



Podręczne informacje

Niniejszy dodatek przedstawia podręczne informacje o jQuery API, w tym wyrażenia selektorów oraz metody. Bardziej szczegółowe omówienie tego zagadnienia można znaleźć w książce *jQuery Reference Guide*, a także na stronie biblioteki jQuery z dokumentacją, dostępnej pod adresem <http://docs.jquery.com>.

Wyrażenia selektorów

Funkcja fabryczna jQuery `$()` wykorzystywana jest do odnajdywania na stronie elementów, z którymi będziemy pracować. Funkcja ta przyjmuje łańcuch znaków złożony ze składni przypominającej CSS i nazywany **wyrażeniem selektora**. Wyrażenia selektorów omówione są szczegółowo w rozdziale 2.

Selektor	Dopasowuje
<code>*</code>	Wszystkie elementy.
<code>#id</code>	Element o podanym identyfikatorze.
<code>element</code>	Wszystkie elementy podanego typu.
<code>..klasa</code>	Wszystkie elementy podanej klasy.
<code>a, b</code>	Elementy dopasowywane przez <i>a</i> lub <i>b</i> .
<code>a b</code>	Elementy <i>b</i> będące potomkami <i>a</i> .
<code>a > b</code>	Elementy <i>b</i> będące dziećmi <i>a</i> .
<code>a + b</code>	Elementy <i>b</i> następujące bezpośrednio po <i>a</i> .
<code>a ~ b</code>	Elementy <i>b</i> będące rodzeństwem <i>a</i> .

Selektor	Dopasowuje
:first	Pierwszy element zbioru wyników.
:last	Ostatni element zbioru wyników.
:not(<i>a</i>)	Wszystkie elementy zbioru wyników, które nie są dopasowane przez <i>a</i> .
:even	Parzyste elementy zbioru wyników (oparte na 0).
:odd	Nieparzyste elementy zbioru wyników (oparte na 0).
:eq(<i>index</i>)	Ponumerowany element zbioru wyników (oparty na 0).
:gt(<i>index</i>)	Wszystkie elementy zbioru wyników po podanym indeksie (większe od niego) (oparte na 0).
:lt(<i>index</i>)	Wszystkie elementy zbioru wyników przed podanym indeksem (mniejsze od niego) (oparte na 0).
:header	Elementy nagłówka (na przykład <h1>, <h3>).
:animated	Elementy z trwającą animacją.
:contains(<i>tekst</i>)	Elementy zawierające podany tekst.
:empty	Elementy bez węzłów potomnych.
:has(<i>a</i>)	Elementy zawierające element potomny dopasowywany przez <i>a</i> .
:parent	Elementy z węzłami potomnymi.
:hidden	Elementy ukryte albo poprzez CSS, albo jako <input type="hidden" />.
:visible	Odwrotność :hidden.
[<i>atr</i>]	Elementy posiadające atrybut <i>atr</i> .
[<i>atr</i> =wartość]	Elementy z atrybutem <i>atr</i> równym <i>wartość</i> .
[<i>atr</i> !=wartość]	Elementy z atrybutem <i>atr</i> , który nie jest równy <i>wartość</i> .
[<i>atr</i> ^=wartość]	Elementy z atrybutem <i>atr</i> rozpoczynającym się od <i>wartość</i> .
[<i>atr</i> \$=wartość]	Elementy z atrybutem <i>atr</i> kończącym się na <i>wartość</i> .
[<i>atr</i> *=wartość]	Elementy z atrybutem <i>atr</i> zawierającym podłańcuch znaków <i>wartość</i> .
:nth-child(<i>indeks</i>)	Elementy będące kolejnym (określonym przez <i>indeks</i>) dzieckiem elementu nadrzędnego (oparte na 1).
:nth-child(even)	Elementy będące parzystym dzieckiem elementu nadrzędnego (oparte na 1).
:nth-child(odd)	Elementy będące nieparzystym dzieckiem elementu nadrzędnego (oparte na 1).
:nth-child(<i>wzór</i>)	Elementy będące kolejnym (<i>wzór</i>) dzieckiem elementu nadrzędnego (oparte na 1). Wzory mają postać $an+b$ dla liczb całkowitych a i b .
:first-child	Elementy będące pierwszym dzieckiem elementu nadrzędnego.
:last-child	Elementy będące ostatnim dzieckiem elementu nadrzędnego.
:only-child	Elementy będące jedynym dzieckiem elementu nadrzędnego.
:input	Wszystkie elementy <input>, <select>, <textarea> oraz <button>.

Selektor	Dopasowuje
:text	Elementy <input> z atrybutem type="text".
:password	Elementy <input> z atrybutem type="password".
:radio	Elementy <input> z atrybutem type="radio".
:checkbox	Elementy <input> z atrybutem type="checkbox".
:submit	Elementy <input> z atrybutem type="submit".
:image	Elementy <input> z atrybutem type="image".
:reset	Elementy <input> z atrybutem type="reset".
:button	Elementy <input> z atrybutem type="button" oraz elementy <button>.
:file	Elementy <input> z atrybutem type="file".
:enabled	Włączone elementy formularza.
:disabled	Wyłączone elementy formularza.
:checked	Zaznaczone przyciski opcji i pola wyboru.
:selected	Wybrane elementy <option>.

Metody przechodzenia drzewa DOM

Po utworzeniu obiektu jQuery za pomocą `$()` możemy zmodyfikować zbiór dopasowanych elementów, z którymi pracujemy, wywołując jedną z poniższych **metod przechodzenia drzewa DOM**. Metody przechodzenia drzewa DOM omówione są szczegółowo w rozdziale 2.

Metoda	Zwraca obiekt jQuery zawierający
<code>.filter(selektor)</code>	Wybrane elementy pasujące do podanego selektora.
<code>.filter(funkcja-zwrotna)</code>	Wybrane elementy, dla których funkcja zwrotna zwraca <code>true</code> .
<code>.eq(indeks)</code>	Wybrany element o podanym indeksie (rozpoczynającym się od 0).
<code>.slice(początek, [koniec])</code>	Wybrany element z podanego przedziału (dla indeksów rozpoczynających się od 0).
<code>.not(selektor)</code>	Wybrane elementy niepasujące do podanego selektora.
<code>.add(selektor)</code>	Wybrane elementy wraz z dodatkowymi elementami pasującymi do podanego selektora.
<code>.find(selektor)</code>	Elementy potomne pasujące do podanego selektora.
<code>.contents()</code>	Węzły dzieci (włącznie z węzłami tekstowymi).
<code>.children([selektor])</code>	Węzły dzieci, opcjonalnie filtrowane po selektorze.
<code>.next([selektor])</code>	Element rodzeństwa następujący bezpośrednio po każdym z wybranych elementów, opcjonalnie filtrowany po selektorze.

Metoda	Zwraca obiekt jQuery zawierający
<code>.nextAll([selektor])</code>	Wszystkie elementy rodzeństwa następujące po każdym z wybranych elementów, opcjonalnie filtrowane po selektorze.
<code>.prev([selektor])</code>	Element rodzeństwa bezpośrednio poprzedzający każdy z wybranych elementów, opcjonalnie filtrowany po selektorze.
<code>.prevAll([selektor])</code>	Wszystkie elementy rodzeństwa poprzedzające każdy z wybranych elementów, opcjonalnie filtrowane po selektorze.
<code>.siblings([selektor])</code>	Wszystkie elementy rodzeństwa, opcjonalnie filtrowane po selektorze.
<code>.parent([selektor])</code>	Element rodzica każdego z wybranych elementów, opcjonalnie filtrowany po selektorze.
<code>.parents([selektor])</code>	Wszystkie elementy przodków, opcjonalnie filtrowane po selektorze.
<code>.closest(selektor)</code>	Pierwszy element dopasowany przez selektor, zaczynając od wybranego elementu i przechodząc w górę przez jego przodków w drzewie DOM.
<code>.offsetParent()</code>	Pozycjonowany element rodzica (na przykład <i>relative</i> , <i>absolute</i>) dla pierwszego wybranego elementu.
<code>.andSelf()</code>	Wybrane elementy plus poprzedni zbiór wybranych elementów z wewnętrznego stosu jQuery.
<code>.end()</code>	Poprzedni zbiór wybranych elementów z wewnętrznego stosu jQuery.
<code>.map(funkcja-zwrotna)</code>	Wynik funkcji zwrotnej wywołanej na każdym z wybranych elementów.

Metody zdarzeń

By zareagować na działania użytkownika, musimy zarejestrować nasze programy obsługi zdarzeń za pomocą poniższych **metod zdarzeń**. Warto zauważyć, iż wiele ze zdarzeń DOM ma zastosowanie jedynie do pewnych typów elementów. Te subtelne szczegóły nie są tutaj omówione. Metody zdarzeń omówione są szczegółowo w rozdziale 3.

Metoda zdarzenia	Opis
<code>.ready(program-obsługi)</code>	Dowiązuje program obsługi zdarzenia, który ma być wywołany po pełnym załadowaniu drzewa DOM oraz arkusza CSS.
<code>.bind(typ, [dane], program-obsługi)</code>	Dowiązuje program obsługi zdarzenia, który ma być wywołany, kiedy zdarzenie określonego typu zostaje przesłane do elementu.
<code>.one(typ, [dane], program-obsługi)</code>	Dowiązuje program obsługi zdarzenia, który ma być wywołany, kiedy zdarzenie określonego typu zostaje przesłane do elementu. Usuwa dowiązanie po wywołaniu programu obsługi zdarzenia.
<code>.unbind([typ], [program-obsługi])</code>	Usuwa dowiązania dla elementu (dla określonego typu zdarzenia, określonego programu obsługi lub wszystkich dowiązań).

Metoda zdarzenia	Opis
<code>.live(typ, program-obsługi)</code>	Dowiązuje program obsługi zdarzenia, który ma być wywołany, kiedy zdarzenie określonego typu zostaje przesłane do elementu za pomocą delegacji zdarzeń.
<code>.die(typ, [program-obsługi])</code>	Usuwa dowiązania dla elementu dołączone wcześniej za pomocą metody <code>.live()</code> .
<code>.blur(program-obsługi)</code>	Dowiązuje program obsługi zdarzenia, który ma być wywołany, kiedy element przestaje być aktywny dla klawiatury.
<code>.change(program-obsługi)</code>	Dowiązuje program obsługi zdarzenia, który ma być wywołany, kiedy zmienia się wartość elementu.
<code>.click(program-obsługi)</code>	Dowiązuje program obsługi zdarzenia, który ma być wywołany, kiedy element zostaje kliknięty.
<code>.dblclick(program-obsługi)</code>	Dowiązuje program obsługi zdarzenia, który ma być wywołany, kiedy element zostaje dwukrotnie kliknięty.
<code>.error(program-obsługi)</code>	Dowiązuje program obsługi zdarzenia, który ma być wywołany, kiedy element otrzymuje błąd zdarzenia (uzależniony od przeglądarki).
<code>.focus(program-obsługi)</code>	Dowiązuje program obsługi zdarzenia, który ma być wywołany, kiedy element zaczyna być aktywny dla klawiatury.
<code>.keydown(program-obsługi)</code>	Dowiązuje program obsługi zdarzenia, który ma być wywołany, kiedy klawisz zostaje naciśnięty, a element jest aktywny dla klawiatury.
<code>.keypress(program-obsługi)</code>	Dowiązuje program obsługi zdarzenia, który ma być wywołany, kiedy za pomocą klawiatury zostanie wprowadzony znak, a element jest aktywny dla klawiatury.
<code>.keyup(program-obsługi)</code>	Dowiązuje program obsługi zdarzenia, który ma być wywołany, kiedy klawisz zostaje zwolniony, a element jest aktywny dla klawiatury.
<code>.load(program-obsługi)</code>	Dowiązuje program obsługi zdarzenia, który ma być wywołany, kiedy element skończy się ładować.
<code>.mousedown(program-obsługi)</code>	Dowiązuje program obsługi zdarzenia, który ma być wywołany, kiedy klawisz myszy zostaje naciśnięty wewnątrz elementu.
<code>.mouseenter(program-obsługi)</code>	Dowiązuje program obsługi zdarzenia, który ma być wywołany, kiedy kursor myszy wchodzi nad element. Nie wpływa na nią propagacja zdarzeń.
<code>.mouseleave(program-obsługi)</code>	Dowiązuje program obsługi zdarzenia, który ma być wywołany, kiedy kursor myszy opuszcza element. Nie wpływa na nią propagacja zdarzeń.
<code>.mousemove(program-obsługi)</code>	Dowiązuje program obsługi zdarzenia, który ma być wywołany, kiedy kursor myszy porusza się wewnątrz elementu.
<code>.mouseout(program-obsługi)</code>	Dowiązuje program obsługi zdarzenia, który ma być wywołany, kiedy kursor myszy opuszcza element.

Metoda zdarzenia	Opis
<code>.mouseover(program-obsługi)</code>	Dowiązuje program obsługi zdarzenia, który ma być wywołany, kiedy kursor myszy wchodzi do elementu.
<code>.mouseup(program-obsługi)</code>	Dowiązuje program obsługi zdarzenia, który ma być wywołany, kiedy klawisz myszy zostaje zwolniony wewnątrz elementu.
<code>.resize(program-obsługi)</code>	Dowiązuje program obsługi zdarzenia, który ma być wywołany, kiedy rozmiar elementu się zmienia.
<code>.scroll(program-obsługi)</code>	Dowiązuje program obsługi zdarzenia, który ma być wywołany, kiedy zmienia się pozycja przewijania elementu.
<code>.select(program-obsługi)</code>	Dowiązuje program obsługi zdarzenia, który ma być wywołany, kiedy zaznaczany jest tekst wewnątrz elementu.
<code>.submit(program-obsługi)</code>	Dowiązuje program obsługi zdarzenia, który ma być wywołany, kiedy przesyłany jest element formularza.
<code>.unload(program-obsługi)</code>	Dowiązuje program obsługi zdarzenia, który ma być wywołany, kiedy element zostaje usunięty z pamięci.
<code>.hover(wejście, wyjście)</code>	Dowiązuje program obsługi zdarzenia wejście, który ma być wywołany, kiedy kursor myszy wchodzi do elementu, i wyjście, który ma być wywołany, kiedy kursor myszy opuszcza element.
<code>.toggle(program-obsługi1, program-obsługi2, ...)</code>	Dowiązuje program-obsługi1, który ma być wywołany po kliknięciu elementu myszą, natomiast po kolejnym kliknięciu wywołany ma zostać program-obsługi2 i tak dalej.
<code>.trigger(typ, [dane])</code>	Uruchamia programy obsługi zdarzenia dla elementu i wykonuje działanie domyślne dla zdarzenia.
<code>.triggerHandler(typ, [dane])</code>	Uruchamia programy obsługi zdarzenia dla elementu bez wykonywania jakichkolwiek działań domyślnych dla zdarzenia.
<code>.blur()</code>	Wywołuje zdarzenie blur.
<code>.change()</code>	Wywołuje zdarzenie change.
<code>.click()</code>	Wywołuje zdarzenie click.
<code>.dblclick()</code>	Wywołuje zdarzenie dblclick.
<code>.error()</code>	Wywołuje zdarzenie error.
<code>.focus()</code>	Wywołuje zdarzenie focus.
<code>.keydown()</code>	Wywołuje zdarzenie keydown.
<code>.keypress()</code>	Wywołuje zdarzenie keypress.
<code>.keyup()</code>	Wywołuje zdarzenie keyup.
<code>.select()</code>	Wywołuje zdarzenie select.
<code>.submit()</code>	Wywołuje zdarzenie submit.

Metody efektów

Poniższe **metody efektów** można wykorzystać do wykonania animacji na elementach drzewa DOM. Metody efektów omówione są szczegółowo w rozdziale 4.

Metoda efektu	Opis
<code>.show()</code>	Wyświetla dopasowane elementy.
<code>.hide()</code>	Ukrywa dopasowane elementy.
<code>.show(szybkość, [funkcja-zwrotna])</code>	Wyświetla dopasowane elementy, wykonując animację wysokości, szerokości oraz nieprzezroczystości.
<code>.hide(szybkość, [funkcja-zwrotna])</code>	Ukrywa dopasowane elementy, wykonując animację wysokości, szerokości oraz nieprzezroczystości.
<code>.toggle([szybkość], [funkcja-zwrotna])</code>	Wyświetla lub ukrywa dopasowane elementy.
<code>.slideDown([szybkość], [funkcja-zwrotna])</code>	Wyświetla dopasowane elementy za pomocą zsuwania.
<code>.slideUp([szybkość], [funkcja-zwrotna])</code>	Ukrywa dopasowane elementy za pomocą zsuwania.
<code>.slideToggle([szybkość], [funkcja-zwrotna])</code>	Wyświetla lub ukrywa dopasowane elementy za pomocą zsuwania.
<code>.fadeIn([szybkość], [funkcja-zwrotna])</code>	Wyświetla dopasowane elementy za pomocą stopniowego przyrastania ich nieprzezroczystości.
<code>.fadeOut([szybkość], [funkcja-zwrotna])</code>	Ukrywa dopasowane elementy za pomocą stopniowego przyrastania ich przezroczystości.
<code>.fadeTo(szybkość, nieprzezroczystość, [funkcja-zwrotna])</code>	Zmienia nieprzezroczystość dopasowanych elementów.
<code>.animate(trybuty, [szybkość], [przejście], [funkcja-zwrotna])</code>	Wykonuje dostosowaną do własnych potrzeb animację wymienionych trybutów CSS.
<code>.animate(trybuty, opcje)</code>	Niskopoziomowy interfejs dla metody <code>.animate()</code> pozwalający kontrolować kolejkę animacji.
<code>.stop([wyczyść-kolejkę], [przejdź-na-koniec])</code>	Zatrzymuje będącą w trakcie animację, a następnie rozpoczyna skolejkowane animacje, jeśli są one obecne.
<code>.queue()</code>	Pobiera kolejkę animacji dla pierwszego dopasowanego elementu.
<code>.queue(funkcja-zwrotna)</code>	Dodaje funkcję zwrotną na końcu kolejki.
<code>.queue(nowa-kolejka)</code>	Zastępuje kolejkę nową.
<code>.dequeue()</code>	Wykonuje kolejną animację z kolejki.

Metody edycji drzewa DOM

Metody edycji drzewa DOM omówione są szczegółowo w rozdziale 5.

Metoda	Opis
<code>.attr(<i>klucz</i>)</code>	Pobiera atrybut o nazwie <i>klucz</i> .
<code>.attr(<i>klucz</i>, <i>wartość</i>)</code>	Ustawia atrybut o nazwie <i>klucz</i> na <i>wartość</i> .
<code>.attr(<i>klucz</i>, <i>funkcja</i>)</code>	Ustawia wartość atrybutu o nazwie <i>klucz</i> na wynik <i>funkcji</i> wywoływanej osobno na każdym z dopasowanych elementów.
<code>.attr(<i>tablica-asocjacyjna</i>)</code>	Ustawia wartości atrybutów, podając je w postaci par klucz-wartość.
<code>.removeAttr(<i>klucz</i>)</code>	Usuwa atrybut o nazwie <i>klucz</i> .
<code>.addClass(<i>klasa</i>)</code>	Przypisuje podaną klasę do każdego dopasowanego elementu.
<code>.removeClass(<i>klasa</i>)</code>	Usuwa podaną klasę z każdego dopasowanego elementu.
<code>.toggleClass(<i>klasa</i>)</code>	Usuwa podaną klasę, jeśli jest ona obecna, z każdego dopasowanego elementu lub ją dodaje, jeśli nie jest.
<code>.hasClass(<i>klasa</i>)</code>	Zwraca <code>true</code> , jeśli dowolny z dopasowanych elementów ma przypisaną podaną klasę.
<code>.html()</code>	Zwraca zawartość kodu HTML dla pierwszego z dopasowanych elementów.
<code>.html(<i>wartość</i>)</code>	Ustawia treść kodu HTML każdego dopasowanego elementu na podaną <i>wartość</i> .
<code>.text()</code>	Pobiera zawartość tekstową wszystkich dopasowanych elementów w postaci jednego łańcucha znaków.
<code>.text(<i>wartość</i>)</code>	Ustawia treść tekstową każdego dopasowanego elementu na podaną <i>wartość</i> .
<code>.val()</code>	Pobiera atrybut <code>value</code> dla pierwszego z dopasowanych elementów.
<code>.val(<i>wartość</i>)</code>	Ustawia atrybut <code>value</code> każdego z elementów na <i>wartość</i> .
<code>.css(<i>klucz</i>)</code>	Pobiera atrybut CSS o nazwie <i>klucz</i> .
<code>.css(<i>klucz</i>, <i>wartość</i>)</code>	Ustawia atrybut CSS o nazwie <i>klucz</i> na <i>wartość</i> .
<code>.css(<i>tablica-asocjacyjna</i>)</code>	Ustawia wartości atrybutów CSS, podając je w postaci par klucz-wartość.
<code>.offset()</code>	Pobiera współrzędne <code>top</code> oraz <code>left</code> w pikselach dla pierwszego dopasowanego elementu względem części strony widocznej w przeglądarce.
<code>.position()</code>	Pobiera współrzędne <code>top</code> i <code>left</code> w pikselach dla pierwszego dopasowanego elementu względem elementu zwracanego przez metodę <code>.offsetParent()</code> .
<code>.scrollTop()</code>	Pobiera pozycję przewijania w pionie dla pierwszego z dopasowanych elementów.

Metoda	Opis
<code>.scrollTop(wartość)</code>	Ustawia pozycję przewijania w pionie dla wszystkich dopasowanych elementów na <i>wartość</i> .
<code>.scrollLeft()</code>	Pobiera pozycję przewijania w poziomie dla pierwszego z dopasowanych elementów.
<code>.scrollLeft(wartość)</code>	Ustawia pozycję przewijania w poziomie dla wszystkich dopasowanych elementów na <i>wartość</i> .
<code>.height()</code>	Pobiera wysokość pierwszego z dopasowanych elementów.
<code>.height(wartość)</code>	Ustawia wysokość wszystkich dopasowanych elementów na <i>wartość</i> .
<code>.width()</code>	Pobiera szerokość pierwszego z dopasowanych elementów.
<code>.width(wartość)</code>	Ustawia szerokość wszystkich dopasowanych elementów na <i>wartość</i> .
<code>.innerHeight()</code>	Pobiera wysokość pierwszego z dopasowanych elementów wraz z dopełnieniem, ale bez obramowania.
<code>.innerWidth()</code>	Pobiera szerokość pierwszego z dopasowanych elementów wraz z dopełnieniem, ale bez obramowania.
<code>.outerHeight(z-marginesem)</code>	Pobiera wysokość pierwszego z dopasowanych elementów wraz z dopełnieniem, obramowaniem oraz opcjonalnie z marginesem.
<code>.outerWidth(z-marginesem)</code>	Pobiera szerokość pierwszego z dopasowanych elementów wraz z dopełnieniem, obramowaniem oraz opcjonalnie z marginesem.
<code>.append(zawartość)</code>	Wstawia <i>zawartość</i> na końcu wnętrza każdego dopasowanego elementu.
<code>.appendTo(selektor)</code>	Wstawia dopasowane elementy na końcu wnętrza elementów dopasowanych przez selektor.
<code>.prepend(zawartość)</code>	Wstawia <i>zawartość</i> na początku wnętrza każdego dopasowanego elementu.
<code>.prependTo(selektor)</code>	Wstawia dopasowane elementy na początku wnętrza elementów dopasowanych przez selektor.
<code>.after(zawartość)</code>	Wstawia <i>zawartość</i> po każdym dopasowanym elemencie.
<code>.insertAfter(selektor)</code>	Wstawia dopasowane elementy po każdym z elementów dopasowanych przez selektor.
<code>.before(zawartość)</code>	Wstawia <i>zawartość</i> przed każdym dopasowanym elementem.
<code>.insertBefore(selektor)</code>	Wstawia dopasowane elementy przed każdym z elementów dopasowanych przez selektor.
<code>.wrap(zawartość)</code>	Opakowuje każdy z dopasowanych elementów zawartością.
<code>.wrapAll(zawartość)</code>	Opakowuje wszystkie dopasowane elementy jako całość wewnątrz zawartości.
<code>.wrapInner(zawartość)</code>	Opakowuje zawartość wewnętrzną każdego z dopasowanych elementów wewnątrz zawartości nawiasów.

Metoda	Opis
<code>.replaceWith(zawartość)</code>	Zastępuje dopasowane elementy zawartością nawiasów.
<code>.replaceAll(selektor)</code>	Zastępuje elementy dopasowane przez selektor elementami dopasowanymi.
<code>.empty()</code>	Usuwa wszystkie węzły dzieci każdego dopasowanego elementu.
<code>.remove([selektor])</code>	Usuwa dopasowane węzły (opcjonalnie filtrowane za pomocą selektora) z drzewa DOM.
<code>.clone([z-programami-obsługi])</code>	Kopiuje wszystkie dopasowane elementy, opcjonalnie również ich programy obsługi zdarzeń.
<code>.data(klucz)</code>	Pobiera element danych o nazwie <i>klucz</i> powiązany z pierwszym dopasowanym elementem.
<code>.data(klucz, wartość)</code>	Ustawia element danych o nazwie <i>klucz</i> powiązany ze wszystkimi dopasowanymi elementami na <i>wartość</i> .
<code>.removeData(klucz)</code>	Usuwa element danych o nazwie <i>klucz</i> powiązany z każdym z dopasowanych elementów.

Metody Ajaksa

Informacje możemy pobierać z serwera bez odświeżania strony, wywołując jedną z poniższych **metod opartych na Ajaksie**. Metody Ajaksa omówione są szczegółowo w rozdziale 6.

Metoda Ajaksa	Opis
<code>\$.ajax(opcje)</code>	Wykonuje oparte na Ajaksie żądanie z użyciem podanego zbioru opcji. Jest to metoda niskopoziomowa, wywoływana zazwyczaj za pośrednictwem pozostałych metod pomocniczych.
<code>.load(URL, [dane], [funkcja-zwrotna])</code>	Wykonuje oparte na Ajaksie żądanie do adresu <i>URL</i> i umieszcza odpowiedź w dopasowanych elementach.
<code>\$.get(URL, [dane], [funkcja-zwrotna])</code>	Wykonuje oparte na Ajaksie żądanie do adresu <i>URL</i> z użyciem metody GET.
<code>\$.getJSON(URL, [dane], [funkcja-zwrotna])</code>	Wykonuje oparte na Ajaksie żądanie do adresu <i>URL</i> , interpretując wyniki jako strukturę danych JSON.
<code>\$.getScript(URL, [funkcja-zwrotna])</code>	Wykonuje oparte na Ajaksie żądanie do adresu <i>URL</i> , wykonując odpowiedź jako kod w JavaScriptcie.
<code>\$.post(URL, [dane], [funkcja-zwrotna], [zwracany-typ])</code>	Wykonuje oparte na Ajaksie żądanie do adresu <i>URL</i> z użyciem metody POST.
<code>.ajaxComplete(program-obsługi)</code>	Dowiązuje program obsługi, który ma być wywołany, kiedy zakończona zostanie dowolna transakcja oparta na Ajaksie.

Metoda Ajaksa	Opis
<code>.ajaxError(program-obsługi)</code>	Dowiązuje program obsługi, który ma być wywołany, kiedy dowolna transakcja oparta na Ajaksie zakończona zostanie błędem.
<code>.ajaxSend(program-obsługi)</code>	Dowiązuje program obsługi, który ma być wywołany, kiedy rozpocznie się dowolna transakcja oparta na Ajaksie.
<code>.ajaxStart(program-obsługi)</code>	Dowiązuje program obsługi, który ma być wywołany, kiedy rozpocznie się dowolna transakcja oparta na Ajaksie, a żadne inne nie są właśnie w trakcie.
<code>.ajaxStop(program-obsługi)</code>	Dowiązuje program obsługi, który ma być wywołany, kiedy dowolna transakcja oparta na Ajaksie się zakończy, a żadne inne nie są właśnie w trakcie.
<code>.ajaxSuccess(program-obsługi)</code>	Dowiązuje program obsługi, który ma być wywołany, kiedy dowolna transakcja oparta na Ajaksie zakończy się powodzeniem.
<code>\$.ajaxSetup(opcje)</code>	Ustawia opcje domyślne dla wszystkich kolejnych transakcji opartych na Ajaksie.
<code>.serialize()</code>	Koduje wartości zbioru kontroltek formularza w łańcuch znaków zapytania.
<code>.serializeArray()</code>	Koduje wartości zbioru kontroltek formularza w strukturę danych JSON.
<code>\$.param(tablica-asocjacyjna)</code>	Koduje dowolną tablicę asocjacyjną wartości w łańcuch znaków zapytania.

Pozostałe metody

Poniższe metody pomocnicze nie pasują do żadnej z powyższych kategorii, jednak często przydają się przy pisaniu skryptów z użyciem biblioteki jQuery.

Metoda lub właściwość	Opis
<code>\$.support</code>	Zwraca tablicę asocjacyjną właściwości wskazującą, czy przeglądarka obsługuje różne opcje i standardy.
<code>\$.each(zbiór, funkcja-zwrotna)</code>	Wykonuje iterację po zbiorze, wykonując funkcję zwrotną na każdym z elementów.
<code>\$.extend(cel, dodatek, ...)</code>	Modyfikuje obiekt docelowy, dodając właściwości z innych podanych obiektów.
<code>\$.grep(tablica, funkcja-zwrotna, [odwrócenie])</code>	Filtruje tablicę, wykorzystując funkcję zwrotną jako test.
<code>\$.makeArray(obiekt)</code>	Przekształca obiekt w tablicę.

Metoda lub właściwość	Opis
<code>\$.map(tablica, funkcja-zwrotna)</code>	Konstruuje nową tablicę składającą się z wyniku funkcji zwrotnej wywołanej na każdym z elementów.
<code>\$.isArray(wartość, tablica)</code>	Ustala, czy <i>wartość</i> znajduje się w tablicy.
<code>\$.merge(tablica1, tablica2)</code>	Łączy zawartość tablic <i>tablica1</i> oraz <i>tablica2</i> .
<code>\$.unique(tablica)</code>	Usuwa wszystkie powtarzające się elementy DOM z tablicy.
<code>\$.isFunction(obiekt)</code>	Ustala, czy obiekt jest funkcją.
<code>\$.trim(łańcuch-znaków)</code>	Usuwa wszelkie białe znaki znajdujące się na końcach łańcucha znaków.
<code>\$.noConflict([extreme])</code>	Przywraca \$ definicji sprzed jQuery.
<code>.hasClass(nazwa-klasy)</code>	Ustala, czy dowolny z dopasowanych elementów ma przypisaną podaną klasę.
<code>.is(selektor)</code>	Ustala, czy dowolny z dopasowanych elementów zostaje dopasowany przez podane wyrażenie selektora.
<code>.each(funkcja-zwrotna)</code>	Wykonuje iterację po dopasowanych elementach, wykonując podaną funkcję zwrotną dla każdego z elementów.
<code>.length</code>	Zwraca liczbę dopasowanych elementów.
<code>.get()</code>	Pobiera tablicę węzłów drzewa DOM odpowiadających dopasowanemu elementom.
<code>.get(indeks)</code>	Pobiera węzeł drzewa DOM odpowiadający dopasowanemu elementowi znajdującemu się pod podanym indeksem.
<code>.index(element)</code>	Pobiera indeks podanego węzła drzewa DOM wewnątrz zbioru dopasowanych elementów.