

Java Script

jQuery

jQuery

Aby zacząć używać jQuery, będziemy musieli standardowo dołączyć jej plik do naszej strony. Możemy albo ściągnąć plik z biblioteką na dysk, albo odwołać się do wystawionej w necie wersji.

jQuery – lekka biblioteka programistyczna dla języka JavaScript, ułatwiająca korzystanie z JavaScriptu (w tym manipulację drzewem DOM). Kosztem niewielkiego spadku wydajności w stosunku do profesjonalnie napisanego kodu w niewspomagany JavaScript pozwala osiągnąć interesujące efekty animacji, dodać dynamiczne zmiany strony, wykonać zapytania AJAX.

Znak \$ i składnia łańcuchowa

jQuery ma bardzo zwięzłą konstrukcję skryptów, do której wykorzystuje składnię łańcuchową. Każdy taki łańcuch rozpoczyna się znakiem \$, a poszczególne polecenia są łączone w łańcuchy za pomocą kropki. Poniższy przykład ilustruje zastosowanie jquery:

```
$('#pierwszy').css({color : 'red'}).hide('slow').show(3000);
```

//wcale nie musisz wszystkiego pisać w jednej linii

```
$('#pierwszy')
```

```
  .css({background : 'red'})
```

```
  .hide('slow')
```

```
  .show(3000);
```

**Aby pobrać element ze strony, tak samo jak w przypadku querySelectorAll()
korzystamy ze składni CSS:**

```
$('.div.czerwony') //pobranie divów o klasie czerwony  
$('.div p') //wszystkie p zawierające się w div  
$('.div#cnt > p') //wszystkie p mieszczące się bezpośrednio w div #cnt  
$('.p:first-of-type') //pobranie pierwszego p  
$('.img[width=100]') //pobranie wszystkich img, które mają width=100  
$('.checkbox ~ label') //wszystkie labele, które następują po checkboxach
```

JQuery udostępnia też sporo metod dodatkowych, które są odpowiednikami selektorów CSS3.

Wśród nich najczęściej używanymi są:

eq(index) - zwraca z listy element o danym indexie

first() - zwraca pierwszy element z listy elementów

last() - zwraca ostatni element z listy elementów

lt(index) - zwraca elementy o indeksie mniejszym od danego

gt(index) - zwraca elementy o indeksie większym od danego

not() - zwraca element, jeżeli nie jest on ...

is() - zwraca element, jeżeli jest on ...

filter() - filtruje listę elementów przez zadane kryterium

Pokazywanie i ukrywanie obiektów w jQuery

jQuery udostępnia nam kilka metod służących do pokazywania i ukrywania elementów na stronie.

Pierwszymi z nich są `show(szybkość, funkcja_zwrotna)` która pokazuje i `hide(szybkość, funkcja_zwrotna)` która ukrywa dany element. Parametry tych metod są opcjonalne (i w praktyce mało kiedy używane). Szybkość możemy podać zarówno słownie: "slow", "normal", "fast", lub liczbowo w milisekundach.

```
$('#someDiv').show();
```

```
$('#someDiv').hide();
```

Ajax w jQuery

Obsługa Ajaxa w jQuery jest jak zwykle bajecznie prosta.

Do najprostszych metody jQuery służących do obsługi ajaxa należą:

```
$.get(adres_skryptu, data_wysylana, funkcja_zwrotna)
```


Przykład:

```
$.get( "zapisz_do_bazy.php",  
      { imie: "Johnny", nazwisko: "Bravo" },  
      function(dane){ alert("Dane otrzymane: " + dane); }  
);
```

Poza adresem do skryptu na serwerze reszta atrybutów jest opcjonalna. Nikt nie każe nam wysyłać ani danych ani wykonywać żadnych akcji po otrzymaniu odpowiedzi z serwera.

Bardzo podobne działanie ma metoda **\$.post**, jednak jak sama nazwa wskazuje wysyła ona dane za pomocą post:

```
$.post( "saveToDatabase.php",  
        { name: "Johnny",    surname: "Bravo"  },  
        function(data){    alert("Dane otrzymane: " + data);  }  
);
```

Powyższe metody mimo, że bardzo proste w użyciu, nie dają nam pełnej kontroli nad połączeniem. O wiele lepszym do tego celu jest metoda

\$.ajax:

```
$.ajax({
    type : "POST",
    url  : "saveToDatabase.php",
    data : {
        name : 'Marcin',
        country : 'Polska'
    },
    success: function(ret) {
        //ten fragment wykona się po
        pomyślnym zakończeniu połączenia -
        //atrybut ret zawiera dane zwrócone z
        serwera },
        complete: function() {
            //ten fragment wykona się po
            zakończeniu łączenia - nie ważne czy
            wystąpił błąd, czy sukces },
            error: function(jqXHR, errorThrown,
            errorThrown) { //ten fragment
            wykona się w przypadku BŁĘDU }
        });
```

jQuery daje niesamowite możliwości w budowaniu strony WWW.

Więcej w sieci (manual jQuery)

KONIEC