

Zdarzenia w jQuery

Aby przypiąć nasłuch zdarzenia do pobranego lub utworzonego elementu korzystamy z metody będącej nazwą zdarzenia, lub skorzystać z metody **on()**.

```
1 //podpinam kliknięcie dla wszystkich buttonów
2
3 $("button").click(function() {
4     console.log("test");
5 })
6
7 //lub
8
9 $("button").on("click", function() {
10     console.log("test");
11 })
```

Podobnie jak przy metodzie [.css\(\)](#) i [.attr\(\)](#) także przy zdarzeniach możemy je deklarować pojedynczo, ale i zbiorczo podając obiekt z odpowiednimi kluczami:

```
1 $("button").on("click", function() {...})
2 $("button").on("mouseover", function() {...})
3
4 //lub
5
6 $("button").on({
7     "click" : function() {
8         ...
9     },
10    "mouseover" : function() {
11        ...
12    }
13 });
```

Przy czym znacznie częściej spotkasz pierwszy zapis.

This

Aby odwołać się do obiektu wywołującego, wskazujemy go identycznie jak w przypadku klasycznego Javascript czyli poprzez [this](#), [e.target](#) lub [e.currentTarget](#).

```
1  $("button").on("click", function() {  
2      $(this).text("Kliknięto!");  
3  });  
4  
5  //w przypadku funkcji strzałkowej nie możemy użyć this  
6  $("button").on("click", () => {  
7      $(e.currentTarget).text("Kliknięto!");  
8  });
```

Zauważ, jak objęliśmy nasze zmienne [składnią dolara](#).

Jeżeli dana zmienna zawiera odwołanie do jakiegoś elementu (np. window, e.target, this itp.), możemy na takich elementach wykonywać metody i właściwości z czystego Javascript.

Aby na takim obiekcie wykonywać operacje jQuerowe (np. find(), parent(), css()), wystarczy takie elementy objąć nawiasami:

```
1 //przykłady elementów w czystym Javascript
2 document.body
3 window
4 const btn = document.querySelector("button");
5
6
7 //mogę dla nich wykonywać operacje w czystym Javascript
8 document.body.style.color = "red";
9 window.addEventListener("click", () => { console.log("klik"); });
10 btn.innerText = "klik";
11
12
13 //ale nie mogę dla nich odpalać metod jQuery
14 document.body.css("color", "red"); //błąd
15 window.on("click", () => { console.log("klik"); }); //błąd
16 btn.fadeOut(); //błąd
17
18
19 //możemy je jednak zmienić na odpowiednik w jQuery okrywając je nawiasami
20 $(document.body).css("color", "red");
21 $(window).on("click", () => { console.log("klik"); });
22 $(btn).fadeOut();
```

```
23
24
25 //odmienić na czysty Javascript także możemy
26 const $btn = $("button");
27
28 //użyję jednego z 2 sposobów
29 $btn.get(0); //wybieram pierwszy button w kolekcji
30 $btn[0]; //pierwszy button w kolekcji
31 //lub
32 $("button").get(0);
33 $("button")[0]
```

Odpinanie zdarzenia

Aby odpiąć dane zdarzenie, skorzystamy z metody **off()**. Działa ona praktycznie tak samo jak [removeEventListener](#), czyli jeżeli chcemy odpiąć jakąś funkcję, powinniśmy przekazać ją jako parametr:

```
1 function showSomething() {
2     console.log("Kliknięto");
3 }
4
5 $(".element").on("click", showSomething); //podpinam funkcję showSomething
6 $(".element").off("click", showSomething); //odpinam ją
7
8 $(".element").off("click"); //odpinam wszystkie funkcje podpięte dla click
9 $(".element").off(); //odpinam wszystkie zdarzenia
```

[W jQuery w przeciwieństwie do JavaScript możemy też odpinać funkcje anonimowe](#). Wystarczy przy ich podpinaniu podać dla zdarzenia nazwę:

```
1 $(".element").on("click", function() {
2     console.log("Kliknięto 1");
3 });
4
5 $(".element").on("click.klik2", function() {
6     console.log("Kliknięto 2");
7 });
8
9 //usuwam nasłuchiwanie o nazwie .klik2
10 $(".element").off("click.klik2");
```

Jeżeli chcemy do elementu podpiąć event, który wykona się tylko jeden raz (a zaraz potem zostanie usunięty), skorzystamy z funkcji **.one()**:

```
1 $(".element").one("click", function() {
2     console.log("Wykonam się tylko jeden raz po czym się odłączę");
3 });
```

Jest to równoznaczne z odpisaniem eventu z jego wnętrza (co jest możliwe także tylko w JQuery):

```
1  $(".element").one("click", function(e) {  
2      console.log("Wykonam się tylko jeden raz po czym się odłączę");  
3      $(this).off(e);  
4  });
```

Zadania

1. Stwórz na stronie dwa przyciski. Za pomocą jQuery obsłuż ich kliknięcia poprzez wyświetlenie w konsoli nazwy metody, która została wykorzystana. (metody **click** i **on**)
2. Dodaj do strony zdjęcie. Obsłuż zdarzenie najechania na niego myszką (**mouseenter**). Wykorzystując jeden ze sposobów odwołania się do obiektu, na którym wywołujemy zdarzenie, zmień obrazek na inny. (wykorzystaj np. **this** i **on**)
3. Rozwiń zadanie 2 o aspekt, w którym po zjechaniu myszką ze zdjęcia obrazek powraca do pierwotnego stanu – przed zmianą. (**mouseleave**)
4. Dodaj jeden przycisk oraz ukryty div (atrybut **hidden**) z zawartością. Wywołaj na nim zdarzenie, które wykona się tylko raz: zmień/usuń atrybut **hidden**.
5. Zmień napis przycisku po kliknięciu, tak by pokazywał ile razy go kliknięto. Np. „Naciśnij mnie”, po kliknięciu pokaże się „Licznik: 1”. Po odświeżeniu strony licznik się wyzeruje. (**text**)