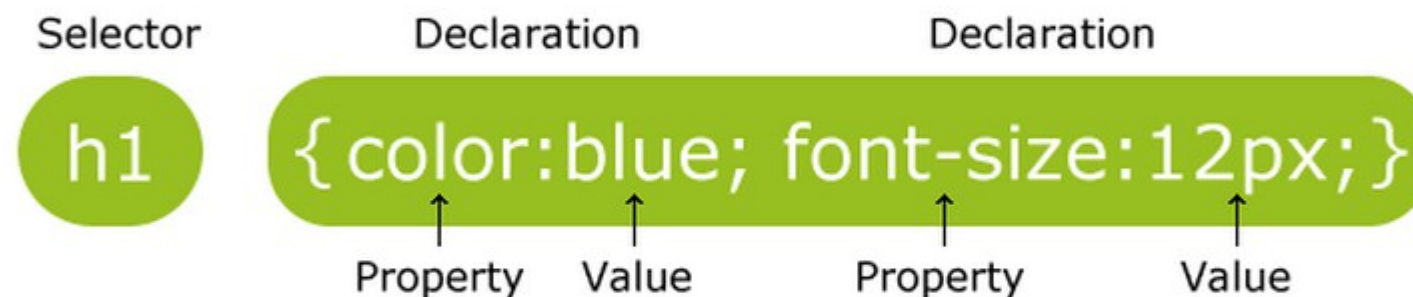


# CSS Cascading Style Sheets

*"Kaskadowy Arkusz Stylów" jest to plik z rozszerzeniem .css, w którym znajdują się reguły CSS, które określają wygląd różnych elementów HTML znajdujących się na danej stronie internetowej.*

## CSS Syntax

A CSS rule set consists of a selector and a declaration block:



The selector points to the HTML element you want to style.

The declaration block contains one or more declarations separated by semicolons.

Each declaration includes a property name and a value, separated by a colon.

# background

CSS properties used for background effects:

- background-color
- background-image
- background-repeat
- background-attachment
- background-position

# background-color

# Tło elementu body

```
7 <body bgcolor="rgb(24,12,250)">
```



Atrybut bgcolor nie jest wspierany w HTML5  
(występuje w pytaniach egzaminacyjnych)

## Użycie CSS

inline

```
<body style="background-color: rgb(7,12,250);">
```

W sekcji head dokumentu HTML

```
<style>
body{
background-color: rgb(127,12,250);
}
</style>
```

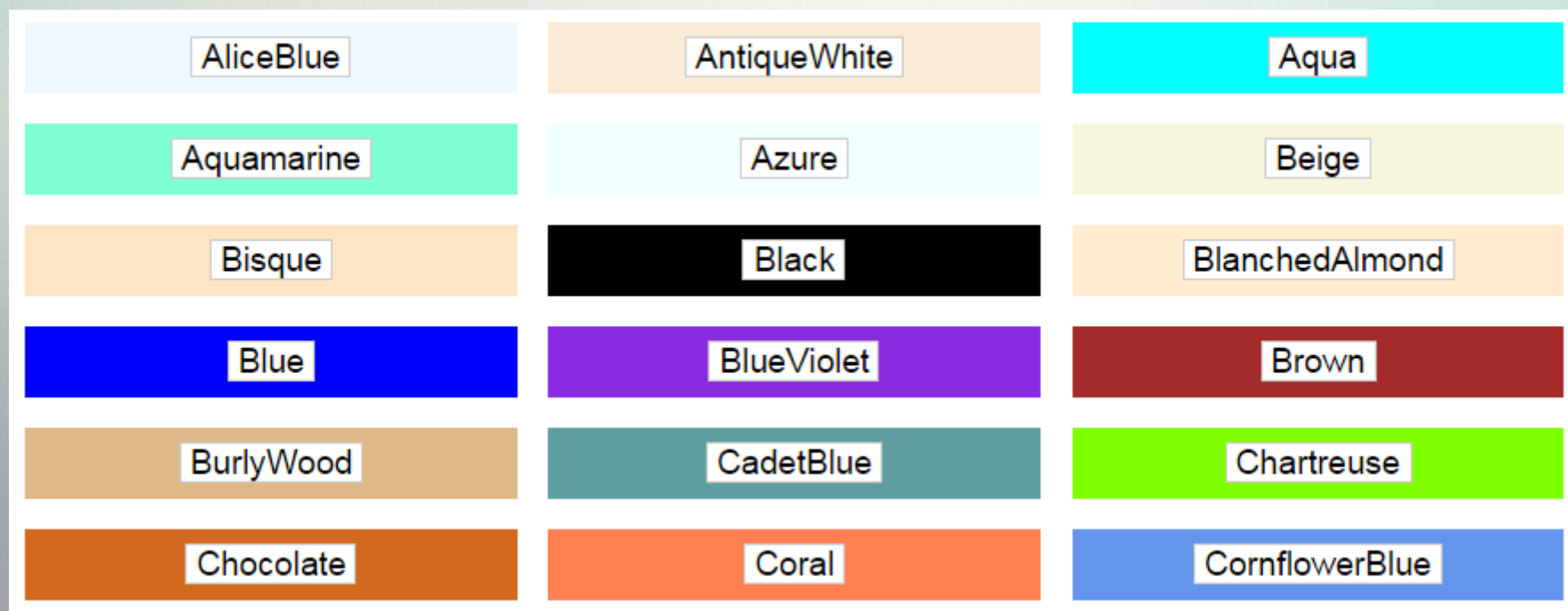
W sekcji head dokumentu HTML-  
Odwołanie do zewnętrznego pliku

```
1 body {
2   background-color: rgb(123,100,2);
3 }
```

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

# Kolory w CSS3

**1. paleta kolorów podstawowych, predefiniowanych - do określenia koloru możemy posłużyć się jego angielską nazwą. Paleta kolorów podstawowych, predefiniowanych jest mocno ograniczona (około 140 kolorów).**



# Kolory w CSS3- zapis heksadecymalny

*2. zapis heksadecymalny – szesnastkowy. Składa się on z sześciu znaków poprzedzonych znakiem #. Pierwsze dwa znaki definiują kolor czerwony, kolejne dwa znaki kolor zielony, ostatnie dwa znaki kolor niebieski.*

```
body { background-color: #EFD9F4; }
```

# Kolory w CSS3-rgb

## 3. format rgb:

- `rgb(red, green, blue)`- w nawiasie znajdują się trzy liczby oddzielone przecinkiem. Każda z tych liczb określa intensywność barw, kolejno: czerwonej, zielonej i niebieskiej. Do określania tej intensywności możemy używać wyłącznie liczb z zakresu 0 - 255 lub wyłącznie wartości procentowych z zakresu 0% - 100%.

`rgb(255, 0, 0)` - kolor czerwony

`rgb(0, 255, 0)` - kolor zielony

`rgb(0, 0, 255)` - kolor niebieski

`rgb(100%, 0%, 0%)` - kolor czerwony

`rgb(0%, 100%, 0%)` - kolor zielony

`rgb(0%, 0%, 100%)` - kolor niebieski

```
body { background-color: rgb(212,34,123); }
```



# Kolory w CSS3-rgba

## 4. format rgba:

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

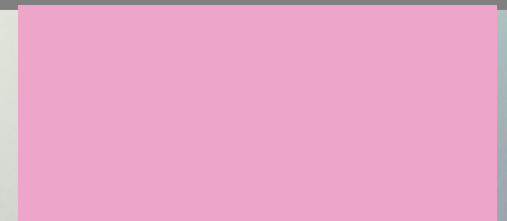
Parametr alpha jest liczbą zawartą między 0.0 (fully transparent) and 1.0 (fully opaque).

```
rgba(255, 0, 0, 0.2);  
rgba(255, 0, 0, 0.4);  
rgba(255, 0, 0, 0.6);  
rgba(255, 0, 0, 0.8);
```

```
body { background-color: rgb(212,34,123); }
```



```
body { background-color: rgba(212,34,123,0.4); }
```



RGBA colors:

Red

Green

Blue

```
{background-color: rgba(255, 0, 0, 0.3);} /* red with opacity */  
{background-color: rgba(0, 255, 0, 0.3);} /* green with opacity */  
{background-color: rgba(0, 0, 255, 0.3);} /* blue with opacity */
```



# Kolory w CSS3-hsl

## 5. format hsl - *hsl(odcień, nasycenie, jasność)* *hsl(hue, saturation, lightness)*

**HSL** – jeden z modeli opisowych dla kolorów postrzeganych przez ludzi. Ten sposób opisowy miał polegać na tym, że każdej barwie postrzeganej przez człowieka jest przyporządkowany jeden punkt w przestrzeni trójwymiarowej identyfikowany przez trzy składowe:  $(h,s,l)$ . Model pojawił się w okresie startu telewizji – pierwsze demonstracje w latach 1926-1930.

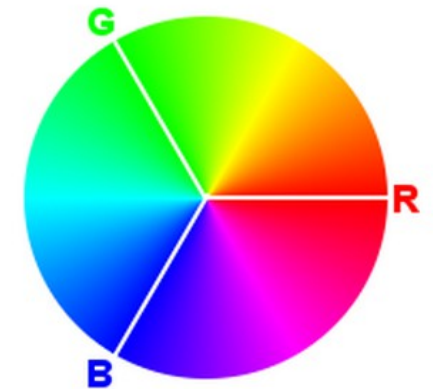
Znaczenie i zakresy współrzędnych:







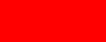
**H:** Hue – (z ang. *odcień*, barwa), o wartościach z przedziału: od 0 do 360 stopni.

**S:** Saturation – *nasycenie koloru*. z przedziału 0...1 albo 0...100%.






**L:** Lightness – *średnie światło białe*, z przedziału 0...1 albo 0...100%.



Sterowanie nasyceniem barwy czerwonej:

Definicja	Kolor
<code>hsl(0, 0%, 50%)</code>	
<code>hsl(0, 25%, 50%)</code>	
<code>hsl(0, 50%, 50%)</code>	
<code>hsl(0, 75%, 50%)</code>	
<code>hsl(0, 100%, 50%)</code>	

Sterowanie jasnością barwy czerwonej:

Definicja	Kolor
<code>hsl(0, 100%, 0%)</code>	
<code>hsl(0, 100%, 25%)</code>	
<code>hsl(0, 100%, 50%)</code>	
<code>hsl(0, 100%, 75%)</code>	
<code>hsl(0, 100%, 100%)</code>	

# Kolory w CSS3-hsla

## 5. format hsla - *hsla(odcień, nasycenie, jasność, stopień przezroczystości)* *hsl(hue, saturation, lightness, alpha)*

An HSLA color value is specified with: `hsla(hue, saturation, lightness, alpha)`, where the alpha parameter defines the opacity. The alpha parameter is a number between 0.0 (fully transparent) and 1.0 (fully opaque).

```
hsla(0, 100%, 30%, 0.3);  
hsla(0, 100%, 50%, 0.3);  
hsla(0, 100%, 70%, 0.3);  
hsla(0, 100%, 90%, 0.3);
```

HSLA colors:

Green

Light green

Dark green

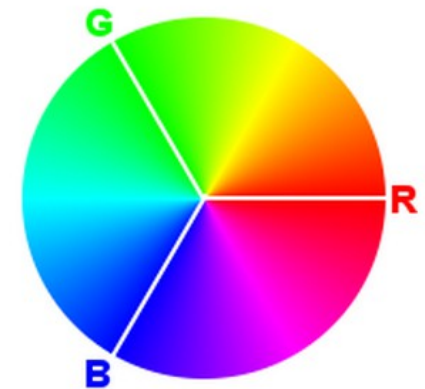
Pastel green

Violet

Pastel violet



```
{background-color:hsla(120,100%,50%,0.3);}  
{background-color:hsla(120,100%,75%,0.3);}  
{background-color:hsla(120,100%,25%,0.3);}  
{background-color:hsla(120,60%,70%,0.3);}  
{background-color:hsla(290,100%,50%,0.3);}  
{background-color:hsla(290,60%,70%,0.3);}
```



# Kolory w CSS3-rgb+opacity

## 6. format rgb + opacity:

The CSS3 opacity property sets the opacity for a specified RGB value.

The opacity property value must be a number between 0.0 (fully transparent) and 1.0 (fully opaque).

```
rgb(255, 0, 0);opacity:0.2;
```

```
rgb(255, 0, 0);opacity:0.4;
```

```
rgb(255, 0, 0);opacity:0.6;
```

```
rgb(255, 0, 0);opacity:0.8;
```

RGB colors with opacity:

Red

Green

Blue

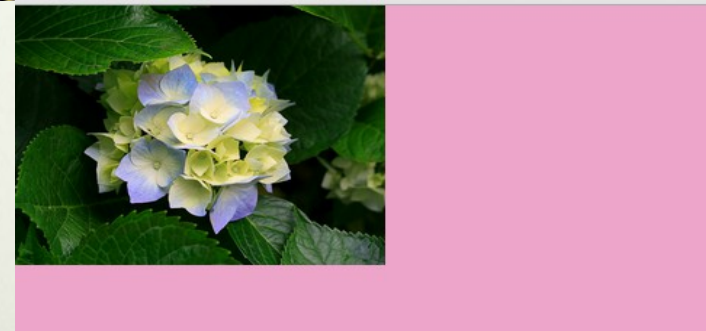
```
{background-color:rgb(255,0,0);opacity:0.6;} /* red with opacity */  
{background-color:rgb(0,255,0);opacity:0.6;} /* green with opacity */  
{background-color:rgb(0,0,255);opacity:0.6;} /* blue with opacity */
```

# background-image

```
body {  
  background-color: rgba(212, 34, 123, 0.4);  
  background-image: url("kwiat.jpg");  
}
```



```
1 body {  
2   background-color: rgba(212, 34, 123, 0.4);  
3   background-image: url("kwiat.jpg");  
4   background-repeat: no-repeat;  
5 }
```

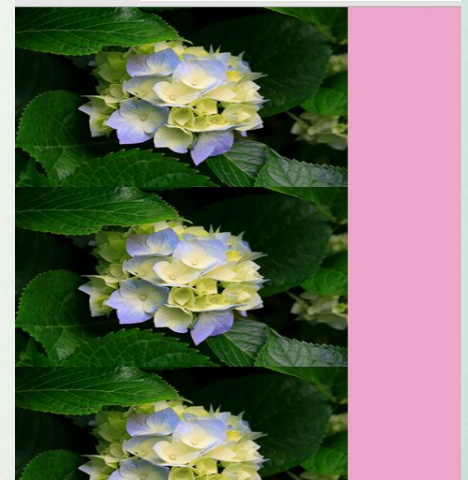


```
1 body {  
2   background-color: rgba(212, 34, 123, 0.4);  
3   background-image: url("kwiat.jpg");  
4   background-repeat: repeat-x;  
5 }
```



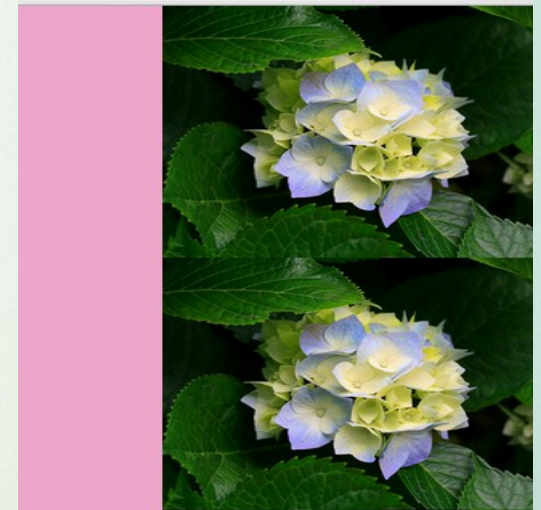
# background-image

```
body {  
  background-color: rgba(212, 34, 123, 0.4);  
  background-image: url("kwiat.jpg");  
  background-repeat: repeat-y;  
}
```

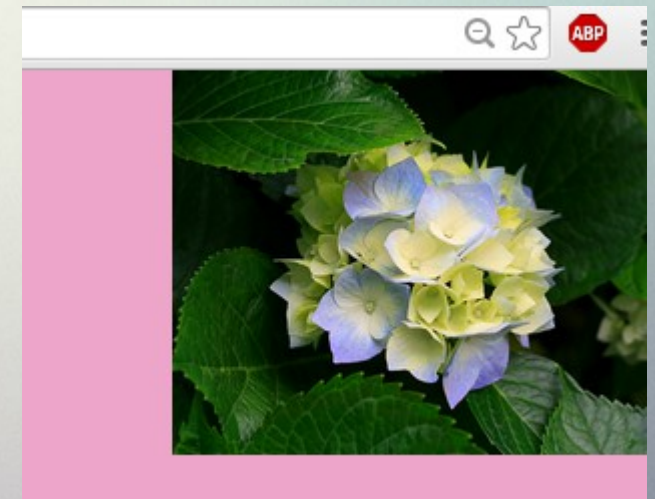


# background-position

```
1 body {  
2   background-color: rgba(212, 34, 123, 0.4);  
3   background-image: url("kwiat.jpg");  
4   background-repeat: repeat-y;  
5   background-position: right;  
6 }
```



```
1 body {  
2   background-color: rgba(212, 34, 123, 0.4);  
3   background-image: url("kwiat.jpg");  
4   background-repeat: no-repeat;  
5   background-position: right top;  
6 }
```



# background-position

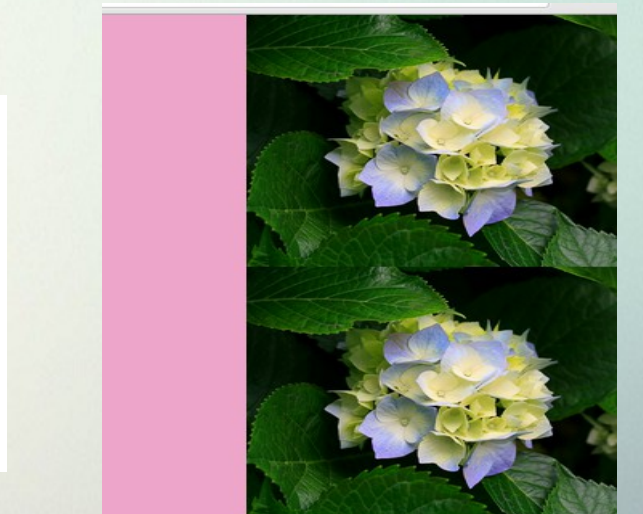
```
background-position:  
2cm 2cm
```

Obrazek jest ustawiony 2 cm od lewej i 2 cm w dół

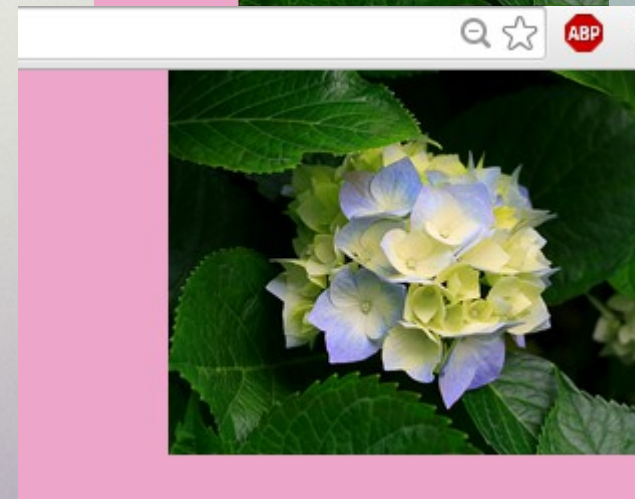
```
background-position:  
top right
```

Obrazek jest ustawiony w górnym prawym rogu strony

```
1 body {  
2   background-color: rgba(212, 34, 123, 0.4);  
3   background-image: url("kwiat.jpg");  
4   background-repeat: repeat-y;  
5   background-position: right;  
6 }
```

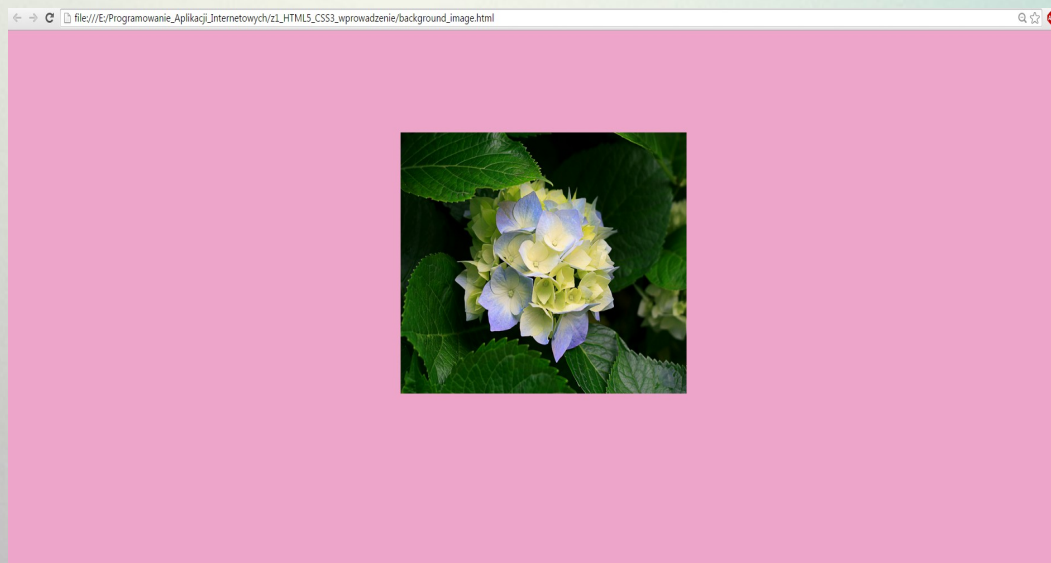


```
1 body {  
2   background-color: rgba(212, 34, 123, 0.4);  
3   background-image: url("kwiat.jpg");  
4   background-repeat: no-repeat;  
5   background-position: right top;  
6 }
```



# background-position

```
1 body {  
2   background-color: rgba(212, 34, 123, 0.4);  
3   background-image: url("kwiat.jpg");  
4   background-repeat: no-repeat;  
5   background-position: 50% 300px;  
6 }
```





# background-attachment

Background-attachment: scroll	Obrazek porusza się razem ze stroną - odblokowany
Background-attachment: fixed	Obrazek jest zablokowany

```
body{  
  background-color: rgba(212,34,123,0.4);  
  background-image: url("kwiat.jpg");  
  background-repeat: no-repeat;  
  background-position: right top;  
  background-attachment: fixed;  
}
```

# Background-wersja skrócona

The shorthand property for background is simply "`background`":

```
body {  
    background: #ffffff url("img_tree.png") no-repeat right top;  
}
```

When using the shorthand property the order of the property values is:

- `background-color`
- `background-image`
- `background-repeat`
- `background-attachment`
- `background-position`