

# Zdarzenia czasowe i canvas

Anna Gogolińska

# Zdarzenia czasowe

- Funkcje mogą być wywoływane po określonym czasie lub w określonych odstępach czasu przez przeglądarkę.
- `window.setTimeout(function, milliseconds);` - polecenie to spowoduje wywołanie funkcji podanej jako pierwszy argument po czasie określonym jako drugi argument (w milisekundach)
  - [https://www.w3schools.com/js/tryit.asp?filename=tryjs\\_timing1](https://www.w3schools.com/js/tryit.asp?filename=tryjs_timing1)

# Zdarzenia czasowe

- `window.setInterval(function, milliseconds);` - funkcja podana jako pierwszy argument będzie wywoływana co podaną ilość milisekund
  - [https://www.w3schools.com/js/tryit.asp?filename=tryjs\\_setinterval2](https://www.w3schools.com/js/tryit.asp?filename=tryjs_setinterval2)
- W przypadku funkcji `setInterval()` i `setTimeout()` (oraz `clearInterval()` – nastęny slajd) można pominąć `window`.

# Zdarzenia czasowe

- `setInterval()` i `setTimeout()` zwracają wartości. Aby zatrzymać wywołanie należy te wartości dać jako argument dla `window.clearInterval()`

```
myVar = setTimeout(function, milliseconds);  
clearTimeout(myVar);
```

```
myVar = setInterval(function, milliseconds);  
clearInterval(myVar);
```

# Canvas

- Obiekt canvas służy do rysowania elementów.
- Tworzenie canvas:

```
<canvas id="myCanvas" width="200" height="100"></canvas>
```

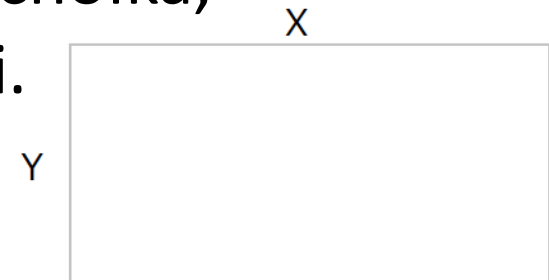
- Można dodać jeszcze obramowanie (przy użyciu styli).

# Rysowanie po canvas

- Aby rysować w skrypcie po canvas najpierw trzeba odwołać się do obiektu canvas.  
`var canvas = document.getElementById("myCanvas");`
- Następnie trzeba stworzyć obiekt rysujący.  
`var ctx = canvas.getContext("2d");`
- Dalej można już rysować. Osobne polecenia rysują osobne elementy. Zazwyczaj ustawianie koloru i rysowanie to oddzielne polecenia.

# Rysowanie

- Rysowanie prostokąta:
  - Najpierw należy ustawić kolor:  
**ctx.fillStyle = "#FF0000";**
  - Następnie rysowanie: **ctx.fillRect(0, 0, 150, 75);**
  - Kolor można podać w dowolnym z przyjętych formatów.
  - Przy rysowaniu pierwsze dwie liczby to współrzędne górnego lewego wierzchołka, następne dwie to długości krawędzi.



# Rysowanie

- Rysowanie linii:
  - Na początek należy rozpocząć rysowanie:  
**ctx.beginPath();**
  - Później aby narysować linię najpierw należy określić początek linii:  
**ctx.moveTo(0, 0);** - współrzędne początku.
  - Następnie współrzędne końca:  
**ctx.lineTo(200, 100);**
  - Można jeszcze ustawić kolor linii:  
**ctx.strokeStyle = '#ff0000';**
  - Następnie właściwe rysowanie to:  
**ctx.stroke();**

# Rysowanie

- Rysowanie okręgu/łuku:
  - Na początek należy rozpocząć rysowanie:  
**ctx.beginPath();**
  - Potem polecenie `arc()` rysuje:  
**ctx.arc(95, 50, 40, 0, 2 \* Math.PI);** - parametry po kolei to współrzędne środka (95, 50), promień (40), kąt początku (0) oraz końca ( $2 * \text{Math.PI}$ ) – to polecenie narysuje koło.
  - Można ustawić kolor i nastęnie właściwe rysowanie: **ctx.strokeStyle = '#ff0000';**  
**ctx.stroke();**

# Przykłady

- [https://www.w3schools.com/graphics/tryit.asp?filename=trycanvas\\_draw](https://www.w3schools.com/graphics/tryit.asp?filename=trycanvas_draw)
- [https://www.w3schools.com/graphics/tryit.asp?filename=trycanvas\\_line](https://www.w3schools.com/graphics/tryit.asp?filename=trycanvas_line)
- [https://www.w3schools.com/graphics/tryit.asp?filename=trycanvas\\_circle](https://www.w3schools.com/graphics/tryit.asp?filename=trycanvas_circle)
- „Czyszczenie” canvasu – należy cały canvas zamalować na kolor tła (np. **ctx.fillStyle = "#FFFFFF"**; i później **ctx.fillRect(0, 0, szerokość, wysokość)**; gdzie za wysokość i szerokość trzeba podać odpowiednie wartości.
- Należy pamiętać że przed każdym rysowaniem poza prostokątem trzeba użyć **ctx.beginPath()**; aby zadziałała zmiana koloru.