

Lekcja 10 – Zdarzenia - Myszka - Klawiatura

plotkarka.eu (kodujący stwórz, lekcja 56)

```
var b1 = new Kwadrat(30, 30, 120);  
b1.rysuj();
```

Zdarzenia, to funkcje, które wykonywane są automatycznie podczas przeglądania strony. Większość zdarzeń wywoływana jest przez użytkownika: klikanie, poruszanie myszką, naciskanie klawiszy. Istnieją też zdarzenia inne: ładowanie, zamykanie, zmiana rozmiarów strony, itp. W Javie mamy wiele różnych zdarzeń, spośród nich tylko kilkanaście jest często używanych:

Jak wykorzystać zdarzenia? Po prostu „podpinamy się” do określonego elementu na stronie internetowej. Musimy pamiętać, aby skrypt JS nie uruchomił się przed wczytaniem podpiętego do niego elementu strony – nie będzie czego obsługiwać!

onblur	gdy obiekt przestał być aktywny
onchange	gdy obiekt zmienił swoją zawartość
onclick	gdy obiekt został kliknięty
ondblclick	gdy podwójnie klikniemy na obiekt
onfocus	gdy obiekt stał się wybrany
onkeyoown	gdy został naciśnięty klawisz na klawiaturze
oninput	uruchamiane w czasie trzymania klawisza
onkeyop	gdy puścimy klawisz na klawiaturze
onload	gdy obiekt został załadowany
onmouseover	gdy kursor znalazł się na obiekcie
onmouseout	gdy kursor opuścił obiekt
onmousedown	przycisk myszki naciśnięty
onmouseup	przycisk myszki puszczony
onmousemove	kursor się porusza
onresize	rozmiar okna przeglądarki jest zmieniany
onselect	gdy zawartość obiektu została zaznaczona
onsubmit	gdy formularz został wysłany
onunload	gdy użytkownik opuszcza daną stronę

Rejestrowanie w kodzie HTML

Zdarzenia **onclick**, **checkbox**, **onchange** zostały omówione w lekcji

7. Kliknięcie w przycisk z napisem START uruchamia funkcję JS o nazwie **OdNowa**.

```
<input type="button" value="START" onclick= OdNowa(>
```

Zmiana własności kwadracika **rys0** czyści ekran. Przesunięcie suwaka **RopS**

```
if (rys0.checked) {c.clearRect(0, 0, w, h);}
```

```
<input type="range" id="RopS" min=990 max=1000 onchange=OporPow(>  
<input type="text" id="RopT" size=2>  
function OporPow(){opor=RopS.value/1000; RopT.value=opor;}
```

uruchamia funkcję **OporPow**. Wartość ustawiona na suwaku wpisywana jest do pola tekstowego **RopT**.

Położenie myszki

Przeglądarki udostępniają kilkanaście właściwości, które powinny dawać pozycję myszki

screenX i **screenY** - pozycja myszki na ekranie

clientX i **clientY** - pozycja myszki w obiekcie, do którego podpięto zdarzenie. Najłatwiej przechowywać położenie myszki w zmiennej **MYSZKA**, która posiada dwa pola (x,y),

Zdarzenie **onmousemove** przypięto do obiektu **canvas**.

UWAGA – wskaźnik myszki jest przesunięty o 8 pikseli, dlatego niewielka korekta w funkcji.

```
ekran X:<input type=text id=Ex size=2>  
ekran Y:<input type=text id=Ey size=2>  
canvas X:<input type=text id=Cx size=2>  
canvas Y:<input type=text id=Cy size=2>  
// ...  
var MYSZKA = {x: 0, y: 0};  
c.canvas.onmousemove = function(evt) {  
  MYSZKA.x = evt.clientX - 8;  
  MYSZKA.y = evt.clientY - 8;  
  Cx.value = MYSZKA.x;  
  Cy.value = MYSZKA.y;  
  Ex.value = evt.screenX;  
  Ey.value = evt.screenY;  
};
```

Błądząca piłeczka

Kolorowa kulka podąża za wskaźnikiem myszki, im dalej wskaźnik od piłki, tym szybciej. UWAGA - instrukcje wstawiono do pętli głównej animacji.

```
vx=(MYSZKA.x-x)/10;  
vy=(MYSZKA.y-y)/10;  
x=x+vx;  
y=y+vy;  
pilka(x,y,r,"red");
```

Strzał

Kliknięcie w canvas spowoduje powiększenie promienia piłki. w Pętli głównej stopniowo przywracamy promieniowi poprzednią wartość.

```
c.canvas.onclick = function(evt) {r=100;}  
// ...  
if (r>10){r=r-5}
```