

<b>Przedmiot:</b>	Technologie multimedialne		Studia stacjonarne Semestr 5, 2021
<b>Temat:</b>	<b>Animacje</b>		
<b>Numer lab.:</b>	3	<b>Data wykonania:</b>	2021.11.26
<b>Prowadzący:</b>	dr inż. Piotr Grad	<b>Data oddania:</b>	2021.11.26
<b>Autor:</b>	Łukasz Mencil	<b>Indeks:</b>	114828

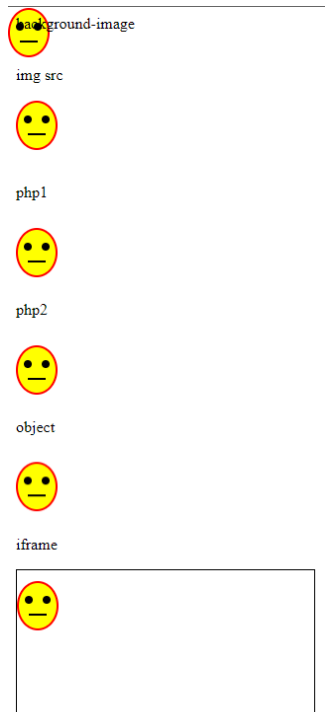
### Cel Ćwiczenia

Zapoznanie się z tworzeniem webowych animacji wektorowych z użyciem SVG oraz CSSa.  
Integracja animacji z bazą danych.

Wszystkie zadania są dostępne pod adresem: [mencel.tmultimedialne.pl](https://mencel.tmultimedialne.pl)

#### 4. Animacja emoji

<https://mencel.tmultimedialne.pl/z3/index1.php>



Testy na różnych przeglądarkach nie wykazały widocznych zmian w działaniu aplikacji.  
Aby animacja działała właściwie należy pamiętać o odpowiednim podlinkowaniu pliku css.

#### 5. Animacja ust ikonki Emoji

<https://mencel.tmultimedialne.pl/z3/index2.php>

php1



php2

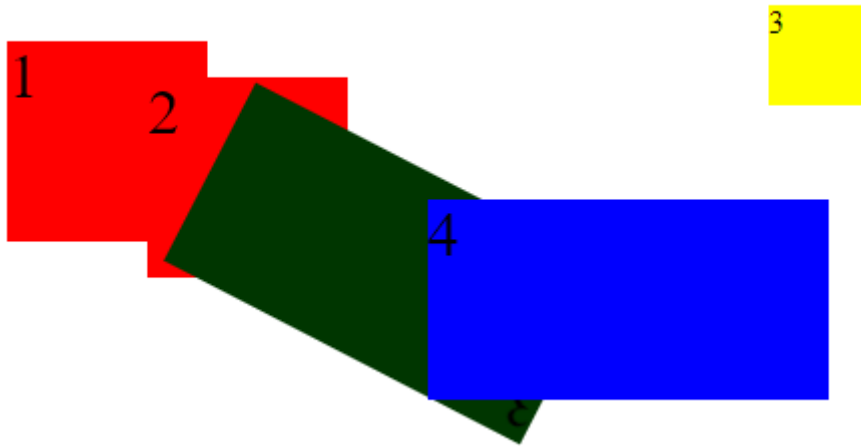


```
.mouth {  
  animation-timing-function: ease-in-out;  
  animation-duration: 2s;  
  animation-iteration-count: infinite;  
  animation-name: mouth;  
}  
  
@keyframes mouth { 50% { transform: skewY(15deg); } }
```

Mina ikonki emoji się zmienia poprzez użycie transform:skewY dzięki czemu linia symbolizująca usta obraca się o 15 stopni na osi Y.

## 6. Dodanie i animacja divów

<https://mencel.tmultimedialne.pl/z3/index3.php>



```
#rotate_green {
  width:100px; height:50px; background-color:green; margin:10px; float:left;
  transform:translate(-15px,90px) scale(2);transform-origin:50% 50%;transition: all 2s ease;
}

#rotate_green:hover {
  background-color:black; transform:translate(-15px,90px) scale(2) rotate(360deg);
}

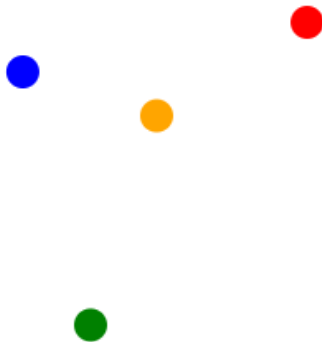
#rotate_blue {
  width:100px; height:50px; background-color:blue; margin:10px; float:left;
  transform:translate(-15px,90px) scale(2);transform-origin:50% 50%;transition: all 2s ease;
}

#rotate_blue:hover {
  background-color:black; transform:translate(-15px,90px) scale(2) rotate(720deg);
}
```

Kluczowe jest tutaj zmienienie wartości *transform-origin* tak by obiekt obracała się względem środka diva. W hoverze (co ma się dzieć gdy najedziemy na diva) zmieniamy wartość *rotate* o zadany obrót.

## 9. Modyfikacja kodu animacji aby grafiki pojawiały się na podstawie wartości z bazy danych.

<https://mencel.tmultimedialne.pl/z3/index5.php>



id	x0	y0	x_delta	y_delta	begin	diameter	time	color
1	10	10	200	10	0	10	2	red
2	200	40	10	40	0	10	2	blue
3	60	60	60	200	0	10	2	green
4	100	200	100	60	0	10	2	orange

```
$rezultat = mysqli_query($polaczenie, "SELECT * FROM animacje1") or die ("Błąd zapytania do bazy: $dbname");
while ($wiersz = mysqli_fetch_array ($rezultat)) {
    $id = $wiersz[0];
    $x0 = $wiersz[1];
    $y0 = $wiersz[2];
    $x_delta = $wiersz[3];
    $y_delta = $wiersz[4];
    $begin = $wiersz[5];
    $diameter = $wiersz[6];
    $time = $wiersz[7];
    $color = $wiersz[8];

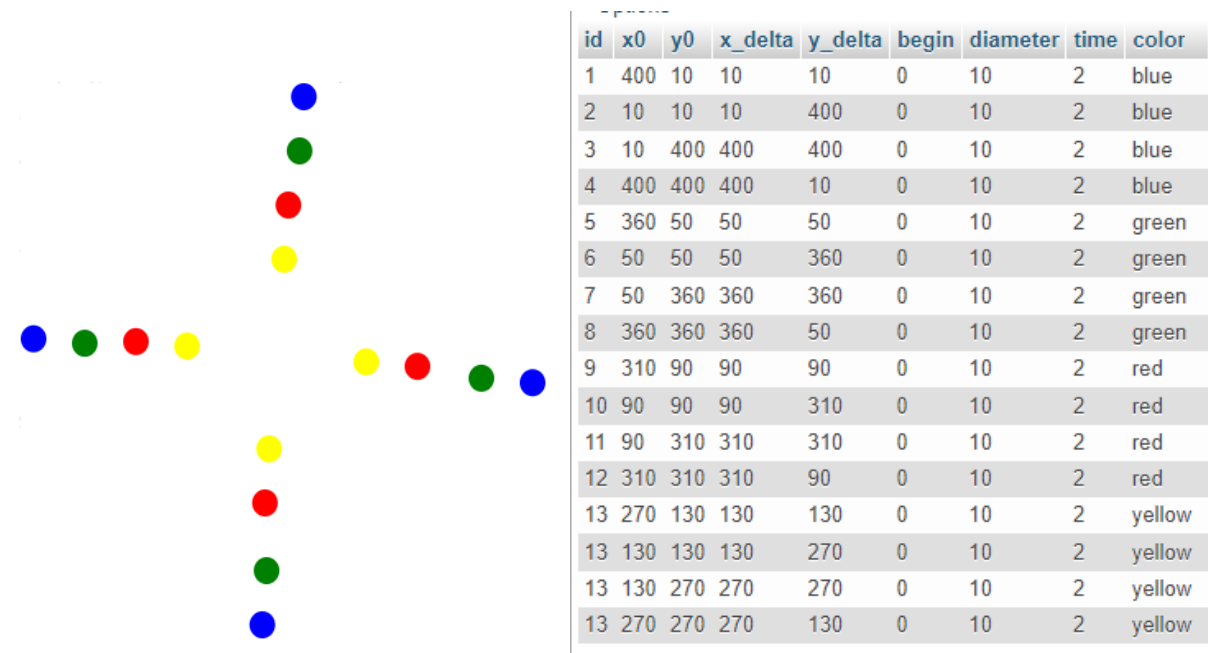
    echo '<div id="layer'.$id.'" style="position:absolute; left: 10px; top:10px; visibility:visible; height:500px; width:500px;">
    <svg viewBox="0 0 500 500" xmlns="http://www.w3.org/2000/svg">
        <circle cx="10" cy="10" r="'.$diameter.'" fill=".$color.'">
            <animate attributeName="cx" from=".$x0." to=".$x_delta." begin=".$begin." dur=".$time." repeatCount=indefinite keyTimes="0;1" />
            <animate attributeName="cy" from=".$y0." to=".$y_delta." begin=".$begin." dur=".$time." repeatCount=indefinite keyTimes="0;1" />
        </circle>
    </svg>
    </div>';
}
mysqli_close($polaczenie);
?>
```

Nasz program w pętli while pobiera dane z bazy danych do odpowiednich zmiennych a następnie te zmienne wstawia w odpowiednie miejsca do „wyprintowanego” diva.

Żeby zrobić to z pomocą pętli musiałem zmienić wartości stylowania *left* i *top* na standardowe *10* i odpowiednio zwiększyć wartości *x0* i *y0*. Dzięki temu wszystko dzieje się automatycznie bez konieczności wypisywania każdego diva ręcznie z odpowiednimi wartościami *left* i *top*.

## 10. Animacja z 16 kulkami

<https://mencel.tmultimedialne.pl/z3/index6.php>



```
$rezultat = mysqli_query($polaczenie, "SELECT * FROM animacje") or die ("Błąd zapytania do bazy: $dbname");
while ($wiersz = mysqli_fetch_array ($rezultat)) {
    $id = $wiersz[0];
    $x0 = $wiersz[1];
    $y0 = $wiersz[2];
    $x_delta = $wiersz[3];
    $y_delta = $wiersz[4];
    $begin = $wiersz[5];
    $diameter = $wiersz[6];
    $time = $wiersz[7];
    $color = $wiersz[8];

    echo '<div id="layer'.$id.'" style="position:absolute; left: 10px; top:10px; visibility:visible; height:500px; width:500px;">
    <svg viewBox="0 0 500 500" xmlns="http://www.w3.org/2000/svg">
        <circle cx="10" cy="10" r=".'.$diameter.'" fill=".'.$color.'">
            <animate attributeName="cx" from=".'.$x0.'" to=".'.$x_delta.'" begin=".'.$begin.'" dur=".'.$time.'" repeatCount=indefinite keyTimes="0;1" />
            <animate attributeName="cy" from=".'.$y0.'" to=".'.$y_delta.'" begin=".'.$begin.'" dur=".'.$time.'" repeatCount=indefinite keyTimes="0;1" />
        </circle>
    </svg>
    </div>';
}
mysqli_close($polaczenie);
?>
```

Od strony kodu mamy tą samą sytuację co w poprzednim zadaniu jedynie wartości w bazie danych się zmieniają. Mamy 16 rekordów. Wyliczenie odpowiednich przebiegów wartości x i y było kluczowe dla jej działania.