





Akademia Developera – edycja FrontDev

Frontend development od podstaw



KAMIL SZKUTNIK

WEB DEVELOPER

Test wstępny

Co już poznaliśmy

- budowa dokumentu HTML
- podstawowe elementy, nagłówki
- tagi semantyczne
- formularze
- podstawy accessibility

Co poznamy dziś?

- składnia css
- osadzanie styli w HTML
- selektory
- pseudo-klasy
- pseudo-elementy
- priorytety stylów
- CSS box model
- deklarowanie kolorów
- właściwość display
- właściwość position
- pozycjonowanie (float)

SKŁADNIA CSS

```
selektor {  
    właściwość: wartość;  
}
```

przykład:

```
div > p + a {  
    font-size: 14px;  
}
```

OSADZENIE STYLI W HTML

Inline - nie polecane!

```
<div style="właściwość: wartość"></div>
```

Wewnętrzny arkusz stylów

```
<head>
  ...
  <style>
    div {
      właściwość: wartość
    }
  </style>
</head>
```


Zewnętrzny arkusz stylów

```
<head>
  ...
  <link rel="stylesheet"
        type="text/css"
        href="style.css">
</head>
```

Kolory

PREDEFINIOWANE

np. background-color: Tomato:

Tomato

Inne przykłady:

Orange DodgerBlue MediumSeaGreen Gray SlateBlue Violet LightGray

Oraz wiele wiele więcej...

HEX

- #rrggbb

gdzie rr(red), gg(green), bb(blue) to heksadecymalna wartość z przedziału 00 - FF

np. dla takiego samego koloru jak Tomato:

#ff6347

RGB & RGBA

- `rgb(red, green, blue)`

każdy parametr może przyjąć wartość 0-255 `rgb(255, 99, 71)`

- `rgba(red, green, blue, alpha)`

gdzie alpha to przezroczystość z przedziału 0.0 do 1.0

`rgba(255, 99, 71, 0.5)`

HSL & HSLA

- `hsl(hue, saturation, lightness)`

hue jest stopniem z koła kolorów z przedziału 0-360 `hsl(9, 100%, 64%)`

- `hsla(hue, saturation, lightness, alpha)`

gdzie alpha to przezroczystość z przedziału 0.0 do 1.0

`hsla(9, 100%, 64%, 0.5)`

Selektory css

PODSTAWOWE

```
// Selektor uniwersalny
```

```
* {}
```

```
// Selektor elementu
```

```
div {}
```

```
// Selektor klasy
```

```
.klasa {}
```

```
// Selektor id
```

```
#identyfikator {}
```


DODATKOWE

```
div, p, .klasa {}
```

```
div .klasa {}
```

```
div > p {}
```

I JESZCZE KILKA

```
div + p {}
```

```
div ~ p {}
```

```
a[title] {}
```

Pseudo-klasy

:first-child oraz :last-child

```
ul li:first-child {  
  color: red;  
}
```

- PGS Software
- PGS Software
- PGS Software
- PGS Software
- PGS Software

:nth-child()

```
ul li:nth-child(2) {  
  color: red;  
}
```

- PGS Software
- **PGS Software**
- PGS Software
- PGS Software
- PGS Software

:not

```
ul li:not(#exampleLi) {  
  color: red;  
}
```

- PGS Software
- PGS Software
- PGS Software
- PGS Software (#exampleLi)
- PGS Software
- PGS Software

:enabled, :disabled oraz :checked

```
input:checked + label {  
    color: #f67628;  
}  
  
input:enabled {  
    background-color: #f67628;  
}  
  
input:disabled {  
    background-color: gray;  
}
```

☐ opcja1 ☐ opcja2

input1

Dynamiczne pseudo-klasy

- :link
- :visited
- :hover
- :active
- :focus

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

Link to: [Akademia Developera](#)



Pseudo-elementy

Pseudo-elementy pozwalają wstawiać do dokumentu fikcyjne elementy.

::before oraz :after

```
.test::before {  
  content: "PGS Software";  
  color: #f67628;  
}
```

PGS SoftwarePrzykładowy tekst

::first-line

```
.test::first-line {  
  text-decoration: underline;  
}
```

Przykładowy tekst

Druga linia

::first-letter

```
.test::first-letter {  
  text-decoration: underline;  
}
```

Przykładowy tekst

JAKIE TŁO BĘDZIE MIAŁ ELEMENT?

```
.test span {  
  background: blue;  
}  
  
div.test > span {  
  background: green;  
}  
  
section footer div span {  
  background: red;  
}
```

WAGI SELEKTORÓW

- Tagi oraz pseudoelementy mają wagę 1
- Klasy, atrybuty selektorów oraz pseudoklasy mają wagę 10
- Identyfikatory mają wagę 100
- Style inline przy elemencie mają wagę 1000
- Użycie !important dodaje 1000 do wagi selektora

Im większa waga, tym większy priorytet. A im większy priorytet, tym ważniejsza reguła.

CSS box model



```
.box {  
  width: 300px;  
  border: 30px solid #ff7726;  
  padding: 30px;  
  box-sizing: border-box lub content-box;  
}
```

300px

Pudelko

WŁAŚCIWOŚĆ DISPLAY:BLOCK

```
.box {  
  display: block;  
  background: #ff7726;  
  font-size: 40px;  
}
```

box 1

box 2

box 3

WŁAŚCIWOŚĆ DISPLAY:INLINE

```
.box {  
  display: inline;  
  background: #ff7726;  
  font-size: 40px;  
}
```

box 1 box 2 box 3

WŁAŚCIWOŚĆ DISPLAY:INLINE-BLOCK

```
.box {  
  display: inline-block;  
  background: #ff7726;  
  font-size: 40px;  
}
```

box 1 box 2 box 3

WŁAŚCIWOŚĆ POSITION

Określa jak pozycjonowany ma być określony element, dostępne opcje:

POSITION: STATIC

```
div {  
  position: static;  
}
```

POSITION: FIXED

```
div {  
  position: fixed;  
}
```

POSITION: ABSOLUTE

```
div {  
  position: absolute;  
}
```

POSITION: RELATIVE

```
div {  
  position: relative;  
}
```


POSITION: STICKY

```
div {  
  position: sticky;  
}
```

Float

Używając float, możemy zrobić elementy pływające.

Elementy te będą dosunięte do lewej lub prawej krawędzi

```
div {  
  float: left;  
}
```

`float: left;`

Lorem ipsum dolor sit amet,
consectetur adipiscing elit,
sed do eiusmod tempor
incididunt ut labore et
dolore magna aliqua. Ut
enim ad minim veniam, quis
nostrud exercitation ullamco
laboris nisi ut aliquip ex ea commodo
consequat.

PROBLEM



ROZWIĄZANIE

Użycie clearfix na kontenerze

```
.clearfix:after {  
  content: "";  
  display: table;  
  clear: both;  
}
```





Dziękuję za uwagę !

Odwiedź:

www.facebook.com/AkademiaDeveloperaRzeszow

KAMIL SZKUTNIK • WEB DEVELOPER