Baza programisty

Pragmatic Bookshelf

HTML5 i CSS3

Standardy przyszłości

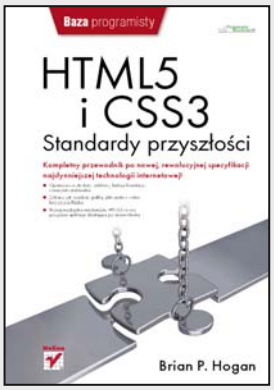
Kompletny przewodnik po nowej, rewolucyjnej specyfikacji najsłynniejszej technologii internetowej!

- Opanuj nowe atrybuty, selektory, funkcje formularzy i znaczniki strukturalne
- Zobacz, jak osadzać grafikę, pliki audio i wideo bez użycia Flasha
- Poznaj niezbędne mechanizmy HTML5 i twórz przyjazne aplikacje działające po stronie klienta

Helion

Brian P. Hogan

HTML5 i CSS3. Standardy przyszłości



Autor: Brian P. Hogan
Tłumaczenie: Daniel Kaczmarek
ISBN: 978-83-246-3028-8
Tytuł oryginału: HTML5 and CSS3: Develop with Tomorrow's Standards Today
Format: 158×235, stron: 304

Kompletny przewodnik po nowej, rewolucyjnej specyfikacji najsłynniejszej technologii internetowej!

- Opanuj nowe atrybuty, selektory, funkcje formularzy i znaczniki strukturalne
- Zobacz, jak osadzać grafikę, pliki audio i wideo bez użycia Flasha
- Poznaj niezbędne mechanizmy HTML5 i twórz przyjazne aplikacje, działające po stronie klienta

Stało się! Choć jeszcze niedawno HTML5 i CSS3 wydawały się technologiami wciąż odległej przyszłości, ich specyfikacje już zostały zaimplementowane w takich przeglądarkach, jak Google Chrome, Safari, Firefox czy Opera. Standardy HTML5 i CSS3, mimo że jeszcze w stadium rozwoju, nie przestają wzbudzać zachwytu administratorów sieci i twórców stron WWW, a ich opublikowanie hucznie zapowiedziało nową generację aplikacji internetowych. Teraz można je łatwiej wdrażać i utrzymywać, a także wyjść naprzeciw potrzebom użytkowników. Mało? Język HTML5 wprowadza także nowe elementy służące do definiowania struktury witryny oraz osadzania w niej treści, a CSS3 udostępnia zaawansowane selektory, rozszerzenia graficzne oraz zapewnia lepszą obsługę czcionek. W świecie internetu, gdzie pod względem rozprzestrzeniania się nowości trzy miesiące to cała epoka, już jutro ta technologia będzie obecna wszędzie. Nie trzeba więc nikogo przekonywać, że aby ten postęp bez zadyszki dogonić, trzeba już dziś… wziąć się za lekturę!

Przed Tobą doskonały podręcznik, w którym przedstawione zostały wszystkie dostępne sposoby korzystania z nowych możliwości standardów HTML5 i CSS3, także tych, które nie są jeszcze obsługiwane przez część przeglądarek. Każdy rozdział skupia się na określonej grupie problemów i zawiera listę niezbędnych znaczników, funkcji lub mechanizmów. Na początek dokładnie poznasz standardy HTML5 i CSS3, nowe atrybuty i znaczniki strukturalne. Przeczytasz co nieco na temat formularzy i odkryjesz, jak wykorzystać niektóre nowe pola i funkcje. Poznasz rewolucje wprowadzone w CSS3 – nowe selektory, cienie, gradienty, transformacje i sposoby korzystania z czcionek. Dowiesz się, jak HTML5 obsługuje dane audio i wideo oraz nauczysz się używać kanw do rysowania rozmaitych kształtów. A potem stworzysz przykładowe aplikacje działające po stronie klienta, a także zobaczysz, jak HTML5 umożliwia przesyłanie komunikatów i danych między domenami.

- Wprowadzenie do HTML5 i CSS3
- Nowe atrybuty i znaczniki strukturalne
- Tworzenie przyjaznych formularzy internetowych
- Tworzenie lepszych interfejsów użytkownika z użyciem CSS3
- Zwiększanie dostępności witryn internetowych
- Rysowanie na kanwach
- Osadzanie danych audio, wideo oraz grafiki wektorowej
- Wykorzystanie cieni, gradientów, transformacji i czcionek
- Przetwarzanie danych po stronie klienta
- Korzystanie z interfejsów API

Jesteś gotowy? Doskonale! Zacznij już dziś poznawać standardy przyszłości!

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

Spis treści

Rozdział 3.
Tworzenie przyjaznych formularzy internetowych

Jeżeli kiedykolwiek tworzyłeś skomplikowany interfejs użytkownika, na pewno wiesz, jak bardzo ograniczone jest działanie podstawowych kontrolek formularzy HTML. Programista jest ograniczony jedynie do pól tekstowych, menu wyboru, przycisków opcji, pól wyboru i czasami korzysta też z jeszcze bardziej prymitywnych **list wielokrotnego wyboru**, których sposób wykorzystania trzeba użytkownikom nieustannie tłumaczyć („Naciśnij klawisz *Ctrl* i klikaj myszą pozycje, które chcesz zaznaczyć, a na komputerze Mac zamiast klawisza *Ctrl* trzymaj wciśnięty klawisz *Cmd*").

W takiej sytuacji robi się to, co robią wszyscy dobrzy programiści — korzysta się z biblioteki Prototype albo jQuery bądź też implementuje się własne kontrolki i funkcje przy użyciu HTML, CSS i JavaScript. Jednak gdy spojrzy się potem na formularz, który składa się z suwaków, kontrolek kalendarza, pokręteł, pól uzupełnianych automatycznie i wizualnych edytorów, szybko dochodzi się do wniosku, że jego utrzymanie będzie koszmarem. Trzeba będzie bowiem zapewnić, że kontrolki umieszczane na stronie nie będą popadać w konflikt z innymi kontrolkami na niej obecnymi

ani z żadną biblioteką kodu JavaScript używaną na stronie. Można spędzić długie godziny na implementowaniu kontrolki kalendarza, a na koniec dowiedzieć się, że w bibliotece Prototype występuje teraz problem wynikający z przejęcia funkcji `$()` przez bibliotekę jQuery. Pierwszym pomysłem jest wtedy użycie metody `noConflict()`, lecz od razu okazuje się z kolei, że kontrolka wyboru koloru przestała działać, ponieważ obsługująca ją wtyczka nie została prawidłowo zaimplementowana.

Jeśli się teraz uśmiechasz, to znak, że już przez to przechodziłeś. Jeśli się wściekasz, przyczyna jest pewnie ta sama. Jest jednak dla nas nadzieja. W tym rozdziale utworzymy kilka formularzy internetowych, które będą zawierać nowe pola formularzy. Zaimplementujemy także mechanizm autofokusu i tekst zastępczy. Na koniec natomiast pokażę, jak używa się nowego atrybutu `contenteditable`, aby za jego pomocą zmienić dowolne pole HTML w kontrolkę na dane wejściowe.

W niniejszym rozdziale opisane zostaną następujące rozwiązania[1]:

Pole adresu poczty elektronicznej [`<input type="email">`]
: Wyświetla pole formularza do wpisywania adresów poczty elektronicznej. [O10.1, IOS]

Pole adresu URL [`<input type="url">`]
: Wyświetla pole formularza do wpisywania adresów URL. [O10.1, IOS]

Pole numeru telefonu [`<input type="tel">`]
: Wyświetla pole formularza do wpisywania numerów telefonów. [O10.1, IOS]

Pole wyszukiwania [`<input type="search">`]
: Wyświetla pole formularza do wpisywania poszukiwanych słów kluczowych. [C5, S4, O10.1, IOS]

Suwak [`<input type="range">`]
: Wyświetla kontrolkę suwaka. [C5, S4, O10.1]

[1] W opisach przedstawionych poniżej wskazano zakres obsługi poszczególnych elementów przy użyciu skrótów umieszczonych w nawiasach kwadratowych. Poszczególne skróty oznaczają: C — Google Chrome; F — Firefox; IE — Internet Explorer; O — Opera; S — Safari; IOS — urządzenia iOS z przeglądarką Mobile Safari; A — przeglądarka Android.

Liczba [<input type="number">]
: Wyświetla pole formularza przeznaczone do wpisywania liczb, często w postaci kontrolki pokrętła. [C5, S5, O10.1, IOS]

Pole daty [<input type="date">]
: Wyświetla pole formularza do wpisywania dat. Obsługuje następujące typy: `date`, `month` oraz `week`. [C5, S5, O10.1]

Daty wraz z godzinami
: Wyświetla pole formularza do wpisywania dat wraz z godzinami. Obsługuje następujące typy: `datetime`, `datetime-local`, `time`. [C5, S5, O10.1]

Kolor [<input type="color">]
: Wyświetla pole formularza do wskazywania kolorów. [C5, S5] (Chrome 5 i Safari 5 rozpoznają pole color, lecz nie wyświetlają żadnego konkretnego koloru).

Obsługa autofokusu [<input type="text" autofocus>]
: Obsługa mechanizmu ustawiania fokusu na konkretnym elemencie formularza. [C5, S4]

Obsługa tekstów zastępczych [<input type="email" placeholder="ja@przyklad.com">]
: Obsługa mechanizmu wyświetlania tekstu zastępczego w polu formularza. [C5, S4, F4]

Obsługa edycji wewnątrz strony [<p contenteditable>lorem ipsum</p>]
: Obsługa mechanizmu edytowania danych z poziomu przeglądarki. [C4, S3.2, IE6, O10.1]

Najpierw poznamy najbardziej przydatne typy pól formularza.

Porada 3. Opisywanie danych za pomocą nowych pól wejściowych

W HTML5 wprowadzono kilka nowych typów pól wejściowych, za pomocą których można bardziej precyzyjnie opisać dane wejściowe wpisywane przez użytkowników. Oprócz standardowych pól tekstowych, przycisków opcji i pól wyboru można definiować nowe elementy, takie jak pola na adresy poczty elektronicznej, kalendarze, kontrolki wyboru koloru, pokrętła i suwaki. Na podstawie tak zdefiniowanych pól przeglądarki mogą udostępniać użytkownikom lepsze kontrolki bez konieczności wykonywania kodu języka JavaScript. W urządzeniach przenośnych oraz w wirtualnych klawiaturach dla tabletów i ekranów dotykowych na podstawie typów pól można wyświetlać klawiatury w różnych układach. Na przykład przeglądarka Mobile Safari obecna w iPhonie odpowiednio dostosowuje układ klawiatury, gdy użytkownik wpisuje dane w polach typu `URL` i `email`. W wyświetlanym wówczas układzie klawiatury łatwiej dostępne stają się klawisze takie jak @, ., : i /.

Ulepszenie formularza na stronie firmy Pełen Wypas S.A.

Firma Pełen Wypas S.A. opracowuje nową aplikację internetową do zarządzania projektami. Witryna ma za zadanie ułatwić programistom i menedżerom monitorowanie postępów prac przy licznych projektach, które są realizowane w firmie. Dla każdego projektu definiuje się nazwę, adres poczty elektronicznej osoby kontaktowej oraz roboczy adres URL strony, na której można śledzić postęp w rozwoju witryny. Dodatkowo na stronie znajdują się pola na datę początku projektu, jego priorytet i szacowaną liczbę godzin roboczych, które trzeba poświęcić na prace projektowe. Menedżer projektu powinien mieć również możliwość przypisywania projektowi określonego koloru, który pozwoli mu szybko identyfikować poszczególne projekty na raportach.

W następnym punkcie opracujemy szablon strony do definiowania preferencji użytkownika. Strona będzie zawierać nowe pola HTML5.

Projekt prostego formularza

Najpierw utworzymy prosty formularz HTML, który będzie przesyłany metodą `POST`. Ponieważ pole, w którym wpisuje się nazwę projektu, nie musi posiadać żadnych szczególnych cech, stworzymy je na podstawie standardowego pola `text`.

html5_formularze/index.html

```
<form method="post" action="/projects/1">

  <fieldset id="personal_information">
    <legend>Informacje na temat projektu</legend>
    <ol>
      <li>
        <label for="name">Nazwa</label>
        <input type="text" name="name" autofocus id="name">
      </li>
      <li>
        <input type="submit" value="Wyślij">

      </li>
    </ol>

  </fieldset>

</form>
```

Zwróćmy uwagę, że kontrolki formularza są opisywane przy użyciu etykiet wchodzących w skład listy uporządkowanej. Etykiety mają kluczowe znaczenie, gdy tworzy się przyjazne formularze. Atrybut `for` etykiety odwołuje się do identyfikatora `id` powiązanego z nią elementu formularza. Dzięki temu rozwiązaniu programy odczytujące zawartość ekranu mogą zidentyfikować pola obecne na stronie. Lista uporządkowana pozwala ułożyć pola bez konieczności definiowania skomplikowanej tabeli ani struktur `div`. Jednocześnie za pomocą listy można wyznaczyć kolejność, w której użytkownicy powinni wypełniać kolejne pola formularza.

Tworzenie suwaka dla zakresu wartości

Suwaki wykorzystuje się po to, by umożliwić użytkownikom zmniejszanie lub zwiększanie wartości liczbowej. Za pomocą tej kontrolki umożliwimy menedżerom szybkie i proste wizualizowanie i zmienianie priorytetu dla projektu. Suwak implementuje się z zastosowaniem typu `range`.

html5_formularze/index.html

```
<label for="priority">Priorytet</label>
<input type="range" min="0" max="10"
       name="priority" value="0" id="priority">
```

Element suwaka należy dodać do formularza wewnątrz elementu `li`, tak samo jak poprzednio zdefiniowane pole nazwy projektu.

Zarówno Chrome, jak i Opera obsługują kontrolkę suwaka, która prezentuje się w sposób widoczny na rysunku 3.1.

Rysunek 3.1. Kontrolka suwaka wyświetlona w przeglądarce Chrome

Jak widać w definicji suwaka, zdefiniowano dla niego również wartości `min` i `max` reprezentowanego zakresu. W ten sposób ograniczyliśmy zakres wartości, jakie można wybrać przy użyciu kontrolki.

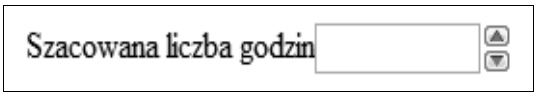
Obsługa liczb za pomocą pola pokrętła

W formularzach często trzeba podawać liczby. Wprawdzie wpisywanie liczb nie jest niczym trudnym, jednak czynność tę można jeszcze nieco ułatwić. Pole pokrętła to kontrolka z przyciskami strzałek, które zwiększają lub zmniejszają wartość widoczną w polu tekstowym. Pola pokrętła użyjemy do wpisywania szacowanej liczby godzin do przepracowania przy projekcie. W ten sposób łatwo będzie zmieniać liczbę godzin.

html5_formularze/index.html

```
<label for="estimated_hours">Szacowana liczba godzin</label>
<input type="number" name="estimated_hours"
       min="0" max="1000"
       id="estimated_hours">
```

Pole pokrętła jest obsługiwane przez przeglądarki Opera i Chrome. Wygląd pola pokrętła w przeglądarce Opera przedstawiono na rysunku 3.2.



Rysunek 3.2. Pole pokrętła wyświetlone w przeglądarce Opera

W polu pokrętła domyślnie można także wpisywać liczby z klawiatury. Ponadto podobnie jak dla suwaków, dla pola pokrętła można definiować wartości minimalną i maksymalną. Jednak wartość minimalna i maksymalna nie będzie brana pod uwagę, jeżeli wartość zostanie wpisana w polu z klawiatury.

Warto zaznaczyć, że za pomocą parametru `step` można zdefiniować postęp wartości w polu — wartość parametru wskazuje, o ile ma się jednorazowo zmienić wartość obecna w polu. Domyślnie postęp ma wartość `1`, lecz można mu przypisać dowolną inną wartość liczbową.

Daty

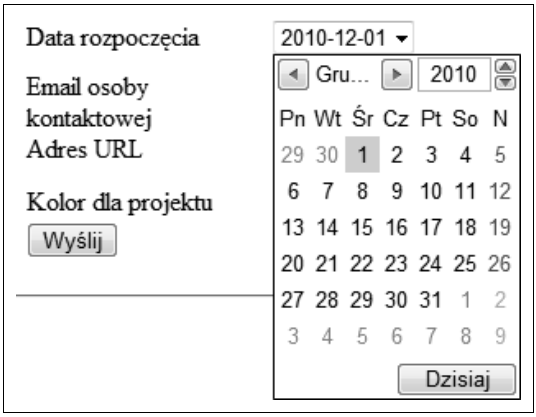
Istotną sprawą jest wskazanie daty rozpoczęcia projektu. Chcemy, by było to jak najprostsze. Doskonale do tego celu nadaje się pole typu `date`.

html5_formularze/index.html

```
<label for="start_date">Data rozpoczęcia</label>
<input type="date" name="start_date" id="start_date"
       value="2010-12-01">
```

W trakcie powstawania tej książki jedyną przeglądarką, która w pełni obsługiwała kontrolkę kalendarza, była przeglądarka Opera.

Na rysunku 3.3 przedstawiono przykładową implementację kontrolki kalendarza w przeglądarce Opera.



Rysunek 3.3. Kontrolka kalendarza wyświetlona w przeglądarce Opera

Przeglądarka Safari 5.0 wyświetla zamiast kontrolki kalendarza pole podobne do pola typu `number`, z przyciskami ze strzałkami, za pomocą których można ustawiać datę. Jeżeli pole jest puste, przyjmowana jest domyślna wartość „1582". Pozostałe przeglądarki zamiast kontrolki kalendarza wyświetlają zwykłe pole tekstowe.

Adres poczty elektronicznej

Specyfikacja HTML5 stanowi, że pole wejściowe typu `email` jest przeznaczone do wpisywania jednego adresu poczty elektronicznej lub listy takich adresów. Pole to posłuży nam zatem do wpisywania w formularzu adresu poczty elektronicznej osoby kontaktowej.

html5_formularze/index.html

```
<label for="email">Email osoby kontaktowej</label>
<input type="email" name="email" id="email">
```

Największą korzyść z zastosowania pola typu `email` uzyskuje się na urządzeniach przenośnych, w których układ wirtualnej klawiatury zmienia się tak, aby łatwiej dostępne były klawisze ze znakami niezbędnymi do wpisania adresu poczty elektronicznej.

Adres URL

Dostępne jest także pole specjalnie przystosowane do wpisywania adresów URL. Sprawdza się ono zwłaszcza wówczas, gdy użytkownik wywołujący stronę korzysta z iPhone'a, ponieważ urządzenie zmienia wówczas układ klawiatury na taki, w którym znajdują się przyciski ułatwiające szybkie wpisywanie adresów internetowych. Jest to rozwiązanie podobne do klawiatury wyświetlanej w trakcie wpisywania adresu URL w pasku adresów w przeglądarce Mobile Safari. W celu umieszczenia na formularzu pola na adres URL wystarczy użyć następującej definicji:

html5_formularze/index.html

```
<label for="url">Adres URL</label>
<input type="url" name="url" id="url">
```

Również wirtualne klawiatury zmieniają swój układ zależnie od typu pola wejściowego.

Kolory

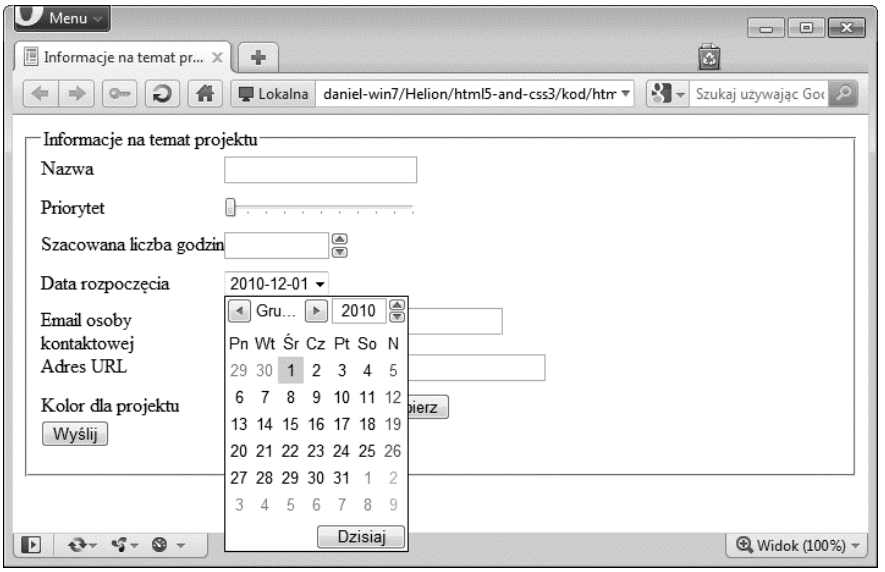
W ostatnim polu formularza należy podać kolor dla projektu. Posłuży do tego pole z typem `color`.

html5_formularze/index.html

```
<label for="project_color">Kolor dla projektu</label>
<input type="color" name="project_color" id="project_color">
```

W trakcie powstawania książki żadna przeglądarka nie obsługiwała pola wejściowego typu `color`, jednak nie jest to powód, by w ogóle z niego rezygnować. Odpowiedni znacznik precyzyjnie opisuje przeznaczenie pola, co przyda się w przyszłości, zwłaszcza gdy zajdzie potrzeba opracowania rozwiązania zastępczego.

Obecnie większość z przedstawionych nowych kontrolek jest obsługiwana przez przeglądarkę Opera, co widać na rysunku 3.4. Jednak gdy tę samą stronę otworzy się w przeglądarce Firefox, Safari albo Google Chrome, będzie widać tylko niewielkie różnice. I te właśnie różnice trzeba będzie wyeliminować.



Rysunek 3.4. Niektóre kontrolki formularza są już obsługiwane przez przeglądarkę Opera

Rozwiązanie zastępcze

Przeglądarki, które nie potrafią rozpoznać pól wejściowych nowych typów, prezentują je jako pola typu `text`. Formularz będzie więc nadal zdatny do użycia. Na chwilę obecną do takich pól można podpinać widżety interfejsu użytkownika z biblioteki jQuery UI albo YUI, które odpowiednio takie pole przekształcą. W miarę upływu czasu coraz więcej przeglądarek będzie obsługiwać pola nowych typów, więc będzie można stopniowo usuwać zastępcze rozwiązania oparte na kodzie JavaScript.

Zastąpienie kontrolki do wskazywania koloru

Kontrolkę do wskazywania koloru można bez trudu zastąpić przy użyciu biblioteki jQuery oraz selektorów atrybutów obecnych w CSS3. Wystarczy w tym celu zlokalizować pole `input` typu `color` i przypisać mu plugin biblioteki jQuery o nazwie SimpleColor.

html5_formularze/index.html

```
if (!hasColorSupport()){
  $('input[type=color]').simpleColor();
}
```

Ponieważ w kodzie znaczników użyto nowych typów dla pól formularzy, nie trzeba definiować dodatkowych nazw klas ani innych znaczników, które identyfikowałyby kontrolkę do wskazywania koloru. Selektory atrybutów i HTML5 w zupełności wystarczą.

Jeżeli przeglądarka obsługuje kontrolkę nowego typu, nie ma potrzeby korzystania z pluginu. Należy więc przy użyciu kodu JavaScript sprawdzić, czy przeglądarka obsługuje pola wejściowe z atrybutem `type` o wartości `color`.

html5_formularze/index.html

```
1  function hasColorSupport(){
2    input = document.createElement("input");
3    input.setAttribute("type", "color");
4    var hasColorType = (input.type !== "text");
5    // obsługa częściowej implementacji Safari/Chrome
6    if(hasColorType){
7      var testString = "foo";
8      input.value=testString;
9      hasColorType = (input.value != testString);
10   }
11   return(hasColorType);
12 }
```

Najpierw w kodzie JavaScript tworzony jest nowy element, a jego atrybutowi `type` przypisywana jest wartość `color`. Następnie odczytujemy atrybut `type`, aby sprawdzić, czy przeglądarka pozwoliła przypisać mu potrzebną nam wartość. Jeżeli odczytaną wartością atrybutu `type` jest `color`, oznacza to, że przeglądarka prawidłowo obsługuje kontrolkę do wskazywania koloru. Jeżeli wartość atrybutu jest inna, wówczas trzeba użyć pluginu.

Ciekawie przedstawia się zwłaszcza szósty wiersz przedstawionego kodu. Safari 5 i Google Chrome 5 częściowo obsługują już typ `color`. Mówiąc bardziej precyzyjnie, przeglądarki obsługują pola tego typu, lecz nie wyświetlają właściwej kontrolki do wskazywania koloru, a zamiast niej umieszczają na formularzu standardowe pole tekstowe. Dlatego w metodzie, która sprawdza, czy kontrolka jest obsługiwana, przypisujemy wartość polu wejściowemu, a następnie sprawdzamy, czy wartość ta zostaje zapamiętana. Jeżeli wartość nie zostanie zapamiętana, jest to znak, że przeglądarka w pełni obsługuje kontrolkę do wskazywania koloru, ponieważ pole wejściowe nie zachowuje się jak standardowe pole tekstowe.

Kod, który zastępuje kontrolkę do wskazywania koloru widżetem, przedstawia się następująco:

html5_formularze/index.html

```
if (!hasColorSupport()){
  $('input[type=color]').simpleColor();
}
```

Tak skonstruowane rozwiązanie działa, lecz jest dość ograniczone. Jest ono przeznaczone tylko dla niektórych przeglądarek i rozwiązuje problem jedynie z kontrolką do wskazywania koloru. Z innymi kontrolkami wiążą się inne problemy, o których warto wiedzieć. Na szczęście istnieje rozwiązanie alternatywne.

Modernizr

Biblioteka Modernizr[2] potrafi rozpoznawać, czy dostępna jest obsługa większości funkcji obecnych w HTML5 i CSS3. Jeżeli przeglądarka nie obsługuje danej funkcji, biblioteka jej w tym nie zastępuje, lecz udostępnia szereg rozwiązań podobnych do zaimplementowanego przez nas w poprzednim punkcie rozwiązania rozpoznającego pola formularza. Co więcej, rozwiązania dostępne w bibliotece Modernizr są znacznie bardziej niezawodne.

Zanim zaczniemy wykorzystywać Modernizr we własnych projektach, należy najpierw poświęcić trochę czasu na zrozumienie, jak biblioteka działa. Programista jest bowiem odpowiedzialny za kod użyty w projekcie bez względu na to, czy jest jego autorem, czy nie. Modernizr nie potrafi jeszcze rozpoznawać częściowej obsługi pola do wskazywania koloru, obecnej w przeglądarce Safari. Gdy udostępniona zostanie nowa wersja przeglądarki Chrome lub Firefox, być może trzeba będzie samodzielnie opracować nowe rozwiązanie. Kto wie, może Twoje rozwiązanie stanie się częścią biblioteki Modernizr?

Rozwiązania zastępcze dla kontrolek takich jak kontrolki kalendarza oraz suwaki implementuje się w podobny sposób. Suwaki i kontrolki kalendarza to komponenty dostępne w bibliotece jQuery UI[3]. Bibliotekę jQuery UI należy dołączyć do strony, sprawdzić, czy przeglądarka sama obsługuje wybraną kontrolkę, a jeżeli tej obsługi brakuje, użyć implementacji analogicznej kontrolki w języku JavaScript.

Za jakiś czas będzie można całkowicie zrezygnować z kontrolek JavaScript i korzystać wyłącznie z mechanizmów zaimplementowanych w przeglądarkach. Biblioteka Modernizr jest o tyle pomocna, że zdejmuje z programisty obowiązek tworzenia kodu rozpoznającego obsługę poszczególnych typów kontrolek, który zwykle jest dość skomplikowany. Tym niemniej

[2] *http://www.modernizr.com/*

[3] *http://jqueryui.com/*

w dalszej części książki nadal będziemy implementować kod rozpoznający obecność poszczególnych funkcji w przeglądarce, aby pokazać istotę jego działania.

Oprócz nowych typów pól formularza HTML5 definiuje kilka nowych atrybutów pól formularzy, które zwiększają ich użyteczność. W następnym punkcie omówię atrybut `autofocus`.

Porada 4. Przechodzenie do pierwszego pola formularza przy użyciu atrybutu autofocus

Czynność wypełniania formularza można znacznie przyspieszyć, gdy w momencie ładowania strony umieści się kursor w pierwszym polu formularza. W wielu wyszukiwarkach internetowych dokonuje się tego za pomocą odpowiedniego kodu JavaScript. Obecnie natomiast HTML5 udostępnia podobną funkcję jako element języka.

Całe rozwiązanie sprowadza się do tego, by do wybranego pola formularza dodać atrybut `autofocus` w taki sam sposób, jak zrobiliśmy to już na stronie profilu utworzonej w poradzie „Opisywanie danych za pomocą nowych pól wejściowych" na stronie 56.

html5_formularze/index.html

```
<label for="name">Nazwa</label>
<input type="text" name="name" autofocus id="name">
```

Aby atrybut `autofocus` działał poprawnie, powinien występować tylko jeden raz na stronie. Jeżeli atrybut wystąpi więcej niż jeden raz, przeglądarka umieści kursor w ostatnim polu z tym atrybutem.

Rozwiązanie zastępcze

Jeżeli przeglądarka użytkownika nie obsługuje atrybutu `autofocus`, można ten fakt rozpoznać za pomocą odpowiednio sformułowanego kodu JavaScript. Następnie należy użyć biblioteki jQuery i umieścić kursor w odpowiednim polu formularza. Jest to najprawdopodobniej najprostsze rozwiązanie zastępcze, które znajduje się w książce.

html5_formularze/autofocus.js

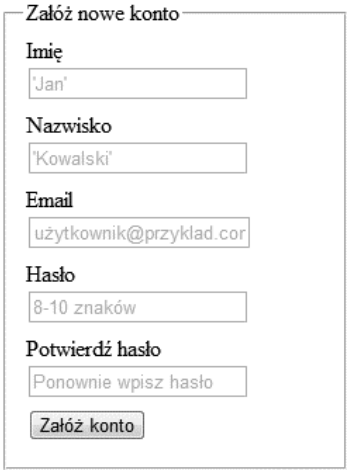
```
function hasAutofocus() {
  var element = document.createElement('input');
  return 'autofocus' in element;
}

$(function(){
   if(!hasAutofocus()){
     $('input[autofocus=true]').focus();
   }
});
```

Wystarczy umieścić zaprezentowany kod JavaScript na stronie, aby uzyskać odpowiednią obsługę atrybutu `autofocus`.

Autofokus ułatwia użytkownikom wypełnianie formularza. Jeszcze większym ułatwieniem będzie jednak udostępnienie informacji na temat tego, jakich konkretnie danych oczekujemy w poszczególnych polach formularza. Dlatego w następnym punkcie zajmiemy się atrybutem `placeholder`.

Porada 5. Wyświetlanie wskazówek w postaci tekstu zastępczego

Tekst zastępczy stanowi wskazówkę dla użytkownika, która mówi, jakie dane należy wpisywać w poszczególnych polach. Na rysunku 3.5 przedstawiono formularz do tworzenia nowego konta, w którym zastosowano tekst zastępczy. Formularz ten zdefiniujemy w następnym punkcie.



Rysunek 3.5. Teksty zastępcze wskazują użytkownikowi, jakie dane powinien wpisać w poszczególnych polach

Formularz zakładania nowego konta

Na stronie firmy Pełen Wypas S.A. użytkownik musi najpierw założyć nowe konto. Poważnym problemem, który zawsze pojawia się w takim przypadku, jest to, że użytkownicy podają hasła łatwe do odgadnięcia. Dlatego użyjemy tekstu zastępczego, aby wskazać użytkownikowi wymagania co do postaci hasła. Dla zachowania spójności tekst zastępczy umieścimy także w pozostałych polach tekstowych.

Aby umieścić w polu formularza tekst zastępczy, wystarczy do każdego takiego pola dodać atrybut `placeholder` w następujący sposób:

html5_tekstzastepczy/index.html

```
<input id="email" type="email"
       name="email" placeholder="użytkownik@przyklad.com">
```

Kod znaczników całego formularza przedstawiono na poniższym listingu. W każdym polu formularza umieszczono tekst zastępczy.

html5_tekstzastepczy/index.html

```
<form id="create_account" action="/signup" method="post">
  <fieldset id="signup">
    <legend>Załóż nowe konto</legend>
    <ol>
      <li>
        <label for="first_name">Imię</label>
        <input id="first_name" type="text"
               autofocus="true"
               name="first_name" placeholder="'Jan'">
      </li>
      <li>
        <label for="last_name">Nazwisko</label>
        <input id="last_name" type="text"
               name="last_name" placeholder="'Kowalski'">
      </li>
      <li>
        <label for="email">Email</label>
        <input id="email" type="email"
               name="email" placeholder="użytkownik@przyklad.com">
      </li>
      <li>
        <label for="password">Hasło</label>
        <input id="password" type="password" name="password" value=""
               autocomplete="off" placeholder="8-10 znaków" />
      </li>
      <li>
        <label for="password_confirmation">Potwierdź hasło</label>
        <input id="password_confirmation" type="password"
               name="password_confirmation" value=""
               autocomplete="off" placeholder="Ponownie wpisz hasło" />
      </li>
      <li><input type="submit" value="Załóż konto"></li>
    </ol>
  </fieldset>
</form>
```

Wyłączanie automatycznego uzupełniania

Nietrudno zauważyć, że do pól formularza, w których wpisuje się hasło, dodano atrybut `autocomplete`. Atrybut ten jest nowością w HTML5 i wskazuje on przeglądarkom, że nie powinny automatycznie wypełniać tego pola danymi. Niektóre przeglądarki pamiętają bowiem dane wpisane w przeszłości przez użytkownika, a w niektórych sytuacjach pożądane jest, by przeglądarki nie wstawiały ponownie tych samych danych.

Ponieważ również w tym formularzu pola są zdefiniowane w ramach listy uporządkowanej, można dodać kaskadowy arkusz stylów, aby nadać stronie bardziej przyjazny wygląd.

html5_tekstzastepczy/style.css

```
fieldset{
  width: 216px;
}

fieldset ol{
  list-style: none;
  padding:0;
  margin:2px;
}

fieldset ol li{
  margin:0 0 9px 0;
  padding:0;
}

/* Pola wejściowe mają znajdować się w oddzielnych wierszach */
fieldset input{
  display:block;
}
```

Dzięki takiemu rozwiązaniu użytkownicy przeglądarek Safari, Opera i Chrome będą widzieć wskazówki w postaci tekstu zastępczego widocznego w polach wejściowych. Pozostało nam tylko opracować rozwiązanie zastępcze, które da ten sam efekt w przeglądarkach Firefox i Internet Explorer.

Rozwiązanie zastępcze

Za pomocą krótkiego kodu języka JavaScript w polach formularzy można umieszczać teksty zastępcze. W tym celu należy sprawdzić zawartość każdego pola formularza, a jeżeli jest ono puste, trzeba przypisać mu tekst zastępczy. Gdy w polu formularza zostanie umieszczony kursor, tekst zastępczy się usuwa, a gdy kursor zostanie przeniesiony do innego pola, ponownie należy sprawdzić jego zawartość. Jeżeli zawartość pola jest inna niż tekst zastępczy, pozostawia się ją bez zmian. Jeżeli natomiast pole jest puste, wstawia się w nim tekst zastępczy.

Aby sprawdzić, czy przeglądarka obsługuje tekst zastępczy, należy wykorzystać rozwiązanie podobne do tego, za pomocą którego sprawdzaliśmy obsługę autofokusu.

html5_tekstzastepczy/index.html

```
function hasPlaceholderSupport() {
  var i = document.createElement('input');
  return 'placeholder' in i;
}
```

Teraz pozostaje już tylko napisać kod JavaScript, który będzie obsługiwał zmiany tekstu. Do tego celu wykorzystamy rozwiązanie, które opracował Andrew January[4] wraz z zespołem. Za pomocą skryptu wszystkie pola formularza będziemy wypełniać tekstem zastępczym przechowywanym w atrybucie `placeholder`. Gdy użytkownik umieści kursor w polu, umieszczony w nim tekst zostanie usunięty. Całość kodu zaimplementujemy w postaci pluginu biblioteki jQuery, aby łatwo było go wykorzystać w formularzu. Więcej informacji na temat sposobu działania pluginów znajduje się w ramce na stronie 72.

html5_tekstzastepczy/jquery.placeholder.js

```
(function($){

  $.fn.placeholder = function(){

    function valueIsPlaceholder(input){
      return ($(input).val() == $(input).attr("placeholder"));
    }
    return this.each(function() {

      $(this).find(":input").each(function(){

        if($(this).attr("type") == "password"){

          var new_field = $("<input type='text'>");
          new_field.attr("rel", $(this).attr("id"));
          new_field.attr("value", $(this).attr("placeholder"));
          $(this).parent().append(new_field);
          new_field.hide();

          function showPasswordPlaceHolder(input){
            if( $(input).val() == "" || valueIsPlaceholder(input) ){
              $(input).hide();
              $('input[rel=' + $(input).attr("id") + ']').show();
            };
          };
```

[4] Skrypt w oryginalnej postaci jest dostępny na stronie pod adresem *http://www.morethannothing.co.uk/wp-content/uploads/2010/01/placeholder.js*. Skrypt ten nie obsługuje jednak pól haseł w Internet Explorerze.

```
26
27            new_field.focus(function(){
28              $(this).hide();
29              $('input#' + $(this).attr("rel")).show().focus();
30            });
31
32            $(this).blur(function(){
33               showPasswordPlaceHolder(this, false);
34            });
35
36            showPasswordPlaceHolder(this);
37
38          }else{
39
40            // Zastąpienie wartości tekstem zastępczym.
41            // Opcjonalny parametr reload pozwala na zapisywanie wartości pól
42            // w pamięci podręcznej przez FF i IE.
43            function showPlaceholder(input, reload){
44              if( $(input).val() == "" ||
45                ( reload && valueIsPlaceholder(input) ) ){
46                  $(input).val($(input).attr("placeholder"));
47                }
48            };
49
50            $(this).focus(function(){
51              if($(this).val() == $(this).attr("placeholder")){
52                $(this).val("");
53              };
54            });
55
56            $(this).blur(function(){
57               showPlaceholder($(this), false)
58            });
59
60
61            showPlaceholder(this, true);
62          };
63        });
64
65        // Zapobiega wysłaniu formularza z wartościami domyślnymi
66        $(this).submit(function(){
67          $(this).find(":input").each(function(){
68            if($(this).val() == $(this).attr("placeholder")){
69              $(this).val("");
70            }
71          });
72        });
73
74      });
75    };
76
77  })(jQuery);
```

W przedstawionym kodzie źródłowym pluginu znajduje się kilka interesujących rozwiązań, na które warto zwrócić uwagę. W wierszu 45. tekst zastępczy zostaje umieszczony w polu, jeżeli nie ma ono żadnej wartości, ale również wówczas, gdy nastąpiło ponowne załadowanie strony. Firefox i inne przeglądarki zachowują wartości obecne w polach formularza. Skrypt przypisuje atrybutowi `value` pól teksty zastępcze, ponieważ nie chcemy, by pozostała w nim wartość wcześniej wpisana przez użytkownika. W momencie ładowania strony do metody `showPlaceholder()` przekazujemy wartość `true`, co widać w wierszu 61.

Pluginy jQuery

Bibliotekę jQuery można rozszerzać przez implementowanie własnych pluginów. Gdy zaimplementowaną samodzielnie metodę doda się do funkcji jQuery, plugin stanie się automatycznie dostępny dla innych programistów, którzy dołączą do swojego kodu tak rozszerzoną bibliotekę. Oto najprostszy przykład, który powoduje wyświetlenie okna komunikatu JavaScript:

```
jQuery.fn.debug = function() {
  return this.each(function(){
    alert(this.html());
  });
```

Aby wyświetlić okno dla każdego akapitu na stronie, wystarczy wywołać funkcję w następujący sposób:

```
$("p").debug();
```

Pluginy biblioteki jQuery są skonstruowane w taki sposób, że przechodzą przez kolekcję obiektów jQuery oraz zwracają taką kolekcję obiektów. Można więc z uzyskanych w ten sposób obiektów tworzyć łańcuch. Na przykład dzięki temu, że przedstawiony powyżej plugin o nazwie `debug` również zwraca kolekcję jQuery, można wykorzystać metodę `css` biblioteki i zmienić kolor tekstu wszystkich akapitów jednym wierszem kodu.

```
$("p").debug().css("color", "red");
```

W książce jeszcze co najmniej kilka razy skorzystamy z pluginów biblioteki jQuery, aby utrzymać odpowiednią organizację kodu źródłowego stanowiącego rozwiązanie zastępcze. Więcej informacji na temat pluginów jQuery można znaleźć na stronie z dokumentacją biblioteki*.

* *http://docs.jquery.com/Plugins/Authorin*

Pola przeznaczone na wpisywanie haseł zachowują się inaczej niż pozostałe pola formularza, dlatego też trzeba je inaczej potraktować. Spójrzmy na wiersz 12. Najpierw rozpoznajemy w nim obecność pola hasła. Potem trzeba zmienić jego typ na zwykłe pole tekstowe, aby jego wartość nie została ukryta pod postacią gwiazdek. Niektóre przeglądarki zwracają błąd, gdy próbuje się przekształcić pola haseł, dlatego musimy zastąpić pole hasła zwykłym polem tekstowym. Pole hasła i pole tekstowe będziemy zamieniać ze sobą nawzajem zależnie od czynności wykonywanych przez użytkownika.

Opracowane rozwiązanie zmienia wartości pól na formularzu, dlatego należy zapewnić, że teksty zastępcze nie zostaną przesłane do serwera jako dane formularza. Ponieważ kod do obsługi tekstów zastępczych jest wykonywany jedynie wówczas, gdy w przeglądarce włączona jest obsługa języka JavaScript, możemy przy użyciu tego języka wykryć próbę przesłania formularza na serwer i usunąć z niego wartości, które są identyczne jak teksty zastępcze. W wierszu 66. rozpoznawane jest zdarzenie zatwierdzenia formularza, po czym kod usuwa wartości z wszystkich pól wejściowych, w których znajduje się tekst zastępczy.

Dzięki temu, że nasze rozwiązanie zaimplementowaliśmy w postaci pluginu, możemy je teraz wywołać na stronie przez podłączenie pluginu do formularza w następujący sposób:

html5_tekstzastepczy/index.html

```
$(function(){
  function hasPlaceholderSupport() {
    var i = document.createElement('input');
    return 'placeholder' in i;
  }

  if(!hasPlaceholderSupport()){
    $("#create_account").placeholder();
    // KONIEC rozwiązania zastępczego

    $('input[autofocus=true]').focus();

  };
});
```

W ten sposób powstało całkiem eleganckie rozwiązanie, które pozwala umieszczać teksty zastępcze w formularzach aplikacji internetowych w dowolnej przeglądarce.

Porada 6. Edytowanie danych bezpośrednio na stronie przy użyciu atrybutu contenteditable

Zawsze warto szukać sposobów, dzięki którym użytkownikom łatwiej będzie korzystać z aplikacji. Czasami warto pozwolić użytkownikom na edytowanie danych na ich temat bez konieczności odsyłania ich do dedykowanego formularza. Standardowo mechanizm edycji danych bezpośrednio na stronie implementuje się w taki sposób, że najpierw rozpoznaje się kliknięcia myszą w regionie, w którym znajduje się tekst, a następnie w miejsce regionu wstawia się pole tekstowe. Zmieniona zawartość takiego pola tekstowego jest potem przesyłana za pośrednictwem technologii Ajax z powrotem do serwera. Natomiast obecny w HTML5 znacznik `contenteditable` automatycznie udostępnia możliwość wpisywania danych bezpośrednio na stronie. Również w przypadku zastosowania znacznika `contenteditable` trzeba zaimplementować kod JavaScript, który będzie przesyłał zmienione dane na serwer w celu ich zapisania, lecz nie trzeba już wówczas tworzyć ani przełączać ukrytych pól formularzy.

Celem jednego z projektów realizowanych w firmie Pełen Wypas S.A. jest umożliwienie użytkownikom przeglądania danych na temat ich konta. Na przeznaczonej do tego stronie WWW znajduje się nazwisko użytkownika, miasto, województwo, kod pocztowy oraz adres poczty elektronicznej. Dodamy więc możliwość edytowania tych danych bezpośrednio na stronie. W efekcie powinien powstać interfejs widoczny na rysunku 3.6.

Rysunek 3.6. Łatwe edytowanie danych bezpośrednio na stronie

Zanim przystąpimy do implementacji, chcę zaznaczyć, że implementowanie jakiegokolwiek mechanizmu opartego na kodzie JavaScript bez uprzedniego zaimplementowania kodu działającego na serwerze jest wbrew wszelkim znanym mi dobrym praktykom tworzenia aplikacji internetowych. Przyjąłem jednak taki tryb pracy tylko dlatego, że chcę skupić się na przedstawieniu działania znacznika `contenteditable`, oraz dlatego, że

nie będzie to kod na środowisko produkcyjne. Zawsze, i to naprawdę **zawsze**, najpierw należy zaimplementować tę część, która nie wymaga kodu JavaScript, a dopiero **potem** tworzyć wersję rozwiązania, która opiera się na skryptach JavaScript. Na koniec powinno się także zaimplementować automatyczne testy dla obydwóch wersji rozwiązania, aby zwiększyć prawdopodobieństwo zidentyfikowania wszystkich potencjalnych błędów, które mogą zajść zwłaszcza wówczas, gdy zmianie ulegnie tylko jedna z wersji.

Formularz edycji danych o koncie

W HTML5 udostępniono znacznik `contenteditable`, który jest dostępny niemal dla każdego elementu. Dodanie znacznika `contenteditable` spowoduje, że element przekształci się w pole dostępne do edycji.

html5_edytowaniedanych/show.html

```
<h1>Informacje o użytkowniku</h1>
<div id="status"></div>
<ul>
  <li>
    <b>Imię i nazwisko</b>
    <span id="name" contenteditable="true">Daniel Kaczmarek</span>
  </li>
  <li>
    <b>Miasto</b>
    <span id="city" contenteditable="true">Dowolne</span>
  </li>
  <li>
    <b>Województwo</b>
    <span id="state" contenteditable="true">Śląskie</span>
  </li>
  <li>
    <b>Kod pocztowy</b>
    <span id="postal_code" contenteditable="true">44-100</span>
  </li>
  <li>
    <b>Email</b>
    <span id="email" contenteditable="true">boss@pelenwypas.com</span>
  </li>
</ul>
```

Tak skonstruowany formularz można rozszerzyć o kaskadowy arkusz stylów. Do zidentyfikowania pól dostępnych do edycji wykorzystamy selektory CSS3. Pola te będą zmieniać kolor, gdy użytkownik umieści na nich kursor myszy albo je zaznaczy.

html5_edytowaniedanych/show.html

```
1 ul{list-style:none;}
2
3 li{clear:both;}
```

```
4
5  li>b, li>span{
6    display: block;
7    float: left;
8    width: 100px;
9  }
10
11 li>span{
12   width:500px;
13   margin-left: 20px;
14 }
15
16 li>span[contenteditable=true]:hover{
17   background-color: #ffc;
18 }
19
20 li>span[contenteditable=true]:focus{
21   background-color: #ffa;
22   border: 1px shaded #000;
23 }
```

To już wszystko, jeśli chodzi o część widoczną dla użytkownika. Użytkownicy mogą już teraz łatwo zmieniać dane bezpośrednio na stronie. Pozostaje je tylko zapisać na serwerze.

Zapisywanie danych

Użytkownicy mogą już zmieniać dane, jednak na razie zmiany te zostaną utracone, gdy nastąpi ponowne wywołanie strony bądź też gdy użytkownik przejdzie do innej strony internetowej. Trzeba zapewnić możliwość przesłania wprowadzonych zmian do serwera. Najprościej jest wykorzystać do tego celu bibliotekę jQuery. Jeżeli kiedykolwiek miałeś styczność z technologią Ajax, zastosowane rozwiązanie nie będzie dla Ciebie niczym nowym.

html5_edytowaniedanych/show.html

```
$(function(){
    var status = $("#status");
    $("span[contenteditable=true]").blur(function(){
      var field = $(this).attr("id");
      var value = $(this).text();
      $.post("http://localhost:4567/users/1",
        field + "=" + value,
        function(data){
          status.text(data);
        }
      );
    });

});
```

Do każdej sekcji `span`, która znajduje się na stronie i zawiera atrybut `contenteditable` o wartości `true`, dodany został odbiornik zdarzeń. Potem następuje już tylko przesłanie danych do skryptu działającego na serwerze.

Rozwiązanie zastępcze

W zastosowanym rozwiązaniu użyto szeregu rozwiązań, które w niektórych okolicznościach nie będą działać poprawnie. Przede wszystkim wysłanie zmienionych danych do serwera jest wykonywane przez kod JavaScript, co w zasadzie jest złym pomysłem. Poza tym użyliśmy pseudoklasy `focus`, aby wyróżnić pola, w których znajduje się kursor, a pseudoklasa ta nie jest obsługiwana przez niektóre wersje przeglądarki Internet Explorer. Zajmijmy się więc najpierw działaniem strony, a potem jej wyglądem.

Strona do edycji danych

Zamiast wymyślać różne okoliczności, w których użytkownik może nie mieć możliwości skorzystania z zaimplementowanego przez nas rozwiązania, najlepiej jest po prostu dać mu możliwość przejścia do oddzielnej strony z formularzem. Oczywiście wymaga to napisania dodatkowego kodu, jednak weźmy pod uwagę okoliczności, jakie mogą mieć miejsce:

- Użytkownik nie ma włączonej obsługi języka JavaScript i używa Internet Explorera 7.
- Użytkownik korzysta z przeglądarki, która nie obsługuje HTML5.
- Użytkownik korzysta z najnowszej wersji przeglądarki Firefox, która obsługuje HTML5, lecz nie ma włączonej obsługi języka JavaScript, bo po prostu tego języka nie lubi (to może się zdarzyć zawsze, zdecydowanie częściej, niż można by sobie wyobrażać).

Gdy zastanowimy się nad opisanymi okolicznościami, najbardziej sensownym rozwiązaniem zastępczym okaże się stworzenie formularza, który metodą `POST` wykona tę samą czynność, co zaimplementowany przez nas kod Ajax odpowiedzialny za zapisanie zmian. Sposób wykonania rozwiązania zastępczego zależy tylko od nas, jednak wiele dostępnych platform programistycznych umożliwia wykrywanie typu żądania przez sprawdzenie zawartości nagłówków `accept`. Na tej podstawie można się dowiedzieć, czy żądanie zostało wykonane zwykłą metodą `POST`, czy za pomocą `XMLHttpRequest`.

Kod pozostanie wówczas zgodny z zasadą DRY[5]. Łącze do formularza będzie ukrywane, jeżeli przeglądarka będzie obsługiwać znacznik `contenteditable` i język JavaScript.

Utworzymy więc nową stronę o nazwie *edit.html*, która będzie zawierać standardowy formularz edycji danych. Dane będą przesyłane do tego samego adresu, z którego korzysta zaimplementowany przez nas kod Ajax.

html5_edytowaniedanych/edit.html

```
<!DOCTYPE html>
<html lang="pl-PL">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>Edycja danych o koncie</title>
    <link href="style.css" rel="stylesheet" media="screen">
  </head>
  <body>
    <form action="/users/1" method="post" accept-charset="utf-8">
      <fieldset id="your_information">
        <legend>Informacje o użytkowniku</legend>
        <ol>
         <li>
           <label for="name">Imię i nazwisko</label>
           <input type="text" name="name" value="" id="name">
         </li>
         <li>
           <label for="city">Miasto</label>
           <input type="text" name="city" value="" id="city">
         </li>
         <li>
           <label for="state">Województwo</label>
           <input type="text" name="state" value="" id="state">
         </li>
         <li>
           <label for="postal_code">Kod pocztowy</label>
           <input type="text" name="postal_code" value="" id="postal_code">
         </li>
         <li>
           <label for="email">Email</label>
           <input type="email" name="email" value="" id="email">
         </li>
        </ol>

      </fieldset>

      <p><input type="submit" value="Zapisz"></p>
```

[5] DRY to skrót angielskich słów *Don't Repeat Yourself*, czyli „nie powtarzaj się". Określenie to zostało sformułowane przez Dave'a Thomasa i Andy'ego Hunta w książce *The Pragmatic Programmer*. [HT00]

```
    </form>

  </body>
</html>
```

Następnie na stronie *show.html* należy dodać łącze do nowej strony *edit.html*.

html5_edytowaniedanych/show.html

```
<h1>Informacje o użytkowniku</h1>
<section id="edit_profile_link">
  <p><a href="edit.html">Edycja danych o użytkowniku</a></p>
</section>
<div id="status"></div>
```

Po dodaniu łącza trzeba nieco zmodyfikować skrypt. Łącze do strony edycji danych ma być ukrywane, a kod Ajax wykonywany jedynie wówczas, gdy przeglądarka będzie obsługiwać znacznik `contenteditable`.

html5_edytowaniedanych/show.html

```
if(document.getElementById("edit_profile_link").contentEditable != null){
```

Skrypt rozszerzony o mechanizm wykrywania obsługi znacznika `contenteditable` ma następującą postać:

html5_edytowaniedanych/show.html

```
$(function(){
  if(document.getElementById("edit_profile_link").contentEditable != null){
    $("#edit_profile_link").hide();
    var status = $("#status");
    $("span[contenteditable=true]").blur(function(){
      var field = $(this).attr("id");
      var value = $(this).text();
      $.post("http://localhost:4567/users/1",
        field + "=" + value,
        function(data){
          status.text(data);
        }
      );
    });
  }

});
```

Dzięki takiej implementacji użytkownicy mogą korzystać ze standardowego interfejsu edycji danych bądź też z szybszego trybu edycji danych bezpośrednio na stronie. Wiesz już, w jaki sposób zaimplementować taki interfejs, pamiętaj więc, aby najpierw stworzyć rozwiązanie zastępcze. W odróżnieniu bowiem od rozwiązań zastępczych dla innych przypadków, brak rozwiązania zastępczego w tej sytuacji ograniczy zbiór funkcji dostępnych dla użytkownika.

Przyszłość

Jeśli obecnie na stronie dodana zostanie kontrolka kalendarza zaimplementowana w języku JavaScript, użytkownicy będą musieli poznać sposób jej działania. Jeżeli kiedyś kupowałeś w internecie bilety na samolot i rezerwowałeś pokój w hotelu, zdajesz sobie sprawę, jak różnorodne są sposoby implementowania formularzy na stronach WWW. Podobnie jest z bankomatami, których interfejsy różnią się między sobą czasem tak znacznie, że różnice te mogą spowalniać czynności korzystających z nich osób.

Wyobraźmy sobie jednak, że na każdej witrynie znajduje się pole daty zaimplementowane w HTML5, a odpowiedzialność za stworzenie odpowiedniego interfejsu spoczywa na przeglądarce. Na każdej witrynie odwiedzanej przez użytkowników kontrolka kalendarza miałaby identyczną postać. W oprogramowaniu odczytującym zawartość ekranu można by wręcz zaimplementować standardowy mechanizm, który ułatwiłby osobom niewidomym łatwe wpisywanie dat. Zwróćmy też uwagę, jak bardzo przydatne dla użytkowników staną się teksty zastępcze i autofokus, gdy będą powszechnie obsługiwane. Na podstawie tekstów zastępczych oprogramowanie odczytujące zawartość ekranu może wyjaśniać użytkownikom przeznaczenie poszczególnych pól formularzy, zaś dzięki mechanizmowi autofokusu łatwiej będzie takim osobom nawigować po formularzu bez użycia myszy. Będzie to znaczne ułatwienie zarówno dla osób niewidomych i niedowidzących, jak również dla tych użytkowników, którym ograniczenia ruchowe nie pozwalają na korzystanie z myszy lub je utrudniają.

Możliwość przekształcania dowolnego elementu w region dostępny do edycji znacznie ułatwia edytowanie danych bezpośrednio na stronie. Jednocześnie może to doprowadzić do zmiany sposobu tworzenia interfejsów dla systemów do zarządzania treścią.

Współczesny internet skupia się na interakcji, a formularze są tej interakcji kluczowym elementem. Rozszerzenia dostępne w HTML5 dają programistom cały zbiór nowych narzędzi, dzięki którym można znacznie ułatwić życie użytkownikom.

Skorowidz

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

Ź

HTML5 i CSS3
Standardy przyszłości

Stało się! Choć jeszcze niedawno HTML5 i CSS3 wydawały się technologiami wciąż odległej przyszłości, ich specyfikacje już zostały zaimplementowane w takich przeglądarkach, jak Google Chrome, Safari, Firefox czy Opera. Standardy HTML5 i CSS3, mimo że jeszcze w stadium rozwoju, nie przestają wzbudzać zachwytu administratorów sieci i twórców stron WWW, a ich opublikowanie wkrótce zapowiedziało nową generację aplikacji internetowych. Teraz można je łatwiej wdrażać i utrzymywać, a także wyjść naprzeciw potrzebom użytkowników. Mało? Język HTML5 wprowadza także nowe elementy służące do definiowania struktury witryny oraz osadzania w niej treści, a CSS3 udostępnia zaawansowane selektory, rozszerzenia graficzne oraz zapewnia lepszą obsługę czcionek. W świecie internetu, gdzie pod względem rozprzestrzeniania się nowości trzy miesiące to cała epoka, już jutro ta technologia będzie obecna wszędzie. Nie trzeba więc nikogo przekonywać, że aby ten postęp bez zadyszki dogonić, trzeba już dziś... wziąć się za lekturę!

Przed Tobą doskonały podręcznik, w którym przedstawione zostały wszystkie dostępne sposoby korzystania z nowych możliwości standardów HTML5 i CSS3, także tych, które nie są jeszcze obsługiwane przez część przeglądarek. Każdy rozdział skupia się na określonej grupie problemów i zawiera listę niezbędnych znaczników, funkcji lub mechanizmów. Na początek dokładnie poznasz standardy HTML5 i CSS3, nowe atrybuty i znaczniki strukturalne. Przeczytasz co nieco na temat formularzy i odkryjesz, jak wykorzystać niektóre nowe pola i funkcje. Poznasz rewolucje wprowadzone w CSS3 – nowe selektory, cienie, gradienty, transformacje i sposoby korzystania z czcionek. Dowiesz się, jak HTML5 obsługuje dane audio i wideo, oraz nauczysz się używać kanw do rysowania rozmaitych kształtów. A potem stworzysz przykładowe aplikacje działające po stronie klienta, a także zobaczysz, jak HTML5 umożliwia przesyłanie komunikatów i danych między domenami.

- Wprowadzenie do HTML5 i CSS3
- Nowe atrybuty i znaczniki strukturalne
- Tworzenie przyjaznych formularzy internetowych
- Tworzenie lepszych interfejsów użytkownika z użyciem CSS3
- Zwiększanie dostępności witryn internetowych
- Rysowanie na kanwach
- Osadzanie danych audio, wideo oraz grafiki wektorowej
- Wykorzystanie cieni, gradientów, transformacji i czcionek
- Przetwarzanie danych po stronie klienta
- Korzystanie z interfejsów API

Jesteś gotowy? Doskonale! Zacznij już dziś poznawać standardy przyszłości!

Pragmatic Bookshelf

helion.pl
księgarnia internetowa

Nr katalogowy: 5874

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 49,00 zł

ISBN 978-83-246-3028-8

9 788324 630288

Informatyka w najlepszym wydaniu