

CSS

Magdalena Lemańska (korzystam m.in z kursu Mateusza
Maya- <https://www.udemy.com/>)

Co to jest CSS?

Kaskadowe arkusze stylów CSS (ang. Cascading Style Sheets) to kod służący do nadawania wyglądu strony. To język, który wnosi styl do stron internetowych poprzez interakcję z elementami HTML.

Co to jest CSS?

Kaskadowe arkusze stylów CSS (ang. Cascading Style Sheets) to kod służący do nadawania wyglądu strony. To język, który wnosi styl do stron internetowych poprzez interakcję z elementami HTML.

Podczas gdy HTML jest używany do tworzenia dokumentów internetowych (definiowania takich rzeczy jak nagłówki i akapity oraz osadzania obrazów, filmów i innych mediów), CSS określa styl strony (kolory i czcionki dokumentu). CSS nie jest językiem programowania.

Co to jest CSS?

Kaskadowe arkusze stylów CSS (ang. Cascading Style Sheets) to kod służący do nadawania wyglądu strony. To język, który wnosi styl do stron internetowych poprzez interakcję z elementami HTML.

Podczas gdy HTML jest używany do tworzenia dokumentów internetowych (definiowania takich rzeczy jak nagłówki i akapity oraz osadzania obrazów, filmów i innych mediów), CSS określa styl strony (kolory i czcionki dokumentu). CSS nie jest językiem programowania.

Budowa CSS

Dyrektywa (reguła) składa się z:

- **selektora** (nazwa tagu HTML / .class / #ID),
- **właściwości** (np. color / font-size / padding / position),
- **wartości** (np. yellow / 16px).

Przykład:

```
p {  
  color: blue;  
  font-size: 30px;  
}
```

Co to jest CSS?

Kaskadowe arkusze stylów CSS (ang. Cascading Style Sheets) to kod służący do nadawania wyglądu strony. To język, który wnosi styl do stron internetowych poprzez interakcję z elementami HTML.

Podczas gdy HTML jest używany do tworzenia dokumentów internetowych (definiowania takich rzeczy jak nagłówki i akapity oraz osadzania obrazów, filmów i innych mediów), CSS określa styl strony (kolory i czcionki dokumentu). CSS nie jest językiem programowania.

Budowa CSS

Dyrektywa (reguła) składa się z:

- **selektora** (nazwa tagu HTML / .class / #ID),
- **właściwości** (np. color / font-size / padding / position),
- **wartości** (np. yellow / 16px).

Przykład:

```
p {  
  color: blue;  
  font-size: 30px;  
}
```

Selektor (jak sama nazwa wskazuje: selekcja = wybór) określa jednoznacznie do których elementów z kodu HTML zostaną zastosowane właściwości podane w tzw. ciele selektora (czyli pomiędzy nawiasami klamrowymi).

Selektor uniwersalny

Odnosi się do wszystkich tagów w html. W przykładzie wszystkie linki, nagłówki, paragrafy mają kolor czerwony i font-size 30 px. Pod spodem: odniesienie wyłącznie do paragrafu.

Uniwersalny selektor

* to uniwersalny selektor w CSS.

Przykład:

```
* {  
  color: red;  
  font-size: 30px;  
}
```

```
p {  
  color: yellow;  
  font-size: 50px;  
}
```

Gdzie umieszczać css?

Za css odpowiada następująca linijka w kodzie html (w pliku index.html):

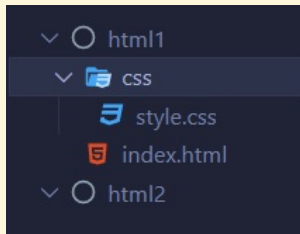
```
<link rel="stylesheet" href="style.css" />
```

Gdzie umieszczać css?

Za css odpowiada następująca linijka w kodzie html (w pliku index.html):

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

Stwórzmy teraz nowy katalog **css** oraz przenieśmy do niego plik **style.css**:

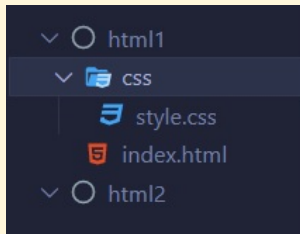


Gdzie umieszczać css?

Za css odpowiada następująca linijka w kodzie html (w pliku index.html):

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

Stwórzmy teraz nowy katalog **css** oraz przenieśmy do niego plik **style.css**:

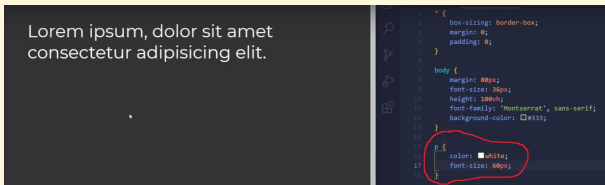


Teraz linijka w kodzie html wygląda tak:

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

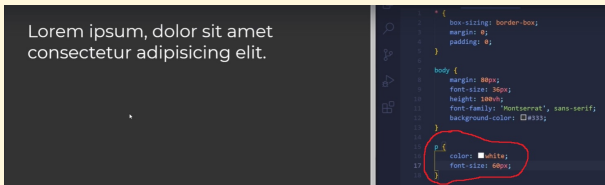
Przykład

Jeśli w html umieścimy sobie linijkę z paragrafem: `< p > Lorem ipsum, dolor sit amet consectetur adipisicing elit < /p >` ,to po umieszczeniu w pliku css tekstu zaznaczonego na czerwono otrzymamy następujący efekt:



Przykład

Jeśli w html umieścimy sobie linijkę z paragrafem: `< p > Lorem ipsum, dolor sit amet consectetur adipisicing elit < /p >`, to po umieszczeniu w pliku css tekstu zaznaczonego na czerwono otrzymamy następujący efekt:



Podobny efekt można uzyskać korzystając ze stylów w html i umieszczając poniższy kod w pliku **index.html** (poniżej linijki dotyczącej css), ale nie jest to zalecane. Lepiej wszystko dotyczące css zamieszczać w pliku css.

```
<style>
  p {
    font-size: 100px;
  }
</style>
```

Komentarze

Komentarz pozostaje w kodzie, ale nie widzimy go na stronie. Komentarz można zrobić podświetlając tekst i stosując skrót klawiszowy **Ctrl + Slash**.

Komentarze

Komentarz pozostaje w kodzie, ale nie widzimy go na stronie. Komentarz można zrobić podświetlając tekst i stosując skrót klawiszowy **Ctrl + Slash**.

Skracanie kodu w css- przykład

```
<div>Lorem ipsum dolor sit amet.</div>
<p>Lorem ipsum dolor sit amet.</p>
<span>Lorem ipsum dolor sit amet.</span>
```

Lorem ipsum dolor sit amet.
 Lorem ipsum dolor sit amet.
 Lorem ipsum dolor sit amet.

```
div {
  font-size: 80px;
}
p {
  font-size: 80px;
}
span {
  font-size: 80px;
}

div, p, span {
  font-size: 80px;
}
```

Klasy w CSS

Selektor **klasy** odszukuje każdy element HTML, który posiada atrybut **class** o podanej przez nas wartości. Selektor klasy tworzymy za pomocą znaku . (kropka), po którym należy umieścić wartość atrybutu class interesującego nas elementu HTML, dzięki czemu selektor klasy może dokonać selekcji.

Klasy w CSS

Selektor **klasy** odszukuje każdy element HTML, który posiada atrybut **class** o podanej przez nas wartości. Selektor klasy tworzymy za pomocą znaku . (kropka), po którym należy umieścić wartość atrybutu class interesującego nas elementu HTML, dzięki czemu selektor klasy może dokonać selekcji.

Klasy

- Nazwę klasy wymyślamy sami (klasa **musi być** po angielsku).
- Tę samą klasę możemy dodać do wielu tagów HTML.
- W CSS do klasy odwołujemy się za pomocą kropki.
- Możemy umieścić kilka klas w jednym elemencie.

Przykład:

```
.main {}
```

Klasy w CSS

Selektor **klasy** odszukuje każdy element HTML, który posiada atrybut **class** o podanej przez nas wartości. Selektor klasy tworzymy za pomocą znaku **.** (kropka), po którym należy umieścić wartość atrybutu class interesującego nas elementu HTML, dzięki czemu selektor klasy może dokonać selekcji.

Klasy

- Nazwę klasy wymyślamy sami (klasa **musi być** po angielsku).
- Tę samą klasę możemy dodać do wielu tagów HTML.
- W CSS do klasy odwołujemy się za pomocą kropki.
- Możemy umieścić kilka klas w jednym elemencie.

Przykład:

```
.main {}
```

Przykład zastosowania klasy:

```
<div class="text">Lorem ipsum dolor sit amet.</div>
<p class="text">Lorem ipsum dolor sit amet.</p>
<span class="text">Lorem ipsum dolor sit amet.</span>

<p>ten paragraf ma mieć małą czcionkę</p>
```

```

Lorem ipsum dolor sit amet.
Lorem ipsum dolor sit amet.
Lorem ipsum dolor sit amet.
ten paragraf ma mieć małą czcionkę
  
```

```
.text {
  font-size: 60px;
}
```


Przykład zastosowania klasy

HTML:

```
<h1 class="center">Red and center-aligned heading</h1>  
<p class="center">Red and center-aligned paragraph.</p>
```

CSS:

```
.center {  
    text-align: center;  
    color: red;  
}
```

EFEKT:

Red and center-aligned heading

Red and center-aligned paragraph.

Przykład zastosowania klasy

HTML:

```
<h1>This heading will not be affected</h1>  
<p class="center">This paragraph will be red and center-aligned.</p>  
<p class="centerlarge">This paragraph will be red, center-aligned, and in a large font-size</p>
```

CSS:

```
.center {  
  text-align: center;  
  color: red;  
}  
  
.centerlarge {  
  text-align: center;  
  color: red;  
  font-size: 300%;  
}
```

EFEKT:

This heading will not be affected

This paragraph will be red and center-aligned.

This paragraph will be red, center-aligned, and in a large font-size.

ID

Selektor ID wyszukuje każdy element z atrybutem id o takiej samej nazwie w dokumencie HTML. Atrybut ID musi jednoznacznie (unikalnie) identyfikować jeden konkretny element witryny (klasy mogą kilka). ID zarezerwowany jest raczej dla Java Script.

ID

- Nazwę ID wymyślamy sami (ID **musi być** po angielsku).
- **ID jest unikalne!** Tylko jedno na stronę!
- W CSS do ID odwołujemy się za pomocą hash.
- Nie stylujemy za pomocą ID.

Przykład:

```
#main {}
```

```
<div id="main"></div>
```

```
<p id="main2"></p>
```

(Gdybyśmy używali klasy, to w paragrafie mogłaby się znowu pojawić ta sama klasa- w przypadku ID to nie zadziała!)

Czcionki w html

Domyślną czcionką jest czcionka szeryfowa.

TEKST Tekst tekst
 TEKST Tekst tekst iw IW
 TEKST Tekst tekst iw IW
 TEKST Tekst tekst

```

1  <!DOCTYPE html> "UTF-8" />
2  <html lang="pl">
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,
7        initial-scale=1.0" />
8      <title>MMC School</title>
9      <link rel="preconnect" href="https://fonts.gstatic.com" />
10     <link href="https://fonts.googleapis.com/css2?
11       family=Montserrat:wght@100;400;700&display=swap"
12       rel="stylesheet" />
13
14     <link rel="stylesheet" href="./css/style.css" />
15   </head>
16   <body>
17     <p>TEKST Tekst tekst</p>
18     <p class="sans-serif">TEKST Tekst tekst iw IW</p>
19     <p class="monospace">TEKST Tekst tekst iw IW</p>
20     <p class="cursive">TEKST Tekst tekst</p>
21   </body>
22 </html>

```

Czcionki w html

Domyślną czcionką jest czcionka szeryfowa.

TEKST Tekst tekst
 TEKST Tekst tekst iw IW
 TEKST Tekst tekst iw IW
 TEKST Tekst tekst

```

1  <!DOCTYPE html>
2  <html lang="pl">
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,
7        initial-scale=1.0" />
8      <title>MMC School</title>
9      <link rel="preconnect" href="https://fonts.gstatic.com" />
10     <link href="https://fonts.googleapis.com/css2?
11       family=Montserrat:wght@100;400;700&display=swap"
12       rel="stylesheet" />
13
14     <link rel="stylesheet" href="/css/style.css" />
15   </head>
16   <body>
17     <p>TEKST Tekst tekst</p>
18     <p class="sans-serif">TEKST Tekst tekst iw IW</p>
19     <p class="monospace">TEKST Tekst tekst iw IW</p>
20     <p class="cursive">TEKST Tekst tekst</p>
21   </body>
22 </html>
  
```

Aby to zadziało, w pliku **style.css** musimy wpisać następujący kod:

```

.sans-serif {
  font-family: sans-serif;
}

.monospace {
  font-family: monospace;
}

.cursive {
  font-family: cursive;
}
  
```

Jeśli chcemy coś zmienić (np pogrubić czcionkę) w obrębie paragrafu, używamy do tego **span** (definiując odpowiednią klasę w pliku **style.css**):

```
<p>Lorem ipsum dolor sit amet consectetur adipisicing elit.  
<span class="bold-text">Necessitatibus nihil velit ipsam  
dolorum doloribus itaque, error harum distinctio!</span>  
Voluptatem delectus nulla quia error repudiandae iusto labore  
dolore dolorem fuga laudantium.</p>
```

```
.bold-text {  
    font-weight: bold;  
}
```

Stylowanie tekstu

Po wpisaniu jakiegoś tekstu w paragrafe w **index.html**, można z tym tekstem w pliku **style.css** robić różne rzeczy, na przykład:

```
p {  
  color: ■gold;  
  
  text-align: left;  
  text-align: right;  
  text-align: center;  
  text-align: justify;  
  text-decoration: overline;  
  text-decoration: underline;  
  text-decoration: line-through;  
}
```

Stylowanie tekstu

Po wpisaniu jakiegoś tekstu w paragrafie w **index.html**, można z tym tekstem w pliku **style.css** robić różne rzeczy, na przykład:

```
p {  
  color: ■gold;  
  
  text-align: left;  
  text-align: right;  
  text-align: center;  
  text-align: justify;  
  text-decoration: overline;  
  text-decoration: underline;  
  text-decoration: line-through;  
}
```

Gdybyśmy chcieli zmienić kolor linka zdefiniowanego w **index.html** i usunąć z niego podkreślenie, w pliku **style.css** należy wpisać:

```
a {  
  color: ■white;  
  text-decoration: none;  
}
```


Litery małe, duże i mieszane

TO JEST MAŁY TEKST
to jest duży tekst
To Jest Trzeci Tekst

```
<p class="small">to jest mały tekst</p>  
<p class="big">TO JEST DUŻY TEKST</p>  
<p class="text">to jest trzeci tekst</p>
```

```
.small {  
  text-transform: uppercase;  
}  
  
.big {  
  text-transform: lowercase;  
}  
  
.text {  
  text-transform: capitalize;  
}
```

Litery małe, duże i mieszane

TO JEST MAŁY TEKST
to jest duży tekst
To Jest Trzeci Tekst

```
<p class="small">to jest mały tekst</p>
<p class="big">TO JEST DUŻY TEKST</p>
<p class="text">to jest trzeci tekst</p>
```

```
.small {
  text-transform: uppercase;
}

.big {
  text-transform: lowercase;
}

.text {
  text-transform: capitalize;
}
```

Odstępy między literami:

LoRem ipsum dolor sit
amet.

```
{
  color: gold;
  letter-spacing: 20px;
}
```

Litery małe, duże i mieszane

TO JEST MAŁY TEKST
to jest duży tekst
To Jest Trzeci Tekst

```
<p class="small">to jest mały tekst</p>
<p class="big">TO JEST DUŻY TEKST</p>
<p class="text">to jest trzeci tekst</p>
```

```
.small {
  text-transform: uppercase;
}

.big {
  text-transform: lowercase;
}

.text {
  text-transform: capitalize;
}
```

Odstępy między literami:

Lo rem ip sum do lor sit
a met.

```
p {
  color: gold;
  letter-spacing: 20px;
}
```

Rozmycie tekstu:

Lo rem ip sum do lor sit amet.

```
p {
  color: gold;
  text-shadow: 5px 5px 5px #fff;
}
```

Kolory i obrazy



Kolory i obrazki



Odniesienie w css do obrazka zdefiniowanego w html (obrazki zawsze zachowują swoje proporcje! Wystarczy podać tylko jeden wymiar):

```

img {
  width: 600px;
}
  
```

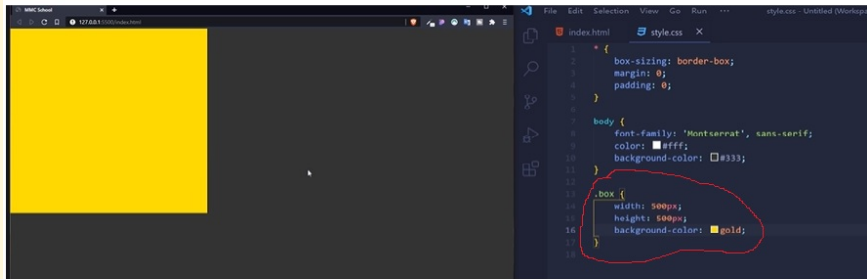
Jak wyodrębnić kształty określone koloru?

Gdybyśmy chcieli potem tworzyć więcej takich kształtów, lepiej to zrobić za pomocą klas:

```
<body>  
  <div class="box"></div>  
</body>
```



w pliku index.html



Box model- model pudełkowy

Można narzucić w całym dokumencie "pudełka", które są "przyklejone" do lewej strony (tekst jest również "przyklejony" do lewej strony):

```
* {  
  box-sizing: border-box;  
  margin: 0;  
  padding: 0;  
}
```

Box model- model pudełkowy

Można narzucić w całym dokumencie "pudełka", które są "przyklejone" do lewej strony (tekst jest również "przyklejony" do lewej strony):

```
* {  
  box-sizing: border-box;  
  margin: 0;  
  padding: 0;  
}
```

Można jednak manipulować poszczególnymi "pudełkami", zgodnie z zasadą **content** → **padding** → **border** → **margin** ("padding" oznacza odstęp tekstu od krawędzi, "border" ramkę, a "margin" marginesy (odpowiednio **top**, **right**, **bottom**, **left**):



Stylowanie przycisków

```
<button>Jestem zwykłym przyciskiem... 😞</button>  
<button class="btn">Jestem ostylowanym przyciskiem! 😊</button>
```

Jestem zwykłym przyciskiem... 😞

Jestem ostylowanym
przyciskiem! 😊

```
7  
8 body {  
9   margin: 50px;  
10  font-family: 'Montserrat', sans-serif;  
11  background-color: #333;  
12  font-size: 36px;  
13 }  
14 .btn {  
15   padding: 10px 20px;  
16   background-color: #f08080; /* tomato */  
17   border: none;  
18   border-radius: 6px;  
19   cursor: pointer;  
20   transition: .3s;  
21 }  
22  
23 .btn:hover {  
24   background-color: #f0e68c; /* gold */  
25   box-shadow: 0 0 5px #f0e68c; /* gold */  
26 }
```

Stylowanie przycisków

```
<button>Jestem zwykłym przyciskiem... 😞</button>
<button class="btn">Jestem ostylowanym przyciskiem! 😊</button>
```

Jestem zwykłym przyciskiem... 😞

Jestem ostylowanym
przyciskiem! 😊

```

7  body {
8    margin: 50px;
9    font-family: 'Montserrat', sans-serif;
10   background-color: #333;
11   font-size: 36px;
12 }
13
14 .btn {
15   padding: 10px 20px;
16   background-color: #f08080;
17   border: none;
18   border-radius: 6px;
19   cursor: pointer;
20   transition: .3s;
21 }
22
23 .btn:hover {
24   background-color: #f0e68c;
25   box-shadow: 0 0 5px #f0e68c;
26 }
```

padding- odpowiada za odsunięcie tekstu od krawędzi; **border radius**- zaokrąglenie rogów;

cursor: pointer- podpowiedź dla użytkownika, że jest to element klikalny;

transition (na elemencie animowanym) - oznacza czas zmiany koloru;

pseudoklasa **hover** (odwołanie za pomocą dwukropka) - zmiana koloru przycisku po kliknięciu w niego.

Responsywne strony internetowe

RWD – z angielskiego: "Responsive Web Design" jest techniką projektowania stron WWW, tak aby ich układ i wygląd automatycznie dostosowywał się do rozmiaru okna przeglądarki, na której strona jest wyświetlana. Strona WWW stworzona w technice RWD jest uniwersalna, co oznacza, że zarówno na komputerach jak i nośnikach mobilnych wyświetla się w ten sam sposób.

Responsywne strony internetowe

RWD – z angielskiego: "Responsive Web Design" jest techniką projektowania stron WWW, tak aby ich układ i wygląd automatycznie dostosowywał się do rozmiaru okna przeglądarki, na której strona jest wyświetlana. Strona WWW stworzona w technice RWD jest uniwersalna, co oznacza, że zarówno na komputerach jak i nośnikach mobilnych wyświetla się w ten sam sposób.

Strona responsywna jest wygodna do przeglądania, dzięki niej wielkość strony jest dopasowana do wyświetlacza używanego urządzenia i nie musimy przybliżać treści zawartych w witrynie. Co istotne, strony responsywne są lepiej postrzegane przez algorytmy Google, co wpływa na wyższą pozycję w wyszukiwarce i lepsze pozycjonowanie strony WWW.

Responsywne strony internetowe

RWD – z angielskiego: "Responsive Web Design" jest techniką projektowania stron WWW, tak aby ich układ i wygląd automatycznie dostosowywał się do rozmiaru okna przeglądarki, na której strona jest wyświetlana. Strona WWW stworzona w technice RWD jest uniwersalna, co oznacza, że zarówno na komputerach jak i nośnikach mobilnych wyświetla się w ten sam sposób.

Strona responsywna jest wygodna do przeglądania, dzięki niej wielkość strony jest dopasowana do wyświetlacza używanego urządzenia i nie musimy przybliżać treści zawartych w witrynie. Co istotne, strony responsywne są lepiej postrzegane przez algorytmy Google, co wpływa na wyższą pozycję w wyszukiwarce i lepsze pozycjonowanie strony WWW.

Za responsywność strony odpowiada m.in. metatag "viewport":

```
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
```

Z viewportem i bez

touch.facebook.com przedtem:

Welcome to Facebook | Facebook

facebook

Email or Phone:
mbrubeck@limpet.net

Password:
••••••••

Login

[Sign up](#) [Forgot your password?](#) [Full site](#)

touch.facebook.com teraz:

Welcome to Facebook | Facebook

facebook

Email or Phone:
mbrubeck@limpet.net

Password:
••••••••

Login

Z viewportem i bez

touch.facebook.com przedtem:



touch.facebook.com teraz:



@media

Podsumowanie

`@media` określa typ docelowych mediów (oddzielonych przecinkami) dla zestawu reguł (ograniczonych klamrami).

Składnia

```
@media <typy mediów> {
    /* reguły specyficzne dla medium */
}
```

Z viewportem i bez

touch.facebook.com przedtem:



touch.facebook.com teraz:



@media

Podsumowanie

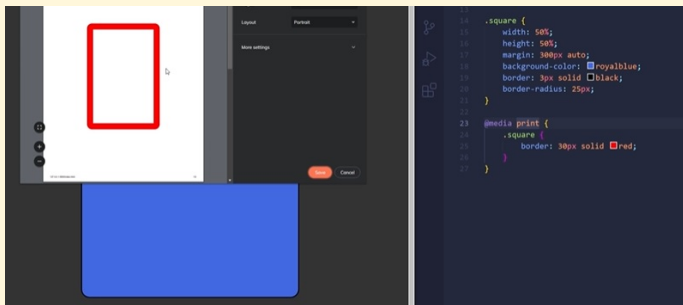
`@media` określa typ docelowych mediów (oddzielonych przecinkami) dla zestawu reguł (ograniczonych klamrami).

Składnia

```
@media <typy mediów> {
    /* reguły specyficzne dla medium */
}
```

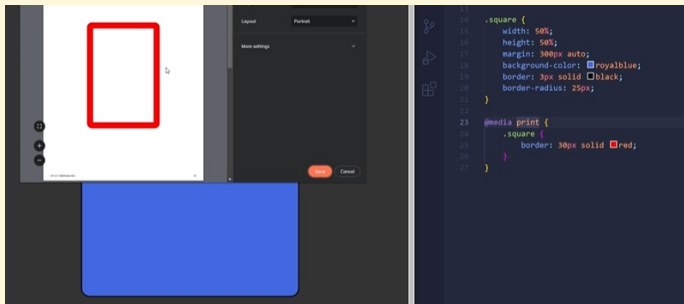
Jak nie ma wpisanego typu mediów, to jest domyślnie "all"

Przykład



Na stronie tego nie widać, ale przy wydruku przy kwadracie pojawia się czerwona ramka.

Przykład



Na stronie tego nie widać, ale przy wydruku przy kwadracie pojawia się czerwona ramka.

Przykład

```

@media (min-width: 500px) {
  .square {
    background-color: gold;
  }
}
  
```

W momencie, gdy szerokość ekranu przekroczy 500px, kwadrat zmienia kolor na żółty.