Tchnij życie w Twoje strony internetowe!

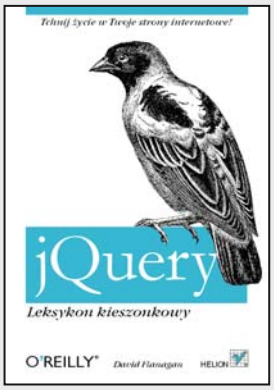
jQuery

Leksykon kieszonkowy

O'REILLY® *David Flanagan* HELION

jQuery. Leksykon kieszonkowy

Autor: David Flanagan
Tłumaczenie: Rafał Downarowicz
ISBN: 978-83-246-3364-7
Tytuł oryginału: jQuery Pocket Reference
Format: 115×170, stron: 160



Tchnij życie w Twoje strony internetowe!

- Jak wycisnąć z JavaScriptu jeszcze więcej?
- Jak operować na polach formularza?
- Jak łatwiej wykorzystać możliwości technologii AJAX?

Historia języka JavaScript pełna jest zakrętów – okresów popularności oraz odrzucenia. W momentach zachwytu język ten był wręcz nadużywany, a gdy przychodziły gorsze dni, był przez użytkowników masowo blokowany. Jego prawdziwy potencjał został natomiast wykorzystany w technologii AJAX. Od tego dnia jego popularność nie maleje. Powstają liczne rozwiązania oparte o JavaScript. Wśród nich znajduje się – ostatnio najmodniejsza – biblioteka jQuery. Jej możliwości naprawdę robią wrażenie!

Biblioteka jQuery pozwala przy użyciu zwięzłej składni wyprawiać w przeglądarce niestworzone rzeczy! Zjawiskowe pokazy slajdów, okna dialogowe, efekty specjalne to tylko niektóre z atrakcji wizualnych dostarczanych przez jQuery. Narzędzie to pozwala na banalnie prostą manipulację wszystkimi elementami drzewa DOM, ich atrybutami oraz własnościami.

Dzięki tej książce zawsze będziesz miał pod ręką ściągę pozwalającą Ci błyskawicznie wykorzystać każdą funkcjonalność jQuery. Dowiesz się, jak rejestrować i obsługiwać zdarzenia, pobierać elementy, rozszerzać funkcjonalność za pomocą wtyczek oraz usprawnić korzystanie z technologii AJAX. Biblioteka jQuery to potężne narzędzie, dzięki któremu tchniesz życie w Twoje strony internetowe!

- Pobieranie elementów
- Określanie klas CSS
- Operowanie na wartościach formularzy
- Manipulowanie strukturą dokumentu
- Obsługa i generowanie zdarzeń
- Efekty animacji
- Obsługa technologii AJAX
- Metody wybierania
- Mechanizm wtyczek – rozszerzanie możliwości jQuery

Wykorzystaj potencjał języka JavaScript!

Helion SA
ul. Kościuszki 1c
44-100 Gliwice
tel. 32 230 98 63
e-mail: helion@helion.pl

Spis treści

Rozdział 2.
Pobieranie i ustawianie elementów

Do najprostszych operacji na obiektach biblioteki jQuery należy pobieranie i ustawianie wartości atrybutów HTML, stylów CSS, zawartości oraz geometrii elementów. Powalające na to metody mają kilka cech charakterystycznych:

- Jeden typ metody pełni rolę zarówno procedury pobierającej (funkcja `get`), jak i ustawiającej (funkcja `set`). Jeżeli przekażesz do metody nową wartość, zostanie ona ustawiona, a jeżeli nie — zwrócona zostanie aktualna.
- Funkcja `set` ustawia wartości dla każdego elementu w obiekcie jQuery i zwraca ten obiekt, umożliwiając utworzenie łańcucha metod.
- Funkcja `get` wysyła zapytanie dotyczące tylko pierwszego elementu (jeżeli chcesz je rozszerzyć na wszystkie elementy, użyj metody `map()`) i zwraca pojedynczą wartość. Ponieważ funkcja `get` nie zwraca obiektu, na którym została wywołana, może pojawić się wyłącznie na końcu łańcucha metod.
- W przypadku funkcji `set` argumentem może być obiekt. Definiuje on zestaw właściwości, które mają zostać ustawione.
- W przypadku funkcji `set` argumentem może być funkcja pozwalająca na obliczenie wartości, które zostaną ustawione. Wartość `this` określa obiekt, dla którego wykonywane są obliczenia.

Pamiętaj o tych zastrzeżeniach, czytając dalszą część rozdziału. W każdej z jego sekcji została wyjaśniona ważna kategoria metod służących do pobierania i ustawiania elementów.

Pobieranie i ustawianie atrybutów HTML

Metoda `attr()` służy do pobierania i ustawiania atrybutów HTML. Radzi sobie z niekompatybilnością przeglądarek i wyjątkami. Możesz w niej stosować nazwy atrybutów HTML lub ich ekwiwalentów w języku JavaScript (na przykład `for` lub `htmlFor` czy też `class` lub `className`). Funkcja `removeAttr()` służy do usuwania atrybutu dla wszystkich wybranych elementów. Oto kilka przykładów:

```
// Wyślij zapytanie o atrybut action pierwszego formularza
$("form").attr("action");
// Ustaw atrybut src elementu z id „icon"
$("#icon").attr("src", "icon.gif");
// Ustaw cztery atrybuty naraz
$("#banner").attr({src:"banner.gif",
                   alt:"Advertisement",
                   width:720, height:64});
// Niech wszystkie łącza ładują się w nowych oknach
$("a").attr("target", "_blank");
// Wyznacz atrybut docelowy tak, aby lokalne łącza
// załadowały się w bieżącym oknie, a łącza spoza strony w nowym oknie
$("a").attr("target", function() {
    if (this.host == location.host) return "_self"
    else return "_blank";
});
// Można również przekazywać funkcje w taki sposób
$("a").attr({target: function() {...}});
// Niech wszystkie łącza załadują się w tym oknie
$("a").removeAttr("target");
```

Pobieranie i ustawianie atrybutów CSS

Metoda `css()` różni się od metody `attr()` tym, że pozwala na pracę ze stylami CSS elementu. Po wysłaniu zapytania o wartość stylu zwracany jest aktualny (lub wyliczony) styl elementu: zwrócona wartość może pochodzić z atrybutu `style` lub z ar-

kusza stylów. Zauważ, że zapytania o style złożone (np. `font` czy `margin`) trzeba zastępować zapytaniami o indywidualne (np. `font-weight`, `font-family`, `margin-top` czy `margin-left`). Podczas ustawiania stylu metoda po prostu doda go do atrybutu `style` elementu. Możesz stosować nazwy stylów CSS z myślnikami (`background-color`) lub nazwy stylów o notacji *camel case* języka JavaScript (`backgroundColor`). W przypadku pobierania wartości stylu metoda `css()` zwraca wartości numeryczne pod postacią łańcucha znaków z członem określającym jednostki, zaś w przypadku ustawiania tych wartości metoda `css()` konwertuje liczby na łańcuchy znaków i w razie potrzeby dodaje człon `px` (piksele):

```
$("h1").css("font-weight"); // Pobierz styl font-weight dla pierwszego
                            // znacznika <h1>
$("h1").css("fontWeight");  // Camel case również działa
$("h1").css("font");        // BŁĄD: zapytanie o styl złożony nie jest możliwe
$("h1").css("font-variant", // Ustaw styl wszystkich znaczników <h1>
            "smallcaps");
$("div.note").css("border", // Można ustawić styl złożony
                  "solid black 2px");
// Ustaw wiele stylów naraz
$("h1").css({ backgroundColor: "black",
              textColor: "white",
              fontVariant: "small-caps",
              padding: "10px 2px 4px 20px",
              border: "dotted black 4px" });
// Zwiększ rozmiar czcionki dla znaczników <h1> o 25%
$("h1").css("font-size", function(i,curval) {
             return Math.round(1.25*parseInt(curval));
            });
```

Pobieranie i ustawianie klas CSS

Przypomnijmy, że wartość atrybutu `class` (właściwości `className` w JavaScript) interpretuje się jako oddzieloną odstępami listę nazw klasy CSS. Ponieważ częściej operuje się na poszczególnych jej elementach niż zastępuje jedną listę drugą, w bibliotece jQuery są zdefiniowane wygodne metody służące do pracy z atrybutem `class`.

Metoda `addClass()` dodaje, a `removeClass()` usuwa klasy z wybranych elementów; `toggleClass()` dodaje klasy do nieposiadających ich jeszcze, a usuwa z posiadających je już elementów, natomiast `hasClass()` sprawdza, czy element posiada określoną klasę. Oto kilka przykładów:

```
// Dodaj klasę CSS do wszystkich znaczników <h1>
$("h1").addClass("hilite");
// Dodaj dwie klasy do znaczników <p> po <h1>
$("h1,p").addClass("hilite firstpara");
// Przekaż funkcję, aby dodać wyliczoną klasę do każdego elementu.
$("section").addClass(function(n) {
   return "section" + n;
});

// Usuń klasę z wszystkich znaczników <p>
$("p").removeClass("hilite");
// Można stosować wiele klas
$("p").removeClass("hilite firstpara");
// Usuń wyliczone klasy ze znaczników
$("section").removeClass(function(n) {
   return "section" + n;
});
// Usuń wszystkie klasy ze wszystkich znaczników <div>
$("div").removeClass();

// Ustaw klasę CSS: dodaj klasę, jeżeli
// jej nie ma, lub usuń, jeżeli jest
$("tr:odd").toggleClass("oddrow");
// Ustaw dwie klasy naraz
$("h1").toggleClass("big bold");
// Ustaw wyliczanie klasy lub klas
$("h1").toggleClass(function(n) {
   return "big bold h1-" + n;
});
$("h1").toggleClass("hilite", true);  // Jak addClass
$("h1").toggleClass("hilite", false); // Jak removeClass

// Test na obecność klasy CSS: czy któryś znacznik <p> posiada tę klasę?
$("p").hasClass("firstpara")
// To samo wykonuje ten
$("#lead").is(".firstpara")
// Metoda is() jest bardziej elastyczna niż hasClass()
$("#lead").is(".firstpara.hilite")
```

Metoda `hasClass()` zwraca wartość `true`, jeżeli któryś z wybranych elementów posiada określoną klasę CSS, a wartość `false` — jeżeli nie. Zauważ, że jest ona mniej elastyczna niż klasy `addClass()`, `removeClass()` i `toggleClass()` oraz klasa `is()` (zobacz „Zapytania i ich wyniki" na stronie 17), którą można stosować w podobnym celu.

Te metody jQuery różnią się od właściwości `classList` HTML5 tym, że działają we wszystkich przeglądarkach i że umożliwiają pracę na wielu elementach i tworzenie łańcucha metod.

Pobieranie i ustawianie wartości dla formularzy HTML

Metoda `val()` służy do pobierania i ustawiania wartości atrybutu `value` dla elementów formularzy HTML, a także wykonywania podobnych operacji w odniesieniu do zaznaczeń pól wyboru, przycisków opcji i elementów `<select>`:

```
// Pobierz wartość z pola tekstowego surname
$("#surname").val()
// Pobierz pojedynczą wartość ze znacznika <select>
$("#usstate").val()
// Pobierz tablicę wartości ze znacznika <select multiple>
$("select#extras").val()
// Pobierz wartość val zaznaczonego przycisku radio
$("input:radio[name=ship]:checked").val()
// Ustaw wartość pola tekstowego
$("#email").val("Błędny adres email")
// Zaznacz wszystkie pola wyboru z tymi nazwami czy wartościami
$("input:checkbox").val(["opt1", "opt2"])
// Przywróć wartość początkową wszystkim polom tekstowym
$("input:text").val(function() {
    return this.defaultValue;
})
```

Pobieranie i ustawianie zawartości elementów

Metody `text()` i `html()` służą do pobierania i ustawiania (hiper)tekstowej zawartości elementu. Metoda `text()` wywołana bez argumentu zwraca tekstową zawartość wszystkich węzłów potomnych dla wszystkich wybranych elementów (nawet w przeglądarkach nieobsługujących właściwości `textContent` czy `inner`↪`Text`).

Metoda `html()` w podobnym przypadku zwraca zawartość hipertekstową tylko pierwszego z wybranych elementów. jQuery korzysta w tym celu z właściwości `innerHTML`: funkcja `x.html()` daje ten sam efekt co `x[0].innerHTML`.

Po przekazaniu łańcucha znaków do `text()` lub `html()` zastąpi on wszelką (hiper)tekstową zawartość elementu. Możesz też przekazać funkcję, która utworzy ten łańcuch za Ciebie:

```
var t = $("head title").text(); // Pobierz tytuł dokumentu
var hdr = $("h1").html()        // Pobierz hipertekst dla pierwszego
                                // znacznika <h1>
// Nadaj każdemu nagłówkowi numer sekcji
$("h1").text(function(n, current) {
    return "§" + (n+1) + ": " + current
});
```

Pobieranie i ustawianie geometrii elementów

Czasem trudności może sprawić właściwe określenie rozmiaru i położenia elementu zwłaszcza w przeglądarkach nieobsługujących `getBoundingClientRect()`. Metody jQuery pozwalają rozwiązać ten problem. Zauważ, że wszystkie opisane tu metody mogą być stosowane do ustawiania wartości, a tylko niektóre do ich pobierania.

Metoda `offset()` służy do pobierania położenia elementu i ustawiania jego współrzędnych. Najpierw oblicza względne położenie elementu w dokumencie, a następnie zwraca je pod postacią

obiektu o właściwościach `left` i `top`, przechowującego współrzędne *X* i *Y*. Przekazując obiekt zawierający właściwości `left` i `top` do tej metody, możesz ustawić położenie obiektu poprzez ustawienie atrybutu CSS:

```
var elt = $("#sprite"); // Element, który ma zostać przemieszczony
var pos = elt.offset(); // Pobierz jego aktualne położenie
pos.top += 100;         // Zmień współrzędną
elt.offset(pos);        // Ustaw nowe położenie
// Przemieść wszystkie znaczniki <h1> w prawo na odległość
// zależną od ich położenia w dokumencie
$("h1").offset(function(index,curpos) {
    return {
        left: curpos.left + 25*index,
        top: curpos.top
    };
});
```

Metoda `position()` różni się od metody `offset()` tym, że służy tylko do pobierania i że zwraca położenie elementów względem ich rodziców (określonych dla każdego elementu w drzewie DOM za pośrednictwem właściwości `offsetParent`). Umiejscowione elementy pełnią zawsze rolę rodziców względem swoich potomków. Chociaż w niektórych przeglądarkach funkcję tę mogą pełnić inne elementy, takie jak komórki tabeli, jQuery nie uznaje ich za rodziców. Metoda `offsetParent()` dokonuje mapowania każdego elementu obiektu na najbliższy umiejscowiony element potomny lub na element `<body>`. Zwróć uwagę na niefortunne nazewnictwo dla tych metod: metoda `offset()` zwraca absolutne położenie obiektu, a metoda `position()` zwraca offset elementu w stosunku do jego `offsetParent()`.

Istnieją trzy funkcje `get` służące do pobierania szerokości elementu i trzy do pobierania jego wysokości. Metoda `width` zwraca podstawową szerokość, a metoda `height` podstawową wysokość bez odstępu wewnętrznego (*padding*), ramek czy marginesów. Natomiast metody `innerWidth()`, `innerHeight()`, `outerWidth()` i `outerHeight()` zwracają sumę podstawowej szerokości (wysokości) i szerokości (wysokości) odstępu wewnętrznego elementu,

a dwie ostatnie sumują poprzednie wartości z wymiarami ramki. Dodając do którejkolwiek z tych metod wartość `true`, można dzięki nim zmieniać także rozmiar marginesów. Oto przykładowe zastosowanie tych metod:

```
var body = $("body");
// Cztery różne szerokości zależnie od zawartych danych
var contentWidth = body.width();
var paddingWidth = body.innerWidth();
var borderWidth = body.outerWidth();
var marginWidth = body.outerWidth(true);
// Sumy l i r odstępu wewnętrznego, ramek i marginesów
var padding = paddingWidth-contentWidth;
var borders = borderWidth-paddingWidth;
var margins = marginWidth-borderWidth;
```

Metody `width()` i `height()` posiadają cechy, których nie mają inne metody omawiane w tej sekcji. Po pierwsze jeżeli pierwszym elementem obiektu jQuery jest obiekt `Window` czy `Document`, zostanie zwrócony rozmiar okna roboczego lub pełny rozmiar dokumentu. Inne metody nie pozwalają na pracę na oknach ani na dokumentach.

Poza tym metody `width()` i `height()` służą zarówno do pobierania, jak i ustawiania wartości elementów. Jeżeli przekażesz do jednej z nich wartość, zostanie ona ustawiona dla każdego elementu w obiekcie jQuery. (Nie można w ten sposób ustawić szerokości ani wysokości obiektów `Window` i `Document`). Jeżeli przekażesz liczbę, zostanie zinterpretowana jako wymiar wyrażony w liczbie pikseli, a jeśli przekażesz łańcuch znaków, zostanie wykorzystany jako wartość atrybutu CSS `width` lub `height`, która będzie mogła pełnić rolę dowolnej jednostki CSS. Można także, jak dla każdej funkcji `set`, przekazać im funkcję, która będzie wywoływana w celu obliczania szerokości i wysokości.

Funkcja `get` metod `width()` i `height()` zwraca wymiary bez odstępu wewnętrznego (*padding*), ramek i marginesów, zaś ich funkcja `set` ustawia atrybuty `width` i `height` w CSS. Domyślnie metody te ustawiają również wymiary kontenera zawierającego

obiekt. Jednak jeżeli atrybut CSS `box-sizing` elementu ma wartość `border-sizing`, metody `width()` i `height()` uwzględnią przy podawaniu wymiarów odstęp wewnętrzny i ramkę. Mimo że dla elementu *e*, który używa modelu `content-box` wywołanie `$(e).width(x).width()` spowoduje zwrócenie wartości *x*, nie jest to zjawisko powszechne dla wszystkich elementów korzystających z tego modelu.

Ostatnią parą metod służących do określania geometrii jest para `scrollTop()` i `scrollLeft()`, która służy do pobrania położenia paska przesuwania dla wszystkich elementów. Metody te można wykorzystywać zarówno przy pracy z obiektem `Window`, jak i z elementami dokumentu oraz obiektem `Document`. W tym ostatnim przypadku można ustawić lub pobrać położenie pasków przesuwania dla okna, w którym znajduje się dany dokument.

Korzystając z metody `scrollTop()` w połączeniu z metodą `height()`, możesz zdefiniować metodę służącą do przewijania okna w górę lub w dół o wybraną przez Ciebie liczbę stron:

```
// Przewiń okno o n stron
// n może mieć postać ułamka lub liczby ujemnej
function page(n) {
    // Opakuj okno w obiekcie jQuery
    var w = $(window);
    // Pobierz rozmiar strony
    var pagesize = w.height();
    // Pobierz aktualne położenie
    var current = w.scrollTop();
    // Przewiń o n stron w dół
    w.scrollTop(current + n*pagesize);
}
```

Pobieranie i ustawianie danych elementów

Metoda `data()` (z funkcją `get` i `set`) pobiera i ustawia dane związane z każdym elementem dokumentu lub z obiektem `Document` czy `Window`. Dzięki wywoływaniu mechanizmów kolejkowania pozwala na rejestrację procedury obsługi zdarzeń.

Aby powiązać dane z elementami w obiekcie jQuery, wywołaj metodę `data()` jako metodę `set`, przekazując jako argumenty nazwę i wartość. Możesz również przekazać pojedynczy obiekt do metody `data()`. Każda właściwość tego obiektu zostanie wówczas zinterpretowana jako para nazwa-wartość, którą można będzie powiązać z elementem lub elementami jQuery. Pamiętaj jednak, że po przekazaniu obiektu do metody `data()` właściwości obiektu zastąpią wszystkie dane wcześniej powiązane z tym elementem. Metoda `data()` nie wywołuje przekazywanych funkcji, lecz je przechowuje tak jak każdą inną wartość.

Metoda `data()` może być również wykorzystana jako funkcja `get`. Kiedy wywołasz ją bez argumentów, zwróci obiekt zawierający wszystkie pary nazwa-wartość powiązane z pierwszym elementem w obiekcie jQuery. Natomiast gdy zrobisz to samo z jednym argumentem o postaci łańcucha znaków, zwróci powiązaną z nim wartość dla pierwszego elementu.

Metoda `removeData()` służy do usuwania danych z elementu. (Ustawienie przy użyciu metody `data()` danej wartości na `null` lub `undefined` nie jest równoznaczne z usunięciem tej wartości). Przekazanie łańcucha znaków do metody `removeData()` spowoduje usunięcie wszystkich wartości powiązanych z tym łańcuchem dla tego elementu, a wywołanie tej metody bez argumentów da w wyniku usunięcie wszystkich danych powiązanych z tym elementem.

```
$("div").data("x", 1);              // Ustaw dane
$("div.nodata").removeData("x"); // Usuń dane
var x = $('#mydiv').data("x");    // Pobierz dane
```

W bibliotece jQuery metody `data()` oraz `removeData()` mogą przyjąć formę funkcji. Możesz na przykład powiązać dane z pojedynczym elementem *e*, stosując metodę `data()` pod postacią funkcji lub metody:

```
$(e).data(...) // Forma metody
$.data(e, ...) // Forma funkcji
```

W jQuery dane nie są przechowywane jako właściwości elementów. Zamiast tego do każdego elementu, z którym powiązane są dane, dodawana jest specjalna właściwość. Ponieważ w niektórych przeglądarkach nie można dodawać właściwości do elementów `<applet>`, `<object>` i `<embed>`, jQuery nie pozwala na powiązanie danych z tymi elementami.

Skorowidz

I

J

K

M

P

R

S

W

Z

WWW

O'REILLY®

jQuery. Leksykon kieszonkowy

Historia języka JavaScript pełna jest zakrętów — okresów popularności oraz odrzucenia. W momentach zachwytu język ten był wręcz nadużywany, a gdy przychodziły gorsze dni, był przez użytkowników masowo blokowany. Jego prawdziwy potencjał został natomiast wykorzystany w technologii AJAX. Od tego dnia jego popularność nie maleje. Powstają liczne rozwiązania oparte na JavaScripcie. Wśród nich znajduje się — ostatnio najmodniejsza — biblioteka jQuery. Jej możliwości naprawdę robią wrażenie!

Biblioteka jQuery pozwala przy użyciu zwięzłej składni wyprawiać w przeglądarce niestworzone rzeczy! Zjawiskowe pokazy slajdów, okna dialogowe, efekty specjalne to tylko niektóre z atrakcji wizualnych dostarczanych przez jQuery. Narzędzie to pozwala na banalnie prostą manipulację wszystkimi elementami drzewa DOM, ich atrybutami oraz własnościami. Dzięki tej książce zawsze będziesz miał pod ręką ściągę pozwalającą Ci błyskawicznie wykorzystać każdą funkcjonalność jQuery. Dowiesz się, jak rejestrować i obsługiwać zdarzenia, pobierać elementy, rozszerzać funkcjonalność za pomocą wtyczek oraz usprawnić korzystanie z technologii AJAX. Biblioteka jQuery to potężne narzędzie, dzięki któremu tchniesz życie w Twoje strony internetowe!

- Pobieranie elementów
- Określanie klas CSS
- Operowanie na wartościach formularzy
- Manipulowanie strukturą dokumentu
- Obsługa i generowanie zdarzeń
- Efekty animacji
- Obsługa technologii AJAX
- Metody wybierania
- Mechanizm wtyczek — rozszerzanie możliwości jQuery

Wykorzystaj potencjał języka JavaScript!

helion.pl
księgarnia internetowa

Nr katalogowy: 6623

Księgarnia internetowa:
http://helion.pl

Zamówienia telefoniczne:
0 801 339900
0 601 339900

Helion
Sprawdź najnowsze promocje:
http://helion.pl/promocje
Książki najchętniej czytane:
http://helion.pl/bestsellery
Zamów informacje o nowościach:
http://helion.pl/nowosci

Helion SA
ul. Kościuszki 1c, 44-100 Gliwice
tel.: 32 230 98 63
e-mail: helion@helion.pl
http://helion.pl

Cena 24,90 zł

ISBN 978-83-246-3364-7

Informatyka w najlepszym wydaniu