WYKŁAD 6
JQUERY - zdarzenia

dr inż. Jacek Paluszak

JQUERY

Zdarzenia

Przeglądarki są zaprogramowane tak, aby wykrywały podstawowe zjawiska, takie jak:

- wczytanie strony
- przeniesienie kursora
- wpisanie znaku
- zmiana wielkości okna.
- itp.

Wszystkie zmiany związane ze stroną WWW to zdarzenia. Aby utworzyć interaktywną stronę, należy przygotować skrypty reagujące na zdarzenia.

JQUERY

Zdarzenia

Zdarzenie reprezentuje moment zajścia określonego zjawiska.

Moment poinformowania przez przeglądarkę o zajściu zdarzenia nazywany jest także jego **zgłoszeniem**.

Zdarzenia związane z myszą

- **click** - zdarzenie jest zgłaszane po zwolnieniu przycisku myszy przy kliknięciu. Zdarzenie **click** jest uruchamiane także po wybraniu odnośnika za pomocą klawiatury oraz kiedy przejdziesz do odsyłacza za pomocą klawisza Tab, a następnie wciśniesz klawisz Enter.

- **dblclick** - kiedy użytkownik dwukrotnie wciśnie i zwolni przycisk służy na przykład do otwierania katalogów i plików na pulpicie.

- **mousedown** - odpowiada pierwszej części kliknięcia przyciśnięciu przycisku myszy przed jego zwolnieniem. Jest ono przydatne przy przenoszeniu elementów na stronie.

JQUERY

Zdarzenia związane z myszą

- **mouseup** - odpowiada drugiej części kliknięcia — zwolnieniu przycisku myszy. Pozwala ono wykryć np. moment upuszczenia przenoszonego elementu.

- **mouseover** - kiedy umieścisz kursor nad elementem strony. Przy jego użyciu można przypisać uchwyty zdarzeń do przycisków nawigacyjnych i wyświetlać menu rozwijane po umieszczeniu kursora nad danym przyciskiem. (w CSS – zdarzenie **hover**).

- **mouseout** - przeniesienie kursora poza dany element.

Zdarzenia związane z myszą

mousemove - zdarzenie jest wyzwalane przy każdym ruchu myszą, czyli prawie cały czas.

Można go używać do sprawdzania aktualnej pozycji kursora na ekranie. Ponadto zdarzenie to można przypisać do określonego znacznika strony, na przykład , i reagować tylko na ruchy myszą w obrębie danego elementu.

JQUERY

Zdarzenia związane z dokumentem i oknem

- **load -** zgłaszane po wczytaniu przez przeglądarkę wszystkich plików strony: pliku HTML, a także dołączonych rysunków, filmów oraz zewnętrznych plików CSS i JavaScript.
  jQuery udostępnia szybciej reagujący zastępnik zdarzenia: **ready()**, który czeka tylko na pobranie kodu HTML, a następnie wykonuje skrypty.

- **resize** - kiedy zmienisz rozmiar okna przeglądarki przez kliknięcie przycisku maksymalizacji lub przeciągnięcie krawędzi okna – warto wtedy sprawdzić szerokość okna - jeśli jest duża, można zmodyfikować układ i zapełnić puste miejsce nowymi kolumnami.

Zdarzenia związane z dokumentem i oknem

- **scroll** - zgłaszane po przeciągnięciu suwaka albo użyciu klawiatury (strzałek w górę i dół, klawiszy *Home* lub *End* itd.) lub rolki myszy do przewijania strony.

- **unload** - kiedy klikniesz odnośnik, aby przejść do nowej strony, albo zamkniesz zakładkę lub okno przeglądarki. Jest to ostatnia informacja dla skryptu JavaScript, która umożliwia wykonanie końcowych operacji przed opuszczeniem strony przez użytkownika.

JQUERY

Zdarzenia związane z formularzami

- **submit** - zgłaszane przy przesyłaniu formularza. Użytkownik może to zrobić przez kliknięcie przycisku **Wyślij** lub wciśnięcie klawisza **Enter**, kiedy kursor znajduje się w polu tekstowym.

- **reset** - umożliwia anulowanie wszystkich zmian wprowadzonych w formularzu i przywrócenie jego wyjściowego stanu

- **change -** wiele pól formularza zgłasza zdarzenie **change** przy zmianie stanu, na przykład w wyniku kliknięcia przycisku opcji lub wybrania odnośnika z menu rozwijanego. Zdarzenie to umożliwia natychmiastowe sprawdzenie wybranego odsyłacza lub zaznaczonego przycisku.

Zdarzenia związane z formularzami

- **focus** - kiedy klikniesz pole tekstowe lub przejdziesz do niego za pomocą klawisza **Tab**, aktywujesz je. Oznacza to, że przeglądarka **„skoncentruje uwagę”** na tym polu

- **blur -** jest to **przeciwieństwo** zdarzenia **focus**. Przeglądarka zgłasza zdarzenie **blur**, kiedy użytkownik opuści aktywne pole za pomocą klawisza **Tab** lub kliknięcia myszą. Zdarzenie jest przydatne przy walidacji formularzy.

Zdarzenia związane z klawiaturą

- **keypress** - zdarzenie jest zgłaszane w momencie wciśnięcia klawisza. Następnie przeglądarka wciąż zgłasza je do czasu zwolnienia klawisza.

- **keydown** - zdarzenie działa podobnie jak zdarzenie **keypress** — jest zgłaszane przy wciśnięciu klawisza. Zachodzi **tuż przed** zdarzeniem **keypress**.

- **keyup** - zdarzenie przeglądarka zgłasza w momencie zwolnienia klawisza.

JQUERY

Zdarzenia - przypisywanie

aby przypisać zdarzenie do elementu, wystarczy dodać kropkę, nazwę zdarzenia i parę nawiasów.

Na przykład można dodać zdarzenie **mouseover** do każdego odnośnika na stronie:

```
$('a').mouseover();
```

Poniższy kod dodaje zdarzenie **click** do elementu o identyfikatorze **menu**:

```
$('#menu').click();
```

Zdarzenia - przekazywanie funkcji

Do zdarzenia można przekazać nazwę wcześniej zdefiniowanej funkcji:

```
$('#start').click(startSlideShow);
```

Podczas przypisywania funkcji do zdarzeń należy **pominąć nawiasy**, standardowo umieszczane po nazwie funkcji w celu jej wywołania.

Oznacza to, że **poniższy zapis jest nieprawidłowy**:

```
$('#start').click(startSlideShow());
```

Jednak najczęściej używanym sposobem obsługi zdarzeń jest przypisywanie do nich **funkcji anonimowych**.

JQUERY

Zdarzenia - przekazywanie funkcji - przykład

1. Pobieranie odnośnika menu:

```
$('#menu')
```

2. Dołączanie zdarzenia:

```
$('#menu').mouseover();
```

3. Dodawanie funkcji anonimowej:

```
$('#menu').mouseover(function() {
    $('#submenu').show();
}); // koniec mouseover
```

JQUERY

Zdarzenia specyficzne dla jQuery

- **ready()** - czeka na pobranie samego kodu HTML, a następnie wykonuje skrypty. Umożliwia natychmiastowe manipulowanie stroną bez konieczności oczekiwania na wolno wczytujące się rysunki i filmy.

- **hover() -** skrótowy zapis uwzględniający dwa zdarzenia:
  - **mousover**
  - **mouseout**

```
$('#selektor').hover(funkcja1, funkcja2);
```

Pierwsza funkcja jest uruchamiana po umieszczeniu kursora nad elementem, a druga — kiedy użytkownik przeniesie kursor w inne miejsce.

Zdarzenia specyficzne dla jQuery

przykład funkcji **hover():**

```
$('#menu').hover(
    function() {
        $('#submenu').show();
    }, // koniec mouseover
    function() {
        $('#submenu').hide();
    } // koniec mouseout
); // koniec hover
```

```
function showSubmenu() {
    $('#submenu').show();
}
function hideSubmenu() {
    $('#submenu').hide();
}
$('#menu').hover(showSubmenu,
hideSubmenu);
```

JQUERY

Zdarzenia specyficzne dla jQuery

- **on()** - elastyczne zarządzanie zdarzeniami - pozwala na określenie zdarzenia i reagującej na nie funkcji, oraz na **przekazanie dodatkowych danych do funkcji obsługującej zdarzenie**.

```
.on( events [, selector ] [, data ], handler )
```

```
np. $('#table').on('click', 'tr', myData, functionName);
```

Dane przekazywane do funkcji – np. literał obiektowy lub zmienna zawierająca taki literał. Literały obiektowe to listy nazw i wartości właściwości:

```
{
    firstName : 'Jan',
    lastName : 'Kowalski',
}
```

Zdarzenia specyficzne dla jQuery

- **on()** c.d.

```
$('#selektor').on('click', myData, functionName);
```

Literał obiektowy można zapisać w zmiennej w następujący sposób:

```
var linkVar = {message:'Pozdrowienia od odnośnika'};
```

Ostatni argument funkcji **on()** to funkcja uruchamiana po zgłoszeniu zdarzenia. Można użyć tu funkcji anonimowej lub standardowej, podobnie jak przy używaniu zwykłych zdarzeń biblioteki jQuery.

Zdarzenia specyficzne dla jQuery

- **on()** c.d.

  Przekazywanie danych w funkcji **on()** nie jest konieczne.
  Można użyć jej do dołączenia zdarzenia i funkcji do elementu oraz pominąć zmienną z danymi:

```
$('selektor').on('click', functionName);
```

  Ten kod działa tak samo jak poniższa instrukcja:

```
$('selektor').click(functionName);
```

Zdarzenia specyficzne dla jQuery

on() c.d. przykład - chcemy wyświetlić okno dialogowe w reakcji na zgłoszenie zdarzenia, jednak komunikat ma być dopasowany do elementu powiązanego z tym zdarzeniem.

Aby uzyskać ten efekt, można utworzyć zmienne przechowujące różne literały obiektowe, a następnie przekazywać te zmienne do funkcji **on()**, powiązane z różnymi elementami:

```
var linkVar = { message:'Pozdrowienia od odnośnika'};
var pVar = { message:'Pozdrowienia od akapitu'};
function showMessage(event) {
    alert(event.data.message);
}
$('a').on('click',linkVar,showMessage);
$('p').on('mouseover',pVar,showMessage);
```

Zdarzenia specyficzne dla jQuery

- **on()** c.d. – przykłady:

```
$('#theElement').on('click', function() {
    // tu robimy coś interesującego
}); // koniec on
$('#theElement'). on('mouseover', function() {
    // tu robimy coś interesującego
}); // koniec on
```

```
$('#theElement'). on({
    'click' : function() {
        // tu robimy coś interesującego
    }, // koniec funkcji click
    'mouseover' : function() {
        // tu robimy coś interesującego
    }; // koniec funkcji mouseover
}); // koniec on
```

Obiekt reprezentujący zdarzenie

Kiedy przeglądarka zgłasza zdarzenie, rejestruje informacje na jego temat i zapisuje go w obiekcie ***zdarzenia***.

Zawiera dane zebrane w momencie wystąpienia zdarzenia, np.:

- współrzędne kursora myszy,
- element powiązany ze zdarzeniem,
- informacje o tym, czy wciśnięty był klawisz *Shift*.

Obiekt reprezentujący zdarzenie

Obiekt zdarzenia jest dostępny w funkcji odpowiedzialnej za obsługę danego zdarzenia.

Obiekt ten jest **przekazywany** do funkcji, dlatego aby uzyskać do niego dostęp, należy użyć parametru.

Poniższy kod sprawdza, jakie były współrzędne X i Y kursora w momencie kliknięcia dowolnego fragmentu strony:

```
$(document).click(function(event) {
    var xPos = event.pageX;
    var yPox = event.pageY;
    alert('X:' + xPos + ' Y:' + yPos);
}); // koniec click
```

Obiekt reprezentujący zdarzenie

```
…click(function(event) {…
```

W momencie wywołania funkcji (w wyniku kliknięcia dowolnego miejsca w oknie przeglądarki) program zapisuje obiekt zdarzenia w zmiennej ***event***.

JQUERY

Obiekt reprezentujący zdarzenie

Właściwość zdarzenia	Opis
pageX	Odległość w pikselach kursora myszy od lewej krawędzi okna przeglądarki.
pageY	Odległość w pikselach kursora myszy od górnej krawędzi okna przeglądarki.
screenX	Odległość w pikselach kursora myszy od lewej krawędzi monitora.
screenY	Odległość w pikselach kursora myszy od górnej krawędzi monitora.
shiftKey	Ma wartość **true**, jeśli w momencie wystąpienia zdarzenia wciśnięty był klawisz ***Shift***.
which	Należy jej używać w zdarzeniu **keypress**. Pozwala sprawdzić kod wciśniętego klawisza.
target	Obiekt docelowy zdarzenia. Na przykład kliknięty element w zdarzeniu **click()**.
data	Obiekt jQuery użyty w funkcji **on()** do przekazania danych do funkcji obsługi zdarzenia.

Obiekt reprezentujący zdarzenie

Za pomocą właściwości **which** obiektu zdarzenia **keypress()** można pobrać kod wciśniętego klawisza.

Np. jeśli chcesz ustalić, jaki znak wcisnął użytkownik (a, K, 9 i tak dalej), musisz przekazać wartość właściwości **which** do metody języka JavaScript, która przekształci numer klawisza na literę, liczbę lub symbol:

```
String.fromCharCode(event.which)
```

DZIĘKUJĘ ZA UWAGĘ