

*Aplikacje mobilne dla każdego!*

# jQuery Mobile



HELION

O'REILLY®

*Jon Reid*

Tytuł oryginału: jQuery Mobile

Tłumaczenie: Radosław Meryk

ISBN: 978-83-246-4772-9

© 2012 Helion S.A.

Authorized Polish translation of the English edition of jQuery Mobile 9781449306687

© 2011 Jonathan Reid

This translation is published and sold by permission of O'Reilly Media, Inc., which owns or controls all rights to publish and sell the same.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from the Publisher.

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiegokolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie książki na nośniku filmowym, magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi bądź towarowymi ich właścicieli.

Wydawnictwo HELION dołożyło wszelkich starań, by zawarte w tej książce informacje były kompletne i rzetelne. Nie bierze jednak żadnej odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub autorskich. Wydawnictwo HELION nie ponosi również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Wydawnictwo HELION

ul. Kościuszki 1c, 44-100 GLIWICE

tel. 32 231 22 19, 32 230 98 63

e-mail: [helion@helion.pl](mailto:helion@helion.pl)

WWW: <http://helion.pl> (księgarnia internetowa, katalog książek)

Pliki z przykładami omawianymi w książce można znaleźć pod adresem:

<ftp://ftp.helion.pl/przyklady/jquemo.zip>

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

<http://helion.pl/user/opinie/jquemo>

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Printed in Poland.

- [Kup książkę](#)
- [Poleć książkę](#)
- [Oceń książkę](#)

- [Księgarnia internetowa](#)
- [Lubię to! » Nasza społeczność](#)

---

# Spis treści

<b>Przedmowa .....</b>	<b>7</b>
<b>1. Poznajemy jQuery Mobile .....</b>	<b>13</b>
Ogólny opis biblioteki jQuery Mobile	13
Jak działa jQuery Mobile?	14
Tworzymy pierwszą aplikację jQuery Mobile	15
Jak to działa: niestandardowy selektor jqmData()	20
<b>2. Struktura aplikacji i sposoby nawigacji .....</b>	<b>21</b>
Strony	21
Strony wewnętrzne	22
Strony zewnętrzne	24
Jak to działa? Inicjalizacja stron w jQuery Mobile	27
Zdarzenia związane z ukrywaniem i wyświetlaniem stron	29
Okna dialogowe	32
Nawigacja i historia	33
Przejścia	35
Jak to działa? Animacje w aplikacjach jQuery Mobile	36
<b>3. Elementy stron .....</b>	<b>39</b>
Jak to działa? Wtyczki i widżety jQuery	39
Listy	40
Proste listy	40
Zaawansowane listy	43
Jak to działa? Aktualizowanie list	52
Paski narzędzi	54
Paski nawigacji	54
Pozycjonowanie nagłówka i stopki	57

Przyciski	59
Grupowanie przycisków	60
Ikony przycisków	62
Elementy formularzy	64
Dostęp do elementów formularzy za pomocą JavaScript	65
Pola wyboru i przełączniki	65
Przełączniki dwustanowe	67
Pola i obszary tekstowe	69
Pola wyszukiwania	69
Menu select	71
Suwaki	75
Siatki układu dokumentu	77
<b>4. Motywy w jQuery Mobile .....</b>	<b>81</b>
Motywy i próbki	81
Jak to działa? Personalizacja próbek	91
Motywy list	95
<b>5. API jQuery Mobile .....</b>	<b>97</b>
Metody biblioteki jQuery Mobile	97
changePage	97
pageLoading	98
silentScroll	99
addResolutionBreakpoints	99
Zdarzenia	99
Zdarzenia związane z dotykiem	99
Zdarzenia inicjalizacji	102
Zdarzenia związane z ukrywaniem i wyświetlaniem stron	103
Zdarzenia związane z przewijaniem dokumentu	104
Zdarzenia związane ze zmianą orientacji urządzenia	104
API responsywnego układu dokumentu	104
Selektory CSS	105
Konfigurowanie biblioteki jQuery Mobile	108
Dostępne opcje	108
Modyfikowanie opcji za pomocą zdarzenia mobileinit	110
Jak to działa? Przestrzenie nazw atrybutów data-	111

<b>6. jQuery Mobile w akcji .....</b>	<b>113</b>
Strony aplikacji	113
Inicjalizacja aplikacji	116
Metoda initMainPage	118
Metoda initSettings	123
Metoda initDetailPage	124
Okno dialogowe do wyświetlania komunikatów o błędach	124
jqmTweet — pierwsza odsłona	125
Ulepszanie interfejsu użytkownika	127
Usprawnienia CSS	127
Usprawnienia interakcji	128
Podejście ogólne	133
<b>Skorowidz .....</b>	<b>135</b>



# Elementy stron

Podobnie jak jQuery UI, biblioteka jQuery Mobile oferuje różnorodne elementy interfejsu użytkownika. Wiele z nich bazuje na popularnych elementach interfejsu. Tworzy się je w standardowy dla jQuery Mobile sposób: najpierw należy napisać semantyczny kod, a następnie dodać do elementu atrybut `data-`. Po inicjalizacji jQuery Mobile ulepszy element w miarę potrzeb.

## Jak to działa? Wtyczki i widżety jQuery

jQuery Mobile intensywnie korzysta z dwóch standardowych wzorców projektowych biblioteki jQuery: wtyczek i widżetów.

Mechanizm wtyczek jQuery to sposób na rozszerzanie samej biblioteki jQuery. Dzięki temu mechanizmowi użytkownik może dodawać własne metody. Metody te mogą być wywoływane w identyczny sposób jak wszystkie pozostałe metody jQuery. Mechanizm wtyczek szczegółowo omówiono w dokumentacji biblioteki jQuery. Osobom nieznanąym wzorca wtyczek jQuery gorąco polecam przestudiowanie tego tematu i korzystanie z mechanizmu wszędzie tam, gdzie jest to możliwe. Osobiście używam wtyczek jQuery na co dzień, wykonując swoje zadania programistyczne. W kontekście aplikacji jQuery Mobile wzorzec ten oferuje użyteczny mechanizm enkapsulacji aplikacji, funkcji inicjalizacji stron oraz menedżerów danych.

Mechanizm widżetów biblioteki jQuery jest nieco bardziej skomplikowany od mechanizmu wtyczek, ale również daje olbrzymie możliwości. Widżety jQuery tworzy się za pomocą klasy-fabryki `jQuery.widget`. Podobnie jak w przypadku wtyczek, w rezultacie otrzymujemy metodę jQuery, którą można wywołać dla dowolnego selektora. Jedną z głównych korzyści stosowania

fabryki widżetów jest zdolność do utrzymania stanu nawet po zakończeniu działania wynikowej metody. Fabryka umożliwia działanie mechanizmu odświeżania, który zapobiega „wyciekom pamięci” w przeglądarkach.

Dla przykładu rozważmy jeden z moich ulubionych widżetów UI biblioteki jQuery — akordeon. W celu stworzenia akordeonu najpierw należy napisać kod strony, a następnie zastosować do niego metodę `jQuery.accordion()`. Podczas procesu inicjalizacji widżet akordeonu ulepsza kod, tworzy zdarzenia, do których można się dowiązywać, oraz udostępnia metody, które można wykorzystać w celu zapewnienia interakcji i modyfikowania wynikowego „akordeonu”.

Brzmi znajomo? Powinno, bowiem to jest dokładnie to, co wykonuje biblioteka jQuery Mobile. Wiele elementów stron, które będziemy poznawać, tworzy się tak, jak widżety jQuery. Sposób postępowania z nimi jest identyczny jak z widżetami jQuery.

## Listy

Biblioteka jQuery Mobile pozwala na tworzenie wizualnie sformatowanych list, które są bardzo podobne do tych występujących w aplikacjach natywnych.

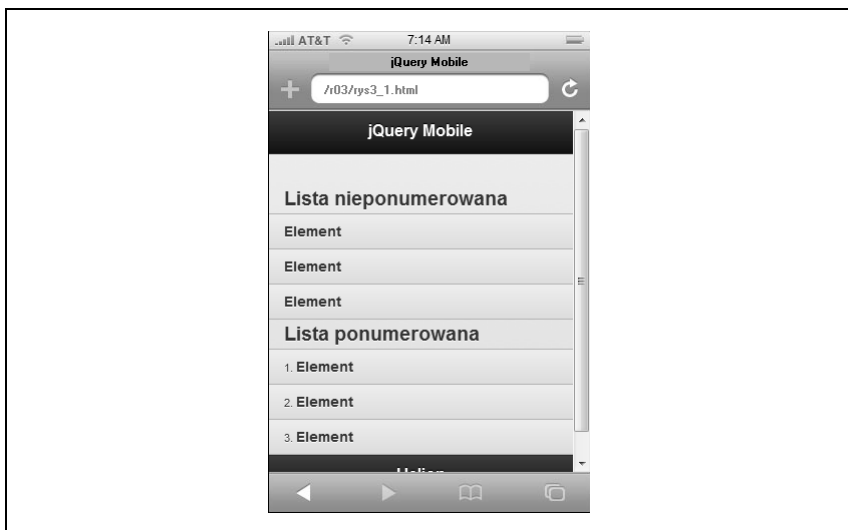
### Proste listy

Biblioteka jQuery Mobile pozwala na ulepszanie list zarówno ponumerowanych, jak i nieponumerowanych. Wystarczy dodać do listy atrybut `data-role="listview"`, a resztą zajmie się jQuery Mobile (listing 3.1, rysunek 3.1).

*Listing 3.1. Prosta lista*

```
<h3>Lista nieponumerowana</h3>
<ul data-role="listview">
  <li>Element</li>
  <li>Element</li>
  <li>Element</li>
</ul>
<h3>Lista ponumerowana</h3>
<ol data-role="listview">
  <li>Element</li>
  <li>Element</li>
  <li>Element</li>
</ol>
```





Rysunek 3.1. Prosta lista

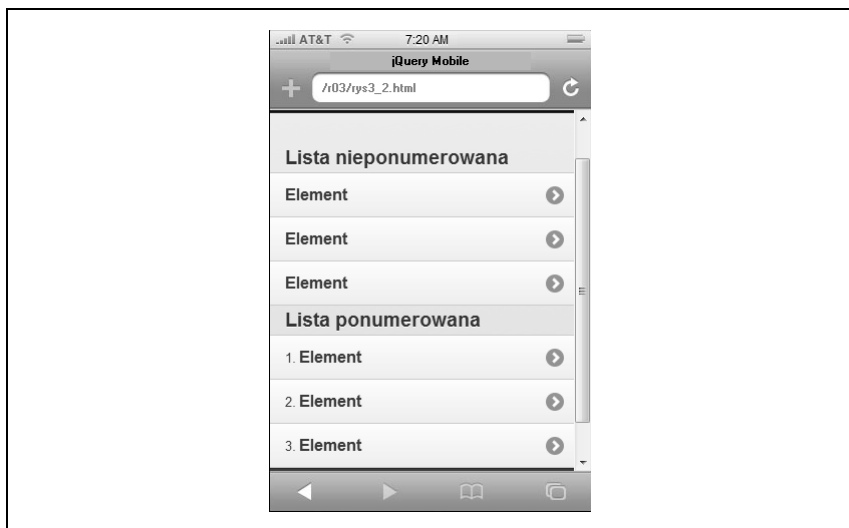
## Listy złożone z przycisków

Domyślnie elementy list nie mają właściwości interaktywnych: kliknięcie lub przeciągnięcie elementu listy nie przynosi żadnego efektu. W interfejsach użytkownika aplikacji mobilnych powszechnie stosuje się listy złożone z przycisków, które można klikać. Aby uzyskać taki efekt za pomocą jQuery Mobile, należy dołączyć znaczniki zakotwiczenia wewnątrz kodu elementów listy (listing 3.2).

Listing 3.2. Listy złożone z przycisków

```
<h3>Lista nieponumerowana</h3>
<ul data-role="listview">
  <li><a href="#">Element</a></li>
  <li><a href="#">Element</a></li>
  <li><a href="#">Element</a></li>
</ul>
<h3>Lista ponumerowana</h3>
<ol data-role="listview">
  <li><a href="#">Element</a></li>
  <li><a href="#">Element</a></li>
  <li><a href="#">Element</a></li>
</ol>
```

Teraz elementy list można klikać (patrz rysunek 3.2) (oczywiście w tym przykładzie łączy prowadzą donikąd; w pełni funkcjonalny przykład Czytelnik znajdzie w dalszej części książki).



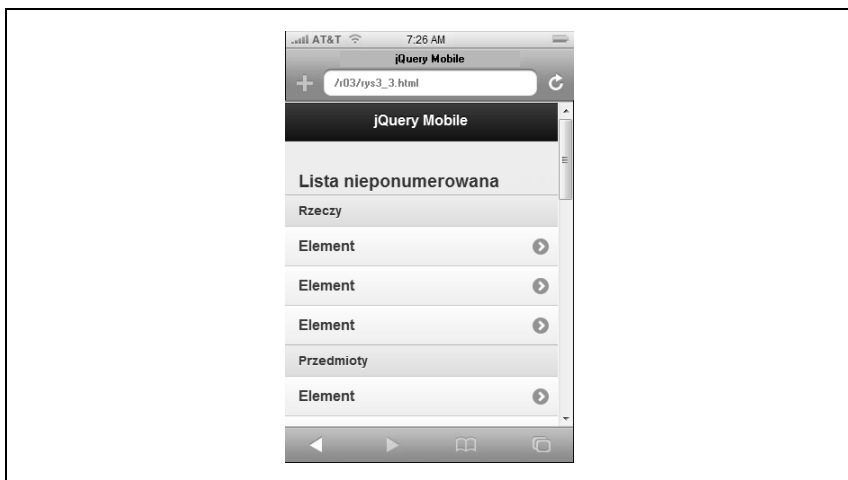
Rysunek 3.2. Lista z przyciskami

## Separatory list

Bardzo często listy są rozdzielane nagłówkami sekcji. Tego rodzaju efekt można uzyskać za pomocą jQuery Mobile poprzez zastosowanie atrybutu `data-role="list-divider"` do dowolnego elementu listy, który ma pełnić rolę separatora. Sposób użycia takiego separatora pokazano na listingu 3.3, natomiast efekty jego zastosowania na rysunku 3.3.

Listing 3.3. Listy z separatorami

```
<h3>Lista nieponumerowana</h3>
<ul data-role="listview">
  <li data-role="divider">Rzeczy</li>
  <li><a href="#">Element</a></li>
  <li><a href="#">Element</a></li>
  <li><a href="#">Element</a></li>
  <li data-role="divider">Przedmioty</li>
  <li><a href="#">Element</a></li>
  <li><a href="#">Element</a></li>
  <li data-role="divider">Różne</li>
  <li><a href="#">Element</a></li>
  <li><a href="#">Element</a></li>
```



Rysunek 3.3. Listy z separatorami

```

<li><a href="#">Element</a></li>
<li><a href="#">Element</a></li>
<li><a href="#">Element</a></li>
</ul>
<h3>Lista ponumerowana</h3>
<ol data-role="listview">
  <li data-role="divider">Grupa</li>
  <li><a href="#">Element</a></li>
  <li><a href="#">Element</a></li>
  <li><a href="#">Element</a></li>
  <li data-role="divider">Grupa</li>
  <li><a href="#">Element</a></li>
  <li><a href="#">Element</a></li>
  <li data-role="divider">Grupa</li>
  <li><a href="#">Element</a></li>
  <li><a href="#">Element</a></li>
  <li><a href="#">Element</a></li>
  <li><a href="#">Element</a></li>
  <li><a href="#">Element</a></li>
</ol>

```

## Zaawansowane listy

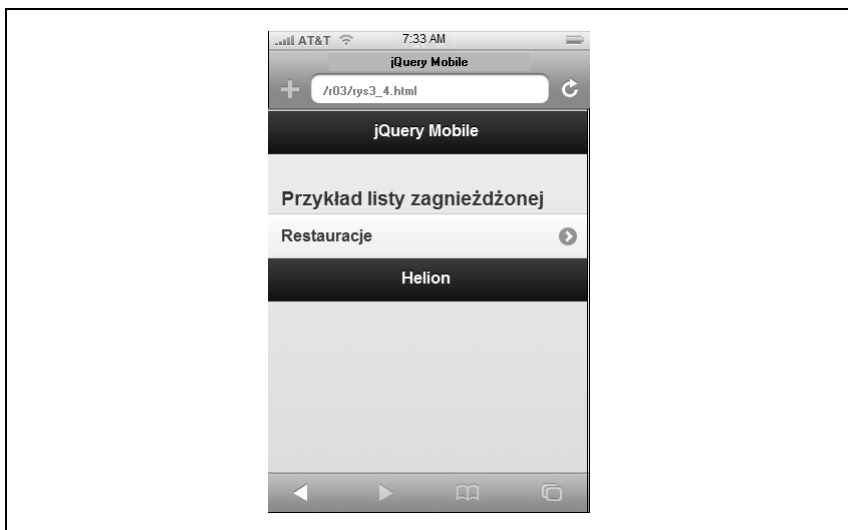
Listy są bardzo często wykorzystywanym elementem interfejsu użytkownika w aplikacjach mobilnych. Z tego powodu biblioteka jQuery Mobile pozwala na generowanie popularnych modyfikacji za pomocą niewielkich zmian w kodzie strony.

## Listy zagnieżdżone

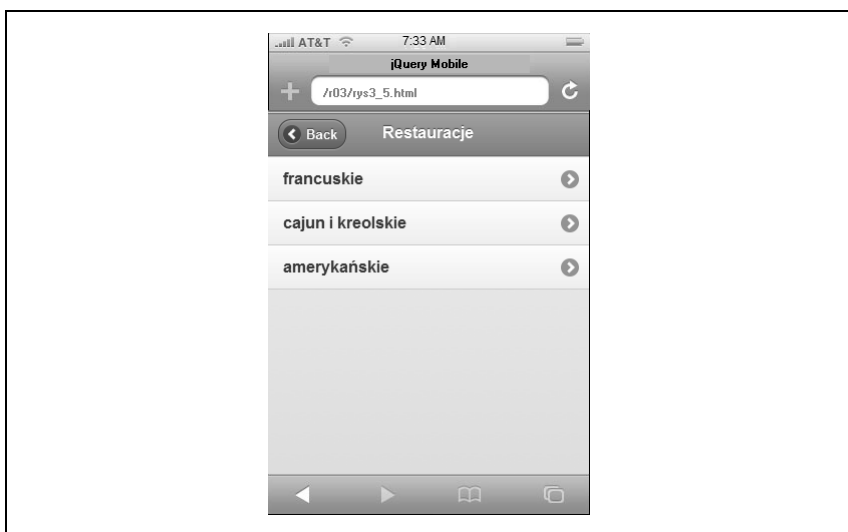
jQuery Mobile przetwarza zagnieżdżone listy i na tej podstawie tworzy interaktywne widoki, w które użytkownik może się zagłębiać, klikając pojedyncze elementy. Pierwszy z widoków składa się z elementów listy najwyższego poziomu. Kliknięcie jednego z tych elementów powoduje wyświetlenie listy podrzędnej itd. Kod takiej listy zamieszczono na listingu 3.4. Wynikową listę pokazano na rysunkach 3.4, 3.5 i 3.6. jQuery Mobile automatycznie wstawia przycisk *Wstecz* oraz zarządza mapowaniem adresów URL i przejściami pomiędzy stronami.

Listing 3.4. Listy zagnieżdżone

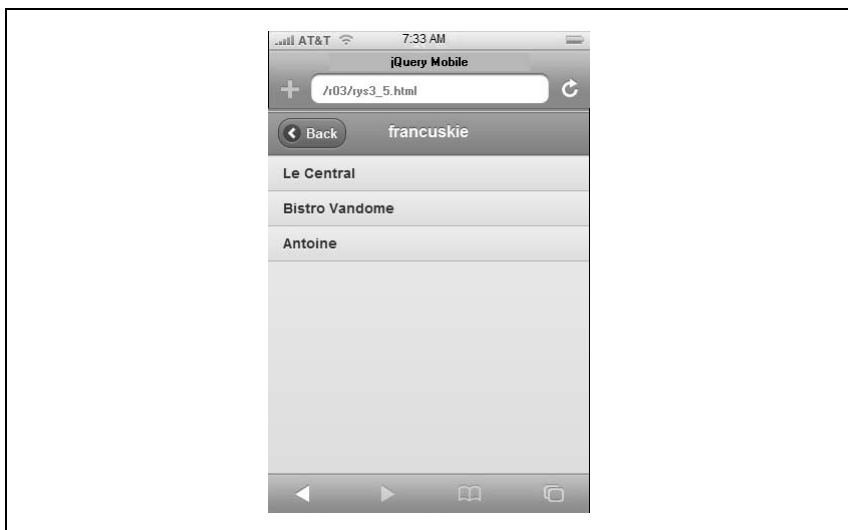
```
<h3>Przykład listy zagnieżdżonej</h3>
<ul data-role="listview">
  <li>Restauracje
    <ul>
      <li>francuskie
        <ul>
          <li>Le Central</li>
          <li>Bistro Vandome</li>
          <li>Antoine</li>
        </ul>
      </li>
      <li>cajun i kreolskie
        <ul>
          <li>Bayou Bob's</li>
          <li>Pappadeaux</li>
          <li>Lucile's</li>
        </ul>
      </li>
      <li>amerykańskie
        <ul>
          <li>Dixon</li>
          <li>Vesta Dipping Grill</li>
          <li>Steuben</li>
        </ul>
      </li>
    </ul>
  </li>
</ul>
```



Rysunek 3.4. Lista zagnieżdżona (strona 1)



Rysunek 3.5. Lista zagnieżdżona (strona 2)



Rysunek 3.6. Lista zagnieżdżona (strona 3)

## Listy złożone z przycisków dzielonych

Biblioteka jQuery Mobile pozwala na tworzenie list złożonych z przycisków dzielonych: przycisków składających się z dwóch różnych aktywnych obszarów. Główny obszar po lewej stronie przycisku jest szerszy, natomiast obszar po stronie prawej jest węższy. Przyciski dzielone są powszechnie stosowane w interfejsach użytkownika aplikacji mobilnych. Dzięki nim jeden element listy może spełniać dwie różne funkcje. Można to wykorzystać na przykład do podglądu szczegółów elementu listy lub do skonfigurowania elementu w określony sposób albo — tak jak w przypadku kodu z listingu 3.5 — do przeglądania szczegółów restauracji lub dokonywania rezerwacji.

Aby stworzyć przycisk dzielony, należy dodać dwa znaczniki zakotwiczenia do elementu listy.

Listing 3.5. Tworzenie przycisków dzielonych

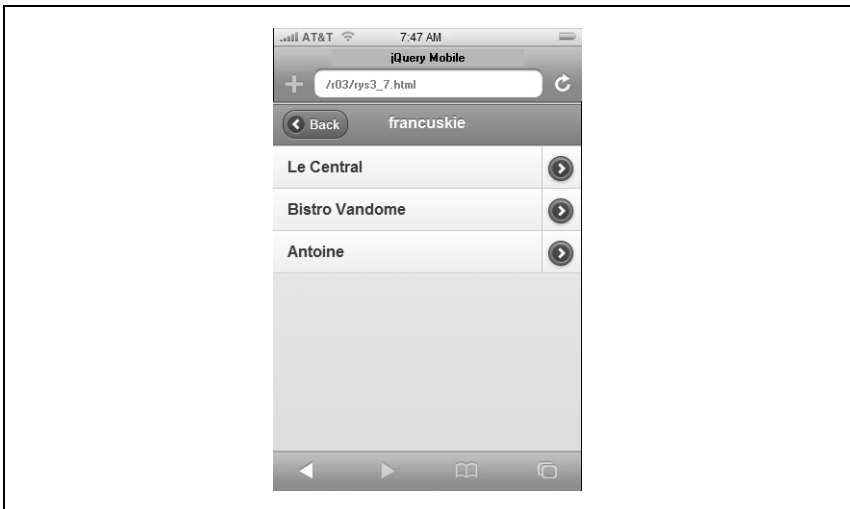
```
<li>francuskie
  <ul>
    <li>
      <a href="/lecentral.html">Le Central</a>
      <a href="/reservations.php?restaurant=903">Zarezerwuj stolik</a>
    </li>
    <li>
      <a href="/bistrovandome.html">Bistro Vandome</a>
    </li>
  </ul>
</li>
```

```

        <a href="reservations.php?restaurant=904">Zarezerwuj stolik</a>
    </li>
    <li>
        <a href="antoinnes.html">Antoine</a>
        <a href="reservations.php?restaurant=905">Zarezerwuj stolik</a>
    </li>
</ul>
</li>

```

jQuery Mobile automatycznie przypisuje pierwszy znacznik zakotwiczenia jako główne łącze przycisku oraz ostatni znacznik zakotwiczenia jako mniejszy obszar z prawej strony. Przykład zaprezentowano na rysunku 3.7.



Rysunek 3.7. Lista z przyciskami dzielonymi

Z elementem listy można powiązać więcej niż dwa znaczniki zakotwiczenia, ale jQuery Mobile wykorzysta tylko pierwszy i ostatni. Pozostałe znaczniki zakotwiczenia staną się częścią głównego przycisku tak jak zwykłe łącza.

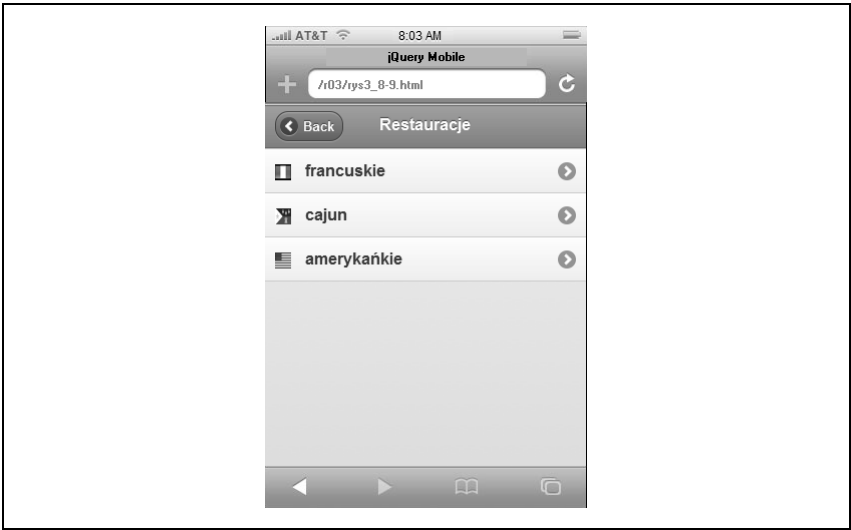
## Miniaturki i ikony

Można też określić obrazek miniaturki bądź ikony dla każdego elementu listy. W tym celu należy dołączyć w kodzie strony odpowiednie odwołanie do obrazka (listing 3.6). Miniaturki i obrazki są wyświetlane po lewej stronie elementu listy, natomiast ikony są mniejsze i są wyrównywane w pionie do środka elementu listy. Biblioteka jQuery Mobile udostępnia klasę CSS, którą można zastosować do obrazka w celu wskazania, że to jest ikona (rysunek 3.8), a nie miniaturka (rysunek 3.9).

### Listing 3.6. Ikony i miniaturki na liście

```
<ul data-role="listview">
<li>Restauracje
  <ul>
    <li>
      <a href="#">
        
          francuskie
        </a>
      <ul>
        <li>
          <a href="lecentral.html">
            
            Le Central
          </a>
          <a href="reservations.php?restaurant=403">Zarezerwuj stolik</a>
        </li>
        <li>
          <a href="bistrovandome.html">
            
            Bistro Vandome
          </a>
          <a href="reservations.php?restaurant=404">Zarezerwuj stolik
          </a>
        </li>
        <li>
          <a href="antoines.html">
            
            Antoine
          </a>
          <a href="reservations.php?restaurant=405">Zarezerwuj
          ↳stolik</a>
        </li>
      </ul>
    </li>
  </ul>
</li>
<li>
  <a href="#">
    
      cajun
    </a>
    <ul>
      <li>Bayou Bob's</li>
      <li>Pappadeaux</li>
      <li>Lucile's</li>
    </ul>
  </li>
</ul>
```





Rysunek 3.8. Lista z ikonami

```

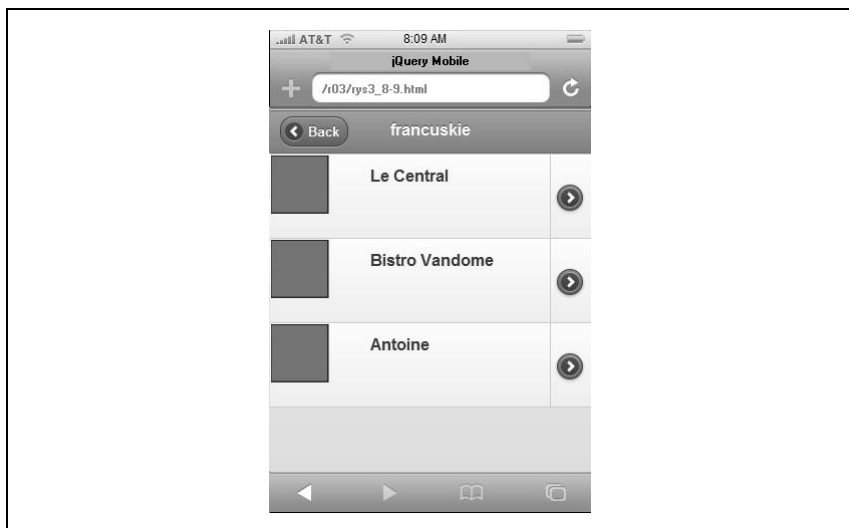
<li>
  <a href="#">
    
    amerykańskie
  </a>
</li>
</ul>
<li>Dixon's</li>
<li>Vesta Dipping Grill</li>
<li>Steuben's</li>
</ul>
</li>
</ul>
</ul>

```



W czasie kiedy powstawała niniejsza książka, w jQuery Mobile ikony i miniaturki były zawarte wewnątrz znaczników kotwicy w obrębie elementów listy nawet wtedy, kiedy element listy nie miał być aktywny<sup>1</sup>. Bez znacznika kotwicy formatowanie nie będzie poprawne. Zamieszczenie znacznika kotwicy, który nie jest semantycznie poprawny, nie stanowi problemu, ponieważ jQuery Mobile ignoruje wszystkie łącza do "#".

<sup>1</sup> Element aktywny to taki, który można klikać — *przyp. tłum.*



Rysunek 3.9. Lista z miniaturkami

## Baloniki liczników

Baloniki liczników to umieszczone na listach ikony zawierające informacje o liczbie elementów z określonej grupy na liście. Aby umieścić ikonę licznika za pomocą jQuery Mobile, należy dołączyć w kodzie strony klasę `ui-li-count`, tak jak pokazano na listingu 3.7. Efekt jest widoczny na rysunkach 3.10, 3.11 i 3.12.

Listing 3.7. Baloniki liczników elementów list

```

<ul data-role="listview">
  <li data-role="divider">Elementy<span class="ui-li-count">10</span></li>
  <li><a href="#">Restauracje</a><span class="ui-li-count">9</span>
    <ul>
      <li>
        <a href="#">
          
          francuskie
          <span class="ui-li-count">3</span>
        </a>
      </li>
      <li>
        <a href="lecentral.html">

```

```

        
        Le Central
    </a>
    <span class="ui-li-count">9</span>
    <a href="reservations.php?restaurant=403">Zarezerwuj
    ↳stolik</a>
</li>
<li>
    <a href="bistrovandome.html">
        
        Bistro Vandome
    </a>
    <a href="reservations.php?restaurant=404">Zarezerwuj
    ↳stolik</a>
</li>
<li>
    <a href="antoines.html">
        
        Antoine
    </a>
    <a href="reservations.php?restaurant=405">Zarezerwuj
    ↳stolik</a>
</li>
</ul>
</li>
<li>
    <a href="#">
        
        cajun
        <span class="ui-li-count">3</span>
    </a>
    <ul>
        <li>Bayou Bob's</li>
        <li>Pappadeaux</li>
        <li>Lucile's</li>
    </ul>
</li>
<li>
    <a href="#">
        
        amerykańskie
        <span class="ui-li-count">3</span>
    </a>
    <ul>
        <li>Dixon's</li>

```

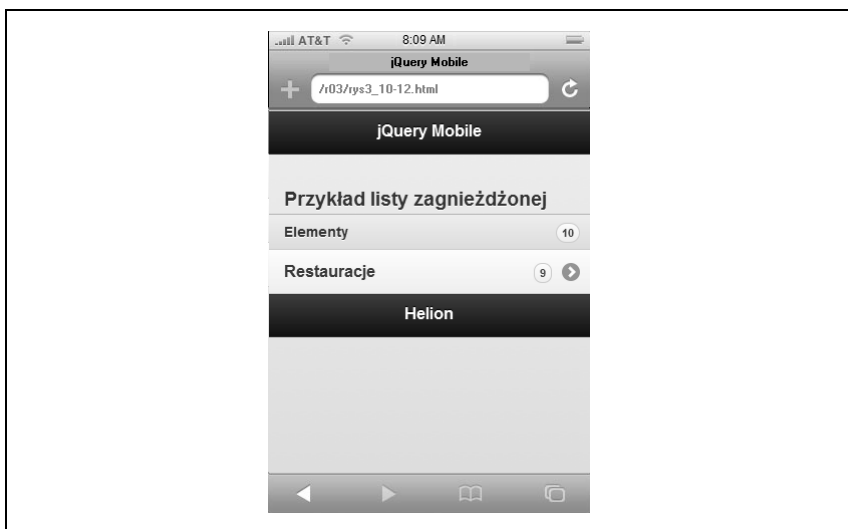
```

<li>Vesta Dipping Grill</li>
<li>Steuben's</li>
</ul>
</li>
</ul>

```



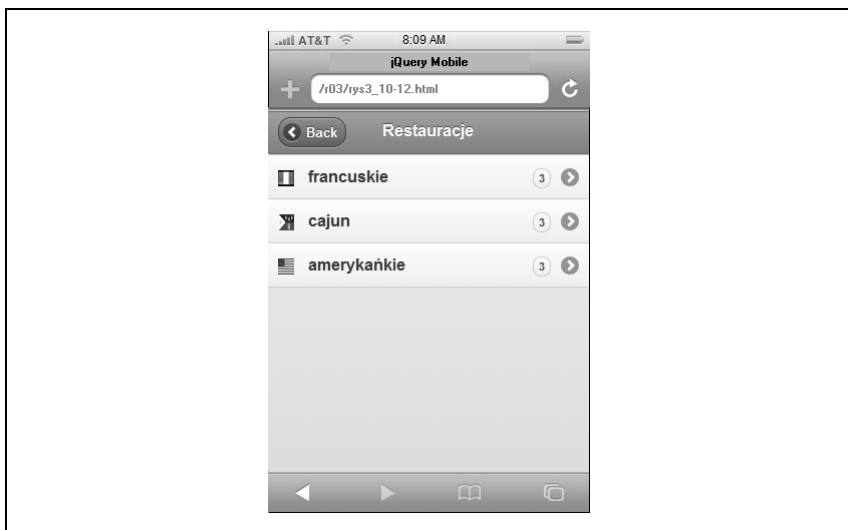
W czasie kiedy powstawała niniejsza książka, liczniki działały tylko wtedy, kiedy zostały umieszczone wewnątrz znaczników kotwicy lub wewnątrz separatorów list (elementów list z atrybutami `data-role="divider"`). W przyciskach dzielonych liczniki nie mogły być umieszczane w ostatnim znaczniku kotwicy.



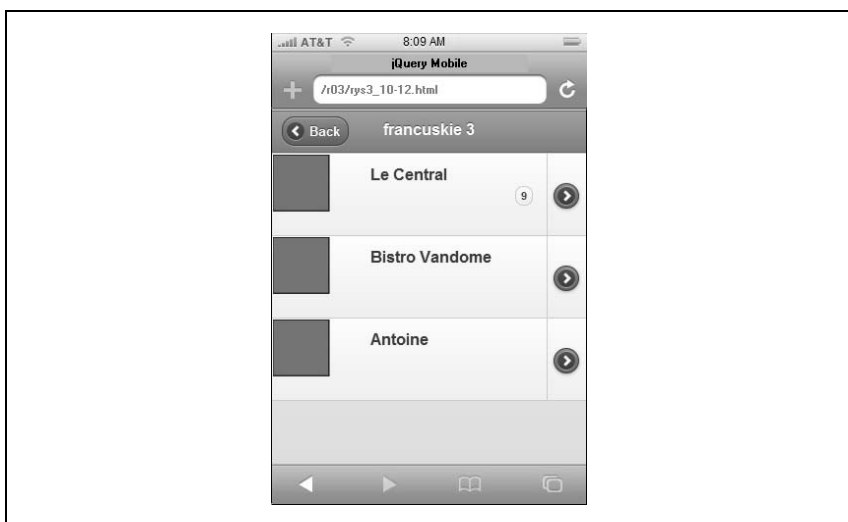
Rysunek 3.10. Baloniki liczników na listach (strona 1)

## Jak to działa? Aktualizowanie list

Listy należą do najbardziej złożonych elementów biblioteki jQuery Mobile. Jednocześnie należą do elementów najczęściej wykorzystywanych w aplikacjach.



Rysunek 3.11. Baloniki liczników na listach (strona 2)



Rysunek 3.12. Baloniki liczników na listach (strona 3)

Listy jQuery Mobile tworzy się za pomocą widżetu listview. Podobnie jak inne widżety jQuery, listview udostępnia metody, które można wywołać po inicjalizacji. W czasie kiedy powstawała niniejsza książka, widżet listview udostępniał metodę refresh służącą do aktualizowania listy. Metoda

ta najbardziej przydaje się w sytuacji, gdy trzeba dodać lub usunąć elementy z wcześniej zainicjowanej listy (listing 3.8).

*Listing 3.8. Aktualizowanie listy — dodawanie nowych elementów*

```
<script>

// Zainicjowanie obiektu listy
$("#ul.twitter-feed").listview();

var updateTweets = function() {
    // Pobranie nowych tweetów, dodanie ich na początek listy
    // i odświeżenie listy
    var strNewTweetsHtml = getNewTweets();
    $("#ul.twitter-feed").prepend(strNewTweetsHtml).listview("refresh");
}

</script>
```

Powyższy przykład pokazuje jedną z zalet wzorca widżetów jQuery. Jest nią dołączanie metod do elementów modelu DOM. Dzięki temu metody te można wywoływać po inicjalizacji.

## Paski narzędzi

Biblioteka jQuery Mobile pozwala na tworzenie kilku różnych typów pasków narzędzi. W pierwszym z prezentowanych przykładów spotkaliśmy nagłówek i stopkę. jQuery Mobile umożliwia także tworzenie pasków nawigacji.

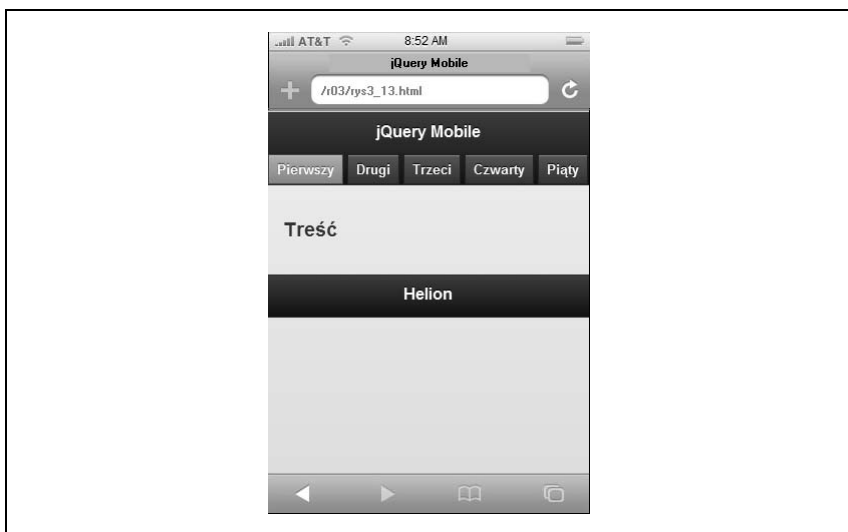
## Paski nawigacji

W interfejsie użytkownika aplikacji mobilnych paski nawigacji zwykle składają się ze zbioru przycisków pozwalających użytkownikom na przemieszczanie się pomiędzy widokami aplikacji. Paski nawigacji można umieszczać wewnątrz nagłówek, stopki lub obszarów treści widoku strony jQuery Mobile. W zależności od miejsca jQuery Mobile odpowiednio formatuje pasek nawigacji.

Aby oznaczyć pasek nawigacji, należy dodać atrybut `data-role="navigation"` do elementu poziomemu bloku, na przykład elementu HTML5 `nav`. Znaczniki kotwic zawarte w oznaczonym elemencie nawigacji są sformatowane jako grupa przycisków. jQuery Mobile automatycznie zarządza zmianami stanów aktywności i nieaktywności przycisków (listing 3.9). Efekty zaprezentowano na rysunku 3.13.

### Listing 3.9. Prosty pasek nawigacji

```
<section id="page1" data-role="page">
  <header data-role="header">
    <h1>jQuery Mobile</h1>
    <nav data-role="navbar">
      <a href="#" class="ui-btn-active">Pierwszy</a>
      <a href="#">Drugi</a>
      <a href="#">Trzeci</a>
      <a href="#">Czwarty</a>
      <a href="#">Piąty</a>
    </nav>
  </header>
  <div class="content" data-role="content">
    <h3>Treść</h3>
  </div>
  <div data-role="content">
    <h3>Helion</h3>
  </div>
</section>
```



Rysunek 3.13. Prosty pasek nawigacji

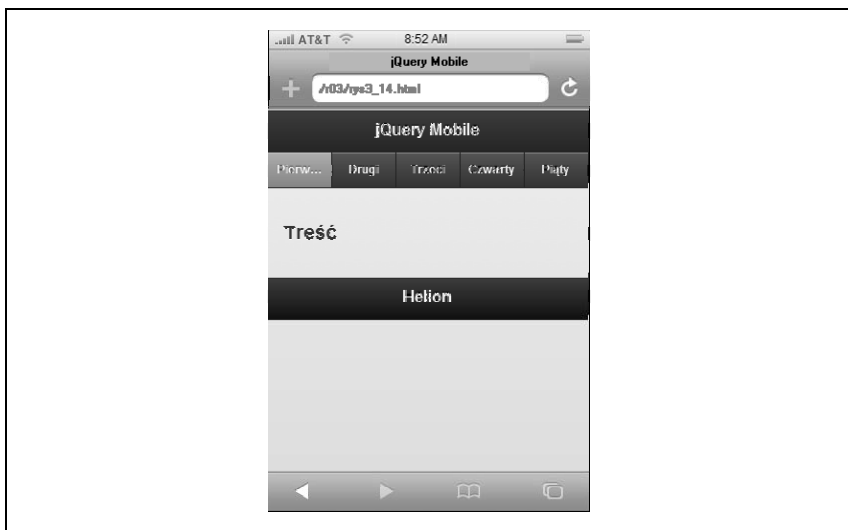


Klasa CSS `ui-btn-active` określa stan aktywności przycisków nawigacji.

jQuery Mobile pozwala także na tworzenie przycisków o równej szerokości. Aby uzyskać taki efekt, należy zamknąć znaczniki kotwicy wewnątrz listy nieponumerowanej (listing 3.10). Efekty zaprezentowano na rysunku 3.14.

Listing 3.10. Sformatowany pasek nawigacji

```
<nav data-role="navbar">
  <ul>
    <li><a href="#" class="ui-btn-active">Pierwszy</a></li>
    <li><a href="#">Drugi</a></li>
    <li><a href="#">Trzeci</a></li>
    <li><a href="#">Czwarty</a></li>
    <li><a href="#">Piąty</a></li>
  </ul>
</nav>
```



Rysunek 3.14. Sformatowany pasek nawigacji

Biblioteka jQuery Mobile pozwala na umieszczenie do pięciu elementów na sformatowanym pasku nawigacji. W przypadku próby wstawienia większej liczby elementów jQuery Mobile rozmieści je w dwóch kolumnach. Przykład zaprezentowano na listingu 3.11, a efekty pokazano na rysunku 3.15.

Listing 3.11. Pasek nawigacji z przyciskami w dwóch kolumnach

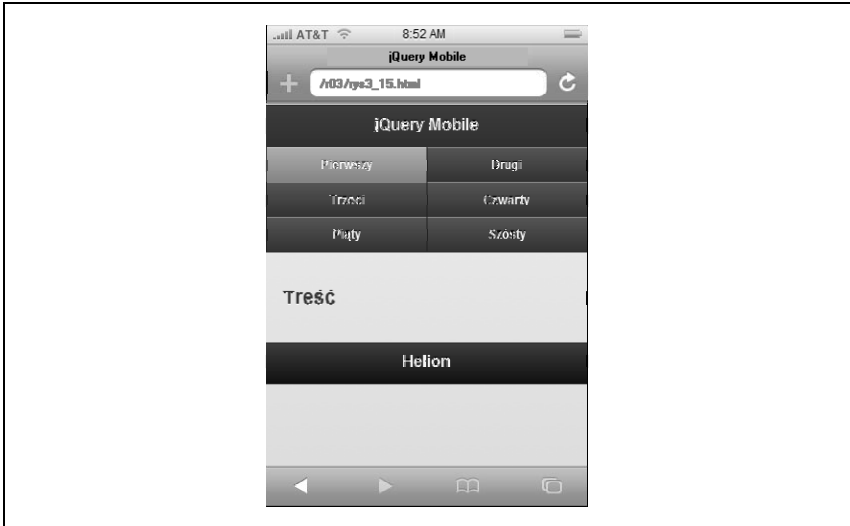
```
<nav data-role="navbar">
  <ul>
    <li><a href="#" class="ui-btn-active">Pierwszy</a></li>
    <li><a href="#">Drugi</a></li>
  </ul>
```



```

<li><a href="#">Trzeci</a></li>
<li><a href="#">Czwarty</a></li>
<li><a href="#">Piąty</a></li>
<li><a href="#">Szósty</a></li>
</ul>
<nav>

```



Rysunek 3.15. Pasek nawigacji z przyciskami w dwóch kolumnach

## Pozycjonowanie nagłówka i stopki

Biblioteka jQuery Mobile oferuje trzy sposoby dynamicznego pozycjonowania pasków narzędzi nagłówka i stopki:

### Standardowy

Paski narzędzi są rozmieszczone zgodnie z „przepływem dokumentu”. Przewijanie dokumentu powoduje znikanie i pojawianie się pasków narzędzi na widocznej części strony. To jest opcja domyślna.

### Ustalony

Nagłówek i stopka są umieszczone w górnej i dolnej części widocznego fragmentu strony. Pozostają widoczne przez cały czas, niezależnie od pozycji przeglądanej dokumentu. Kliknięcie ekranu powoduje ich powrót na standardową pozycję w układzie dokumentu.

## Pełnoekranowy

Nagłówek i stopka są umieszczone wewnątrz widocznego fragmentu strony i pozostają widoczne przez cały czas, niezależnie od pozycji przeglądane dokumentu. Kliknięcie ekranu powoduje ukrycie nagłówka i stopki. W istocie nagłówek i stopka są usuwane z dokumentu i dynamicznie pozycjonowane w dolnej i górnej części widocznego fragmentu strony.

Aby stworzyć nagłówek i stopkę o ustalonej pozycji, należy zastosować dla nich atrybut `data-position="fixed"` (listing 3.12).

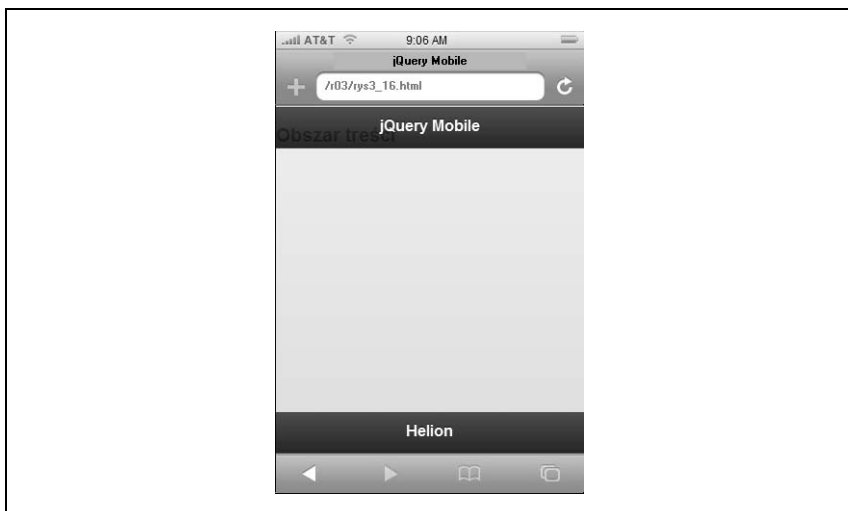
### Listing 3.12. Nagłówek i stopka o ustalonej pozycji

```
<section id="page1" data-role="page">
  <header data-role="header" data-position="fixed">
    <h1>jQuery Mobile</h1>
  </header>
  <div class="content" data-role="content">
    <h3>Obszar treści</h3>
  </div>
</div>
<footer data-role="footer" data-position="fixed">
  <h3>Helion</h3>
</footer>
</section>
```

Aby stworzyć nagłówek i stopkę w trybie pełnoekranowym, należy zastosować atrybut `data-fullscreen="true"` do elementu oznaczonego atrybutem `data-role="page"` oraz równocześnie zastosować atrybut `data-position="fixed"` do elementów nagłówka i stopki:

```
<section id="page1" data-role="page" data-fullscreen="true">
  <header data-role="header" data-position="fixed">
    <h1>jQuery Mobile</h1>
  </header>
  <div class="content" data-role="content">
    <h3>Obszar treści</h3>
  </div>
</div>
<footer data-role="footer" data-position="fixed">
  <h3>Helion</h3>
</footer>
</section>
```

Dynamicznie pozycjonowane paski narzędzi zasłaniają treść, natomiast one same pozostają widoczne (przykład pokazano na rysunku 3.16).



Rysunek 3.16. Nagłówek i stopka z dynamicznie ustaloną pozycją

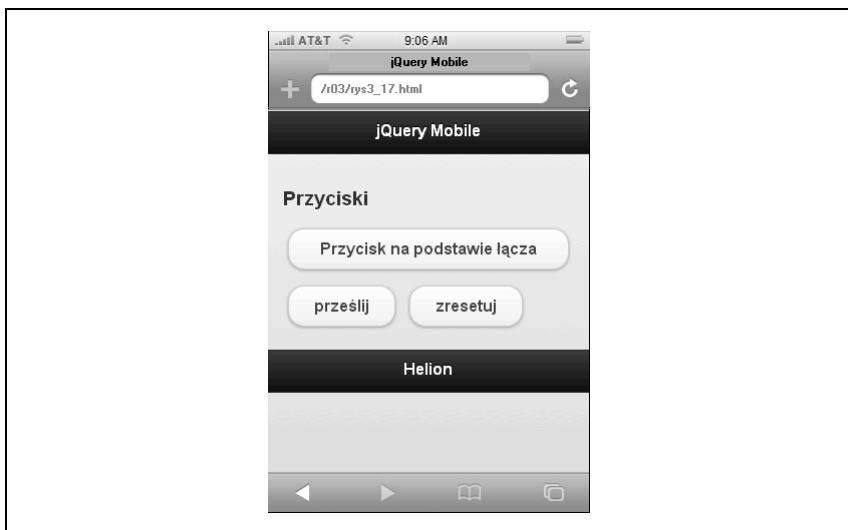
## Przyciski

Biblioteka jQuery Mobile automatycznie tworzy stylizowane przyciski na podstawie zwykłych przycisków formularzy niezależnie od tego, czy zostaną one stworzone za pomocą znaczników `input`, czy `button`. jQuery Mobile potrafi także tworzyć przyciski ze zwykłych łącz (znaczników kotwicy). Wystarczy, że zastosujemy do nich atrybut `data-role="button"`.

Domyślnie przyciski są rozciągane do szerokości elementu-kontenera. Poprzez zastosowanie atrybutu `data-inline="true"` do przycisku można tworzyć przyciski wstawiane o wielkości dostosowanej do treści, która jest na nich umieszczona (listing 3.13, rysunek 3.17).

Listing 3.13. Przyciski, przyciski, komu przyciski?

```
<section id="page1" data-role="page">
  <header data-role="header"><h1>jQuery Mobile</h1></header>
  <div class="content" data-role="content">
    <h3>Przyciski</h3>
    <a href="#" data-role="button">Przycisk na podstawie łącza</a>
    <input type="submit" value="prześlij" data-inline="true">
    <input type="reset" value="zresetuj" data-inline="true">
  </div>
</div>
  <footer data-role="footer"><h1>Helion</h1></footer>
</section>
```



Rysunek 3.17. Przyciski, przyciski, komu przyciski?



W czasie kiedy powstawała niniejsza książka, w dokumentacji jQuery Mobile można było przeczytać, że w wyniku zastosowania atrybutu `data-inline="true"` do elementu-kontenera wszystkie przyciski w nim zawarte będą wyrenderowane jako wstawiane. To nie działa. Atrybut `data-inline="true"` trzeba zastosować do każdego przycisku z osobna. Więcej informacji na ten temat można uzyskać pod adresem <http://jquerymobile.com/demos/1.0a4.1/docs/buttons/buttons-inline.html>. Sądzę, że ten błąd zostanie wyeliminowany do wydania 1.0.

## Grupowanie przycisków

Przyciski mogą być grupowane. W ten sposób można stworzyć zbiór przycisków na jednym pasku. Aby uzyskać taki efekt, wystarczy opakować przyciski w elemencie-kontenera, a następnie zastosować do niego atrybut `data-role="controlgroup"`.

Domyślnie grupy kontrolki tworzą pionowe listy. Jeśli jednak zastosujemy do kontenera atrybut `datatype="horizontal"` (tak jak to zrobiłem na listingu 3.14), to przyciski zostaną wyrenderowane w trybie wstawianym (ang. *inline*), tak jak pokazano na rysunku 3.18.

Listing 3.14. Grupy przycisków

```
<section id="page1" data-role="page">
  <header data-role="header"><h1>jQuery Mobile</h1></header>
  <div class="content" data-role="content">
    <h3>Grupy przycisków</h3>
    <div data-role="controlgroup">
      <a href="#" data-role="button">Tak</a>
      <a href="#" data-role="button">Nie</a>
      <a href="#" data-role="button">Anuluj</a>
    </div>
    <div data-role="controlgroup" data-type="horizontal">
      <a href="#" data-role="button">Tak</a>
      <a href="#" data-role="button">Nie</a>
      <a href="#" data-role="button">Anuluj</a>
    </div>
  </div>
  <div>
  </div>
  <footer data-role="footer"><h1>Helion</h1></footer>
</section>
```



Rysunek 3.18. Grupy przycisków



















Grupy przycisków nie działają tak samo jak przełączniki. W określonym momencie może być aktywny więcej niż jeden przycisk w grupie.

## Ikony przycisków

Często występuje potrzeba umieszczania ikon na przyciskach. Biblioteka jQuery Mobile daje dostęp do zbioru ikon. Istnieje także możliwość decydowania o pozycji, w jakiej ma być umieszczona ikona.

Aby umieścić ikonę na przycisku, należy wykorzystać atrybut `data-icon`. Dozwolone wartości atrybutu `data-icon` zestawiono w tabeli 3.1.

Tabela 3.1. Prawidłowe wartości atrybutu `data-icon`

Wartość	Ikona
alert	
arrow-d	
arrow-l	
arrow-r	
arrow-u	
back	
check	
delete	
forward	
gear	
grid	
home	
info	
minus	
plus	
refresh	
search	
star	

Pozycję ikony można określić za pomocą atrybutu `data-iconpos`. Wartości tego atrybutu zestawiono w tabeli 3.2.

Kod umożliwiający dodanie niektórych spośród ikon przycisków pokazano na listingu 3.15. Uzyskane efekty zaprezentowano na rysunku 3.19.

Tabela 3.2. Prawidłowe wartości atrybutu `data-iconpos`

Wartość	Efekt
bottom	Poniżej tekstu, wyśrodkowana
left	Po lewej stronie przycisku
notext	Tekst ukryty, wyświetla się tylko ikona
right	Po prawej stronie przycisku
top	Powyżej tekstu, wyśrodkowana

Listing 3.15. Ikony przycisków

```

<section id="page1" data-role="page">
  <header data-role="header"><h1>jQuery Mobile</h1></header>
  <div class="content" data-role="content">
    <h3>Ikony przycisków</h3>
    <div data-role="controlgroup" data-type="horizontal" data-inline="true">
      <a href="#" data-role="button" data-icon="home"
        ↪data-iconpos="left">Strona główna</a>
      <a href="#" data-role="button" data-icon="gear"
        data-iconpos="left">Opcje</a>
      <a href="#" data-role="button" data-icon="search"
        data-iconpos="left">Szukaj</a>
    </div>
  </div>
</div>
</div>
<footer data-role="footer"><h1>Helion</h1></footer>
</section>

```



Rysunek 3.19. Ikony przycisków

## Ikony użytkownika

Choć biblioteka jQuery Mobile zawiera zestaw ikon wbudowanych, z łatwością można implementować własne. Wystarczy skorzystać z CSS oraz atrybutu `data-icon`. Jeśli wartość atrybutu `data-icon` nie jest jedną z dostępnych wartości, jQuery Mobile stworzy przycisk, ale zastosuje do niego klasę CSS zdefiniowaną przez użytkownika. Wewnątrz tej klasy powinna znaleźć się wartość atrybutu `data-icon` poprzedzona prefiksem `ui-icon-`. Na przykład, jeśli użyjemy kodu `data-icon="flagicons-french"`, to jQuery Mobile wygeneruje przycisk i utworzy go wewnątrz znacznika `span` z klasą CSS `ui-icon-flagicons-french`.

Klasy ikon przycisków biblioteki jQuery Mobile zaprojektowano na bazie kwadratowych ikon o wymiarach 18×18 pikseli. W przypadku zastosowania systemu motywów biblioteki jQuery Mobile (zobacz rozdział 4.) ikony należy zapisywać w formacie PNG z zastosowaniem przezroczystości alfa.

## Elementy formularzy

jQuery Mobile automatycznie ulepsza natywne elementy formularzy za pomocą komponentów interfejsu użytkownika wzmocnionych o obsługę zdarzeń dotyku.

Najpierw należy stworzyć semantyczny kod strony, розміścić odpowiednie etykiety elementów oraz opakować wszystkie pola formularzy w prawidłowe znaczniki form.

Można także wizualnie pogrupować elementy w obrębie formularza. W tym celu należy zastosować atrybut `data-role="fieldcontain"` do grupy elementów opakowanych właściwym znacznikiem. Idealnie nadają się do tego znaczniki `fieldset` lub `div`.

jQuery Mobile automatycznie obsłuży przesłanie strony za pomocą technologii AJAX oraz podejmie próbę zintegrowania odpowiedzi serwera z modelem DOM aplikacji, dostarczając odpowiednich animowanych przejść. Aby zablokować obsługę formularza za pośrednictwem tej technologii dla wskazanego formularza, należy zastosować do znacznika `form` atrybut `data-ajax="false"`.

Na koniec, jeśli nie chce się, aby biblioteka jQuery Mobile ulepszała elementy formularzy lub całe formularze, należy zastosować do elementu bądź znacznika `form` atrybut `data-role="none"`.



# Dostęp do elementów formularzy za pomocą JavaScript

Podobnie jak inne elementy stron, elementy formularzy jQuery Mobile są zarządzane przez różnorodne widżety jQuery. Jedną z największych zalet mechanizmu widżetów biblioteki jQuery jest możliwość dostępu do metod widżetu już po zainicjowaniu elementu formularza. Programista może skorzystać z udostępnionych metod widżetów do wykonywania różnych użytecznych działań: odczytywania i ustawiania zaznaczonych wartości, modyfikowania informacji, włączania i wyłączania elementów, ich ręcznej inicjalizacji, przywracania ich do stanu sprzed ulepszenia itd.

## Pola wyboru i przełączniki

jQuery Mobile automatycznie ulepsza pola wyboru i przełączniki w interfejsie użytkownika, przekształcając je w elementy przypominające przyciski. Podobnie jak inne przyciski, elementy te mogą być grupowane poprzez zastosowanie w odniesieniu do elementu-kontenera atrybutu `data-role="controlgroup"`. Domyślnie grupy kontrolki tworzą pionowe kolumny przycisków zajmujące pełną szerokość kontenera. Aby stworzyć poziomą grupę kontrolki, należy użyć atrybutu `datatype="horizontal"`. Przykład można znaleźć na listingu 3.16, a uzyskany efekt obejrzeć na rysunku 3.20.

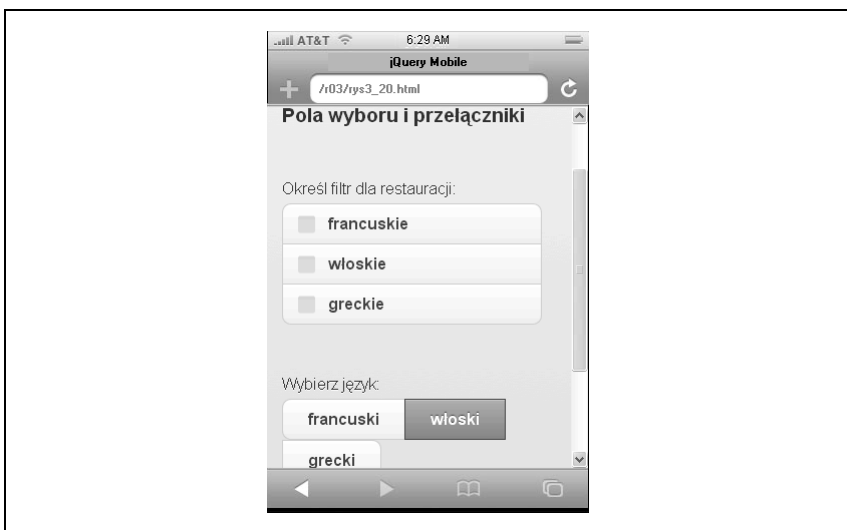
Listing 3.16. Pola wyboru i przełączniki

```
<section id="page1" data-role="page">
  <header data-role="header"><h1>jQuery Mobile</h1></header>
  <div class="content" data-role="content">
    <h3>Pola wyboru i przełączniki</h3>
    <form id="myform" action="formprocessor.php" method="post">
      <div data-role="fieldcontain">
        <fieldset data-role="controlgroup">
          <legend>Określ filtr dla restauracji:</legend>
          <input type="checkbox" name="checkbox-french"
            ↪ id="checkbox-french">
          <label for="checkbox-french">Francuskie</label>
          <input type="checkbox" name="checkbox-italian"
            ↪ id="checkbox-italian">
          <label for="checkbox-italian">Włoskie</label>
          <input type="checkbox" name="checkbox-greek"
            ↪ id="checkbox-greek">
          <label for="checkbox-greek">Greckie</label>
        </fieldset>
      <div data-role="fieldcontain">
        <fieldset data-role="controlgroup" data-type="horizontal">
```

```

<legend>Wybierz język:</legend>
<input type="radio" name="radio-language" id="radio-french">
<label for="radio-french">Francuski</label>
<input type="radio" name="radio-language" id="radio-italian"
checked="checked">
<label for="radio-italian">Włoski</label>
<input type="radio" name="radio-language" id="radio-greek">
<label for="radio-greek">Grecki</label>
</fieldset>
</div>
</form>
</div>
</div>
<footer data-role="footer"><h1>Helion</h1></footer>
</section>

```



Rysunek 3.20. Pola wyboru i przełączniki

## Metody

W bibliotece jQuery Mobile do zarządzania polami wyboru i przełącznikami wykorzystywany jest widżet `checkboxradio`. Można go wykorzystać do włączenia, wyłączenia lub odświeżenia wskazanego pola wyboru lub przełącznika (listing 3.17).

Listing 3.17. Metoda `checkboxradio`

```

<script>
// Zablokowanie przycisku

```

```

$("#checkbox-french").checkboxradio("disable");

// Włączenie zablokowanego przycisku
$("#checkbox-italian").checkboxradio("enable");

// Ręczne uaktywnienie przycisku i odświeżenie jego statusu w interfejsie użytkownika
$("#radio-greek").attr("checked", true).checkboxradio("refresh");
</script>

```

## Przełączniki dwustanowe

Przełączniki dwustanowe (ang. *flip toggle*) to bardzo często stosowane elementy interfejsu użytkownika aplikacji mobilnych. Przełączniki tego rodzaju pozwalają na wybór pomiędzy dwiema wartościami (np. „włączony” i „wyłączony”). Użytkownik może przełączyć stan przełącznika, przeciągając przełącznik bądź stukając w niego.

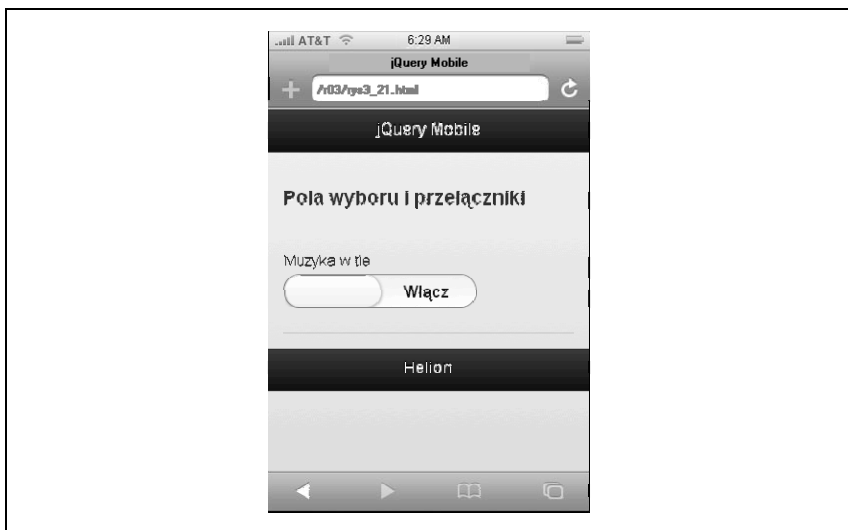
Do tworzenia przełączników dwustanowych w bibliotece jQuery Mobile wykorzystywany jest element formularza HTML `select`. Wystarczy utworzyć element `select` z dwiema opcjami i zastosować do niego atrybut `data-role="slider"`. Przykład pokazano na listingu 3.18, a efekty zaprezentowano na rysunku 3.21.

Listing 3.18. Przełącznik dwustanowy

```

<section id="page1" data-role="page">
  <header data-role="header"><h1>jQuery Mobile</h1></header>
  <div class="content" data-role="content">
    <h3>Pola wyboru i przełączniki</h3>
    <form id="myform" action="formprocessor.php" method="post">
      <div data-role="fieldcontain">
        <label for="slider-music">Muzyka w tle</label>
        <select name="slider-music" id="slider-music" data-role="slider">
          <option value="on">Włącz</option>
          <option value="off">Wyłącz</option>
        </select>
      </div>
    </form>
  </div>
  <footer data-role="footer"><h1>Helion</h1></footer>
</section>

```



Rysunek 3.21. Przełącznik dwustanowy

## Metody

W bibliotece jQuery Mobile do zarządzania przełącznikami dwustanowymi wykorzystywany jest widżet slider. Na listingu 3.19 pokazano, w jaki sposób uzyskać bezpośredni dostęp do przełącznika dwustanowego.

Listing 3.19. Metoda slider

```
<script>
$(document).ready(function() {

    var mySlider = $("#slider-music");

    // Zablokowanie suwaka
    mySlider.slider('disable');

    // Odblokowanie suwaka
    mySlider.slider("enable");

    // Ręczne przełączenie suwaka i zaktualizowanie jego statusu w interfejsie użytkownika
    // (samo przełączenie nie spowoduje automatycznego zaktualizowania interfejsu użytkownika)
    mySlider[0].selectedIndex = 1;
    mySlider.slider("refresh");

});
</script>
```

## Pola i obszary tekstowe

Biblioteka jQuery Mobile udostępnia również pola i obszary tekstowe. Elementy te są domyślnie wizualnie ulepszone przez zaokrąglenie narożników i zastosowanie cieni, choć z łatwością można to zmienić (zobacz rozdział 4.).

Do pól tekstowych można stosować atrybut `type` wraz z określeniem jednej z kilku nowych wartości zdefiniowanych w HTML5 umożliwiających wyświetlenie właściwego rodzaju klawiatury. Na przykład pole tekstowe z atrybutem `type="number"` w większości urządzeń mobilnych spowoduje wyświetlenie klawiatury numerycznej.

Wysokość obszarów tekstowych powiększa się, w miarę jak użytkownik wpisuje w nie tekst. Dzięki temu można uniknąć tworzenia pasków przewijania.

### Metody

W bibliotece jQuery Mobile do zarządzania obszarami i polami tekstowymi wykorzystywany jest widżet `textinput` (listing 3.20).

Listing 3.20. Metoda `textinput`

```
<script>
$(document).ready(function() {

    // Zablokowanie pola tekstowego
    $("#myinput").textinput('disable');

    // Odblokowanie pola tekstowego
    $("#myotherinput").textinput("enable");

});
</script>
```

## Pola wyszukiwania

Pola wyszukiwania to pola tekstowe, dla których zastosowano dodatkowy styl. Przede wszystkim jest na nich ikona wyszukiwania (szkło powiększające) umieszczona z lewej strony pola tekstowego. Kiedy użytkownik zaczyna wpisywać tekst w polu, z jego prawej strony wyświetla się ikona zerowania pola (znak „x”). Kliknięcie tej ikony spowoduje wyczyszczenie treści wprowadzonej w polu.

Aby stworzyć pole wyszukiwania, wystarczy zastosować do pola tekstowego atrybut `type="search"`. Przykład pokazano na listingu 3.21, a uzyskany efekt na rysunku 3.22.

Listing 3.21. Pole wyszukiwania

```
<section id="page1" data-role="page">
  <header data-role="header"><h1>jQuery Mobile</h1></header>
  <div class="content" data-role="content">
    <h3>Pole wyszukiwania</h3>
    <form id="myform" action="formprocessor.php" method="post">
      <div data-role="fieldcontain">
        <label for="search-restaurants">Szukaj restauracji:</label>
        <input type="search" name="search-restaurants"
          ↪ id="search-restaurants" />
      </div>
    </form>
  </div>
  <footer data-role="footer"><h1>Helion</h1></footer>
</section>
```

## Metody

Pole wyszukiwania to nic innego jak pole tekstowe z odpowiednim stylem. W związku z tym do jego obsługi wykorzystuje się widżet `textinput` (zobacz listing 3.20).



Rysunek 3.22. Pole wyszukiwania

## Menu select

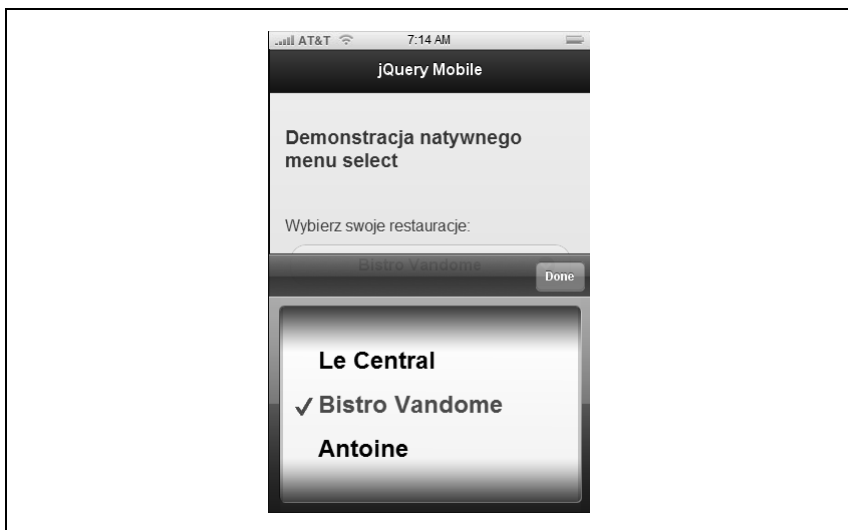
Menu select należą do najbardziej progresywnie ulepszanych elementów w bibliotece jQuery Mobile. Biblioteka pozwala na używanie natywnego dla platformy widżetu select albo na stworzenie własnego menu select z wykorzystaniem stylu zdefiniowanego przez użytkownika.

Menu select bazujące na własnym stylu ma istotną przewagę w porównaniu z używaniem natywnego widżetu select. Na niektórych platformach natywny widżet select jest pozbawiony ważnych właściwości (na przykład w systemie Android brakuje obsługi optgroup). Zastosowanie menu select bazującego na własnym stylu umożliwi odtworzenie tych właściwości. Oprócz tego można dodać nowe właściwości, na przykład możliwość wielokrotnego wyboru. Łatwiej również dopasować styl elementu do wizualnego motywu aplikacji.

Domyślnie biblioteka jQuery Mobile próbuje wykorzystać natywny widżet select. W tym celu nie trzeba robić niczego szczególnego. Wystarczy stworzyć element select oraz zdefiniować jego opcje. Aby zmusić bibliotekę jQuery Mobile do utworzenia menu select bazującego na stylu użytkownika, należy zastosować do elementu select atrybut `data-native-menu="false"`. Przykład pokazano na listingu 3.22, a efekty zaprezentowano na rysunku 3.23.

*Listing 3.22. Natywne menu select*

```
<section id="page1" data-role="page">
  <header data-role="header"><h1>jQuery Mobile</h1</header>
  <div class="content" data-role="content">
    <h3>Demonstracja natywnego menu select</h3>
    <form id="myform" action="formprocessor.php" method="post">
      <div data-role="fieldcontain">
        <label for="select-restaurants">Wybierz swoje restauracje:</label>
        <select id="select-restaurants" name="select-restaurants">
          <option value="lecentral">Le Central</option>
          <option value="bistrovandome">Bistro Vandome</option>
          <option value="antoines">Antoine</option>
        </select>
      </div>
    </form>
  </div>
  <footer data-role="footer"><h1>Helion</h1</footer>
</section>
```



Rysunek 3.23. Natywne menu select

## Menu select bazujące na stylu użytkownika

Menu select bazujące na stylu użytkownika charakteryzują się kilkoma ważnymi właściwościami:

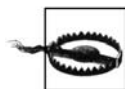
**Elementy zablokowane.** Za pomocą atrybutu `disabled` znacznika `option` można określić elementy zablokowane.

**Wielokrotny wybór.** Menu select bazujące na stylu użytkownika pozwala na wielokrotny wybór. Aby użyć tej własności, należy dodać atrybut `multiple` do elementu `select`. W takiej sytuacji biblioteka jQuery Mobile doda do menu następujące własności:

- Rozwijane menu będzie opatrzone nagłówkiem zawierającym przycisk zamknięcia oraz tekst-rozdzielacz (zobacz „Rozdzielacze” poniżej). Kliknięcie przycisku zamknięcia spowoduje zamknięcie menu. Dzięki temu rozwijane menu może pozostać otwarte w czasie, kiedy użytkownik wybiera z niego wiele pozycji. Standardowo, gdy użytkownik wybierze jeden element z menu select, menu zamknie się automatycznie.
- Po prawej stronie każdej pozycji menu wyświetlą się pola wyboru, które wskazują na to, czy określony element został wybrany, czy nie.
- Jeśli użytkownik wybierze z menu jeden lub więcej elementów, na przycisku wyboru wyświetli się licznik.



- Po zamknięciu rozwijanego menu tekst każdego wybranego elementu zostanie dodany do przycisku wyboru. Teksty zbyt długie, aby mogły się zmieścić na przycisku, zostaną obcięte.
- Jeśli lista wyboru jest zbyt długa, jQuery Mobile wyświetli ją w osobnym oknie zamiast w oknie pop up. Dzięki temu przewijanie okna staje się łatwiejsze.



W momencie powstawania niniejszej książki własność wielokrotnego wyboru była obciążona kilkoma drobnymi błędami. Należy dokładnie przetestować tę własność w aplikacji, aby sprawdzić, czy działa zgodnie z oczekiwaniami.

**Grupy opcji.** Biblioteka jQuery Mobile interpretuje znaczniki `optgroup` służące do grupowania znaczników `option` i tworzenia nagłówek dzielących wynikową listę w oknie pop up.

**Rozdzielacze.** Można je definiować na jeden z trzech sposobów: poprzez pozostawienie pustego znacznika `option` zarówno tekstu, jak i wartości, pozostawienie pustego znacznika `option` co do wartości (ale pozostawienie tekstu) albo zastosowanie atrybutu `data-placeholder="true"` do znacznika `option`.

Jeśli istnieje tekst w opcji `placeholder`, biblioteka jQuery Mobile wyświetli go na przycisku `select`.

Wszystkie własności menu bazujących na stylu użytkownika zestawiono na listingu 3.23. Efekt pokazano na rysunku 3.24.

*Listing 3.23. Menu select bazujące na stylu użytkownika*

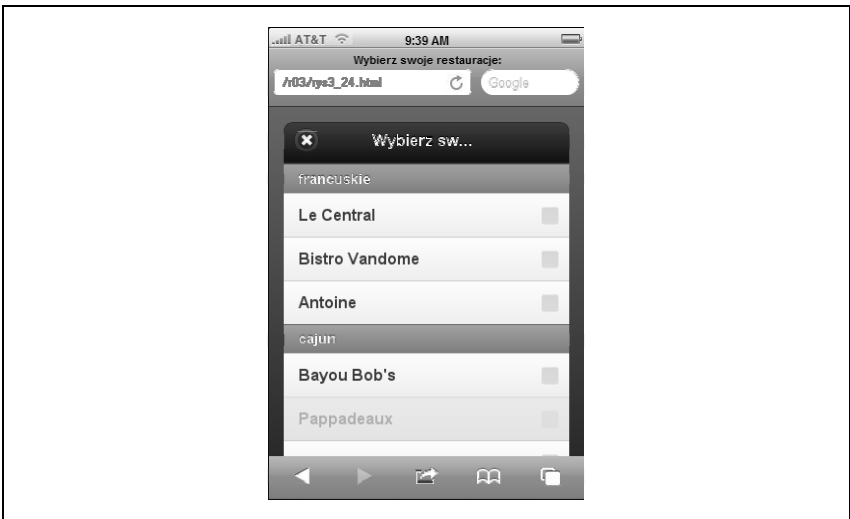
```
<section id="page1" data-role="page">
  <header data-role="header"><h1>jQuery Mobile</h1></header>
  <div class="content" data-role="content">
    <h3>Demonstracja użycia menu select</h3>
    <form id="myform" action="formprocessor.php" method="post">
      <div data-role="fieldcontain">
        <label for="select-restaurants">Wybierz swoje
          ↪ restauracje:</label>
        <select id="select-restaurants"
          name="select-restaurants"
          data-native-menu="false"
          multiple="multiple">
          <option value="choose" data-placeholder="true">
            ↪ Wybierz...</option>
        <optgroup label="francuskie">
          <option value="lecentral">Le Central</option>
          <option value="bistrovandome">Bistro Vandome</option>
          <option value="antoines">Antoine</option>
        </select>
      </div>
    </form>
  </div>
</section>
```

```

</optgroup>
<optgroup label="cajun">
  <option value="bayoubobs">Bayou Bob's</option>
  <option value="pappadeaux" disabled=
    ↳"disabled">Pappadeaux</option>
  <option value="luciles">Lucile's</option>
</optgroup>
</select>
</div>

</form>
</div>
<footer data-role="footer"><h1>Helion</h1></footer>
</section>

```



Rysunek 3.24. Menu select zdefiniowane przez użytkownika

## Metody

W bibliotece jQuery Mobile do zarządzania menu select wykorzystywany jest widżet selectmenu (listing 3.24).

Listing 3.24. Metoda selectmenu

```

<script>
$(document).ready(function() {
  mySelect = $("#select-restaurants");

  // Zablokowanie menu select
  mySelect.selectmenu("disable");

```

```

// Odblokowanie menu select
mySelect.selectmenu("enable");

// Ręczne otwarcie menu select
mySelect.selectmenu("open");

// Ręczne zamknięcie menu select
mySelect.selectmenu("close");

// Ręczne wybranie elementu i odświeżenie jego stanu w interfejsie użytkownika
mySelect[0].selectedIndex = 2;
mySelect.selectmenu("refresh");

// Wymuszenie odbudowania menu select.
// Przydatne w przypadku dynamicznego dodania bądź usunięcia opcji.
mySelect.selectmenu("refresh", true);
});
</script>

```

## Suwaki

Suwaki to elementy pozwalające użytkownikom na wybieranie jednej ze zbioru wartości przez przesuwanie kursora.

Aby stworzyć suwak, należy zastosować do pola tekstowego atrybut `type="range"`, tak jak pokazano na listingu 3.25. Dodatkowo można ustawić atrybuty `min` i `max`. Efekty zaprezentowano na rysunku 3.25 (w czasie powstawania niniejszej książki suwaki nie obsługiwały atrybutu `step`).

Listing 3.25. Suwak

```

<section id="page1" data-role="page">
  <header data-role="header"><h1>jQuery Mobile</h1></header>
  <div class="content" data-role="content">
    <h3>Demonstracja użycia suwaka</h3>
    <form id="myform" action="formprocessor.php" method="post">
      <div data-role="fieldcontain">
        <input type="range" id="slider" name="slider" min="0" max="500">
      </div>
    </form>
  </div>
  <footer data-role="footer"><h1>Helion</h1></footer>
</section>

```

Użytkownik może przesuwać suwakiem w obie strony albo może wprowadzić wartość w polu tekstowym, a pozycja suwaka automatycznie się zaktualizuje.



Rysunek 3.25. Suwak

## Metody

Do obsługi suwaka w bibliotece jQuery Mobile służy metoda `slider` (listing 3.26).

Listing 3.26. Metoda `slider`

```
<script>
$(document).ready(function() {

    // Zablokowanie suwaka
    $("#slider").slider("disable");

    // Odblokowanie suwaka
    $("#slider").slider("enable");

    // Ręczne ustawienie wartości suwaka i odświeżenie jego stanu w interfejsie użytkownika
    $("#slider").val(100).slider("refresh");
});
</script>
```

# Siatki układu dokumentu

Biblioteka jQuery Mobile daje dostęp do zbioru wbudowanych siatek układu zawierających od dwóch do pięciu kolumn. Siatki układu dokumentu przydają się w sytuacjach, w których w dokumencie występują mniejsze elementy, takie jak przyciski bądź pola formularzy, które chcemy rozmieścić z większą dokładnością.

Siatki układu bazują na strukturze złożonej z elementów div: najpierw tworzy się element div, który ma zawierać siatkę, a następnie, wewnątrz, dodaje się elementy div będące komórkami. Element div, który jest kontenerem, ma przypisaną odpowiednią klasę ui-grid: ui-grid-a dla 2 kolumn, ui-grid-b itd. aż do ui-grid-d dla 5 kolumn. Elementy div, które tworzą komórki, mają przypisane klasy ui-blok w zależności od ich pozycji: ui-block-a dla pierwszej, ui-block-b dla drugiej itd. aż do ui-block-e dla piątej.

Aby utworzyć wiele wierszy, należy dodać więcej elementów div komórek i nadać im odpowiednie klasy w taki sposób, aby każda kolumna miała przypisaną własną klasę ui-block (listing 3.27). Efekty pokazano na rysunkach 3.26 i 3.27.

Listing 3.27. Siatki układu

```
<!DOCTYPE html>
<html>
  <head>
    <title>Aplikacja jQuery Mobile</title>
    <link rel="stylesheet" href="http://code.jquery.com/mobile/1.0a4.1/
    ↪jquery.mobile-1.0a4.1.min.css" />
    <script src="http://code.jquery.com/jquery-1.5.2.min.js"></script>
    <script src="http://code.jquery.com/mobile/1.0a4.1/
    ↪jquery.mobile-1.0a4.1.min.js"></script>
    <style>
      .content div div p {
        background-color: #ccc;
        height: 50px;
        border: 1px solid #333;
        margin: 0px;
      }
    </style>
  </head>
  <body>
    <section id="page1" data-role="page">
      <header data-role="header"><h1>jQuery Mobile</h1></header>
      <div class="content" data-role="content">
```

```

<h4>2 kolumny</h4>
<div class="ui-grid-b">
  <div class="ui-block-a"><p>Blok A</p></div>
  <div class="ui-block-b"><p>Blok B</p></div>
</div>

<h4>3 kolumny</h4>
<div class="ui-grid-b">
  <div class="ui-block-a"><p>Blok A</p></div>
  <div class="ui-block-b"><p>Blok B</p></div>
  <div class="ui-block-c"><p>Blok C</p></div>
</div>

<h4>4 kolumny</h4>
<div class="ui-grid-c">
  <div class="ui-block-a"><p>Blok A</p></div>
  <div class="ui-block-b"><p>Blok B</p></div>
  <div class="ui-block-c"><p>Blok C</p></div>
  <div class="ui-block-d"><p>Blok D</p></div>
</div>

<h4>5 kolumn</h4>
<div class="ui-grid-d">
  <div class="ui-block-a"><p>Blok A</p></div>
  <div class="ui-block-b"><p>Blok B</p></div>
  <div class="ui-block-c"><p>Blok C</p></div>
  <div class="ui-block-d"><p>Blok D</p></div>
  <div class="ui-block-e"><p>Blok E</p></div>
</div>

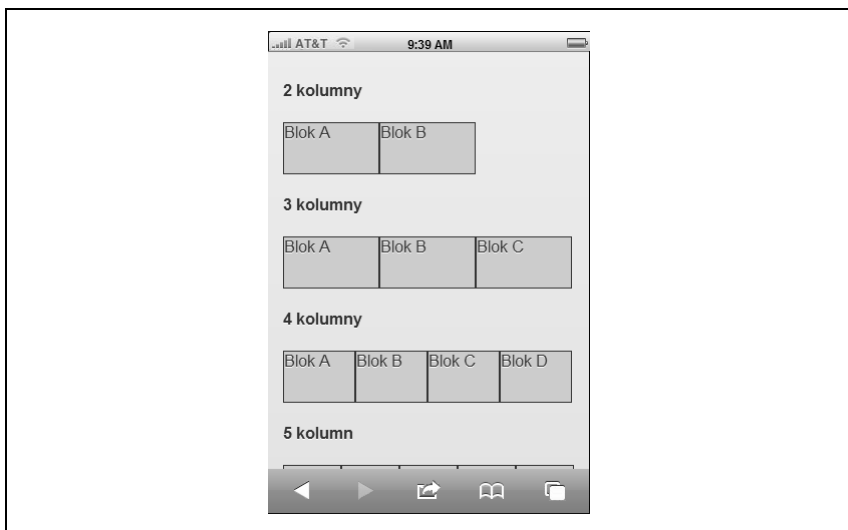
<h4>2 wiersze po 3 kolumny</h4>
<div class="ui-grid-d">
  <div class="ui-block-a"><p>Blok A</p></div>
  <div class="ui-block-b"><p>Blok B</p></div>
  <div class="ui-block-c"><p>Blok C</p></div>
  <div class="ui-block-a"><p>Blok A</p></div>
  <div class="ui-block-b"><p>Blok B</p></div>
  <div class="ui-block-c"><p>Blok C</p></div>
</div>

</div>
<footer data-role="footer"><h1>Helion</h1></footer>
</section>

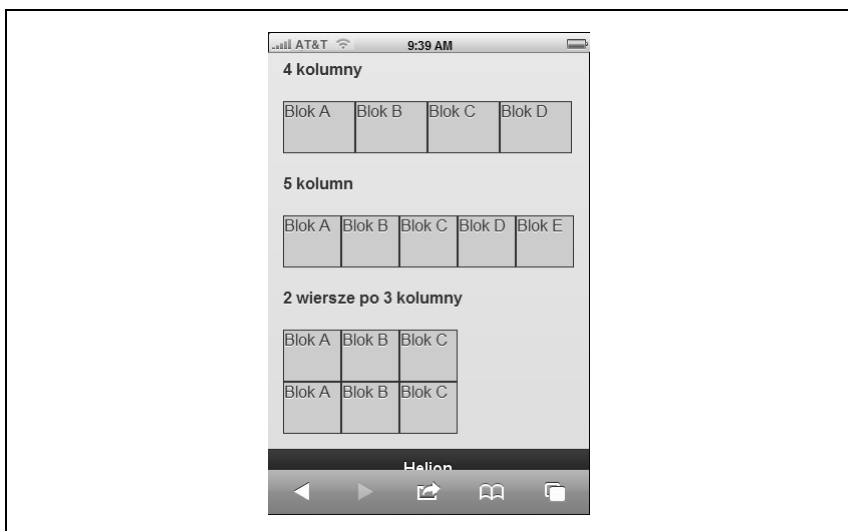
</body>
</html>

```

Na rysunkach 3.26 i 3.27 umieściłem cały dokument zamiast widoku strony po to, aby pokazać regułę CSS, którą dodałem, by bloki stały się widoczne.



Rysunek 3.26. Siatka układu numer 1



Rysunek 3.27. Siatka układu numer 2





---

# Skorowidz

## A

adres URL, 33  
AJAX, 64, 120  
akcent wizualny, 81  
animacja, 14, 19, 35, 36  
API, 97

- responsywnego układu dokumentu, 104
- RESTful, 119
- serwisu Twitter, 120
- wieloplatformowe, 13

aplikacja

- enkapsulacja, 39
- inicjalizacja, 116
- mobilna, 13, 81
- mobilna webowa, 7
- natywna, 7
- stan zajętości, 98

arkusz stylów, 14, 37, 91  
atrybut

- data-split-icon, 95
- ARIA, 27, 28
- data, 27
- data-, 20, 39
- data-count-theme, 95
- data-icon, 62, 64
- data-iconpos, 62
- data-inline, 60
- data-rel, 32
- datarole, 60, 64
- data-role, 42, 54, 65
- data-split-theme, 95
- data-split-icon, 95
- data-theme, 82, 95
- datatype, 65
- dataurl, 34

disabled, 72  
obsługi danych, 14  
type, 69, 70, 75

## B

balonik licznika, 50, 95  
biblioteka Crockforda, 119  
jQuery Mobile, 8, 13, 14, 20, 28, 36, 39, 66, 67, 68, 69, 71, 74, 76, 97, 106

- konfigurowanie, 108
- system motywów, 64

JSON.js, 119  
błędy, 124

## C

CDN, 14, 91  
cień, 81  
czcionka, 81

## E

ekran, 104, 106  
emulator, 17

## F

Firebug, 29  
Firefox, 29  
flip toggle, *Patrz:* przełącznik dwustanowy  
formularz, 64

- blokowanie, 64
- kontrolka, 27

- G**
- GitHub, 20
- I**
- ikona, 47, 62
- przycisku dzielonego, 95
  - użytkownika, 64
  - wyszukiwania, 69
  - zerowania pola, 69
- interfejs użytkownika, 14, 39, 64, 65, 67, 81, 127
- J**
- JavaScript Simple Object Notation,  
*Patrz: JSON*
- JSON, 120
- parsowanie, 119
- K**
- klasa
- CSS, 28, 37, 64, 107
  - motywów, 27
  - ui-nojs, 27
  - ui-page, 27
  - zmiany rozdzielczości, 106
- klasa-fabryka `jQuery.widget`, 39
- kolor, 81, 91
- kontrolka, 27
- L**
- licznik, 52, 72
- balonik, 50, 95
- lista, 40, 52, 121, 122
- aktualizowanie, 53
  - dodawanie elementów, 54
  - motyw, 95
  - nieponumerowana, 40
  - obrazki, 47
  - pionowa, 60
  - ponumerowana, 40
- separator, 42, 52, 95
- wyświetlanie, 127
- zagnieżdżona, 44
- złożona z przycisków, 41, 46
- Ł**
- łącze, 19, 34
- M**
- menu
- rozwijane, 72
  - select, 71, 72
- metoda, 39
- `addResolutionBreakpoints`, 106
  - `changePage`, 34, 97
  - `jQuery(document).ready`, 29
  - `jQuery.ajax`, 120
  - `jQuery.bind`, 30
  - `jQuery.data`, 119
  - `jQuery.delegate`, 30
  - `jQuery.live`, 30
  - `JSON.parse`, 120
  - `JSON.stringify`, 120
  - `localStorage`, 118
  - media, 107
  - `pageLoading`, 98
  - refresh, 53
  - `sessionStorage`, 118
  - `silentScroll`, 99
  - slider, 76
- miniaturka, 47
- motyw, 81, 91, 95
- listy, 95
  - przycisku dzielonego, 95
- multimedia, 107
- N**
- nagłówek, 15, 54, 57, 72
- sekcji, 42
- nawigacja, 15, 21

## O

obszar tekstowy, 69  
okno dialogowe, 32, 98  
orientacja, 105, *Patrz:* urządzenie orientacja

## P

parsowanie, 119  
pasek  
  narzędzi, 27, 54  
  nawigacji, 54, 56  
platform abstraction layer,  
  *Patrz:* warstwa  
pole  
  tekstowe, 69, 75  
  wyboru, 65, 66, 72  
  wyszukiwania, 69, 70  
próbka, 81, 82, 87, 90  
  personalizacja, 91  
przejście, 14, 19, 21, 29, 34, 98  
  animowane, 35, 36, 64  
przełącznik, 61, 65, 66  
  dwustanowy, 67  
przestrzeń nazw, 20, 30, 111  
przewijanie, 104  
przycisk, 27, 34, 54, 65  
  dzielony, 46, 95  
  grupa, 54, 60  
  ikona, 62  
  stylizowany, 59  
  zamknięcia, 72

## R

rola, 16, 19  
rozdzielacz, *Patrz:* tekst-rozdzielacz  
rozdzielczość, 106

## S

Safari, 17, 29  
SDK, 17  
sekcja, 15

selektor  
  CSS, 105  
  filtrowanie, 20  
  jqmData, 20  
  niestandardowy, 20

siatka

  układu, 77

Software Development Kit, *Patrz:* SDK

stopka, 15, 54, 57

strona, 113

  inicjalizacja, 39, 102

  inicjowanie, 27, 123

  kod, 28, 64

  ładowanie, 29, 98

  pobieranie, 27, 29

  przekazywanie danych, 118

  rozmiar, 105

  ukrywanie, 30

  ustawień, 123

  wewnętrzna, 22

  widok, 29

  widżet, 27

  wyświetlanie, 30

  zewnątrzna, 24

styl użytkownika, 72

suwak, 75

system motywów biblioteki jQuery

  Mobile, *Patrz:* biblioteka jQuery Mobile  
  system motywów

szkło powiększające, *Patrz:* ikona  
  wyszukiwania

## T

tablica, 98

  wyników, 121

tekst-rozdzielacz, 72

tekstura, 81

theming class, *Patrz:* klasa motywów

tło, 105

transformacja

  CSS3, 14, 35, 36

  webkit, 36

## U

- urządzenie
  - mobilne, 7, 104
  - orientacja, 105, 107
  - przenośne, 17

## V

- viewport, *Patrz:* wziernik

## W

- warstwa abstrakcji platformy, 7
- webkit, 37
- widżet, 13, 27, 39, 65
  - checkboxradio, 66
  - listview, 53
  - select, 71
  - selectmenu, 74
  - slider, 68
- wtyczka, 13, 39
  - jQuery, 32, 118
- wymiar, 81
- wypełnienie, 81
- wziernik, 99

## X

- XAMPP, 18

## Z

- zdarzenie, 99
  - beforecreate, 27
  - click, 122
  - document ready, 118
  - ładowania strony, 29

- mobileinit, 110
- nasłuchiwanie, 30, 124
- orientationchange, 104, 107
- pagebeforecreate, 102
- pagebeforehide, 29, 103
- pagebeforeshow, 29, 103
- pagecreate, 102
- pagehide, 29, 32, 103
- pageshow, 30, 32, 104
- scroll, 104
- scrollstart, 104
- scrollstop, 104
- swipe, 99
- swipeleft, 100
- swiperight, 100
- tap, 99
- taphold, 99
- ukrywania stron, 30
- wiązanie, 30
- wyświetlania stron, 30
- związane z dotykiem, 99
- związane z przewijaniem dokumentu, 104
- związane ze zmianą orientacji urządzenia, 104
- zmienna globalna, 119
- znacznik
  - div, 15, 28
  - form, 64
  - grupowanie, 73
  - HTML, 14
  - kotwicy, 49, 52, 54, 56, 59
  - optgroup, 73
  - option, 72, 73
  - semantyczny, 15, 19
  - span, 28
  - zakotwiczenia, 46
  - zestaw, 16

# PROGRAM PARTNERSKI

GRUPY WYDAWNICZEJ HELION



- 1. ZAREJESTRUJ SIĘ**
- 2. PREZENTUJ KSIĄŻKI**
- 3. ZBIERAJ PROWIZJĘ**

Zmień swoją stronę WWW  
w działający bankomat!

**Dowiedz się więcej i dołącz już dzisiaj!**

<http://program-partnerski.helion.pl>

GRUPA WYDAWNICZA

 **Helion SA**

# jQuery Mobile



Mimo że na rynku wciąż wiodą prym tylko dwa systemy operacyjne dla urządzeń mobilnych, tworzenie aplikacji natywnych przeznaczonych wyłącznie dla nich jest zbyt czasochłonne. Przyszłość należy do tych aplikacji, które będą kompatybilne z wieloma platformami sprzętowymi. Jednak lista problemów, z jakimi musi się zmierzyć projektant takiego oprogramowania, jest długa. Czy można na to coś poradzić? Tak! Rozwiązaniem są bogate możliwości HTML5 i CSS3! Jeżeli dolożyć do tego genialną bibliotekę jQuery Mobile, otrzymamy wybitne trio, pozwalające stworzyć aplikacje, które będą spójnie wyglądać i działać na wielu różnych urządzeniach.

Dzięki tej książce poznasz możliwości jQuery Mobile i przekonasz się, że tworzenie aplikacji na urządzenia mobilne nie musi być trudne. Dowiesz się, jak przygotować strukturę strony oraz jak wykorzystać dostępne elementy interfejsu użytkownika. Nauczysz się nawigować pomiędzy stronami oraz korzystać z okien dialogowych i formularzy. Znajdziesz tu też informacje na temat mechanizmu zdarzeń w jQuery Mobile oraz poznasz podstawy tworzenia własnych motywów graficznych. Książka ta jest idealną pozycją dla wszystkich osób chcących zaprojektować swoją pierwszą aplikację dla urządzeń mobilnych.

jQuery Mobile pozwala na:

- błyskawiczne tworzenie aplikacji mobilnych
- korzystanie z rozbudowanego mechanizmu zdarzeń
- łatwą nawigację pomiędzy stronami
- tworzenie własnych motywów graficznych

**Wykorzystaj całą zdobytą wiedzę do stworzenia aplikacji mobilnej od początku do końca!**

**helion.pl**  
księgarnia  
internetowa

Nr katalogowy: **11836**



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](mailto:helion@helion.pl)

<http://helion.pl>



KOD KORZYŚCI

ISBN 978-83-246-4772-9



Cena 29,90 zł