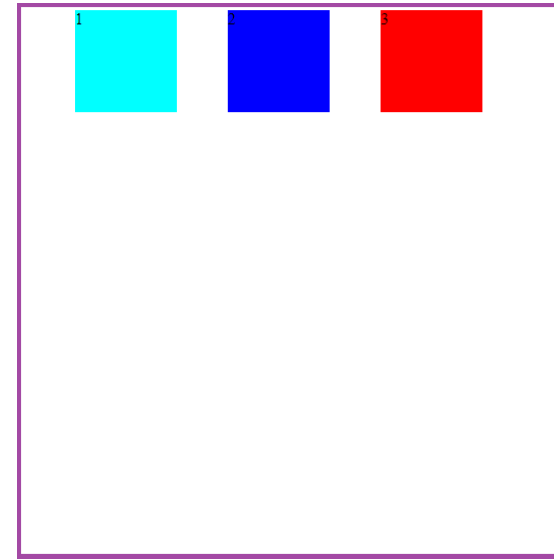
```
strona.html # style.css x
# style.css > .container
1
2 .container{
3   display: flex;
4   justify-content: flex-end;
5 }
6
7 .item-1{
8   background-color: aqua;
9   height: 100px;
10  width: 100px;
11 }
12
13 .item-2{
14   background-color: blue;
15   height: 100px;
16   width: 100px;
17 }
18
19 .item-3{
20   background-color: red;
21   height: 100px;
22   width: 100px;
23 }
24
```



```
strona.html # style.css x
# style.css > .container
1
2 .container{
3   display: flex;
4   justify-content: center;
5   width: 500px;
6   height: 500px;
7 }
8
9 .item-1{
10  background-color: aqua;
11  height: 100px;
12  width: 100px;
13 }
14
15 .item-2{
16  background-color: blue;
17  height: 100px;
18  width: 100px;
19 }
20
21 .item-3{
22  background-color: red;
23  height: 100px;
24  width: 100px;
25 }
26
```



```
strona.html # style.css x
# style.css > .container
1
2 .container{
3   display: flex;
4   justify-content: space-around;
5   width: 500px;
6   height: 500px;
7 }
8
9 .item-1{
10  background-color: aqua;
11  height: 100px;
12  width: 100px;
13 }
14
15 .item-2{
16  background-color: blue;
17  height: 100px;
18  width: 100px;
19 }
20
21 .item-3{
22  background-color: red;
23  height: 100px;
24  width: 100px;
25 }
26
```



Align-items

```
# style.css > .container
1
2 .container{
3   display: flex;
4   justify-content: space-evenly;
5   align-items: flex-start;
6   width: 500px;
7   height: 500px;
8 }
9
10 .item-1{
11   background-color: aqua;
12   height: 100px;
13   width: 100px;
14 }
15
16 .item-2{
17   background-color: blue;
18   height: 100px;
19   width: 100px;
20 }
21
22 .item-3{
23   background-color: red;
24   height: 100px;
25   width: 100px;
26 }
27
```



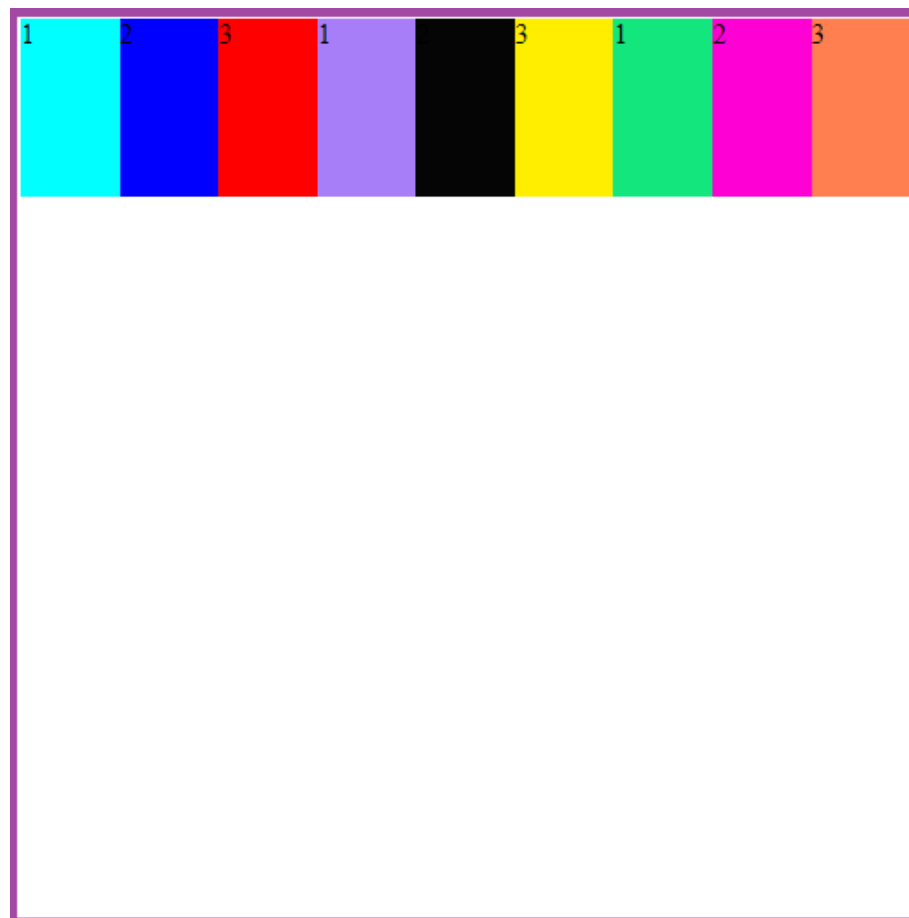
```
strona.html # style.css x
# style.css > .container
1
2 .container{
3   display: flex;
4   justify-content: space-evenly;
5   align-items: center;
6   width: 500px;
7   height: 500px;
8 }
9
10 .item-1{
11   background-color: aqua;
12   height: 100px;
13   width: 100px;
14 }
15
16 .item-2{
17   background-color: blue;
18   height: 100px;
19   width: 100px;
20 }
21
22 .item-3{
23   background-color: red;
24   height: 100px;
25   width: 100px;
26 }
27
```



```
strona.html # style.css x
# style.css > .container
1
2 .container{
3   display: flex;
4   justify-content: space-evenly;
5   align-items: flex-end;
6   width: 500px;
7   height: 500px;
8 }
9
10 .item-1{
11   background-color: aqua;
12   height: 100px;
13   width: 100px;
14 }
15
16 .item-2{
17   background-color: blue;
18   height: 100px;
19   width: 100px;
20 }
21
22 .item-3{
23   background-color: red;
24   height: 100px;
25   width: 100px;
26 }
27
```

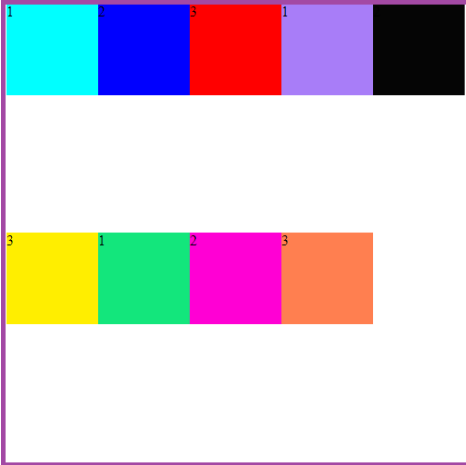


Co gdy dodamy więcej divów?

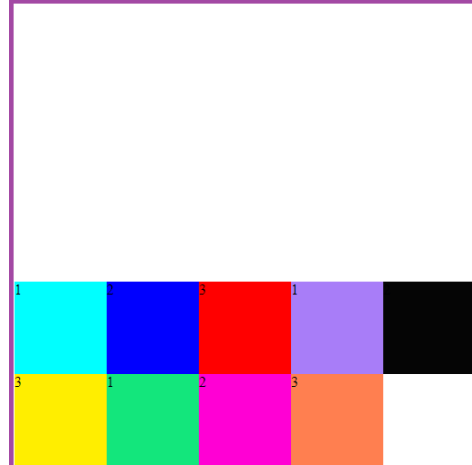


Flex-wrap

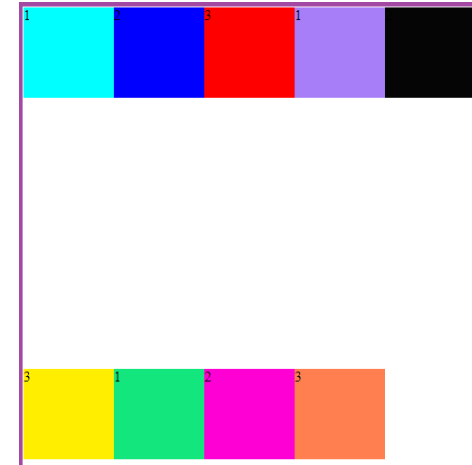
```
# style.css > .container
1 .container{
2   display: flex;
3   /* justify-content: space-evenly; */
4   align-items: flex-end; /*
5   flex-wrap: wrap;
6   align-content: flex-start;
7   width: 500px;
8   height: 500px;
9 }
```



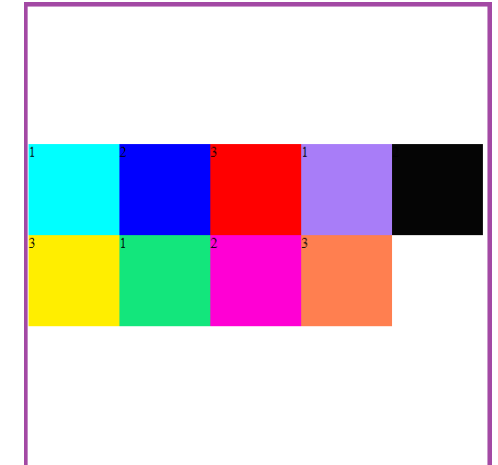
```
.container{
  display: flex;
  /*justify-content: space-evenly;*/
  /*align-items: flex-end;*/
  flex-wrap: wrap;
  align-content: space-between;
  width: 500px;
  height: 500px;
}
```



```
.container{
  display: flex;
  /*justify-content: space-evenly;*/
  /*align-items: flex-end;*/
  flex-wrap: wrap;
  align-content: flex-end;
  width: 500px;
  height: 500px;
}
```

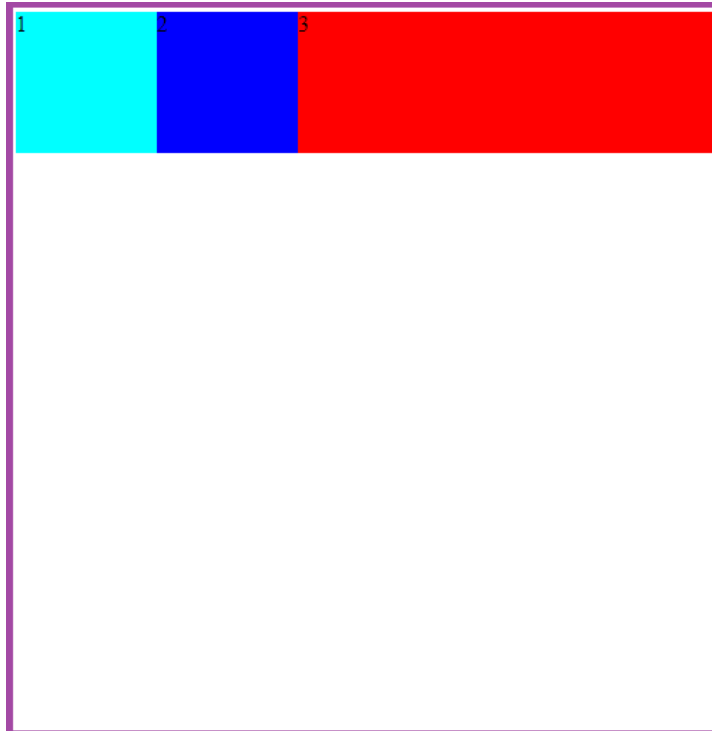


```
.container{
  display: flex;
  /*justify-content: space-evenly;*/
  /*align-items: flex-end;*/
  flex-wrap: wrap;
  align-content: center;
  width: 500px;
  height: 500px;
}
```



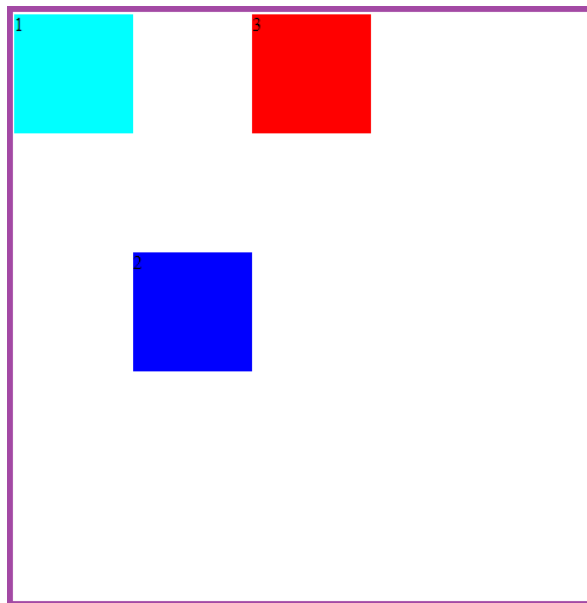
Flex-grow

```
.item-3{  
  background-color: ■ red;  
  height: 100px;  
  width: 100px;  
  flex-grow: 1;  
}
```



Align-self

```
.item-2{  
  background-color: blue;  
  height: 100px;  
  width: 100px;  
  align-self: center;  
}
```



order

```
.item-3{  
  background-color: red;  
  height: 100px;  
  width: 100px;  
  order: -1;  
}
```



Bootstrap – co to jest?

- Bootstrap to jeden z najpopularniejszych frameworków, który umożliwia, tworzenie responsywnych layoutów stron internetowych, zmieniając rozmiar i zachowanie elementów z zależności od rozmiarów ekranu

Importowanie bootstrapa

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <link rel="stylesheet" href="style.css">

  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
  | rel="stylesheet" integrity="sha384-T3c6CoTi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN" crossorigin="anonymous">

<div class = "container" >
  <div class = "item-1">1</div>
  <div class = "item-2">2</div>
  <div class = "item-3">3</div>

  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"
  | integrity="sha384-C6RzsynM9kWDrmNeT87bh950GNYZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL" crossorigin="anonymous"></script>

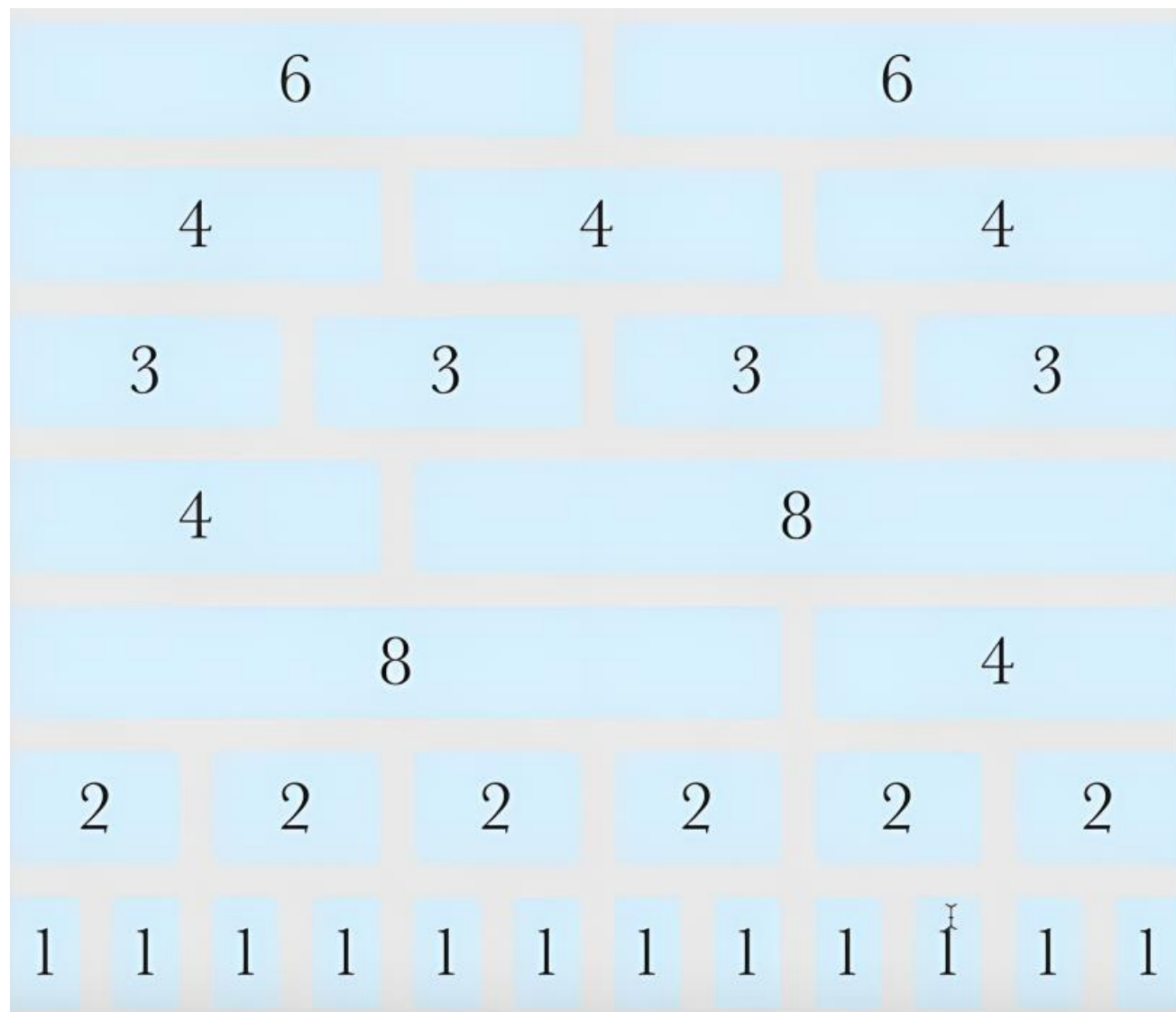
</div>

</head>
<body>

</body>
```

Bootstrap – jak działa

- Bootstrap działa poprzez podzielenie strony w siatkę, gdzie sami możemy ustalać jej wymiary i za pomocą resizowania siatki podczas zmniejszania strony, nadaje jej responsywność



Divy

Wg wzoru col-sizeCode – numberOfCols


```
<div class = „row”>  
  <div class = „col-md-6”>  
    <h2>Tekst</h2>  
  </div>  
  <div class = „col-md-6”>  
    <h2>Tekst</h2>  
  </div>  
</div>
```

Przykładowa siatka

Tekst						Tekst					




**CO TO
WEBSTORM?**

The image features two solid blue squares on the left side. The top square is a lighter shade of blue, while the bottom square is a darker shade. They are positioned vertically, with the top square partially overlapping the bottom one.

Zintegrowane środowisko programistyczne (IDE) stworzone przez firmę **JetBrains**, które jest dedykowane głównie do tworzenia **aplikacji internetowych**, a w szczególności do pracy z **technologiami webowymi**, takimi jak HTML, CSS, JavaScript oraz różnymi frameworkami i bibliotekami, takimi jak Angular, React czy Vue.js.

JETBRAINS

- 
- A decorative graphic on the left side of the slide consisting of two overlapping squares: a larger dark blue square at the bottom and a smaller light blue square on top of it, partially offset to the right.
- Firma **oprogramowania komputerowego** z siedzibą w **Czechach**.
 - Jest znana ze swoich **narzędzi i środowisk programistycznych**, które są szeroko wykorzystywane przez programistów na całym świecie.
 - Narzędzia JetBrains są **dedykowane do konkretnych języków programowania i technologii**, co sprawia, że są one odpowiednie do pracy nad różnymi rodzajami projektów.



FUNKCJE

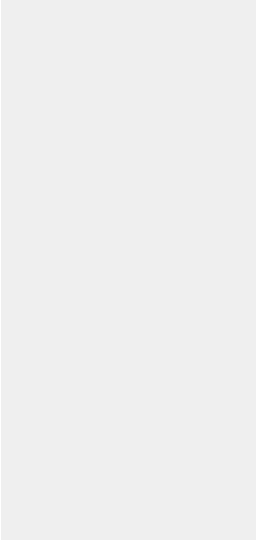


Edytor kodu

- edytor kodu z zaawansowanymi funkcjami
- **podpowiedzi** kodu
- automatyczne **uzupełnianie**
- sprawdzanie składni



Debugowanie

- 
- debugowanie aplikacji internetowych.
 - uruchamianie i śledzenie kodu krok po kroku
 - korzystanie z narzędzi do **analizy** i **monitorowania** aplikacji



Integracja z narzędziami

Integracja z systemami kontroli wersji

- łączenie się z platformą **GIT**
- praca w **zespółach**

Integracja z narzędziami budowania

- różne narzędzia do **automatycznego budowania** projektów
- łatwe tworzenie i dostarczanie aplikacji internetowych

Integracja z bazami danych

- praca z bazą danych **bez konieczności opuszczania programu**

Wsparcie dla frameworków

- obsługuje wiele języków programowania, w tym **HTML, CSS, JS, TypeScript, PHP**
- zaawansowane wsparcie dla popularnych frameworków frontendowych, takich jak **Angular, React, Vue.js, Ember.js**
- zaawansowane **wsparcie** dla frameworków i bibliotek, co przyspiesza rozwijanie aplikacji

Narzędzia do testowania

- WebStorm oferuje narzędzia do tworzenia, uruchamiania i analizy testów
- testy **jednostkowe** oraz testy **integracyjne**

Inteligentne sugestie

- inteligentne sugestie dotyczące **poprawy kodu, optymalizacji i refaktoryzacji**
- pomaga to utrzymać kod źródłowy w dobrej kondycji i zwiększa czytelność.

youtrack-mobile > src > components > issue-summary > issue-summary.js

Project

youtrack-mobile ~ /WebstormProjects

android

ios

node_modules library root

src

components

action-sheet

api

attach-file

attachments-row

auth

cache

color-field

comment

config

custom-field

custom-fields-panel

npm

youtrack-mobile/package.json

start

test

test-ci

test:watch

lint

ios

ios-set-version

issue-summary.js

```
1 import React, {Component, PropTypes} from 'react';
2 import {View, TextInput} from 'react-native';
3 import styles from './issue-summary.styles';
4 import MultilineInput from '../multiline-input/multiline-input';
5
6 export default class AttachmentsRow extends Component {
7
8   static propTypes = {"editable": PropTypes.bool...}
16
17   render() {
18     const {editable, showSeparator, summary, description, ...rest} = this.props;
19
20     return (
21       <View {...rest}>
22         <Te
23           TextInput      ReactNative (react-native.js, react-native)
24           CreateIssue    (create-issue.js, src/views/create-issue)
25           EnterServer    (enter-server.js, src/views/enter-server)
26           ^↓ and ^↑ will move caret down and up in the editor >>
27           multiline={true}
28           editable={editable}
29           autoCapitalize="sentences"
30           placeholder="Description"
31           value={description}
32           onChangeText={this.props.onDescriptionChange} />
33       </View>
34     );
35   }
36 }
37
38 }
```

- WebStorm jest dostępny na różne platformy, w tym **Windows, macOS i Linux.**
- Jest to **narzędzie płatne**, ale dostępna jest także wersja próbna, która pozwala przetestować jego funkcje przed zakupem.
- Dla **programistów webowych i deweloperów** JavaScript WebStorm jest jednym z popularnych wyborów ze względu na bogate funkcje i wsparcie dla najnowszych technologii webowych.

**Dzięki za
uwagę!**