Utopia HTML. Projektowanie w CSS bez użycia tabel

Autor: Dan Shafer
Tłumaczenie: Renata Wójcicka
ISBN: 83-7361-965-8
Tytuł oryginału: HTML Utopia: Designing Without Tables Using CSS
Format: B5, stron: 424

Internet powstał jako medium komunikacyjne umożliwiające wymianę danych pomiędzy ośrodkami badawczymi. Nikt wówczas nie przewidywał ogromnej szybkości, z jaką się rozwinie, i popularności, jaką zdobędzie. Strony WWW były proste, zawierały głównie tekst, a wyświetlanie ich w innych urządzeniach niż komputery stacjonarne kojarzone było raczej z powieściami science-fiction. Dziś internet to potężny zbiór informacji i miliony stron WWW. Jego użytkownicy używają różnych przeglądarek, różnych systemów operacyjnych i różnych urządzeń do przeglądania jego zasobów.
Taka sytuacja wywołała konieczność ustanowienia standardów, w oparciu o które tworzone są strony WWW. Trzeba było także wypracować mechanizmy pozwalające łatwo zmieniać sposób formatowania stron, aby dostosować je do wymagań oraz możliwości przeglądarek i urządzeń.

Czytając książkę „Utopia HTML. Projektowanie w CSS bez użycia tabel", poznasz CSS (kaskadowe arkusze stylów) – technologię, która umożliwia niemal dowolne formatowanie każdego dokumentu HTML. Dowiesz się, jak definiować style CSS i jak stosować je w procesie tworzenia stron WWW. Nauczysz się projektować strony, w których zmiana wyglądu nie będzie wymagać przekonstruowania kodu HTML, a jedynie modyfikacji kilku linijek w arkuszu stylów. Poznasz również niestandardowe zastosowania kaskadowych arkuszy stylów.

- Podstawowe wiadomości o CSS
- Style lokalne i globalne
- Mechanizmy dziedziczenia
- Selektory i pseudoklasy
- Rozmieszczanie elementów strony za pomocą stylów
- Przypisywanie kolorów
- Efekty tekstowe
- Walidacja dokumentów CSS

Książka zawiera również zestawienie wszystkich atrybutów CSS, które można stosować przy definiowaniu stylów.

Wydawnictwo Helion
ul. Chopina 6
44-100 Gliwice
tel. (32)230-98-63
e-mail: helion@helion.pl

Spis treści

Rozdział 4.

Projektowanie serwisu w CSS

Rozwój każdego serwisu rozpoczyna się od jego zaprojektowania. To, co będzie zawarte w serwisie, wymyślasz sam bądź otrzymujesz idee od klienta. Jeśli jesteś projektantem przyzwyczajonym do szczegółowego sporządzania dokumentacji, projektowanie może objąć również stworzenie takiego dokumentu. Będzie to wymagać uwzględnienia przypadków (ang. *case*), czyli możliwych zachowań odwiedzającego stronę. Trzeba będzie również wziąć pod uwagę oficjalne specyfikacje oraz zalecenia, a także listę przeglądarek oraz platform umożliwiających dostęp do strony.

Na tym etapie zwykle buduje się serię prototypów powstałych z projektów na papierze. Następnie z prototypowania wzorów w programie graficznym przechodzi się do stworzenia w pełni funkcjonalnej strony HTML. Jeśli posiadasz pewne doświadczenie w projektowaniu stron, z łatwością przeskoczysz z projektu konceptualnego do kodowania w HTML, odzwierciedlającego Twój pomysł na ekranie.

Rezygnując z tabel na rzecz CSS, poznasz zupełnie nowe zasady projektowania, na podstawie których będziesz budować swoje pierwsze prototypy. W kolejnych rozdziałach tej książki omówię wszystkie te zasady, tak byś zyskał pewne wyobrażenie o ograniczeniach CSS i mógł puścił wodze fantazji.

W ludzkiej naturze leży pewna niechęć do zmian. Gdy zapoznasz się z rzeczami, których CSS nie potrafi dokonać, zamiast ochoty na poznawanie nowego, wspaniałego świata CSS, gdzie układ setek stron może zależeć od jednej reguły, odczujesz pokusę powrotu do topornego projektowania przy użyciu tabel. Będę starał się nakłonić Cię do porzucenia starych przyzwyczajeń, przedstawiając kilka głównych zalet konstruowania stron przy użyciu CSS. Pokażę również parę witryn internetowych, które, podjąwszy stanowcze kroki, zbierają teraz owoce rozplanowywania w CSS.

Zalety tworzenia strony w CSS

W poprzednich rozdziałach omówiłem kilka charakterystycznych cech oraz powodów stosowania reguł CSS przy tworzeniu układu strony. W tej sekcji zbiorę w jednym miejscu wszystkie argumenty „za". Nie tylko mam nadzieję na przekonanie Cię o zaletach używania CSS, lecz także chcę przedstawić parę sposobów na nakłonienie innych do zaakceptowania tej technologii.

W świecie bezwzględnej konkurencji, pracując na umowę-zlecenie, często będziesz musiał przedstawić powody, dla których lepiej niż inni projektanci nadawałbyś się do wykonania określonego projektu. Jeśli tworzenie witryn w CSS należy do Twojego arsenału programistycznego, z pewnością będzie to Twoim dużym atutem. Wiele z zalet projektowania w CSS wykracza daleko poza łatwość późniejszego rozwijania strony i przekłada się na dodatkowe korzyści dla klienta. Niech on o tym się dowie — może stanie się to decydującym powodem wygrania kontraktu.

Zwiększona kontrola składni

Na pierwszy rzut oka kluczową zaletą CSS oraz głównym powodem, dla którego projektanci witryn rozpoczynają pracę z tą technologią, jest możliwość kontroli wielu aspektów wyglądu strony, których nie da się zwyczajnie określać w czystym HTML. Przykładem może być moda na usuwanie podkreślenia z hiperłączy oraz zaznaczanie ich za pomocą innych stylów (np. kolorowania tekstu, pogrubiania czcionki oraz podkreślenia łączy w momencie najechania na nie myszą). Kompletny spis właściwości stylów, które mogą być kontrolowane przez CSS, znajduje się w dodatku C.

Oprócz pokaźnej liczby takich właściwości, CSS pozwala na zastosowanie ich do większego zakresu elementów HTML. W czystym HTML, jeśli zechce się użyć obramowania wokół pewnego obszaru, trzeba będzie stworzyć tabelę, gdyż jedynie ona posiada odpowiedni do tego atrybut. CSS nie tylko umożliwia większą kontrolę nad wyglądem obramowania (możliwości jest wiele: obramowanie może być jednolite (ang. *solid*), wzorzyste (ang. *embossed*), nakrapiane (ang. *dotted*), przerywane (ang. *dashed*), grube bądź cienkie, czerwone lub zielone, itd.), lecz także pozwala na dodawanie obramowania do dowolnie wybranego elementu, a nie tylko tablicy. Celem CSS jest zaoferowanie projektantowi jak najszerszego wachlarza możliwości. Tak więc ideą przyświecającą CSS jest umożliwienie zastosowania reguł stylów wszędzie tam, gdzie ma to rzeczywisty sens.

CSS posiada więcej właściwości stylów, które mogą się odnosić do większej liczby elementów, niż HTML kiedykolwiek pozwalał. Gdybyś miał wybrać pomiędzy CSS a HTML jako środkiem na określenie wyglądu strony, CSS zwyciężyłby bez najmniejszych wątpliwości. Mimo to powszechnie stosuje się HTML do projektowania układu strony, a do CSS ucieka się tylko w szczególnych przypadkach, gdy HTML nie potrafi sobie z pewnymi rzeczami poradzić. Podczas gdy wizualny efekt jest porównywalny, traci się po drodze mnóstwo zalet CSS.

Scentralizowana informacja o rozmieszczeniu elementów na stronie

Jak już to nakreśliłem wcześniej, najlepszą metodą na użycie CSS przy projektowaniu stron WWW jest dołączenie jednego lub więcej plików *.css* zawierających odpowiednie reguły stylów za pomocą znacznika `<link>`. W ten sposób wygląd strony jest wyraźnie oddzielony od jej treści i znajduje się jednym miejscu.

Ideą przyświecającą takiemu postępowaniu jest umożliwienie zmiany treści strony bez konieczności modyfikowania jej wyglądu i odwrotnie. Przy tradycyjnym projektowaniu stron, gdzie znaczniki HTML oraz atrybuty służą do określenia sposobu wyświetlania całości w przeglądarce, kod tych dwóch aspektów jest przemieszany. Jeśli zatem ktokolwiek chciałby zmodyfikować jeden z nich, automatycznie musiałby się znać na obu. Inaczej ryzykowałby popsucie czegoś na stronie. W tym sensie mówimy o powiązaniu treści i wyglądu witryny.

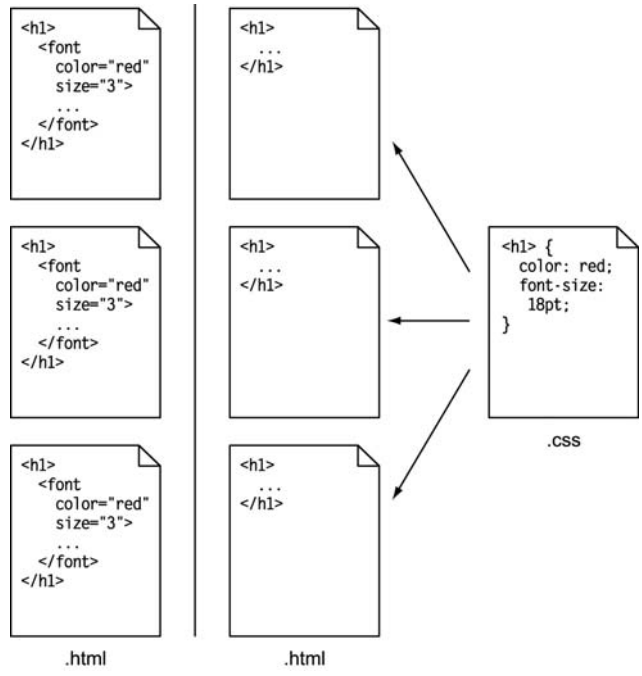
Fakt podzielenia kodu na części służące do różnych celów jest znany w programistycznym świecie (ang. *decoupling*). Gdy oprawa graficzna serwisu jest niezależna od jego treści, projektant witryny łatwo modyfikuje jej wygląd poprzez edytowanie plików *.css*, natomiast osoba zajmująca się treścią dodaje zawartość do plików *.html*.

Taki podział kodu przeciwdziała także jego powielaniu. W projektowaniu opartym na znacznikach HTML, jeśli tytuł znajdujący się na górze każdego artykułu ma być wyświetlany powiększoną, czerwoną czcionką, należy umieścić znacznik `<font>` wewnątrz odpowiedniego znacznika `<h1>` na każdej stronie serwisu. Gdy używa się wzoru w CSS, można w jednym miejscu określić właściwości czcionki dla znaczników `h1`, co znacząco oszczędza czas pisania kodu. W momencie zmiany wyglądu tego typu nagłówków należy jedynie dokonać pewnej modyfikacji pliku CSS — nie ma potrzeby sprawdzania każdego dokumentu HTML z osobna. Na pewno oszczędza to czas i nerwy. Powyższe różnice zilustrowano na rysunku 4.1.

Jeśli przyjrzysz się uważnie rysunkowi 4.1, spostrzeżesz, że oprócz zalet organizacyjnych opisanych wyżej, istnieje dodatkowy atut — przeglądarka ma mniej kodu do pobrania. W przypadku witryn mocno obłożonych kodem lub składających się z setek stron, ograniczenie czasu ładowania się strony może bardzo pozytywnie wpłynąć na ocenę użytkownika i na koszty połączeń.

Znaczniki semantyczne

Gdy używa się plików *.css* do oddzielenia zawartości witryny od jej treści, można zaobserwować ciekawe rzeczy. Ponieważ CSS zapewnia całkowitą kontrolę nad wyglądem poszczególnych składników strony, zaczyna się używać znaczników do oznaczenia struktury i znaczenia elementów na stronie. Kod HTML pozbawiony informacji dotyczących wyglądu i formy prezentacji strony staje się czystym nośnikiem semantycznych treści.



Rysunek 4.1. *CSS centralizuje kod projektu*

Istnieje kilka powodów utrzymywania takiego stanu rzeczy. Jednym z nich jest łatwość dokonywania zmian w zawartości witryny. Najprostszą metodą na zlokalizowanie strony używającej CSS jest spojrzenie w jej źródło — opcja ta dostępna jest w każdej przeglądarce. Jeśli jesteś w stanie określić zawartość strony w przeciągu 10 sekund, całkiem prawdopodobne jest, że nie masz do czynienia z układem strony bazującym na tabelach ani z żadnym innym niesemantycznym kodem HTML.

Użycie semantycznych znaczników HTML znacząco wpływa na mechanizmy wyszukiwania stosowane przez wyszukiwarki internetowe. Im mniej jest znaczników opisujących wygląd strony, tym bardziej jest prawdopodobne, że zwiększy się gęstość występowania na stronie danego słowa kluczowego, która to wartość jest istotnym czynnikiem determinującym popularność strony.

Jak zobaczymy później, CSS pozwala na pozycjonowanie elementu wewnątrz okna przeglądarki w oderwaniu od jego miejsca w dokumencie HTML. Jeśli zatem posiadasz na stronie formularz służący do zamawiania prenumeraty listy dystrybucyjnej bądź inny obszerny fragment kodu HTML, niewnoszący wiele do samej treści dokumentu, możesz swobodnie przenieść część kodu na koniec dokumentu, a następnie użyć CSS do ustawienia formularza na górze okna przeglądarki.

Znacznik `<link>`[1] oferuje możliwości coraz częściej wspierane przez współczesne przeglądarki. Pozwala mianowicie ograniczyć stosowanie dołączonych reguł stylów tylko do przeglądarek i ekranów określonego typu. Przykładowo, mógłbyś dołączyć do strony trzy pliki *.css* — jeden definiowałby wygląd strony wyświetlanej na ekranie

[1] A dokładniej atrybut *media*.

komputera, drugi odnosiłby się do drukowanej strony, natomiast trzeci miałby zastosowanie w przypadku telefonów komórkowych. Tylko poprzez użycie znaczników semantycznych oraz poprzez pozwolenie na to, aby CSS zajął się sposobem wyświetlania strony, można uzyskać takie dostosowanie zawartości do różnych okoliczności.

Na koniec należy wspomnieć o rzeczy nie mniej ważnej, a mianowicie o zwiększonej dostępności do witryn używających znaczników semantycznych. Zajmiemy się tym bardziej szczegółowo w następnej sekcji.

Dostępność

Gdybyś miał kiedyś okazję być świadkiem przeglądania stron WWW przez osoby mające problemy ze wzrokiem, polecam to doświadczenie. Alternatywnie, możesz użyć oprogramowania czytającego zawartość ekranu. Wyłącz monitor i sprawdź na własnej skórze, jak to jest być osobą mającą słaby wzrok, próbującą poruszać się w internecie.

Witryny często korzystające z tabel, rysunków oraz innych niesemantycznych elementów HTML są bardzo niewygodne przy odczytywaniu na głos zawartości strony. Nie jest takie znów niespotykane wśród współczesnych serwisów, że załadowanie strony do momentu wyświetlenia właściwej zawartości zajmuje około 30 sekund. Skoro już teraz nie brzmi to dobrze, pomyśl, że musiałbyś wysłuchiwać takich niepotrzebnych informacji za każdym razem, gdy nowa strona zostałaby załadowana.

Znaczniki semantyczne pozwalają na niemal całkowite pozbycie się tej kakofonii dźwięków, gdyż każdy znacznik dokumentu posiada dzięki nim pewne znaczenie strukturalne, istotne w tym przypadku dla użytkownika (a właściwie słuchacza). Przeglądarki odczytujące zawartość strony na głos ignorują graficzne właściwości elementów interfejsu, toteż niewidomy użytkownik nie powinien być zmuszany do ich odsłuchiwania.

W witrynie używającej znaczników semantycznych osoba z uszkodzonym wzrokiem nie musiałaby zachodzić w głowę, czy pogrubienie tekstu miało jakieś specjalne znaczenie, czy tylko pewien walor estetyczny. Elementy wyświetlone pogrubioną czcionką dla celów estetycznych posiadałyby odpowiednią regułę zdefiniowaną wewnątrz definicji CSS, więc przeglądarka czytająca na głos pominęłaby tę cechę. Elementy, które miałyby się odznaczać na tle innych pod względem semantycznym, wyróżnione byłyby znacznikami `<strong>` oraz `<em>`, czyli tekst byłby pogrubiony i wyświetlony kursywą.

Istnieją osobne poradniki na temat tego, jak należy tworzyć serwisy bardziej dostępne dla użytkowników niepełnosprawnych. Wytyczne dotyczące dostępu do sieci przez osoby niepełnosprawne, czyli inaczej WCAG (ang. *Web Content Accessibility Guidelines*)[2], są pozycją polecaną wszystkim projektantom witryn. W zaleceniach[3] omówiona jest idea unikania znaczników prezentacyjnych na rzecz znaczników semantycznych.

[2] *http://www.w3.org/TR/WCAG10/*

[3] *http://www.w3.org/TR/WCAG10/#g1-structure-presentation*

Zgodność ze standardami

WCAG nie jest jedyną specyfikacją popierającą ideę podziału kodu strony na część prezentacji (CSS) oraz część zawartości (HTML). W rzeczywistości ostatnie standardy HTML[4] były tworzone właśnie w tym duchu.

Konsorcjum World Wide Web[5] (W3C) jest organizacją odpowiedzialną za publikowanie zaleceń (a właściwie standardów) odnoszących się do sieci. Poniżej znajduje się kilka zaleceń związanych z semantycznymi znacznikami oraz z CSS:

HTML 4 (*http://www.w3.org/TR/html4*)

Ostatnia wersja dokumentu oznacza wszystkie niesemantyczne znaczniki i ich atrybuty jako niezalecane[6]. Znacznik `<font>`[7], na przykład, jest w tym standardzie jasno wyszczególniony jako przestarzały. Oto komentarz do tych niezalecanych elementów[8]:

Ogólnie rzecz biorąc, autorzy stron powinni używać zamiast znaczników HTML arkuszy stylów do formatowania i nadawania odpowiedniego kształtu elementom.

XHTML 1.0 (*http://www.w3.org/TR/xhtml1/*)

XHTML, jako HTML 4 przetransformowane do dokumentów XML, pozwala na używanie znaczników oraz atrybutów, jednocześnie wykorzystując funkcjonalność XML (mieszanie języków znaczników, tworzenie własnych itd.).

To zalecenie zawiera te same znaczniki oraz niezalecane elementy co HTML 4.

Wytyczne dotyczące dostępu do sieci (*Web Content Accessibility Guidelines 1.0*) (*http://www.w3.org/TR/WCAG10/*)

Tak jak to zostało opisane w sekcji „Dostępność", WCAG gorąco zaleca używanie CSS oraz znaczników semantycznych do tworzenia projektów stron usprawniających dostępność. Zalecenie to mówi samo za siebie:

Nadużywanie znaczników do osiągnięcia pewnych efektów wizualnych (przykładowo, tabel do budowania układu strony, a nagłówków do zwiększenia rozmiaru czcionki) utrudnia użytkownikom korzystającym ze specjalistycznego oprogramowania poruszanie się po stronie. Co więcej, uniemożliwione jest poprawne wyświetlenie strony na innych urządzeniach, gdy rezygnuje się ze znaczników strukturalnych na rzecz znaczników prezentacji przy przekazywaniu treści.

---

[4] *http://www.w3.org/MarkUp/#recommendations*

[5] *http://www.w3.org/*

[6] Element niezalecany jest oznaczony jako ten do usunięcia ze specyfikacji, w związku z czym nie powinien być dłużej używany. Dokument, który ma być zgodny ze standardami, nie powinien stosować tego typu elementów.

[7] *http://www.w3.org/TR/html4/present/graphics.html#h-15.2.2*

[8] *http://www.w3.org/TR/html4/conform.html#deprecated*

Zdaniem wielu projektantów stron, ścisłe przestrzeganie standardów jest praktycznie niemożliwe. Celem niniejszej książki jest udowodnienie, że nie jest to do końca prawda.

Współczesne przeglądarki działają znacznie lepiej, gdy przestrzega się standardów. Choć błędy oraz problemy ze zgodnością nadal istnieją, nie są one mniej rozwiązywalne niż błędy, z którymi spotyka się projektant bazujący na kodzie niezgodnym ze standardami.

Przykłady zwieńczonego sukcesem zastosowania CSS

Poniższe witryny mogą posłużyć za doskonały przykład na to, co można osiągnąć, budując układ strony wyłącznie na CSS.

SitePoint (*http://www.sitepoint.com/*)

Wiem, wiem, reklamowanie serwisu swoich wydawców nie wydaje się być odpowiednią rzeczą, jednak chłopaki nie tylko wykonali kawał dobrej roboty zamieniając cały układ strony bazujący na tabelach na jego odpowiednik w CSS, lecz także znacząco zwiększyli przy tej okazji funkcjonalność serwisu.

Mimo że na pierwszy rzut oka dobór kolorów może się wydać nieco ubogi, taki „odchudzony" wygląd strony pozwala na jej szybkie ładowanie, nawet w przypadku obszernej treści i bogactwa opcji nawigacji.

A List Apart (*http://www.alistapart.com/*)

Od momentu jej powstania, witryna łącznie z listą dystrybucyjną stała się jednym z głównych źródeł informacji na temat projektowania stron przy użyciu CSS. Serwis jest utrzymany w duchu minimalistycznym, jednak udowadnia, że prosty nie znaczy nudny ani nieciekawy.

Netscape DevEdge (*http://devedge.netscape.com/*)

DevEdge jest serwisem Netscape przeznaczonym dla projektantów rozwijających witryny internetowe. Ponieważ Netscape w wersji 6. oraz 7. był już oparty na mechanizmie zgodnym z obowiązującymi standardami, nie pozostało nic innego jak przerobić witrynę tak, by mogła pełnymi garściami czerpać z tej technologii. W serwisie znajduje się nawet artykuł[9] poświęcony temu przeprojektowaniu.

ESPN (*http://www.espn.com/*)

Pierwsza czołowa, komercyjna witryna internetowa zbudowana na podstawie technik CSS. Jest z pewnością krokiem milowym przy projektowaniu witryn.

[9] *http://devedge.netscape.com/viewsource/2003/devedge-redesign/*

Klienci często będą pytali o inne serwisy, które zostały zaprojektowane w tej samej technologii i odniosły spektakularny sukces.

Do tej pory wszystkie najlepsze przykłady projektów w CSS były albo witrynami stworzonymi przez profesjonalnych projektantów dla innych twórców witryn, albo domowymi serwisami mogącymi ponieść pewne ryzyko niedociągnięć, gdyż nie musiały na siebie zarabiać.

W serwisie Netscape DevEdge[10] znajduje się wyczerpujący wywiad z jednym z projektantów tej witryny.

Fast Company Magazine (*http://www.fastcompany.com/*)

Jest to wersja online popularnego magazynu. Witryna została przebudowana przy użyciu CSS. Aktualna wersja serwisu nie różni się znacząco od tej poprzedniej, jednak dzięki CSS strony ładują się o wiele szybciej.

Przykładowa witryna: Footbag Freaks

W pozostałej części książki, gdzie tylko to będzie możliwe, będę odnosił się do przykładu witryny stworzonej specjalnie na potrzeby tej książki. Tę fikcyjną witrynę, nazwaną Footbag Freaks, będzie można pobrać ze strony: *http://www.footbagfreaks.com/*. Kod źródłowy jest również gotowy do pobrania ze strony książki[11]. Na rysunku 4.2 widać główną stronę serwisu.

Witryna w pełni wykorzystuje CSS zarówno do tworzenia układu strony, jak i do nadawania stylów fragmentom tekstu oraz innym elementom na stronie. Kod HTML jest całkowicie semantyczny. Stronę zaprojektowano i przetestowano pod następującymi przeglądarkami:

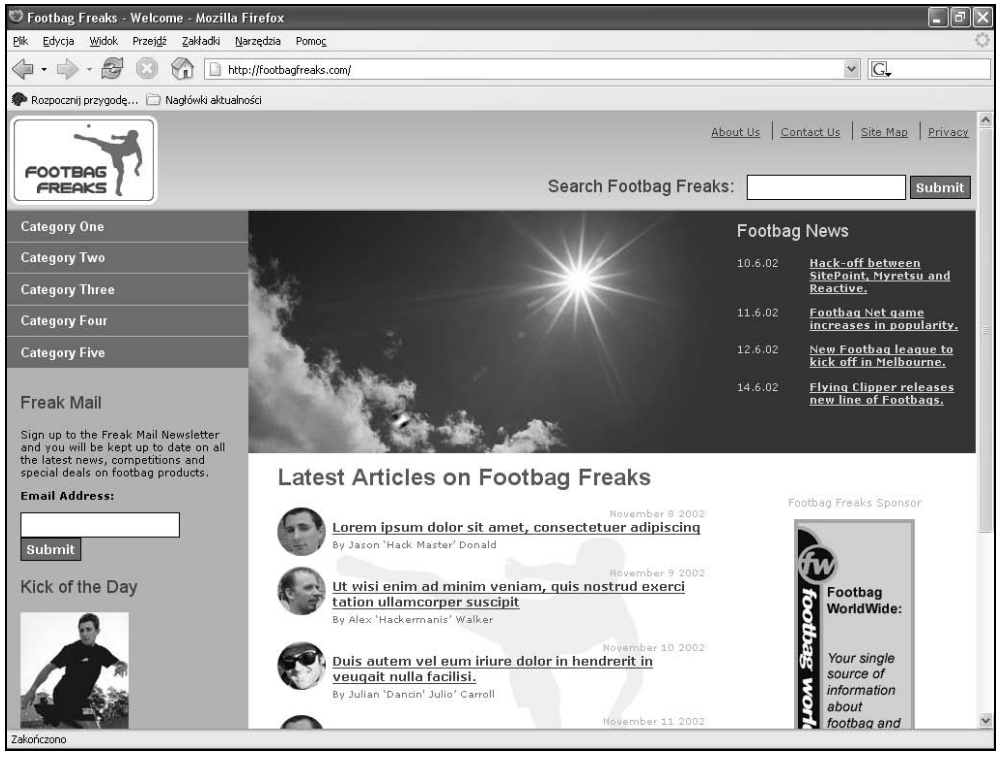
- Internet Explorer 5 lub w wersji wyższej pod Macintosh i Windows,
- Opera 6 lub w wersji wyższej,
- Mozilla 1.0 i wersje późniejsze oraz przeglądarki podobnego typu, łącznie z Netscape 6 i wersjami późniejszymi oraz Camino.

Witryna jest zgodna z następującymi zaleceniami W3C:

- XHTML 1.0 (dokładna kompatybilność),
- WCAG 1.0 (ocena pod względem dostępności do strony),
- CSS 2.0.

[10] *http://devedge.netscape.com/viewsource/2003/espn-interview/01/*

[11] *http://www.sitepoint.com/books/*



Rysunek 4.2. *Strona główna witryny Footbag Freaks*

Podsumowanie

W tym rozdziale podałem kilka najistotniejszych zalet, które CSS ma do zaoferowania przy projektowaniu witryn internetowych. Są nimi przede wszystkim:

- zwiększona kontrola składni,
- scentralizowana informacja o rozmieszczeniu elementów na stronie,
- znaczniki semantyczne,
- dostępność,
- zgodność ze standardami.

Po zaprezentowaniu paru witryn, w których użycie CSS doprowadziło do powstania ciekawych efektów i zakończyło się sukcesem, wprowadziłem naszą własną, stworzoną jedynie na potrzeby tej książki witrynę Footbag Freaks. W pozostałej części książki zajmiemy się szerokim zakresem funkcjonalności oraz technik CSS pozwalających na zbudowanie takiego właśnie serwisu.

Rozdział 5. rozpoczyna się od zbudowania szkieletu strony, a następnie przechodzi do zapełnienia go odpowiednią treścią, przy użyciu jedynie technik CSS.