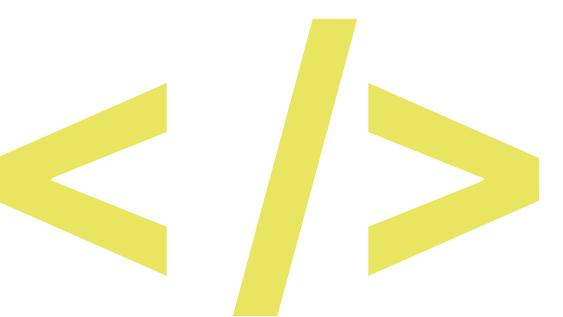


za<pro/>gramowani.com

</>

webmaster



kontakt

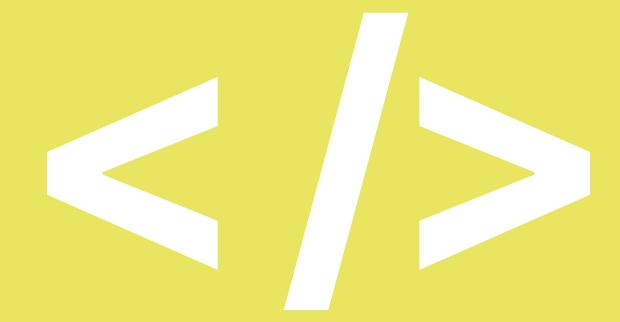
E-mail: hello@filiposinski.com

Discord: <https://discord.gg/eyhAtuxJcv>

GitHub: github.com/zaprogramowaniFO

Zajęcia nr 3

2023/11/07



czym jest CSS

Cascading Style Sheets (Kaskadowe Arkusze Stylów)

to język służący do opisu formy prezentacji naszej strony WWW. Sposobu w jaki się wyświetla.

Możemy w nim ustawić kolory, wielkości liter, marginesy itp.



Arkusze CSS

1. Arkusze CSS mogą być wspólne dla wielu dokumentów
2. Arkusz CSS to lista reguł określających wygląd elementów HTML
3. Przeglądarki różnie interpretują CSS
4. Pozwala na odseparowanie struktury dokumentu od formy jego prezentacji

Warto testować naszą stronę w różnych przeglądarkach aby sprawdzić poprawność wyświetlania naszego kodu. Czasem może się zdarzyć konieczność tworzenia kilku wersji arkusza CSS pod różne przeglądarki.

Zasady tworzenia

Aby utworzyć arkusz css i móc go używać należy:

1. Utworzyć nowy arkusz o rozszerzeniu **CSS**;
2. Arkusz należy “przypiąć” do pliku strony z rozszerzeniem HTML;

Zrobimy to dodając odpowiednią linijkę do naszego kodu strony:

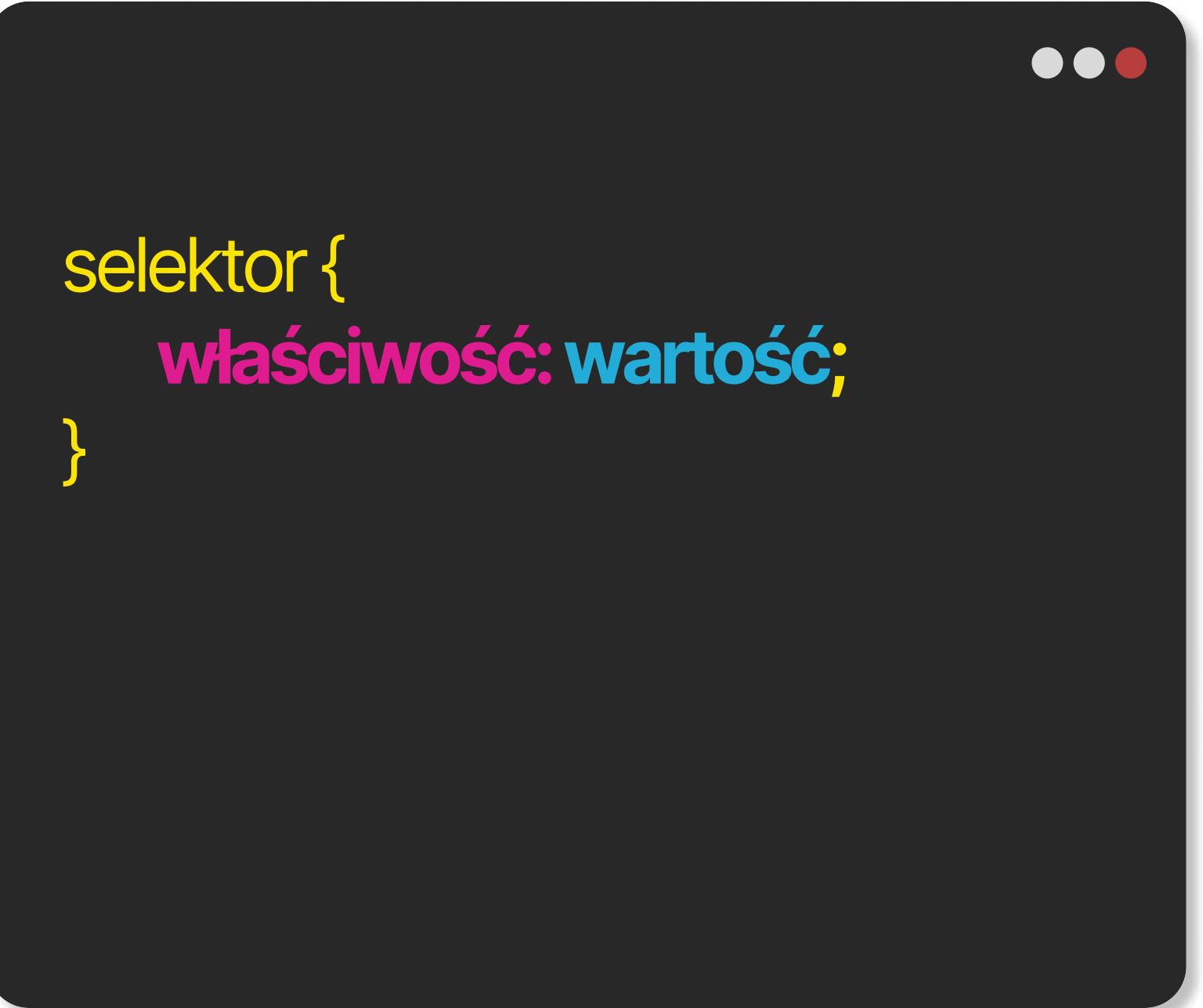
```
<link href="ściezka pliku" rel="stylesheet" type="text/css" />
```

Budowa reguły CSS

Arkusz CSS składa się z poszczególnych **reguł** (odpowiadającym konkretnym znacznikom, które używamy w naszym dokumencie).

Deklaracje dla każdego z selektorów zamykamy w **klamrach {}**

Kolejne **właściwości** należy oddzielać od siebie **średnikami ;**

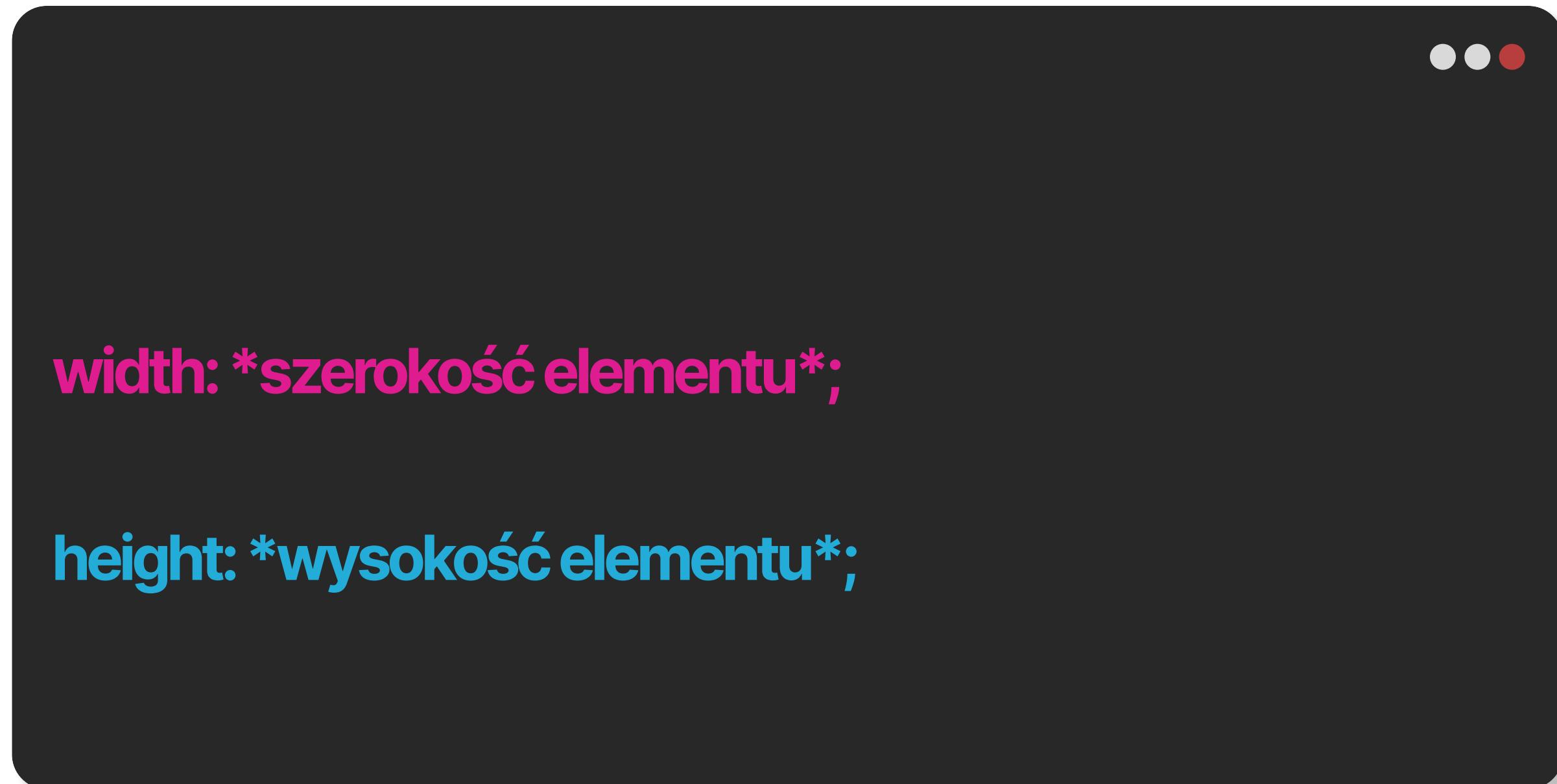


```
selektor {  
    właściwość: wartość;  
}
```



podstawowe właściwości

Parametry wielkości elementu



Możemy elementom nadawać wysokość i szerokość.

Jako jednostek wartości możemy używać m.in. pikseli lub procentów.

Kolor na stronie



color: #FF0000;	system szesnastkowy koloru
color: rgb(255, 0, 0);	liczby składowe koloru
color: rgb(100%, 0%, 0%);	procenty składowe koloru
color: red;	słowo kluczowe koloru

Aby użyć konkretnego koloru w naszym kodzie na stronie możemy użyć jednej z właściwości po lewej stronie.

Kolor na stronie

```
...  
p{  
    color: red;  
}  
  
body{  
    background-color: blue;  
}
```

Aby dodać kolor elementowi opisanym konkretnym **znacznikiem** np. **<p>**, musimy stworzyć dla niego selektor i użyć właściwości **color** i przypisać kolor.

Aby ustawić kolor tła elementu, musimy użyć właściwości **background-color** i przypisać kolor.

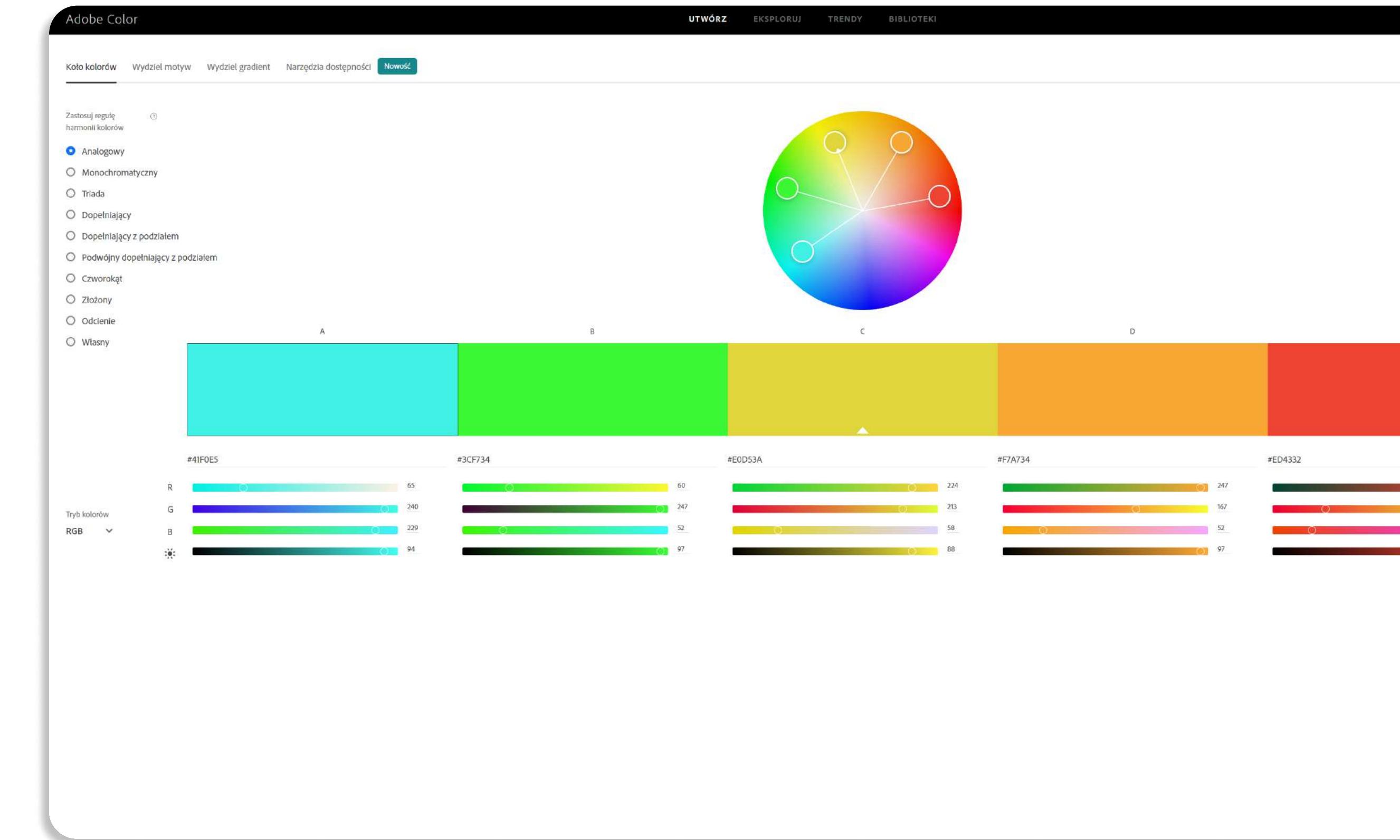
ADOBE COLOR

W doborze kolorów możemy się posłużyć różnymi stronami, np.

<https://color.adobe.com/>

Strona ta pozwala nam wybrać kolor oraz dobrać kolory, które będą się z nim dobrze komponować.

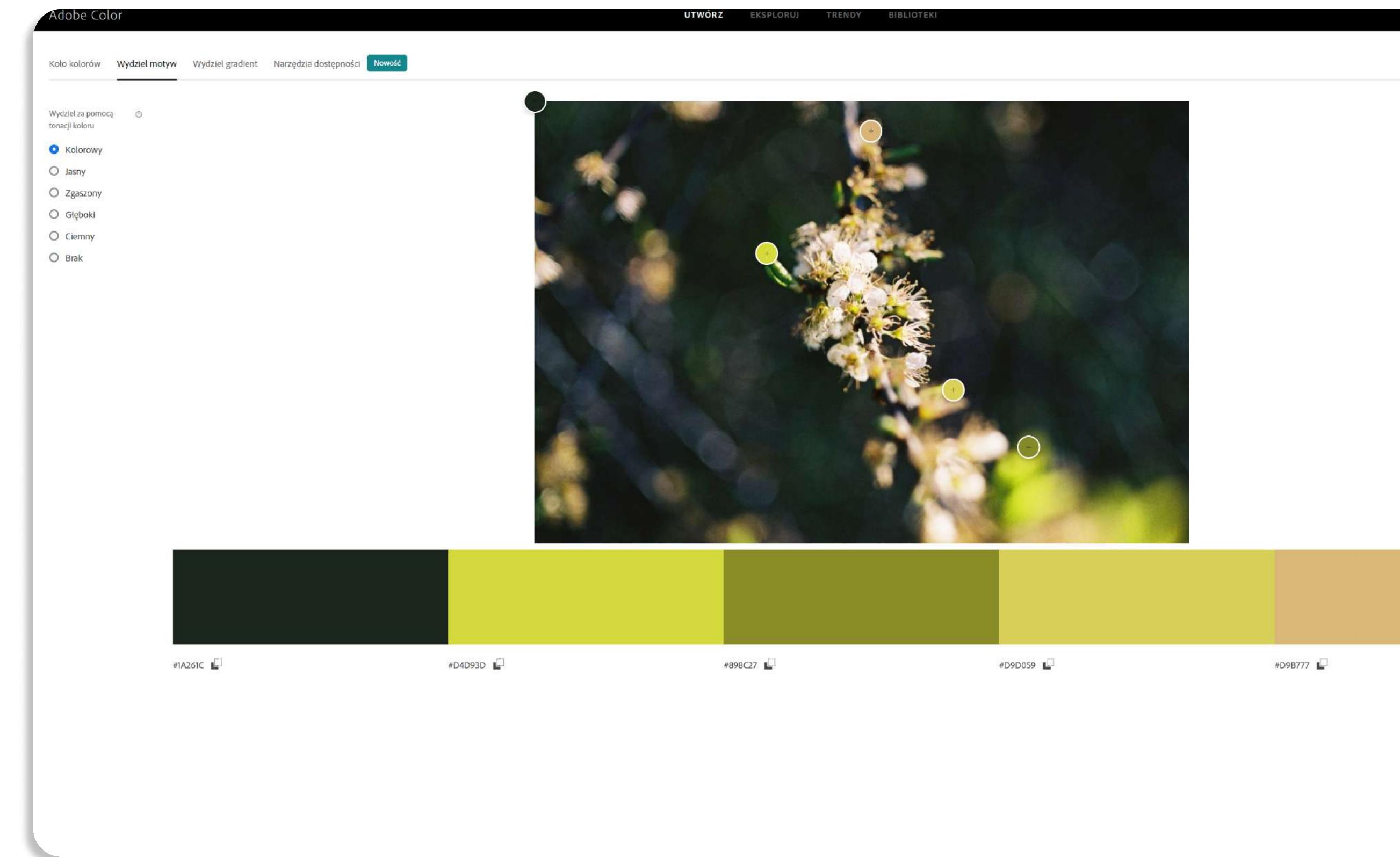
Pozwala również na uzyskanie odpowiedniego oznaczenia wybranego przez nas koloru, który następnie możemy użyć w naszym kodzie.



ADOBE COLOR

Na stronie tej możemy również dobrać kolory na podstawie naszego zdjęcia, które np. chcemy umieścić na naszej stronie i zależy nam aby kolorystyka pasowała do fotografii.

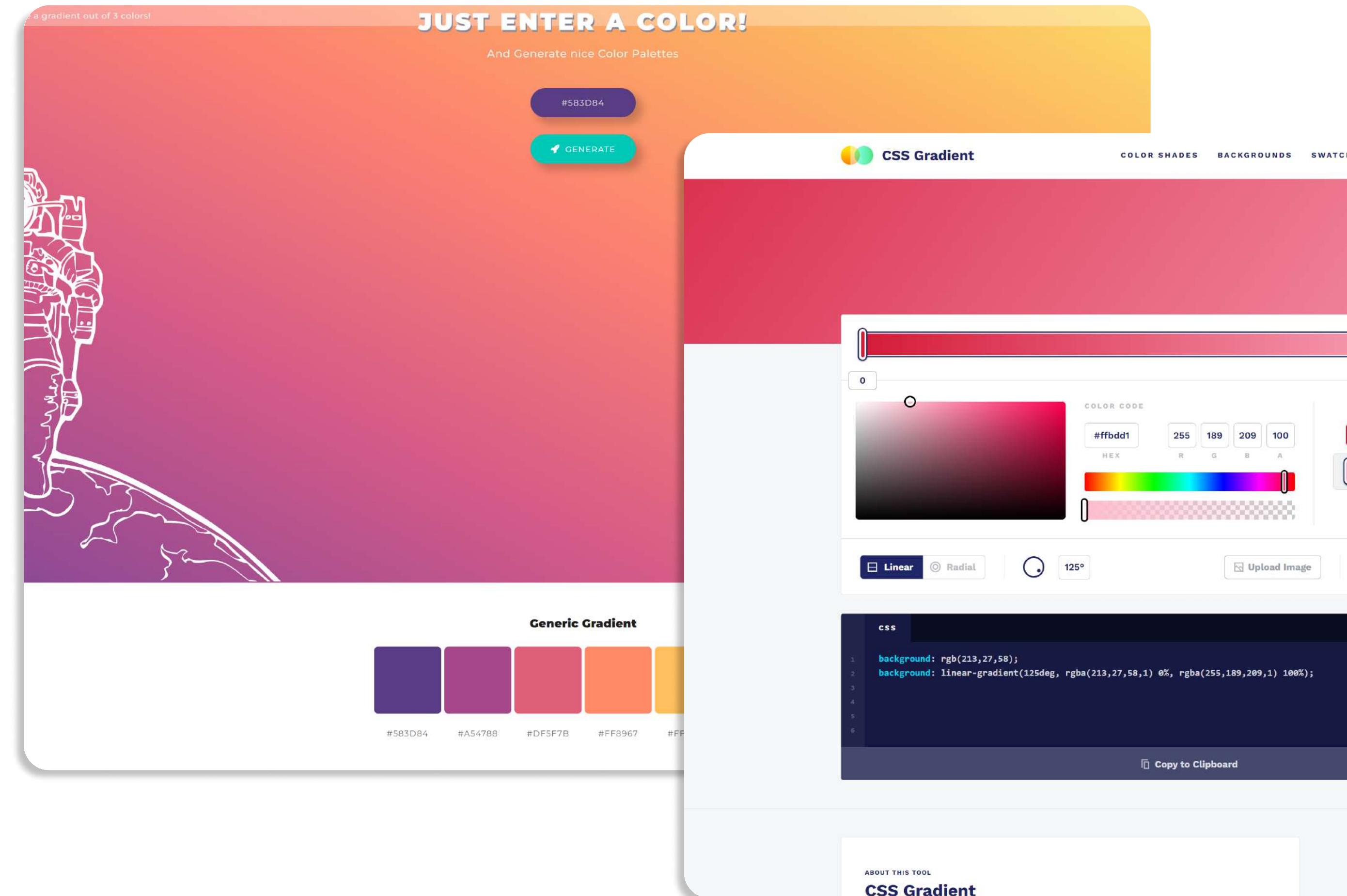
Istnieją również aplikacje mobilne, które pozwalają na to samo



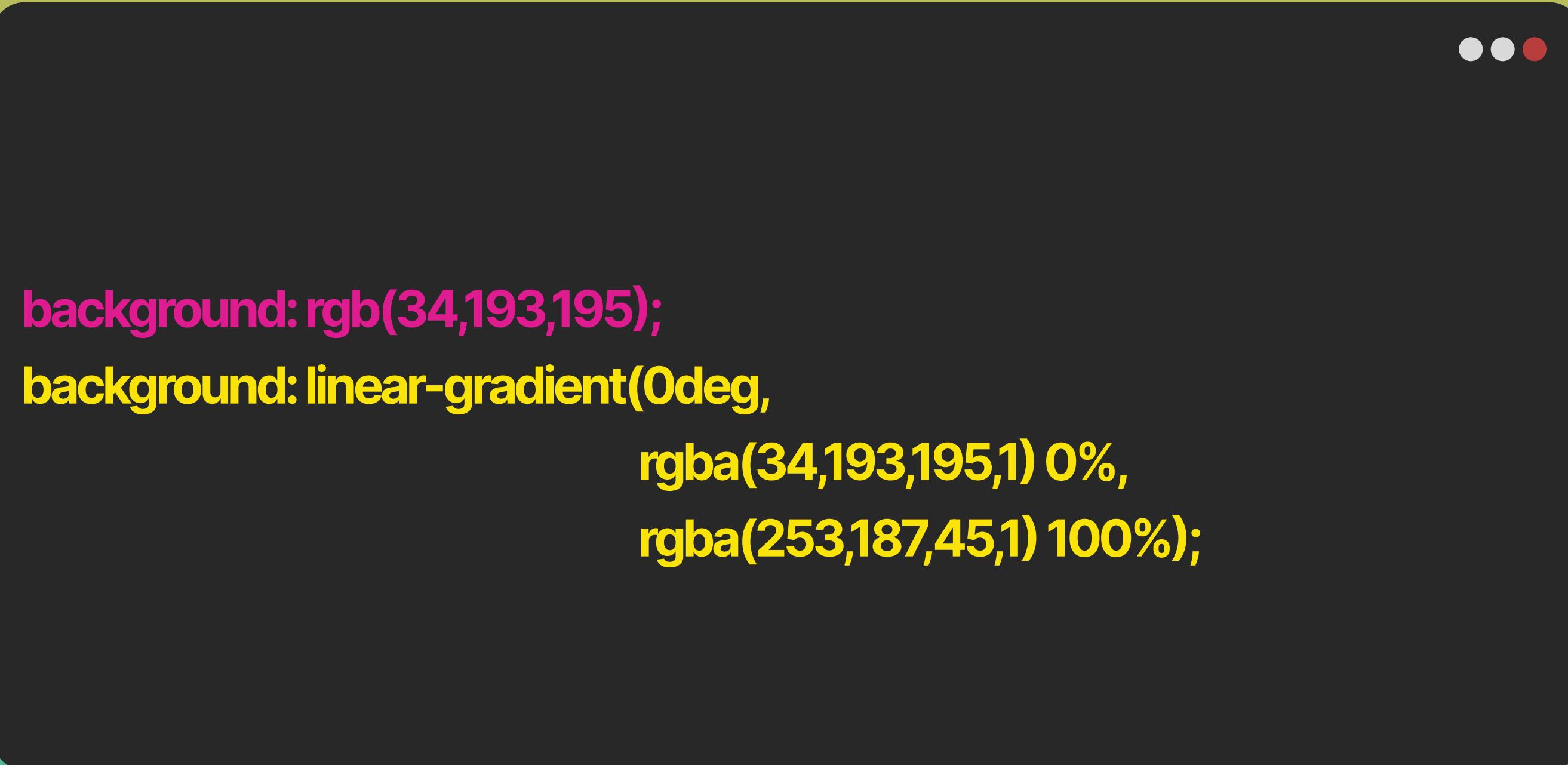
Inne rozwiązania

Innymi wartymi uwagi stronami , które również umożliwiają nam na generowanie ciekawych palet kolorów, oraz gradientów są:

- <https://cssgradient.io>
- <https://mycolor.space>



Gradient



Na stronie oprócz jednolitego koloru w tle możemy również użyć m.in. gradientu.

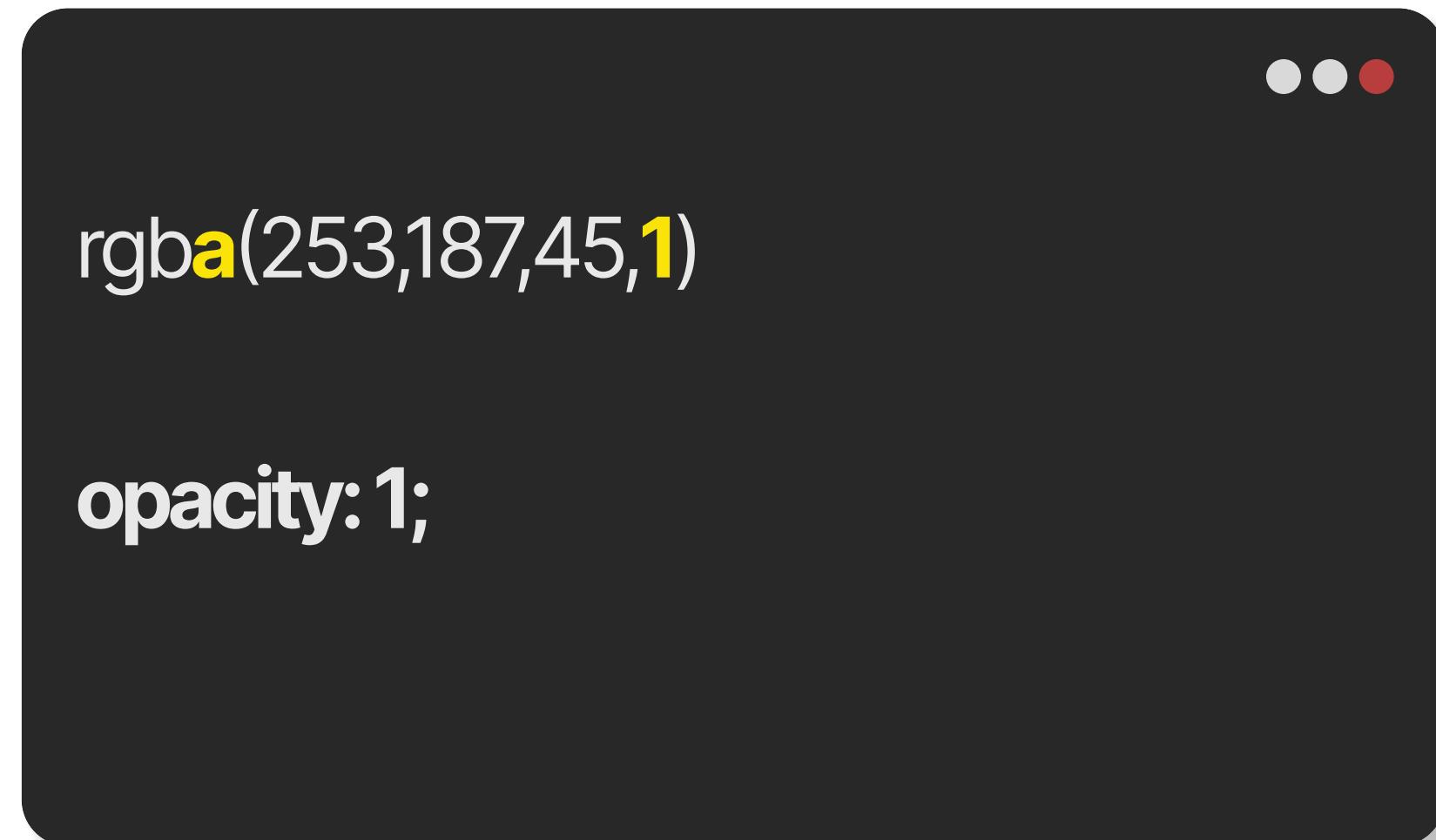
W tym celu posłużymy się właściwością **background**.

Jako wartość podamy wartość **linear-gradient** o wartościach określających **kąt gradientu**

oraz dowolną liczbę kolorów i procentowe miejsce punktu największej intensywności.

Możemy również użyć wartość **circle-gradient** który utworzy nam gradient kołowy.

Kanał Alpha



rgba(253,187,45,**1**)

opacity: 1;

Dodatkowa czwarta wartość to **kanał alfa**, który odpowiada za transparentność koloru.

Wartość tak może być określona od 0 do 1, lub od 0% do 100%

Podobnie działa właściwość **opacity**, która pozwala określenie transparentości dla całego elementu.

Parametry tekstu

Tekst umieszczony na stronie możemy opisać i zdefiniować różnymi właściwościami.



font-size: *rozmiar*;

font-family: 'wybrana rodzina', *szeryfowość*;

color: *kolor*;

text-align: *właściwość*;

line-height: *wartość*;

letter-spacing: *wartość*;

wielkość tekstu;

zdefiniowanie rodzaju

fontu naszego tekstu;

zdefiniowanie koloru tekstu;

charakter tekstu: wyjustowany,

wyśrodkowany, lewa, prawa;

wysokość linii tekstu;

przerwa między literami;

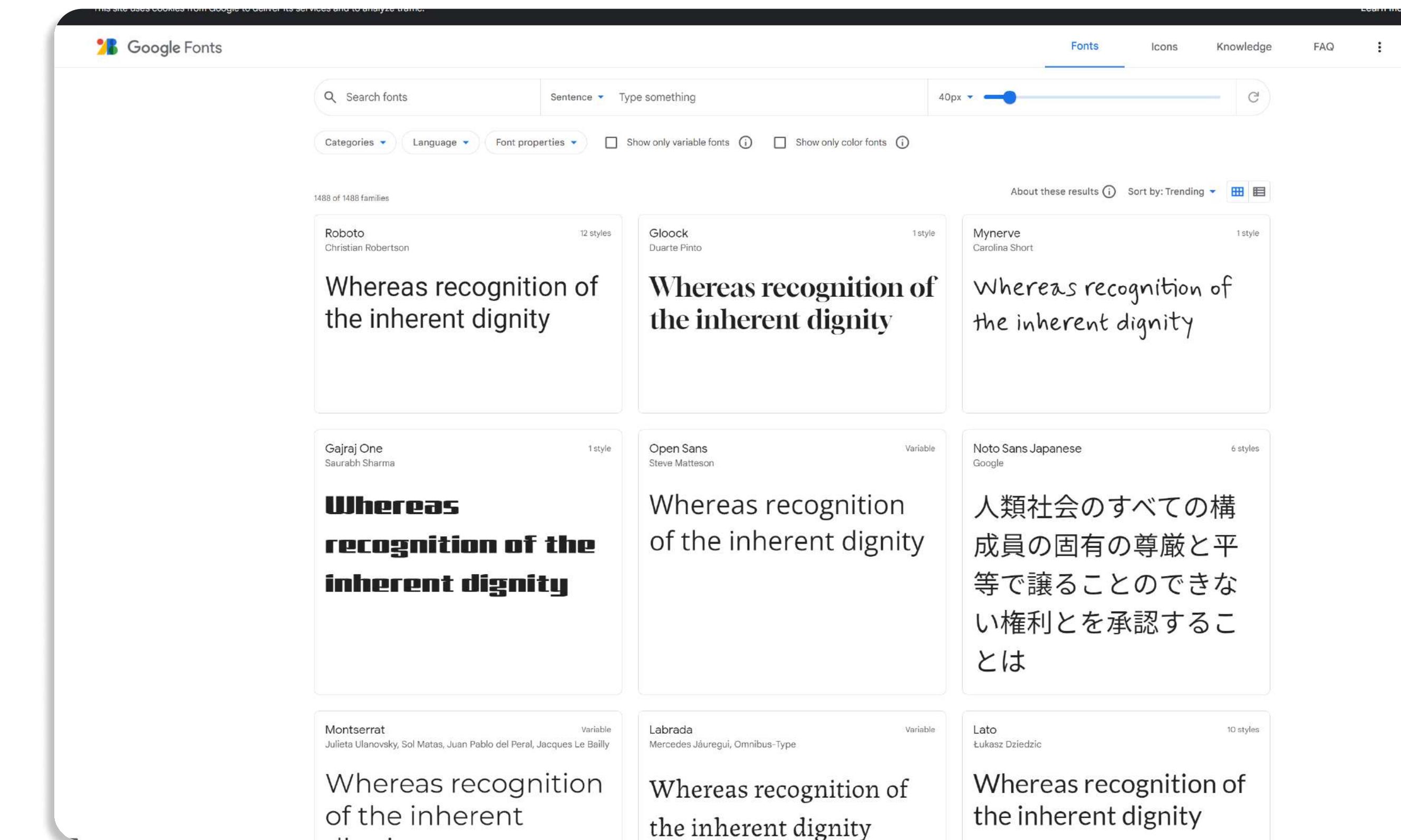
GOOGLE FONTS

W doborze fontów możemy się posłkować różnymi stronami, np.

<https://fonts.google.com>

Strona ta pozwala nam wybrać atrakcyjnego kroju pisma, który będzie odpowiadał naszemu projektowi.

Krój ten możemy w prosty sposób zaimplementować na naszej stronie linkując go w arkuszu stylów.



Grupowanie selektorów

```
p{  
    color: red;  
}  
  
h1{  
    color: red;  
}
```

Jeżeli kilka stworzonych przez nas selektorów ma te same parametry, np kolor, to możemy je zgrupować.

Należy jednak pamiętać, że jeżeli dodamy do tak zgrupowanych selektorów kolejną właściwość, to będzie ona nadana wszystkim zgrupowanym elementom!

```
p, h1{  
    color: red;  
}
```

`</>`klasy oraz ID selektorów

Klasy oraz ID selektorów

Znacznikom możemy nadać klasę lub ID

Identyfikator (ID) jest unikalny dla danego elementu, i może nim zostać zdefiniowany tylko jeden element na stronie.

W kodzie CSS odwołujemy się do identyfikatora poprzez '#'.

Klasa (class) może być przypisana do wielu elementów.

W kodzie CSS odwołujemy się do klasy poprzez '.' .

Nadawanie klas



Nazwy naszych klas należy nadawać w taki sposób aby odwoływały się do treści, którą zawierają.

Dobrą praktyką jest używanie stylowania za pomocą klas wszystkich znaczników!

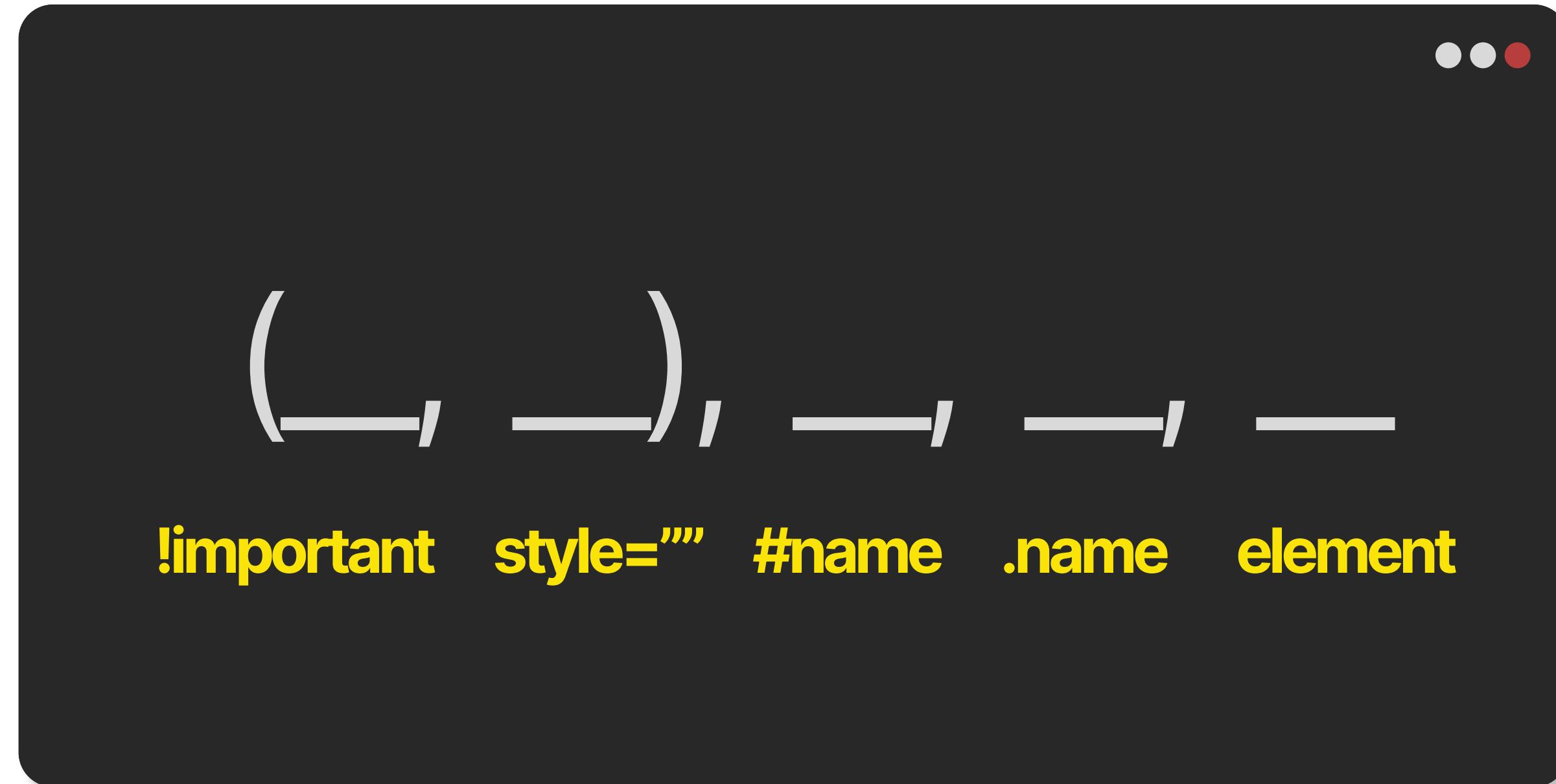
Nie powinno się stylować znaczników używając ID!

Klasy oraz ID selektorów

Zasady konstruowania nazw:

- używaj pojedynczych słów (najlepiej z j. angielskiego);
- nazywaj rzeczy zgodnie z funkcją na stronie;
- nazwy zaczynaj z małej litery;
- nie zaczynaj od cyfr;
- używaj zasady **camelCase**, jeżeli chcesz użyć dwóch słów w nazwie
- staraj się, aby każdy element miał tylko jedną klasę

Specyficzność



Selektory rozpatrujemy od lewej do prawej.

Jeżeli, któryś z selektorów bliżej lewej został użyty, to jest on ważniejszy i zostanie uwzględniony niż kilka tych samych selektorów mniejszej rangi.

Podobnie jak medale na olimpiadzie:

1 złoty medal > 3 medale srebrne

 zasada **BEM**

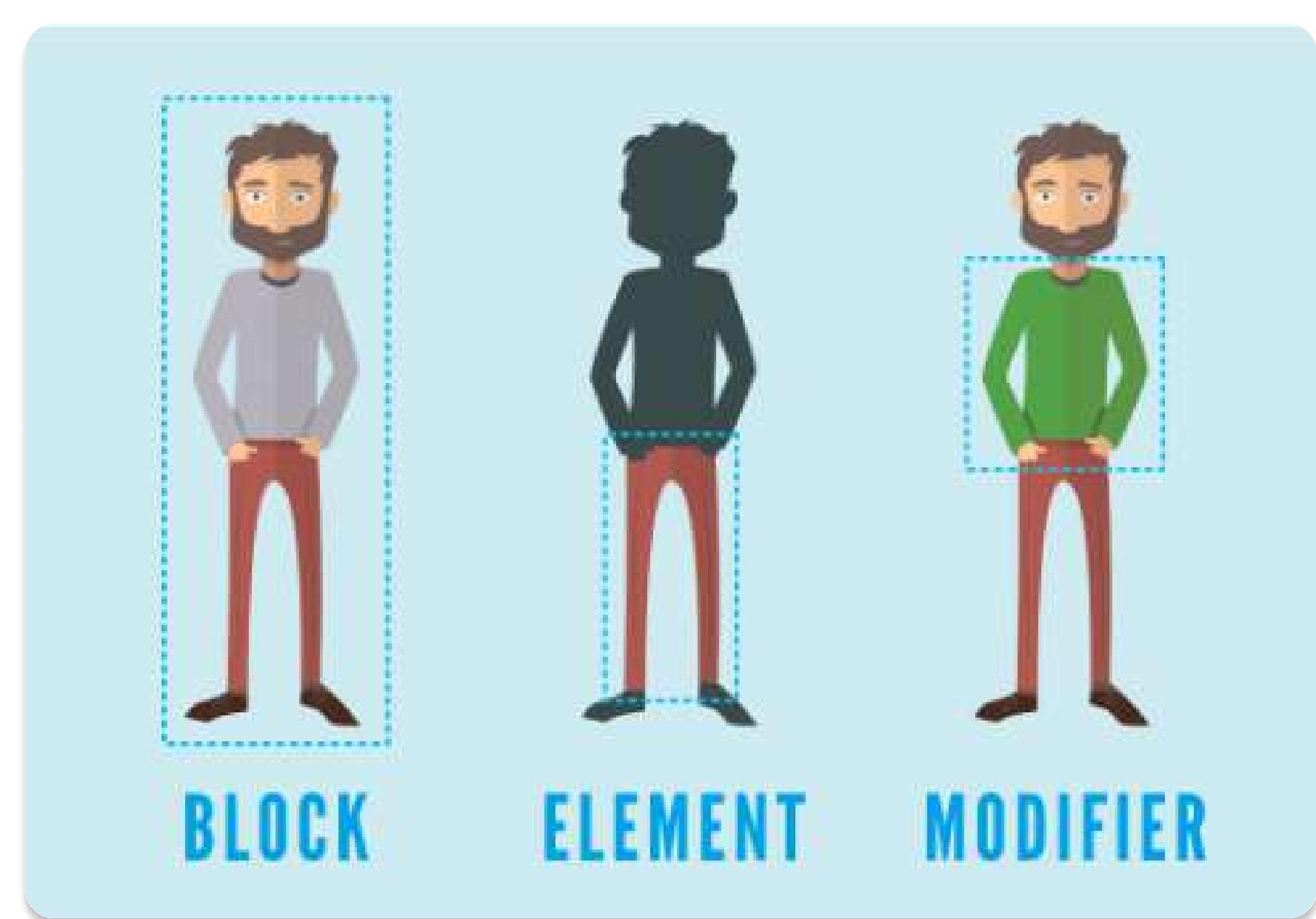
BLOCK_ELEMENT--MODIFIER

Zasady te zostały wprowadzone aby uporządkować i ujednolicić sposób stylowania elementów w html.

Blok - jest to element nadzędny

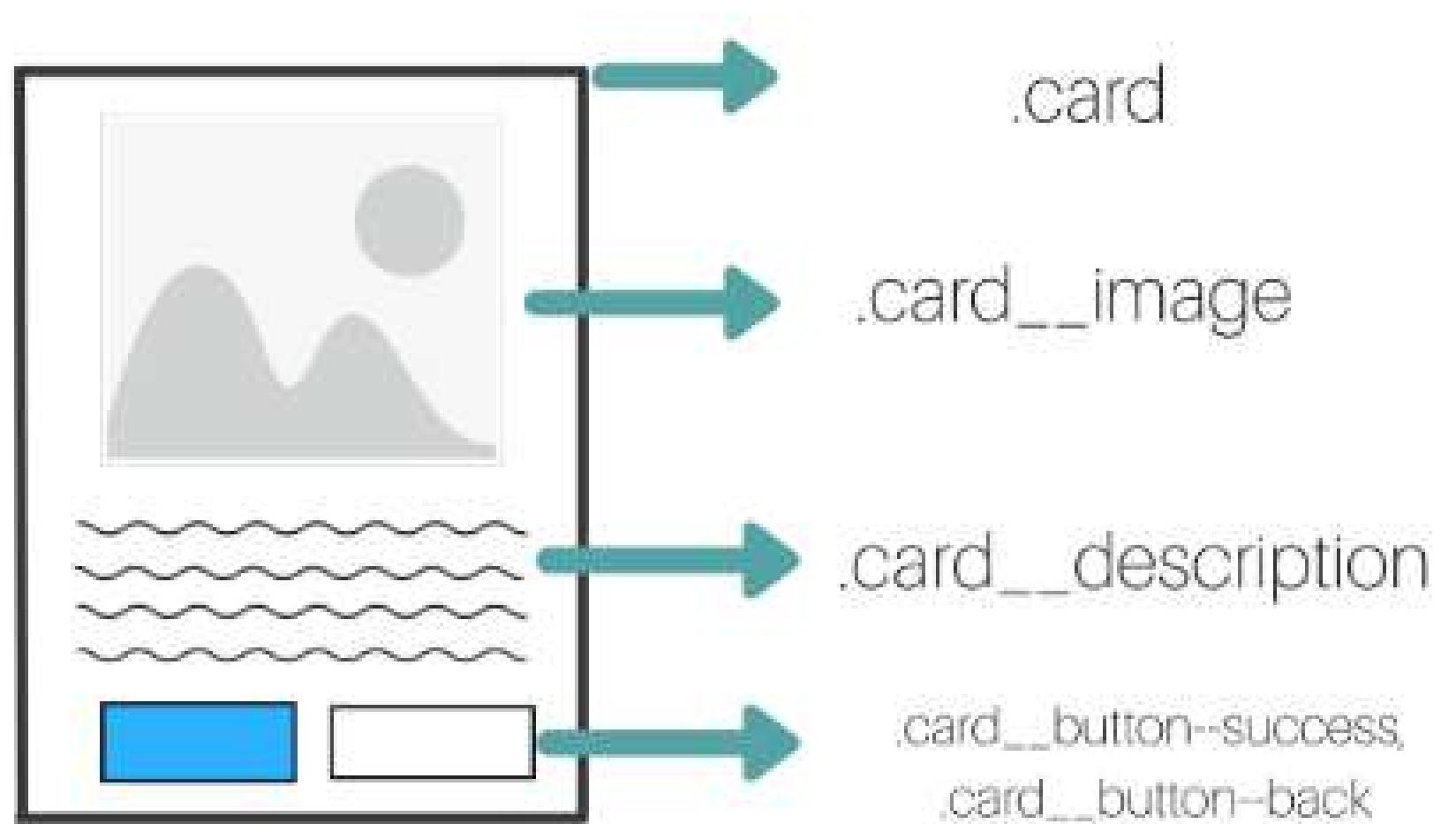
Element - część bloku

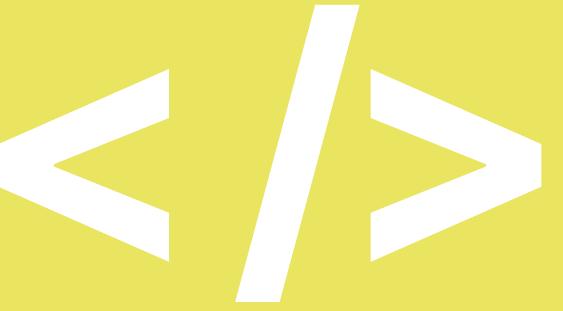
Modyfikator - wyróżniony element



Nadawanie klas

```
<div class="card">  
  <div class="card__image"></div>  
  <div class="card__description"></div>  
  <div class="card__button card__button--success"></div>  
  <div class="card__button card__button--back"></div>  
</div>
```





DIV vs SPAN vs SECTION

Różnice

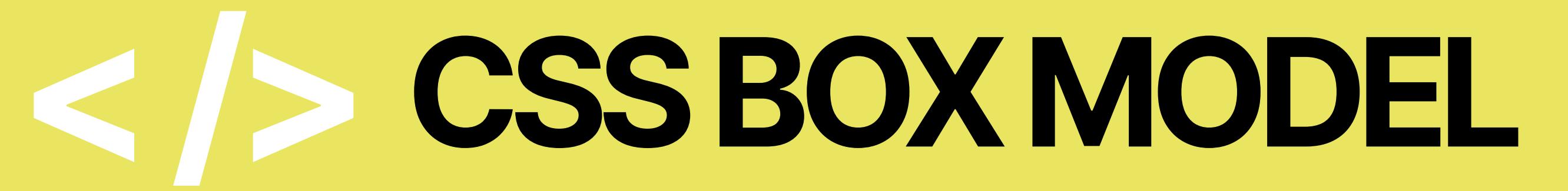
<div>

- pojemnik na treść;
- nie ma znaczenia semantycznego;
- używany do grupowania zawartości;
- element blokowy;

- pojemnik na treść;
- nie ma znaczenia semantycznego;
- używany do stylizacji tekstu;
- element liniowy;

<section>

- pojemnik na treść;
- ma znaczenie semantyczne;
- służy do tematycznego opakowywania;



</> CSS BOX MODEL

CSS BOX MODEL

Model pudełkowy pozwala przedstawiać i opisywać treść. Pozwala na określanie wygląd i układu poszczególnych elementów.

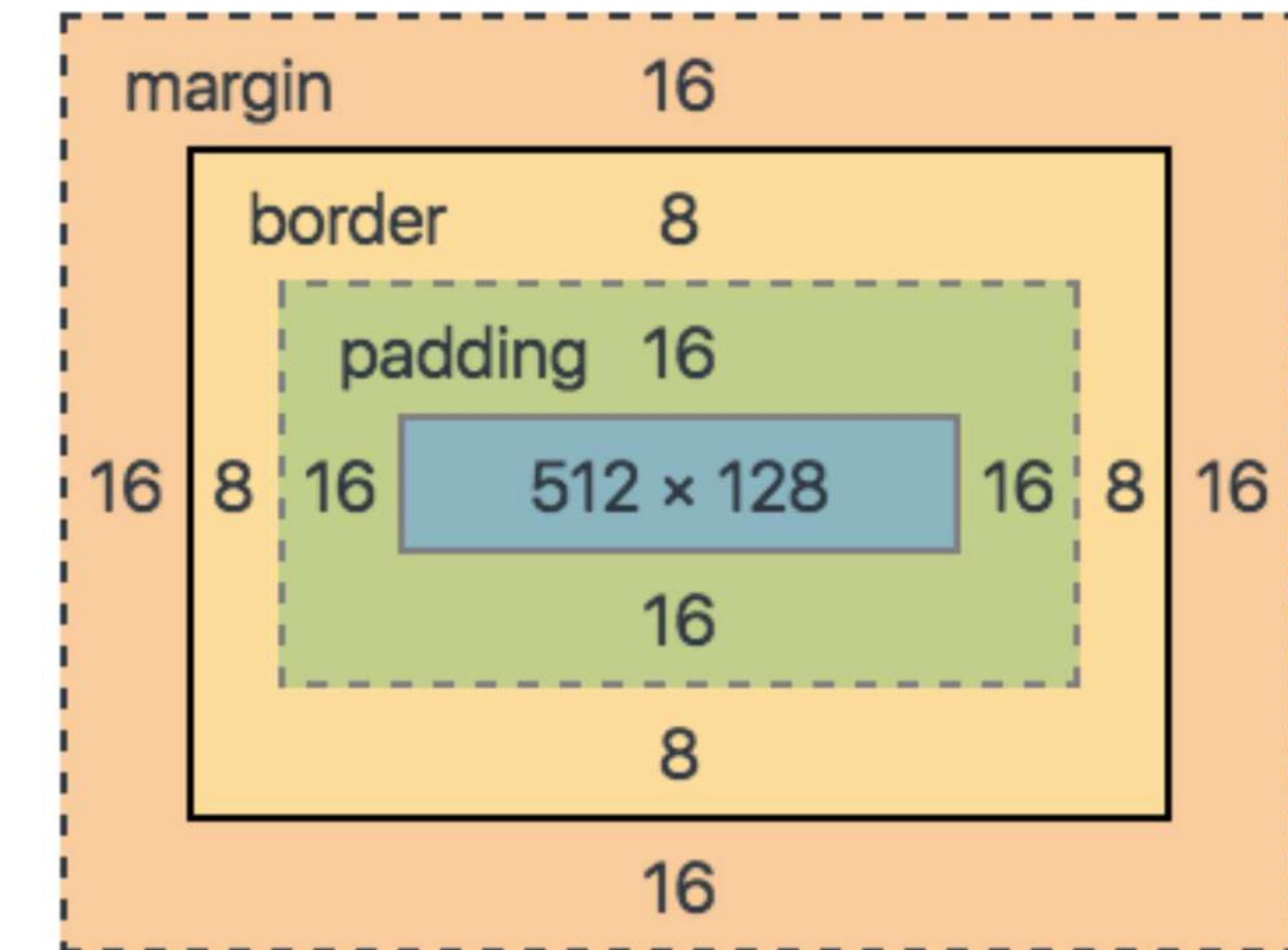
Na model pudełkowy składają się poniższe elementy:

Niebieski kolor to nasza treść.

Kolor zielony to wewnętrzny margines.

Żółty to grubość ramki.

Czerwony to zewnętrzny margines.



Argumenty w selektorach



Każdy z wymiarów możemy opisać pojedynczo, zgrupować w pary lub podać jednej właściwości wszystkie.

Należy pamiętać o zasadzie zegara.

Border, czyli ramka

Border: *rozmiar* ;

Border-style: *wybrany styl*;

border-color: *kolor*;

Border-radius: *wartość zaokrąglenia narożników*;

Border: *wymiar* *styl* *kolor*;

grubość ramki

zdefiniowanie stylu ramki, np. Solid,
dotted, dashed

zdefiniowanie koloru ramki

zaokrąglenie narożników

można również zdefiniować ramkę
w ten sposób

Możemy również
zdefiniować ramkę na
wiele sposobów.

 **JEDNOSTKI**

Jednostki

W zależności od naszych potrzeb rozróżniamy różne jednostki miary, które możemy użyć w naszych projektach, m.in.:

- jednostki relatywne względem fontów
- jednostki relatywne względem okna (viewport)
- jednostki relatywne względem rodzica
- jednostki absolutne
- wartości bezjednostkowe

Jednostki relatywne względem fontów

em - odwołuje się do bezpośredniego rodzica elementu

rem - odwołuje się do globalnej wielkości tekstu

Jednostki relatywne względem fontów

vh - viewport height, relatywny względem wysokości okna

vw - viewport width, relatywny względem szerokości okna

vmin - minimum relatywne względem okna

vmax - maksimum relatywne względem okna

Jednostka relatywna względem rodzica

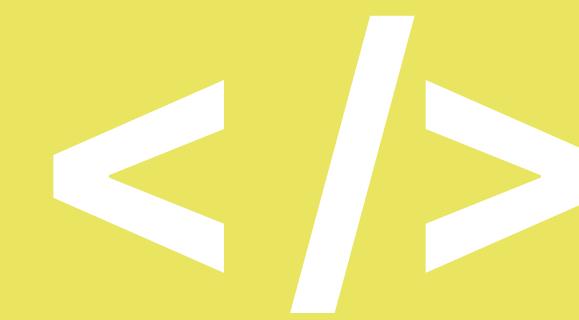
%- jednostka procentowa relatywna względem rodzica

Jednostka absolutna, niezależna względem innych elementów

px - jednostka absolutna niezależna od innych elementów

Wartości bezjednostkowe

**margin: 0;
opacity: 1;
line-height: 1.5;**

 **RESET i NORMALIZACJA**

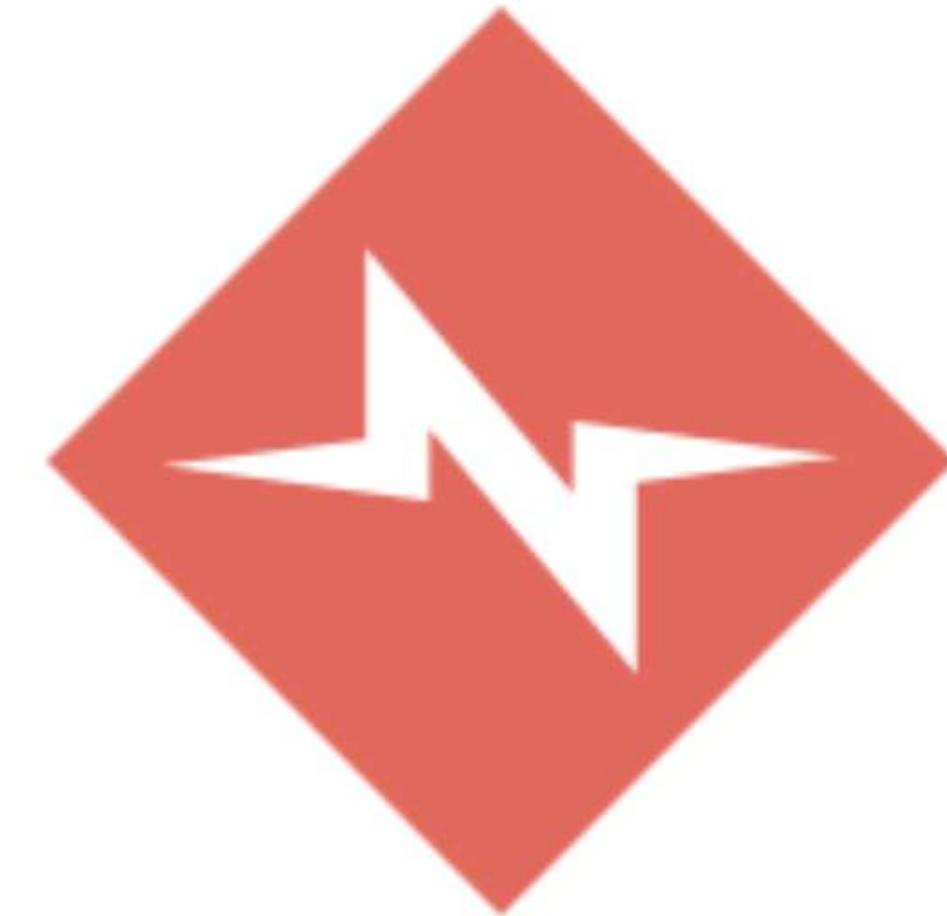
Reset i normalizacja

<https://necolas.github.io/normalize.css/>

Każda przeglądarka ma własne domyślne style CSS. Powoduje to, że nasza strona będzie na każdej przeglądarce wyglądać trochę inaczej.

Dlatego dobrą praktyką jest resetowanie ustawień domyślnych aby uspójnić wygląd.

Aktualnie, najpopularniejszym rozwiązaniem, jest użycie **normalizacji**, która standaryzuje interpretację znaczników przez różne przeglądarki.



Normalize.css

Reset i normalizacja

```
html {  
    box-sizing: border-box;  
}  
  
*,  
*:after,  
*:before {  
    box-sizing: inherit;  
}
```

Dodatkowo, możemy dołożyć w naszym pliku głównym CSS właściwość, która będzie dziedziczyć pozwalać na dziedziczenie wielkości przez elementy.