

IDŹ DO

PRZYKŁADOWY ROZDZIAŁ



SPIS TREŚCI

KATALOG KSIĄŻEK

KATALOG ONLINE

ZAMÓW DRUKOWANY KATALOG

TWÓJ KOSZYK

DODAJ DO KOSZYKA

CENNIK I INFORMACJE

ZAMÓW INFORMACJE
O NOWOŚCIACH

ZAMÓW CENNIK

CZYTELNIA

FRAGMENTY KSIĄŻEK ONLINE

Kuloodporne strony internetowe. Jak poprawi elastyczność z wykorzystaniem XHTML-a i CSS. Wydanie II

Autor: Dan Cederholm

Tłumaczenie: Anna Trojan

ISBN: 978-83-246-1474-5

Tytuł oryginału: [Bulletproof Web Design: Improving flexibility and protecting against worst-case scenarios with XHTML and CSS \(2nd Edition\)](#)

Format: 170x230, stron: 328



Poznaj tajniki projektowania doskonałych stron internetowych

- Jak zbudować elastyczne mechanizmy nawigacyjne?
- W jaki sposób stosować elementy pływające?
- Jak w pełni wykorzystać możliwości kaskadowych arkuszy stylów?

W coraz dynamiczniej rozwijającym się internecie każdego dnia pojawiają się tysiące witryn WWW. Wiele z nich to prawdziwe arcydzieła – zarówno pod względem projektowym, jak i merytorycznym. Często jednak, otwierając swoją ulubioną witrynę WWW nie w tej przeglądarce, co dotychczas, widzimy coś zupełnie innego! Ile razy, podczas projektowania witryny WWW szukamy rozwiązania problemów związanych z różnymi interpretacjami standardów W3C w różnych przeglądarkach? Ile czasu zajmują zmiana układu graficznego strony lub dodanie nowych elementów? Ilu projektantów stron WWW zastanawia się nad tym, jak poprawić ich elastyczność i uniwersalność?

Książka „Kuloodporne strony internetowe. Jak poprawić elastyczność z wykorzystaniem XHTML-a i CSS. Wydanie II” opisuje techniki tworzenia niemal doskonałych witryn WWW. Doskonałych – bo wyglądających równie dobrze w każdej przeglądarce i przy każdej rozdzielczości ekranu, łatwych w nawigacji i prostych w rozbudowie. Czytając ją, poznasz najnowsze standardy XHTML i CSS, nauczysz się odpowiednio dobierać czcionki i ich rozmiary, korzystając z jednostek em; dowiesz się, jak pozycjonować elementy strony i budować uniwersalne mechanizmy nawigacyjne. Odkryjesz sposoby tworzenia uniwersalnych stron internetowych.

- Dobór rozmiaru i jednostek miary tekstu
- Skalowalne menu
- Zastępowanie tabel elementami pływającymi
- Pozycjonowanie składników strony
- Definiowanie stylów dla ramek
- Oddzielanie warstwy prezentacji od treści strony
- Korzystanie z narzędzi do weryfikacji konstrukcji strony
- Budowanie skalującego się układu strony

**Wykorzystaj wiedzę zawartą w tej książce i twórz strony internetowe,
które zachwycą nie tylko wyglądem, ale też uniwersalnością**

Wydawnictwo Helion
ul. Kościuszki 1c
44-100 Gliwice
tel. 032 230 98 63
e-mail: helion@helion.pl

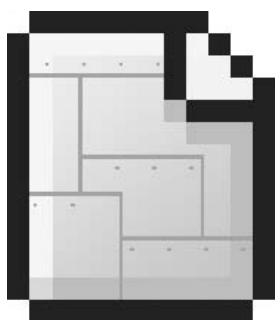


Spis treści

Wprowadzenie	9
Rozdział 1. Elastyczny tekst	15
Rozmiar tekstu należy ustalać przy użyciu słów kluczowych oraz wartości procentowych lub em, dzięki czemu użytkownicy będą mieli nad nim kontrolę	16
Częste rozwiązanie	17
Wszystkie możliwości	20
Rozwiązanie kuloodporne	22
Dlaczego rozwiązanie to jest kuloodporne	27
Elastyczna baza — i co dalej?	28
Stosowanie słów kluczowych i wartości procentowych	32
Elastyczny rozmiar tekstu dzięki em	38
Podsumowanie	40
Rozdział 2. Skalowalna nawigacja	43
Należy zapewnić nawigację, która skaluje się w zależności od ustawionej wielkości tekstu oraz od objętości treści umieszczonych na stronie	44
Częste rozwiązanie	45
Dlaczego rozwiązanie to nie jest kuloodporne	47
Rozwiązanie kuloodporne	49
Dlaczego rozwiązanie to jest kuloodporne	58
Podobny przykład wykorzystujący em	59
Dodatkowe przykłady	61
Podsumowanie	64
Rozdział 3. Elastyczne wiersze	67
Nie należy definiować wysokości poziomych elementów strony i trzeba zaplanować możliwość powiększenia ich w pionie	68
Częste rozwiązanie	69
Dlaczego rozwiązanie to nie jest kuloodporne	70
Rozwiązanie kuloodporne	72
Dlaczego rozwiązanie to jest kuloodporne	87
Inny przykład rozciągania	88
Podsumowanie	93

Rozdział 4. Pomysłowe rozmieszczanie elementów	95
Zamiast stosować tabele, lepiej jest używać elementów pływających	96
Częste rozwiązanie	97
Dlaczego rozwiązanie to nie jest kuloodporne	98
Rozwiązanie kuloodporne	99
Dlaczego rozwiązanie to jest kuloodporne	136
Podsumowanie	136
Rozdział 5. Niezniszczalne ramki	139
Zanim zaczniesz się tworzyć style dla ramek, należy wszystko dokładnie zaplanować	140
Częste rozwiązanie	141
Dlaczego rozwiązanie to nie jest kuloodporne	143
Rozwiązanie kuloodporne	144
Dlaczego rozwiązanie to jest kuloodporne	150
Inne techniki tworzenia zaokrąglonych narożników	151
Iluzja ramki	159
Podsumowanie	166
Rozdział 6. Brak rysunków? Brak CSS? Brak problemów!	169
Należy upewnić się, że treść strony jest czytelna, nawet jeśli strona zostanie pozbawiona rysunków i stylów CSS	170
Częste rozwiązanie	171
Dlaczego rozwiązanie to nie jest kuloodporne	174
Rozwiązanie kuloodporne	174
Dlaczego rozwiązanie to jest kuloodporne	176
Ze stylami lub bez	179
Test Dig Dug	184
Narzędzia do sprawdzania kuloodporności stron	185
Podsumowanie	193
Rozdział 7. Konwersja tabel	197
Należy usunąć z tabel kod prezentacyjny, a ich wygląd zdefiniować za pomocą CSS	198
Częste rozwiązanie	199
Dlaczego rozwiązanie to nie jest kuloodporne	200
Rozwiązanie kuloodporne	201
Dlaczego rozwiązanie to jest kuloodporne	222
Podsumowanie	224
Rozdział 8. Płynny oraz elastyczny układ strony	225
Eksperymentuj z projektowaniem układów stron, które rozszerzają się i zwężają	226
Częste rozwiązanie	227
Dlaczego rozwiązanie to nie jest kuloodporne	229
Rozwiązanie kuloodporne	231

Dlaczego rozwiązanie to jest kuloodporne	264
Układ strony oparty na em	264
Podsumowanie	272
Rozdział 9. Łączenie w całość	275
Zastosuj kuloodporne rozwiązania do projektu całej strony internetowej	276
Cel	277
Dlaczego rozwiązanie to jest kuloodporne	278
Konstrukcja	283
Modyfikacja stylów ze względu na przeglądarkę IE	310
Podsumowanie	314
Skorowidz	315



2

SKALOWALNA NAWIGACJA



Należy zapewnić nawigację, która skaluje się w zależności od ustawionej wielkości tekstu oraz od objętości treści umieszczonych na stronie

Nawigacja często stanowi najważniejszy element projektu witryny internetowej. Komponent ten umieszczony jest w widocznym miejscu i każdy użytkownik witryny z niego korzysta. Tradycyjny sposób projektowania polega na tym, że twórca witryny otwiera ulubiony program graficzny i rysuje przyciski, zakładki czy tekst, które zostaną osadzone wewnątrz kilku zagnieżdżonych tabel. Często stosowany jest też efekt podmiany, który pozwala zastępować jeden zestaw grafik innym, gdy użytkownik najedzie myszą na odpowiedni obszar. Do uzyskania tego efektu wykorzystuje się na przykład skrypty języka JavaScript. Utworzona w taki sposób strona może wyglądać olśniewająco, jednak kryjący się za nią kod może nie być już tak perfekcyjny, co zobaczymy na przykładzie często stosowanego rozwiązania.

Przyjrzyjmy się naszemu przykładowi — stylowym zakładkom zaprojektowanym w tradycyjny sposób i umieszczonym na stronie w postaci obrazków. Najpierw przeanalizujemy wady takiego rozwiązania, a następnie zaprojektujemy bardzo podobny komponent, tyle że będzie on kuloodporny. Nasze rozwiązanie będzie się składać z poprawnego i zwięzłego kodu, trzech niewielkich obrazków oraz stylów CSS.

Częste rozwiązanie

Zakładki nawigacyjne z witryny *LanceArmstrong.com* (rysunek 2.1) wybrałem jako przykład z dwóch powodów. Pierwszy powód jest taki, że jestem fanem Lance’a Armstronga. Drugi — bardziej istotny — to fakt, iż wykorzystanie CSS przy przebudowie tego projektu może znacznie zwiększyć jego elastyczność. Naszym celem jest pomysłowe zastosowanie stylów CSS, dzięki czemu będziemy mogli pozbyć się dużych kawałków kodu, a nowa wersja witryny będzie łatwa w utrzymaniu oraz skalowalna.



Rysunek 2.1. Ładnie zaprojektowane zakładki nawigacyjne w witrynie *LanceArmstrong.com*, w których wyraźnie wyróżniono aktywny i nieaktywny stan zakładek

Przjrzyjmy się dokładnie naszemu przykładowi i sprawdźmy, jakie elementy projektowe się na niego składają.

WYRAŻNE ZAKŁADKI

Zakładki z przykładu charakteryzuje pewien detal, dzięki któremu odróżniają się one od zwykłych zakładek składających się z obramowania oraz tła. Tło każdej zakładki — zarówno w stanie aktywnym, jak i nieaktywnym — posiada wypełnienie gradientowe. Jako tło zakładek wyświetlony jest powtarzany w poziomie wąski obrazek, którego kolor stopniowo zmienia się z ciemniejszego u góry na jaśniejszy u dołu zakładki (biały w przypadku stanu aktywnego oraz jasnożółty w przypadku stanu nieaktywnego). Na samej górze zakładki widoczne jest także podświetlenie o szerokości jednego piksela — jest to pasek, którego kolor jest o jeden odcień jaśniejszy niż kolor zakładki. Pasek ten wprowadza wrażenie trójwymiarowości — tak jakby światło padało z góry strony na dół (rysunek 2.2).

Zanim przejdę do dalszych rozważań, chcę podkreślić, że projekt zakładek wygląda świetnie. Ocena projektu jest sprawą gustu (każdy może mieć inne zdanie), dlatego powinniśmy przynajmniej zgodzić się, że ktoś poświęcił dużo czasu, by nawigacja w witrynie *LanceArmstrong.com* była atrakcyjna i funkcjonalna. Cel został osiągnięty — i jest to kolejny argument za tym, by witryna ta posłużyła nam za przykład.



Rysunek 2.2. Powiększenie, na którym szczegółowo przedstawiono obydwie stany zakładek. Na górze każdej zakładki widoczne jest delikatne podświetlenie, a kolor zakładki stopniowo się zmienia — takie rozwiązanie daje efekt trójwymiarowości zakładki i padającego z góry światła

W witrynie *LanceArmstrong.com* wszystkie zestawy zakładek (każdy z inną wyróżnioną zakładką) to osobne obrazki. Na każdym z nich znajduje się także logo witryny. Na rysunku 2.3 widoczny jest jeden z czterech dostępnych obrazków.



Rysunek 2.3. Cały zestaw zakładek i logo umieszczone są w jednym dużym obrazku. W witrynie wykorzystywane są przynajmniej trzy inne bardzo podobne obrazki (różniące się tylko tym, która zakładka jest podświetlona)

EFEKT PODMIANY

W pasku nawigacyjnym witryny *LanceArmstrong.com* nie jest aktualnie wykorzystywany żaden efekt podmiany. Dodanie go nie jest rzeczą niemożliwą — wystarczy napisać skrypt w języku JavaScript oraz wcześniej załadować pozostałe obrazki — jednak byłoby to związane z dodatkowym zwiększeniem ilości kodu. W dalszej części tego rozdziału, w podrozdziale „Rozwiązanie kuloodporne”, odtworzymy przykładowy projekt i dodamy do niego efekt podmiany obrazków. Uzyskamy go jednak dzięki wprowadzeniu tylko kilku dodatkowych reguł CSS, a nie zwiększając objętość kodu, dodając rysunki i skrypt w JavaScriptcie.

Na pierwszy rzut oka widzimy zbiór zakładek nawigacyjnych, które są zaprojektowane w taki sposób, by pasowały do projektu całej strony. Kliknięcie każdej zakładki powoduje przejście do głównych sekcji wi-

triny, a z każdą zakładką związana jest odpowiednia etykieta tekstowa. Teraz zajrzemy temu rozwiązaniu „pod maskę”, a na koniec dowiemy się, jak można uzyskać podobny efekt w alternatywny sposób.

Dlaczego rozwiązanie to nie jest kuloodporne

Dlaczego opisane rozwiązanie nie jest dobre? Nie mamy zamiaru krytykować projektantów odpowiedzialnych za witrynę *LanceArmstrong.com* — metody zastosowane w tej witrynie można odnaleźć w milionach witryn w całej Sieci. Przedstawiamy tę witrynę jako przykład głównie dlatego, że jej projekt graficzny jest bardzo dobry, a odwołanie się do tej strony przy użyciu CSS i poprawnego, dostępnego kodu będzie proste. Przedstawmy więc negatywne cechy tego rozwiązania.

OGROM KODU

Cecha prześladowająca witryny, w których nawigacji wykorzystywane są obrazki oraz skrypty napisane w JavaScriptcie, to duża ilość kodu potrzebnego do zdefiniowania układu strony i zapewnienia jej dynamiki (innymi słowy — przełączenia stanów przy podmianie). Zazwyczaj każdą zakładkę stanowi osobny obrazek (a tak naprawdę jeden obrazek dla każdego stanu zakładki), który jest osadzany w kilku zagnieżdżonych tabelach. Do rozwiązania dochodzą jeszcze wypełniacze GIF (przezroczyste obrazki stosowane do tworzenia pustych przestrzeni między elementami strony) oraz zbędny kod. Wszystkie te elementy służą tylko i wyłącznie do tworzenia precyzyjnego co do piksela układu strony. W ten sposób uzyskujemy ogromne ilości kodu, co oznacza potencjalne wydłużenie czasu pobierania stron oraz większy rozmiar plików.

W przypadku witryny *LanceArmstrong.com* wszystkie cztery zakładki umieszczone są w jednym pliku graficznym, do którego przypisana jest mapa obrazu, dzięki czemu istnieje możliwość kliknięcia obszaru każdej z zakładek. Z takim rozwiązaniem związana jest mniejsza ilość kodu w porównaniu z opcją, gdy każda zakładka jest *osobną* grafiką umieszczoną w oddzielnej komórce tabeli i obłożoną wypełniaczami GIF. Jednak problemy wymienione w dalszej części tego rozdziału dotyczą obydwu rozwiązań.



UWAGA

Wyjątkiem od ograniczenia związanego ze skalowalnością strony jest opcja powiększenia strony dostępna w przeglądarkach Internet Explorer 7 oraz Opera, która pozwala na powiększenie całego dokumentu.

PROBLEMY Z DOSTĘPNOŚCIĄ

Kolejnym efektem ubocznym stosowania dużych ilości kodu jest dławienie nim przeglądarek tekstowych oraz oprogramowania pomocniczego stosowanego przez użytkowników z wadami wzroku. Cała nawigacja mieści się w jednym obrazku (jak widać na rysunku 2.3), a większość twórców stron nie definiuje tekstu alternatywnego (atrybut `alt`) dla wszystkich obszarów aktywnych w mapie obrazu. Użytkownicy korzystający z czytników ekranu oraz ci, którzy wyłączają wyświetlanie obrazków, by skrócić czas pobierania strony (często stosowana metoda przy wolnym połączeniu z Internetem), będą odczuwali trudności przy nawigacji w takiej witrynie.

PROBLEMY ZE SKALOWALNOŚCIĄ

Ten problem jest bardzo poważny. Zakładki są *rysunkami*, dlatego osoby z wadami wzroku nie mają możliwości powiększenia w przeglądarce rozmiaru tekstu, by zwiększyć jego czytelność. Nawigacja utworzona na podstawie obrazków nie powiększy się, gdy zajdzie taka potrzeba.

BRAK ELASTYCZNOŚCI

W przyszłości redaktorzy witryny *LanceArmstrong.com* będą mieli spory kłopot, gdy będą chcieli zmienić opis którejs z zakładek — na przykład wyrażenie „About Lance” na „Why Lance Rules”. Przeredagowanie, usunięcie czy zamiana opisu którejkolwiek z zakładek wymagają utworzenia w programie graficznym nowych obrazów oraz zmiany wymiarów określonych w kodzie mapy obrazu. Z takimi drobnymi zmianami wiąże się dużo pracy wszystkich osób zaangażowanych w projekt.

Wszystkie zakładki zostały umieszczone w jednym obrazie, stąd wprowadzenie jednej zmiany oznacza konieczność zaktualizowania przynajmniej czterech rysunków (jeden dla każdej wyróżnionej zakładki), co z pewnością każdego projektanta może przyprawić o ból głowy.

Jak widać, w projekcie tym można wiele poprawić. W zależności od naszych własnych priorytetów każda z wymienionych wcześniej wad może być wystarczającym powodem, by zacząć szukać alternatywnego rozwiązania. A więc zrobmy to.

Rozwiązanie kuloodporne

Znając już wady tradycyjnych technik projektowania nawigacji opartej na obrazkach, możemy przebudować omawiany projekt zakładek ze strony *LanceArmstrong.com*. Skorzystamy z prostej, bazującej na standardach metody wykorzystującej style CSS, której implementacja nie wymaga dużych ilości kodu. Celem naszego projektu jest rozwiązanie omówionych wcześniej problemów występujących w wariacie tradycyjnym.

Jak ma to miejsce w przypadku każdego idealnego projektu, najpierw musimy zdecydować się na rozwiązanie, które zapewni nam funkcjonalność, prostotę oraz dostępność kodu. W przypadku nawigacji najbardziej odpowiednią strukturą dla grupy odnośników wydaje się — jak nietrudno odgadnąć — lista.

```
<ul id="nav">
  <li id="t-intro"><a href="/">Introduction</a></li>
  <li id="t-about"><a href="about.html">About
    Lance</a></li>
  <li id="t-news"><a href="news.html">News &
    Events</a></li>
  <li id="t-sponsors"><a href="sponsors.html">
    Sponsors</a></li>
</ul>
```

Do utworzenia kuloodpornego zestawu zakładek nawigacyjnych wystarczy nam zwykła lista wypunktowana (nieuporządkowana). Prosty kod zapewni dobrą czytelność we wszystkich przeglądarkach, urządzeniach i oprogramowaniu pomocniczym, nie wspominając już o tym, że jest on bardzo zwięzły i schludny. Wystarczy porównać ilość kodu w tym przykładzie z ilością kodu użytego do utworzenia zagnieżdżonych tabel. Różnica jest bardzo widoczna.

Każdy element listy ma przypisany unikatowy identyfikator (atrybut id). Identyfikatory przydadzą się nam później, gdy będziemy chcieli wyróżnić zakładkę odpowiadającą stronie, na której obecnie znajduje się użytkownik.

BEZ STYLÓW

Lista wypunktowana, której wyglądu nie określają żadne dodatkowe style, zwykle ma postać pionowej listy elementów; po lewej stronie każdego elementu wyświetlany jest znak wypunktowania (rysunek

- [Introduction](#)
- [About Lance](#)
- [News & Events](#)
- [Sponsors](#)

Rysunek 2.4. Wygląd listy wypunktowanej, której nie przypisano żadnych stylów CSS



WSKAZÓWKI

Kolejną zaletą stosowania listy jest fakt, że czytnik ekranu — przed odczytaniem wszystkich punktów listy — powie użytkownikowi, ile lista ma elementów. Dzięki takiej informacji osoba odwiedzająca stronę jest świadoma tego, co ją czeka. W ten sposób użytkownik popularnej aplikacji czytającej JAWS firmy Freedom Scientific po usłyszeniu liczby elementów znajdujących się na liście może przejść do następnej listy lub kolejnego elementu strony. Więcej informacji na temat sposobu odczytywania stron internetowych w aplikacji JAWS można znaleźć pod adresem www.freedomscientific.com/fs_products/Surfs_Up/Navigating.htm.

2.4). Taką listę nawigacyjną bez żadnych trudności i problemów mogą wyświetlić urządzenia lub przeglądarki tekstowe, które nie obsługują CSS.

DWA MAŁE OBRAZKI

W stosowanym początkowo rozwiązaniu wszystkie zakładki znajdowały się w jednej grafice. Tak naprawdę istniały cztery różne rysunki, a na każdym z nich wyróżniona była zakładka innego działu witryny. Aby uprościć całość rozwiązania, użyjemy tylko dwóch małych obrazków, które będziemy powielać w poziomie wewnątrz każdej zakładki. Obrazki będą powielane, dlatego nie musimy martwić się o ich szerokość ani wysokość — jest to ważny temat, do którego wrócimy w dalszej części tego rozdziału, w podrozdziale „Dlaczego rozwiązanie to jest kuloodporne”.

Ponownie przyjrzymy się oryginalnym zakładkom — przenosząc wzrok od górnej krawędzi w dół zakładki, widzimy, że kolor tła stopniowo się zmienia, a na dole zakładka ma już jednolity kolor, zgodny z kolorem tła strony. Używając naszego ulubionego programu graficznego (na przykład Adobe Photoshop), tworzymy dwa obrazki (jeden dla zakładek w stanie aktywnym, a drugi dla zakładek nieaktywnych), na których widoczne jest stopniowe przejście koloru zakładki w jednolite tło (rysunek 2.5). Jednolity obszar na dole każdego z obrazków wycinamy i ustawiamy jako przezroczysty. W miejscu, gdzie obrazek jest przezroczysty, widoczne będzie tło, którego kolor zdefiniujemy przy użyciu stylów CSS.



Rysunek 2.5. Powiększenie dwóch obrazków, którymi wypełnione będzie tło każdej z zakładek. Obszar obrazka, w którym tło jest jednolite, ustawiliśmy jako przezroczysty (jest to obszar oznaczony wzorem szachownicy). W obszarach tych widoczne będzie tło elementu zdefiniowane w stylach CSS

Obydwa obrazki mają szerokość 10 pikseli. Wzdłuż górnej krawędzi widoczne jest podświetlenie wysokości jednego piksela. Wypełnienie obrazka jest gradientowe i przechodzi w przezroczyste tło na dole obrazka.

STOSOWANIE STYLÓW

Przygotowaliśmy już wszystkie składniki. Teraz musimy wziąć naszą listę wypunktowaną oraz dwa obrazki, którymi zamierzamy wypełnić tło zakładek, i połączyć te składniki w jedną całość, stosując odpowiednie style CSS.

Pierwszy etap zadania polega na utworzeniu deklaracji, dzięki którym lista nawigacyjna uzyska postać poziomą zamiast domyślnej postaci pionowej.

```
#nav {
    margin: 0;
    padding: 10px 0 0 46px;
    list-style: none;
    background: #FFCB2D;
}
#nav li {
    float: left;
    margin: 0 1px 0 0;
    padding: 0;
    font-family: "Lucida Grande", sans-serif;
    font-size: 80%;
}
```

Dzięki zastosowaniu właściwości `float` pozycje listy zostaną ułożone w linii poziomej. W tym miejscu możemy także zadeklarować żółty kolor tła dla paska, na którym zostaną wyświetlone zakładki. Deklaracja koloru tła jest bardzo ważna, ponieważ bez niej cały system (żółty pasek i umieszczone na nim zakładki) nie będzie skalowalny. Poszliśmy także krok do przodu i wyzerowaliśmy domyślne wielkości marginesów (`margin`) i dopełnienia (`padding`), dodając jednak każdej zakładce jednopikselowy margines prawy w celu uzyskania odpowiednich odstępów. Ustawiliśmy także wielkość czcionki na 80% podstawowej wielkości czcionki na stronie. Korzystając z wiedzy zdobytej w rozdziale 1., „Elastyczny tekst”, ustawiliśmy podstawową wielkość czcionki poprzez podanie słowa kluczowego `small` jako wartości właściwości `font-size` w deklaracji dla elementu `<body>`. Teraz możemy spać spokojnie, ponieważ mamy pewność, że użytkownicy przeglądarki IE/Win będą mogli dowolnie zmieniać wielkość tekstu wyświetlanego w zakładkach.

Na rysunku 2.6 przedstawiono aktualny wygląd tworzonego systemu nawigacyjnego.

Wiem, uzyskany efekt nie jest nawet bliski oryginalnego wyglądu. Ale wystarczy tylko kilka deklaracji i będziemy mogli otwierać szampana.

Rysunek 2.6. Na razie system nawigacji wydaje się dosyć niestaranny

PŁYWANIE NA RATUNEK

I pojawił się pierwszy problem, któremu musimy stawić czoła. Elementy `` ustawiliśmy jako pływające, dlatego zostały one wyjęte z normalnego układu dokumentu i nie wypełniają zewnętrznego elementu ``, w którym zdefiniowano kolor tła. Innymi słowy, gdy elementy wewnętrzne ustawione są jako elementy pływające, zewnętrzny element `` nie wie, jaką powinien mieć wysokość.

Istnieje proste rozwiązanie — wystarczy ustawić cały element `` jako pływający. Dzięki temu elementy `` będą mogły wypełnić przestrzeń elementu `` i rozciągnąć jego tło tak, by było za nimi widoczne. Ponieważ element pływający kurczy się bądź rozszerza akurat na tyle, ile potrzeba do wyświetlenia jego zawartości, dodamy również zdefiniowaną wartość szerokości elementu ``, wyobrażając sobie (na potrzeby tego przykładu), że znajduje się on wewnątrz układu strony o stałej szerokości o podobnym rozmiarze (rysunek 2.7).

Rysunek 2.7. Ustawienie elementu `` — oraz wszystkich pozycji listy — jako elementów pływających powoduje rozciągnięcie tła listy i wyświetlenie go za wszystkimi jej pozycjami



UWAGA

Choć dla elementów `#nav` zadeklarowaliśmy szerokość 720 pikseli, w celu wcięcia zakładek przypisujemy im również dopełnienie lewe o wielkości 46 pikseli. Ponieważ dopełnienie dodawane jest do szerokości elementu, całkowita szerokość menu nawigacyjnego wynosi 766 pikseli.

```
#nav {
    float: left;
    width: 100%;
    margin: 0;
    padding: 10px 0 0 46px;
    list-style: none;
    background: #FFCB2D;
}
#nav li {
    float: left;
    margin: 0 1px 0 0;
    padding: 0;
    font-family: "Lucida Grande", sans-serif;
    font-size: 80%;
}
}
```

Przykład wygląda już trochę lepiej. Mamy teraz pewność, że bez względu na wielkość listy żółte tło będzie powiększać się wraz z wyświet-

tlaną treścią. Opisana tu metoda to wygodny sposób na uniknięcie wprowadzania dodatkowych elementów `<div>`, które zapewniłyby prawidłowe wyświetlanie tła. W ten sposób nie musimy także definiować wysokości (`<height>`) listy — czego nie należy robić, jeśli chce się zachować elastyczność projektu.

NADANIE ZAKŁADKOM KSZTAŁTU

Następnie definiujemy style dla odnośników — dodajemy dopełnienie, obramowanie i kolor tła:

```
#nav a {
    float: left;
    display: block;
    margin: 0;
    padding: 4px 8px;
    color: #333;
    text-decoration: none;
    border: 1px solid #9B8748;
    border-bottom: none;
    background: #F9E9A9;
}
```

Zadecydowaliśmy, że odnośniki w zakładkach będą wyświetlane jako elementy blokowe (`display: block;`), dzięki czemu użytkownik będzie mógł kliknąć w dowolnym miejscu zakładki. Elementy blokowe domyślnie wyświetlane są w osobnych wierszach, dlatego korzystając z właściwości `float`, ustawiamy wszystkie zakładki w jednej linii.

Następnie zwiększamy dopełnienie odnośników, zmieniamy ich kolor, wyłączamy dekorację tekstu oraz dodajemy obramowanie na wszystkich bokach poza dolnym. Dla elementów `<a>` ustawiamy także dopełnienie, ponieważ w ten sposób zwiększamy obszar, który można kliknąć (kliknąć będzie można całą zakładkę, a nie tylko jej tekst).

Zakładki zaczynają zyskiwać interesujący kształt dzięki dodaniu tylko tych kilku deklaracji CSS (rysunek 2.8).



Rysunek 2.8. Ustawienie dopełnienia w elemencie `<a>` umożliwia kliknięcie całego obszaru zakładki, co znacznie ułatwia nawigację w witrynie



WSKAZÓWKI

Gdy element o identyfikatorze `#nav` ustawimy jako element pływający, musimy zapewnić, by kolejne elementy strony nie były wyświetlane obok elementów pływających. Dla kolejnego elementu strony (na przykład poziomego wiersza lub bloku treści) musimy zadeklarować regułę `clear: left`, by był on wyświetlany *pod* nawigacją.

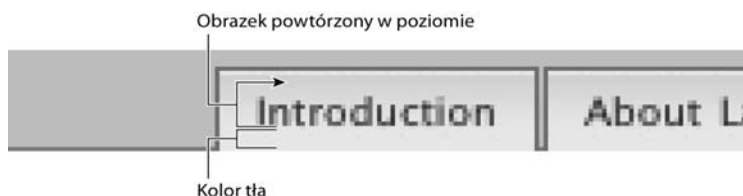
WYRÓWNANIE OBRAZKÓW TŁA

Kolejne zadanie związane jest z utworzonymi wcześniej obrazkami. Chcemy wyświetlić je w taki sposób, by były powielane w poziomie w tle zakładek:

```
#nav a {
  float: left;
  display: block;
  margin: 0;
  padding: 4px 8px;
  color: #333;
  text-decoration: none;
  border: 1px solid #9B8748;
  border-bottom: none;
  background: #F9E9A9 url(img/off_bg.gif) repeat-x
  top left;
}
```

Użyliśmy skrótowej metody definiowania koloru i obrazka tła w jednej regule. Zdefiniowany kolor tła (#F9E9A9) będzie prześwitywał w przezroczystym obszarze obrazka. Kolor ten będzie widoczny na dole obrazka, w miejscu, w którym w programie graficznym ustawiliśmy przezroczystość.

Wyrównaliśmy obrazek do górnego brzegu zakładki i przy użyciu atrybutu `repeat-x` ustawiliśmy jego powielanie w poziomie (rysunek 2.9). W miejscu, w którym kończy się obrazek (włącznie z obszarem przezroczystym), rozpoczyna się tło zdefiniowane jako jednolity kolor.

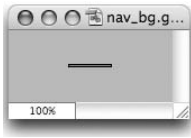


Rysunek 2.9. Umieszczenie w tle obrazka i powielenie go w poziomie wprowadzi efekt pionowej zmiany koloru oraz podświetlenia na górze zakładki. Obrazek jest automatycznie powielany w tle, dlatego zmiana szerokości lub wysokości zakładki nie popsuje wyglądu projektu

Pozostało nam dodanie stylów dla zaznaczonej zakładki oraz dla zakładki, nad którą aktualnie znajduje się kursor myszy (które w naszym przykładzie będą wyglądać tak samo). Jednak najpierw wyświetlimy obramowanie występujące na dole niewybranych zakładek, tworząc wrażenie, że wybrana zakładka jest częścią znajdującej się poniżej białej strony.

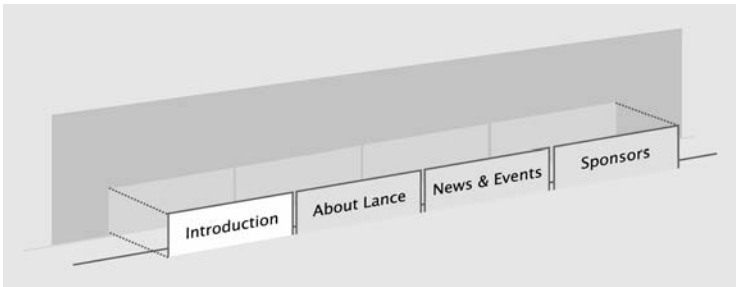
Z mojego doświadczenia wynika, że najłatwiejszym i najlepszym sposobem utworzenia obramowania, które czasami ma być zasłonięte przez inny element, jest utworzenie małego obrazka o wysokości odpowiadającej grubości obramowania, który będzie wielokrotnie powielany w tle elementu, tak by uzyskać długą linię obramowania. W tym miejscu zastosujemy trzeci obrazek, jednak jest on tak mały, że praktycznie się nie liczy (rysunek 2.10).

Obrazek GIF o wymiarach 1×37 pikseli będzie powielany w poziomie wzdłuż dolnej krawędzi zbioru zakładek. Szerokość obrazka nie ma w tym momencie żadnego znaczenia, ponieważ jest on powielany w poziomie. Sam nie mam zielonego pojęcia, dlaczego utworzyłem obrazek o takiej właśnie szerokości.



Rysunek 2.10. Obrazek o wysokości jednego piksela użyty jako dolne obramowanie zakładek. Gdy zakładka staje się aktywna, zasłania obramowanie, dzięki czemu tło wybranej zakładki zlewa się z tłem białej strony znajdującej się poniżej

Na rysunku 2.11 przedstawiono kolejność wyświetlania elementów nawigacji. Najpierw wyświetlane jest tło, nad nim wyświetlane jest obramowanie dolne o grubości jednego piksela, a na wierzchu wyświetlane są zakładki uzupełniające cały projekt.



Rysunek 2.11. Trójwymiarowy obraz, na którym przedstawiono, w jaki sposób można użyć obrazka tła do utworzenia dolnego obramowania paska nawigacyjnego

WSKAZÓWKI

Jeśli w tle wyświetlamy obrazki o większych rozmiarach, zwiększa się wydajność wyświetlania ich w przeglądarce Internet Explorer w systemie Windows. Przeglądarka ta działa czasem ospale przy wyświetlaniu i powielaniu w tle bardzo małych obrazków.

DODANIE DOLNEGO OBRAMOWANIA

Do wcześniejszej deklaracji stylów CSS dla elementu o identyfikatorze #nav dodajemy następujące odwołanie do obrazka tła:

```
#nav {
  float: left;
  width: 720px;
  margin: 0;
  padding: 10px 0 0 46px;
  list-style: none;
  background: #FFCB2D url(img/nav_bg.gif) repeat-x
  bottom left;
}
```

Deklarujemy, że obrazek ma być powielany w poziomie i wyrównany do dolnej krawędzi. Obrazek ma tylko jeden piksel wysokości — zadeklarowane żółte tło (#FFCB2D) będzie widoczne w pozostałej części paska nawigacyjnego (rysunek 2.12).



Rysunek 2.12. Poprzez użycie w rozwiązaniu obrazka linii jako tła paska nawigacyjnego coraz bardziej zbliżamy się do zamierzonego efektu

EFEKT PODMIANY

Dla stanów, w których zakładka jest wybrana lub znajduje się nad nią kursor myszy, spróbujemy odtworzyć stosowany w witrynie *Lance-Armstrong.com* szary kolor zakładki. Wystarczy tylko zamienić beżowy obrazek ustawiony jako domyślne tło odnośników. Obydwie sytuacje możemy ująć w jednej deklaracji CSS. Zaczniemy od utworzenia stylu dla sytuacji, w której kursor myszy znajdzie się nad zakładką:

```
#nav a:hover {
  color: #333;
  padding-bottom: 5px;
  border-color: #727377;
  background: #fff url(img/on_bg.gif) repeat-x top
  left;
}
```

Przyciemniliśmy kolor tekstu i obramowania, ustawiliśmy szarą wersję tła obrazka, która na dole przechodzi w kolor biały (#fff). Zwiększyliśmy także o jeden piksel dolne dopełnienie zakładki (z 4 na 5 pikseli). Dodatkowy piksel dopełnienia spowoduje, że wybrana zakładka oraz zakładki, nad którymi znajduje się kursor myszy, będą zasłaniać

obramowanie dolne zdefiniowane jako tło paska nawigacyjnego (obramowanie to także ma jeden piksel wysokości). Dzięki temu wybrana zakładka będzie wyglądać tak, jakby stanowiła jedną całość z pozostałą częścią strony (rysunek 2.13).



Rysunek 2.13. Zwiększając dolne dopełnienie odnośnika aktywnej zakładki z 4px na 5px, przykrywamy znajdujące się poniżej obramowanie, co daje wrażenie, że zakładka wyświetlana jest na pierwszym planie

WYRÓŻNIONA ZAKŁADKA

Skąd wiadomo, która zakładka ma być wyróżniona? W tym miejscu w grę wchodzi selektory elementów potomnych. Dodamy je do deklaracji określającej styl elementów, nad którymi znajduje się kursor myszy — tak by miały one ten sam styl:

```
#nav a:hover, body#intro #t-intro a
```

Selektor elementu potomnego daje możliwość zdefiniowania stylów dla konkretnego elementu w zależności od jego przodków (elementów zawierających). Podając elementy rozdzielone spacjami, możemy zawęzić listę elementów docelowych w zależności od tego, jakie miejsce zajmują one w drzewie dokumentu. Więcej informacji na temat selektorów można znaleźć w poświęconym im rozdziale specyfikacji CSS dostępnym pod adresem www.w3.org/TR/REC-CSS2/selector.html.

Jeśli zatem do elementu `<body>` strony dodamy atrybut `id` o wartości `intro`, na przykład:

```
<body id="intro">
```

to umieszczoną wyżej deklarację CSS można odczytać tak: „Na stronach, na których identyfikator elementu `body` ma wartość `intro`, zastosuj inne tło, ciemniejsze kolory itp.”. Jest to bardzo przydatny sposób na określenie stylów CSS dla sytuacji typu „jeśli jesteś na danej stronie”, dlatego jest on często wykorzystywany przy projektowaniu nawigacji. Definicję wyglądu zaznaczonej zakładki oraz zakładki, nad którą znajduje się kursor myszy, umieściliśmy w jednej deklaracji CSS.

Dlaczego rozwiązanie to jest kuloodporne

Wzięliśmy ładny projekt zakładek nawigacyjnych ze znanej witryny, rozbiliśmy go na części i utworzyliśmy bardzo podobny system. Nowy system zbudowany jest jednak przy użyciu znacznie mniejszej ilości kodu, kod ten można łatwo aktualizować, a do tego rozwiązanie to jest bardziej dostępne dla szerszej gamy przeglądarek, urządzeń i oprogramowania pomocniczego. I co najważniejsze — nowy system jest *elastyczny* bez względu na wielkość i ilość umieszczonego w nim tekstu.

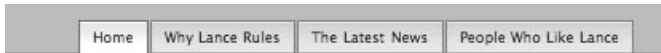
Ze względu na kroki, które podjęliśmy przy wyrównywaniu obrazów tła i przypisywaniu kolorów, cała zakładka ulega *skalowaniu*. Jeśli użytkownik zdecyduje się zwiększyć o kilka punktów rozmiar tekstu, by poprawić jego czytelność, cały projekt będzie się skalował wraz z tekstem (rysunek 2.14).



Rysunek 2.14. Wielkość zakładek można skalować w górę i w dół poprzez zmianę rozmiaru tekstu w przeglądarce

W ciągu kilku minut możemy także zmienić słowa wyświetlane na poszczególnych zakładkach oraz dodać lub usunąć zakładki. W tym celu edytujemy listę wypunktowaną, która nadaje elementom nawigacyjnym odpowiednią strukturę (rysunek 2.15). Wystarczy w kodzie zmienić tekst wewnątrz odpowiedniej pozycji listy:

```
<ul id="nav">
  <li id="t-intro"><a href="/">Home</a></li>
  <li id="t-about"><a href="about.html">Why Lance
    Rules</a></li>
  <li id="t-news"><a href="news.html">The Latest
    News</a></li>
  <li id="t-sponsors"><a href="sponsors.html">People
    Who Like Lance</a></li>
</ul>
```



Rysunek 2.15. Zmiana tekstu wyświetlanego w zakładkach zajmie nam tylko kilka sekund — wystarczy uaktualnić tekst w czytelnej liście wypunktowanej

Podobny przykład wykorzystujący em

W naszej rekonstrukcji zakładek ze strony Lance’a Armstronga podaliśmy dopełnienie wokół tekstu odnośników w pikselach. W części było nam to potrzebne do zwiększenia dopełnienia dolnego o jeden piksel, tak by nakładało się ono na jednopikselowe obramowanie dolne paska nawigacyjnego. Potrzebna była nam dokładność co do piksela.

Jeśli zlikwidujemy konieczność korzystania z tego jednopikselowego obramowania, możemy dopełnienia wokół tekstu odnośnika każdej zakładki zdefiniować w em. Dlaczego em? Jak zostało to pokazane w rozdziale 1., em to jednostka elastyczna, której wielkość uzależniona jest od aktualnej wielkości czcionki. Podając dopełnienie w em zamiast w pikselach, możemy zapewnić, że cała zakładka (a nie tylko znajdujący się w niej tekst) będzie skalowana proporcjonalnie do tekstu, jeśli jego rozmiar zostanie zmieniony przez użytkownika.

Na rysunku 2.16 zaprezentowano uproszczoną wersję poprzedniego przykładu z zakładkami, w których usunięte zostało jednopikselowe obramowanie znajdujące się na dole paska nawigacyjnego, a kolor tła zmienił się na szary.



Rysunek 2.16. Uproszczona wersja nawigacji bez obramowania dolnego

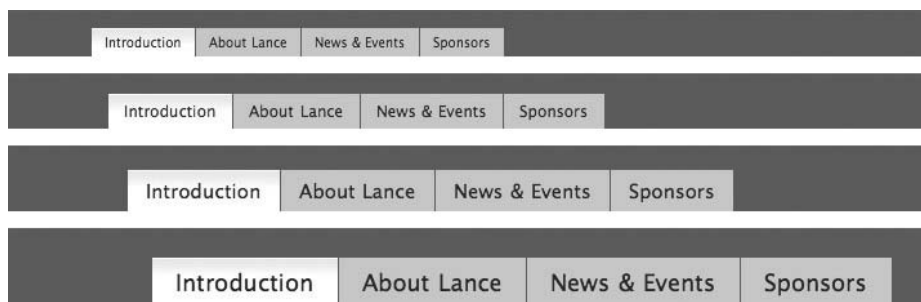
```
#nav {  
    float: left;  
    width: 50em;  
    margin: 0;  
    padding: 1em 0 0 5em;  
    list-style: none;  
    background: #666;  
}
```

Tym razem szerokość ustaliliśmy na 50 em, zakładając, że wielkość ta zmieści się wewnątrz układu strony o tym samym rozmiarze (więcej informacji o układach strony opartych na em znajduje się w rozdziale 8., „Płynny oraz elastyczny układ strony”). Również dopełnienie wokół zakładek podano w em — ma ono wartość 1 em na górze i 5 em po lewej stronie, dzięki czemu powstaje tam niewielkie wcięcie tekstu.

Zmienimy teraz również deklarację `#nav a` w taki sposób, by również wykorzystywała em (dodatkowo zmieniając kolor tła na jasnoszary i usuwając obramowanie znajdujące się wokół zakładek):

```
#nav a {
  float: left;
  display: block;
  margin: 0;
  padding: .5em 1em;
  color: #333;
  text-decoration: none;
  background: #ccc;
}
```

W deklaracji tej widać regułę `padding: .5em 1em;`, którą można przeczytać jako: „Na górze i dole każdego odnośnika znajduje się dopełnienie o wielkości pół em, natomiast po prawej i lewej stronie — jeden em”. Jeśli teraz zmienimy rozmiar tekstu, zauważymy, że tekst zakładek oraz odstęp między zakładkami skalują się w sposób proporcjonalny (rysunek 2.17). Dość sprytne, prawda? Zamiast więc wykorzystywać system, w którym tekst otacza zawsze taka sama liczba pikseli — bez względu na rozmiar tego tekstu — dzięki em uzyskaliśmy w pełni skalowalny system, który nie zmienia się, kiedy rozmiar czcionki się zwiększa lub zmniejsza.



Rysunek 2.17. Wersja zakładek oparta na em, w której zmieniamy rozmiar tekstu w celu pokazania, że proporcje pozostają bez zmian

Wspomniałem o tej wersji projektu przede wszystkim po to, by każdy zaczął sobie przyswajać możliwość zmiany pikseli na em dla dopełnienia, marginesów, wysokości wiersza i podobnych właściwości. Nie zawsze jest to możliwe (przykładem takiej sytuacji jest strona Lance'a Armstronga), ale w zależności od wymagań zastosowanie em może być korzystne i umożliwi nam zachowanie proporcji w całym projekcie strony — bez względu na rozmiar tekstu. Warto spróbować!

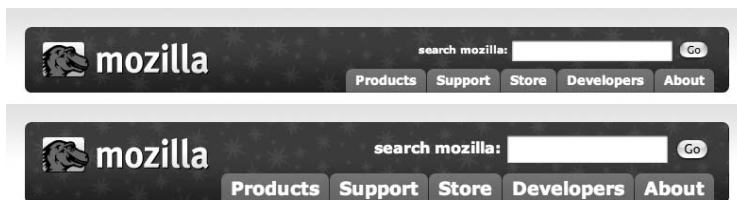
Dodatkowe przykłady

Pomysł przedstawiony w tym rozdziale można rozwinąć. W tym podrozdziale omawiamy inne projekty, które są tak samo elastyczne, jak zakładki nawigacyjne utworzone w tym rozdziale.

MOZILLA.ORG

www.mozilla.org

Niedawno przeprowadzono zmianę wyglądu witryny *Mozilla.org* i wprowadzono zakładki nawigacyjne z zaokrąglonymi narożnikami. Zakładki te oczywiście skalują się wraz z umieszczonym w nich tekstem (rysunek 2.18).



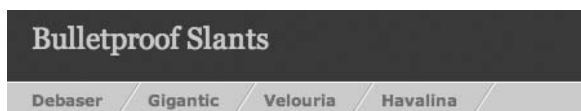
Rysunek 2.18. Zaokrąglone narożniki można w prosty sposób uzyskać przy użyciu techniki „Sliding Doors”

Zespół projektantów witryny Mozilla zastosował technikę opracowaną przez Douglasa Bowmana, znaną pod nazwą „Sliding Doors” (po polsku: „rozsuwane drzwi”) — www.alistapart.com/articles/slidingdoors/. Bowman wymyślił sprytny sposób stosowania dwóch obrazków tła, które rozsuwają się, gdy rozszerza się zawarta między nimi treść. Dwa oddzielne obrazki zaokrąglonych narożników (lub innych form nieprostokątnych) wyrównujemy do krawędzi zakładek, a cały projekt pozostaje elastyczny.

SKOSY

www.simplebits.com/bits/bulletproof_slants.html

Przy realizacji nowego zlecenia dla klienta zaistniała potrzeba zaprojektowania elastycznej nawigacji, w której kolejne elementy byłyby oddzielone skośnymi liniami. Opracowane rozwiązanie wymaga zastosowania tylko jednego obrazka w tle, który będzie wyświetlany w mniejszej lub większej części w zależności od wielkości tekstu umieszczonego w pasku nawigacyjnym (rysunek 2.19).



Rysunek 2.19. Gdy w tle umieścimy za duży obrazek ze skośną linią, to projekt będzie dobrze wyglądał nawet po zwiększeniu rozmiaru tekstu

Jeśli umieszczony w tle obrazek skośnej linii będzie większy, niż jest to konieczne przy domyślnej wielkości tekstu, to po zwiększeniu rozmiaru tekstu wyświetlona zostanie większa część obrazka (rysunek 2.20).



Rysunek 2.20. Powiększenie ukośnego separatora — wyświetlana jest tylko niezbędna część obrazka

WYSZUKIWANIE W WITRYNIE ESPN.COM

www.espn.com

Zakładki stosowane do przełączania filtrów na wykorzystywanym do niedawna projekcie strony wyszukiwania w witrynie *ESPN.com* zostały utworzone przy użyciu metody podobnej do opisywanej w tym rozdziale. Jako tło posłużył duży obrazek z gradientowym wypełnieniem, którego jakaś część jest wyświetlana w zależności od wielkości tekstu.

Na rysunku 2.21 widoczne są zakładki wyświetlone przy dwóch różnych wielkościach czcionki.



Rysunek 2.21. Zakładki na stronie wyszukiwania w poprzedniej wersji witryny ESPN skalują się wraz z wielkością tekstu i jego długością

Aby zrozumieć, w jaki sposób efekt został osiągnięty, przyjrzyjmy się zakładce w stanie nieaktywnym. Na lewym i górnym brzegu obrazka widoczne jest podświetlenie o szerokości jednego piksela. Podświetlenie to widoczne jest tylko na tych dwóch krawędziach, dlatego jako tło wystarczy nam jeden obrazek, który jest wystarczająco duży, by pomieścić tekst napisany nawet bardzo dużą czcionką.

Na rysunku 2.22 widoczna jest zakładka wyświetlona przy domyślnym rozmiarze tekstu — odsłonięta jest wtedy tylko górna lewa część obrazka tła. Natomiast na rysunku 2.23 widoczna jest ta sama zakładka przy większym rozmiarze czcionki — odsłonięta część obrazka tła jest dużo większa. Wyrównując obrazek do górnego lewego rogu elementu <a> w zakładce, możemy pokazać efekty podświetlenia i gradientu bez względu na ilość lub wielkość tekstu umieszczonego wewnątrz zakładki.



Rysunek 2.22. Gdy zakładka wyświetlana jest przy domyślnej wielkości czcionki, widoczna jest tylko górna lewa część obrazka

Zmienna szerokość zakładki była niezbędnym elementem projektu, ponieważ liczba wyników jest wyświetlana obok etykiety zakładki (liczba umieszczona w nawiasie). Ze względu na różną długość liczby wyników zakładce należało zostawić miejsce, tak by mogła się ona rozszerzać lub kurczyć, zachowując przy tym dobry wygląd.



Rysunek 2.23. Zwiększenie wielkości czcionki powoduje odsłonięcie większego obszaru obrazka tła, na górnym i lewym brzegu zakładki nadal widoczne jest delikatne podświetlenie

Podsumowanie

Chęć opracowania nowej metody tworzenia zakładek nawigacyjnych w miejsce powszechnie stosowanej metody wykorzystującej duże ilości grafiki pozwoliła nam zidentyfikować kilka wad stosowania tej ostatniej. Jednocześnie odkryliśmy zalety stosowania prostej listy wypunktowanej i przypisania jej pomysłowych stylów CSS. Zalety te to nie tylko możliwość zmiany wielkości nawigacji wraz ze zmianą wielkości tekstu, ale także możliwość umieszczenia *bardzo różnych ilości* treści w zakładkach.

Poniżej znajduje się kilka wskazówek, o których warto pamiętać przy projektowaniu nawigacji.

- Stosowanie nawigacji opartej na grafice oznacza, że użytkownicy z wadami wzroku nie mogą dostosować wielkości tekstu do swoich potrzeb.
- Prosty kod (na przykład lista wypunktowana) zapewnia lepszą dostępność treści dla szerszego zakresu przeglądarek, urządzeń i oprogramowania dodatkowego.
- Nawigację tekstową dużo prościej się aktualizuje i redaguje — nie trzeba tworzyć nowych rysunków przy wprowadzaniu każdej zmiany.
- Pomysłowe rozmieszczenie obrazków tła może wprowadzić do projektu nawigacji ciekawe efekty, nie pogarszając przy tym jego elastyczności.
- Warto spróbować wykorzystać em do definiowania marginesów, dopełnienia czy wysokości wierszy, tak by cały projekt strony (a nie tylko tekst) był w pełni skalowalny bez względu na rozmiar tekstu.

- Czasami użycie obrazków w nawigacji jest koniecznością. Może to być związane ze specjalną czcionką stosowaną we wszystkich firmowych materiałach lub z ograniczoną przestrzenią do zagospodarowania. Zastosowanie graficznej formy nawigacji to nie koniec świata.

Ostatni punkt w powyższej liście jest bardzo ważny. W świecie rzeczywistym wymagania związane z czcionkami, szerokością lub wysokością paska nawigacyjnego mogą być narzucone z zewnątrz i być niezależne od projektanta witryny. W takiej sytuacji należy uciekać jak najdalej. Nie, niezupełnie — teraz jesteśmy przygotowani, by zaproponować w miejsce takiego projektu bardziej elastyczne rozwiązanie. Nie ma złych odpowiedzi, istnieją tylko rozwiązania działające lepiej od innych, biorąc pod uwagę daną sytuację.

Opracowaliśmy projekt elastycznej nawigacji, teraz możemy przejść do innych często spotykanych komponentów stron internetowych i spróbować je ulepszyć.