

## 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 poprawić elastyczność z wykorzystaniem XHTML-a i CSS

Autor: Dan Cederholm

Tłumaczenie: Agata Bulandra

ISBN: 83-246-0316-6

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

Format: B5, stron: 280



### Poznaj tajniki projektowania doskonałych stron internetowych

- Wykorzystaj możliwości języka XHTML
- Zastosuj standardy CSS
- Zapewnij poprawne działanie strony w różnych przeglądarkach

W coraz dynamiczniej rozwijającym się internecie można znaleźć setki tysięcy witryn WWW. Wiele z nich to prawdziwe perełki – zarówno pod względem projektowym, jak i merytorycznym. Ile razy jednak, otwierając ulubioną witrynę WWW w innej przeglądarce niż dotychczas, widzimy coś zupełnie odmiennego? Jak często projektując witrynę WWW, borykamy się z różnymi interpretacjami standardów w różnych przeglądarkach? Ile czasu trwa wzbogacanie strony o nowe elementy czy też modernizacja układu graficznego? Ilu projektantów stron WWW zastanawia się, jak poprawić elastyczność i uniwersalność swoich dzieł?

Dzięki wiadomościom zawartym w książce „Kuloodporne strony internetowe. Jak poprawić elastyczność z wykorzystaniem XHTML-a i CSS” dowiesz się, jak tworzyć niemal doskonałe witryny WWW. Doskonałe – ponieważ wyglądają równie dobrze w każdej przeglądarce i przy każdej rozdzielczości ekranu. Poza tym są łatwe w nawigacji i proste w rozbudowie. Nauczysz się korzystać z najnowszych standardów – XHTML i CSS, odpowiednio dobierać czcionki i ich rozmiary, pozycjonować elementy strony i budować uniwersalne mechanizmy nawigacyjne. Poznasz sposoby tworzenia kuloodpornych stron WWW.

- Dobór rozmiaru czcionek
- Skalowalne menu
- Rozmieszczanie składników strony
- Definiowanie stylów dla ramek
- Tworzenie uniwersalnych tabel
- Budowanie skalującego się układu strony

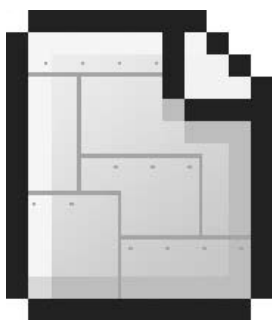
Jeśli wykorzystasz wiadomości z tej książki, tworzone przez Ciebie strony WWW będą naprawdę uniwersalne.



# Spis treści

<b>Wprowadzenie</b> .....	<b>7</b>
<b>Rozdział 1. Elastyczny tekst</b> .....	<b>13</b>
Częste rozwiązanie .....	15
Wszystkie możliwości .....	17
Rozwiązanie kuloodporne .....	19
Dlaczego rozwiązanie to jest kuloodporne .....	25
Elastyczna baza — i co dalej? .....	25
Stosowanie słów kluczowych i wartości procentowych .....	30
Podsumowanie .....	35
<b>Rozdział 2. Skalowalna nawigacja</b> .....	<b>37</b>
Częste rozwiązanie .....	38
Dlaczego rozwiązanie to nie jest kuloodporne .....	40
Rozwiązanie kuloodporne .....	42
Dlaczego rozwiązanie to jest kuloodporne .....	52
Dodatkowe przykłady .....	53
Podsumowanie .....	56
<b>Rozdział 3. Elastyczne wiersze</b> .....	<b>59</b>
Częste rozwiązanie .....	60
Dlaczego rozwiązanie to nie jest kuloodporne .....	62
Rozwiązanie kuloodporne .....	64
Dlaczego rozwiązanie to jest kuloodporne .....	77
Inny przykład rozciągania .....	78
Podsumowanie .....	83
<b>Rozdział 4. Pomysłowe rozmieszczanie elementów</b> .....	<b>85</b>
Częste rozwiązanie .....	87
Dlaczego rozwiązanie to nie jest kuloodporne .....	88
Rozwiązanie kuloodporne .....	89
Dlaczego rozwiązanie to jest kuloodporne .....	119
Podsumowanie .....	119

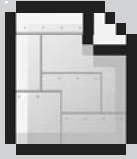
<b>Rozdział 5. Niezniszczalne ramki .....</b>	<b>121</b>
Częste rozwiązanie .....	122
Dlaczego rozwiązanie to nie jest kuloodporne .....	124
Rozwiązanie kuloodporne .....	125
Dlaczego rozwiązanie to jest kuloodporne .....	131
Inne techniki tworzenia zaokrąglonych krawędzi .....	132
Podsumowanie .....	140
<b>Rozdział 6. Brak rysunków? Brak CSS? Brak problemów .....</b>	<b>143</b>
Częste rozwiązanie .....	144
Dlaczego rozwiązanie to nie jest kuloodporne .....	147
Rozwiązanie kuloodporne .....	148
Dlaczego rozwiązanie to jest kuloodporne .....	150
Ze stylami lub bez .....	152
Kuloodporne narzędzia .....	157
Podsumowanie .....	164
<b>Rozdział 7. Konwersja tabel .....</b>	<b>167</b>
Częste rozwiązanie .....	168
Dlaczego rozwiązanie to nie jest kuloodporne .....	170
Rozwiązanie kuloodporne .....	171
Dlaczego rozwiązanie to jest kuloodporne .....	191
Podsumowanie .....	192
<b>Rozdział 8. Płynny układ stron .....</b>	<b>195</b>
Częste rozwiązanie .....	196
Dlaczego rozwiązanie to nie jest kuloodporne .....	199
Rozwiązanie kuloodporne .....	201
Dlaczego rozwiązanie to jest kuloodporne .....	233
Podsumowanie .....	234
<b>Rozdział 9. Łączenie w całość .....</b>	<b>235</b>
Cel .....	236
Dlaczego rozwiązanie to jest kuloodporne .....	237
Konstrukcja .....	242
Modyfikacja stylów ze względu na przeglądarkę IE .....	268
Podsumowanie .....	272
<b>Skorowidz .....</b>	<b>273</b>



1

---

ELASTYCZNY  
TEKST



## Rozmiar tekstu ustal przy użyciu słów kluczowych oraz wartości procentowych, dzięki czemu użytkownicy będą mieli nad nim kontrolę.

Niewiele zagadnień dotyczących projektowania stron internetowych jest tak kontrowersyjnych, jak *zmiana rozmiaru tekstu*. Temat ten zawsze był kłopotliwy dla początkujących, wywoływał dyskusje oraz dzielił narody. Może trochę wyolbrzymiam problem, ale faktycznie wiele osób może się do niego niechętnie odnosić.

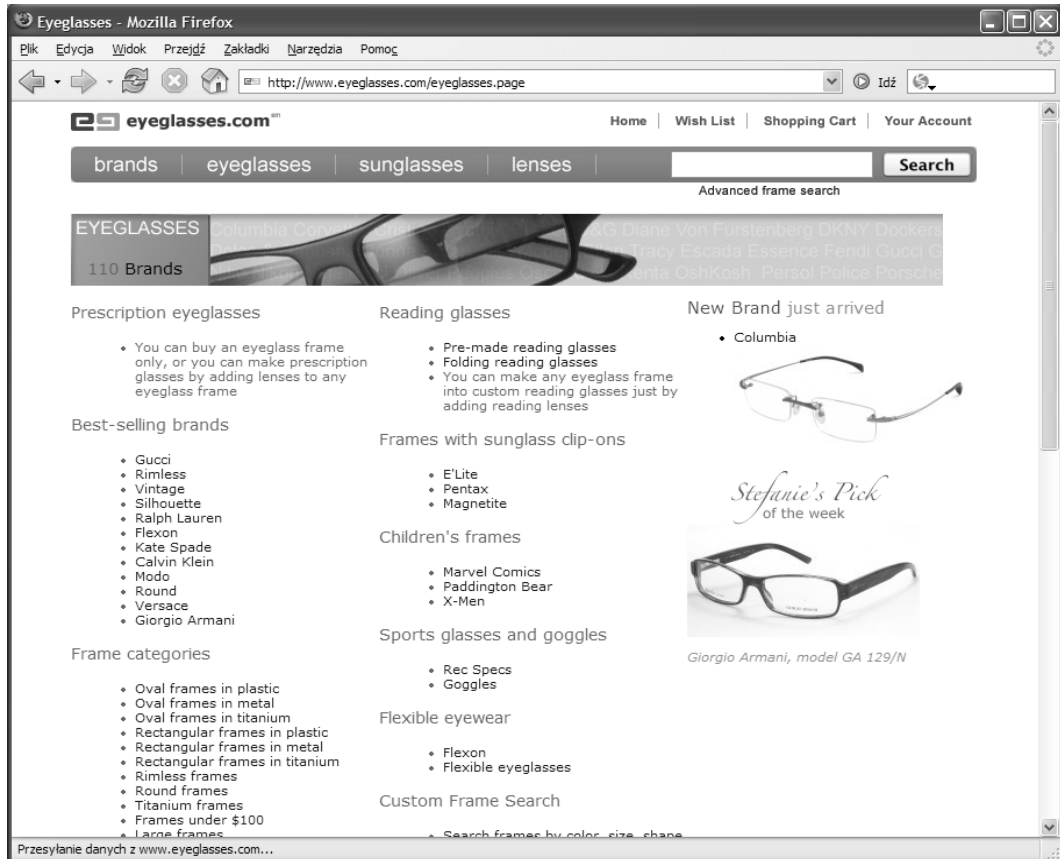
Nie jesteśmy w stanie uwolnić świata od problemu zmiany wielkości wyświetlanego na stronie tekstu. W tym rozdziale pokażę jednak strategię definiowania rozmiaru tekstu, która jest *elastyczna* i łatwa do zaimplementowania, a jednocześnie pozwala projektantom zachować pełną kontrolę nad wyglądem stron.

Wszystkie przykłady w tej książce łączy jedna wspólna cecha — elastyczne dopasowanie się projektu do tekstu każdej wielkości. Umożliwiając *użytkownikowi* kontrolę nad rozmiarem tekstu, zwiększamy czytelność strony dla wszystkich internautów. Prawdziwe wyzwanie stanowi zbudowanie strony internetowej, w której ta elastyczność połączona jest z pomysłowym projektem graficznym. Poprzez zapoznanie się z wszystkimi zamieszczonymi w tej książce przykładami, zyskamy solidne podstawy, by sprostać temu wyzwaniu.

Przyglądając się często stosowanemu rozwiązaniu projektowemu, dowiemy się, dlaczego tak ważna jest możliwość zmiany wielkości tekstu oraz dlaczego zaprezentowane przykładowe rozwiązanie nie jest elastyczne.


## Częste rozwiązanie

Często stosowany sposób definiowania wielkości tekstu prześledzimy na przykładzie witryny *eyeglasses.com* — z pewnością odwiedzające tę stronę osoby z wadami wzroku cieszyłyby się z możliwości sterowania wielkością tekstu oraz z ogólnej czytelności strony (rysunek 1.1).



**Rysunek 1.1.** Często stosowany sposób definiowania rozmiaru tekstu na przykładzie witryny *eyeglasses.com*

Witryna *eyeglasses.com* jest przykładem bardzo dobrego wykorzystania kaskadowych arkuszy stylów. Tak jak w wielu innych ładnie zaprojektowanych stronach internetowych, podstawowa wielkość tekstu określona jest w jednostkach pikselowych poprzez zdefiniowanie wartości dla właściwości `font-size` w deklaracji stylu dla elementu `<body>`:



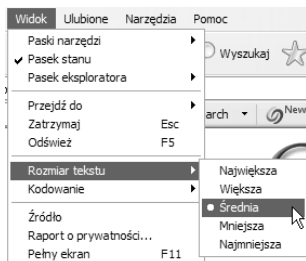
Chcieliśmy zaznaczyć, że nie uważamy strony *eyeglasses.com* za zły projekt. Jest to ładnie zaprojektowana witryna internetowa, utworzona przy użyciu strukturalnego kodu oraz stylów CSS. Właśnie dlatego wybraliśmy ją jako nasz przykład. Tak naprawdę potrzebny był nam pretekst do omówienia różnych sposobów definiowania wielkości tekstu oraz wad i zalet najczęściej stosowanej metody.

```
body {
    font-size: 11px;
}
```

Poprzez ustawienie wielkości czcionki dla elementu `<body>` (w naszym przykładzie 11px) projektanci strony definiują, że wyświetlony na stronie tekst będzie miał wysokość 11 pikseli (chyba że dla jakiegoś fragmentu zostanie określona inna wartość). Zaletą określania rozmiaru tekstu w pikselach jest to, że wielkość czcionki pozostaje bez zmian niezależnie od tego, w jakiej przeglądarce lub urządzeniu wyświetlany jest tekst. Stosowanie wartości pikselowych stało się popularne właśnie ze względu na ich konsekwentną, przewidywalną wielkość. Projektant, chcąc precyzyjnie zdefiniować wielkość tekstu wyświetlanego na stronie, zawsze chętniej użyje wartości pikselowych niż innych jednostek. Jednak istnieje pewien mały problem.

## DLACZEGO TO ROZWIĄZANIE NIE JEST KULOODPORNE

Definiowanie rozmiaru tekstu w wartościach pikselowych daje projektantowi kontrolę nad wielkością czcionki. Niestety, wartości pikselowe stwarzają kłopoty użytkownikom najbardziej popularnej przeglądarki internetowej (Internet Explorer w systemie Windows — IE/Win).



**Rysunek 1.2.** Dzięki menu Widok/Rozmiar tekstu w IE6/Win użytkownik może dostosować rozmiar tekstu wyświetlanego na stronie

Przeglądarki internetowe zwykle umożliwiają użytkownikowi kontrolowanie rozmiaru tekstu poprzez zmianę rozmiaru określonego przez projektanta. Funkcja ta jest bardzo przydatna dla osób z wadami wzroku. Gdy użytkownik chce poprawić czytelność strony, wystarczy że z menu przeglądarki wybierze większy rozmiar tekstu (rysunek 1.2).

Czyż nie jest to wspaniałe? Istnieje jednak pewien haczyk. Jeśli stosowane są wartości pikselowe, użytkownicy przeglądarki Internet Explorer nie mają możliwości zmiany rozmiaru tekstu na inny niż określony przez projektanta. Tak więc korzystanie z wartości pikselowych daje projektantowi największą precyzję przy ustalaniu wielkości czcionki, ale jednocześnie uniemożliwia użytkownikom przeglądarki IE/Win dostosowywanie rozmiaru tekstu do własnych potrzeb. Użytkownicy innych przeglądarek mogą zmieniać rozmiar czcionki bez względu na to, w jaki sposób jest on zdefiniowany.

Możliwość dostosowania rozmiaru tekstu jest *niezbędna* dla osób z wadami wzroku. Będzie ona także mile widziana przez wszystkich chcących zwiększyć czytelność stron internetowych, na których

domyślnie zastosowano małe czcionki. W naturze wszystkich projektantów leży chęć pełnej kontroli nad wyglądem całej witryny. Rezygnując tylko z *odrobiny* tej kontroli w zakresie definiowania rozmiaru tekstu, poprawiamy odbiór witryny przez internautów oraz zwiększamy jej dostępność.

Wróćmy do braku możliwości zmiany wielkości tekstu, którego rozmiar czcionki został zdefiniowany przy użyciu wartości pikselowych. Być może twórcy przeglądarki chcieli pozostać wierni idei, że piksel to piksel —jednostka, która nigdy się nie zmienia, nawet wtedy, gdy użytkownik by tego chciał (a przecież ma do tego prawo). Mimo to wszystkie inne przeglądarki działają w sposób poprawny i umożliwiają użytkownikowi zmianę wielkości tekstu, którego rozmiar został określony w jednostkach pikselowych.

Wszystko byłoby prostsze, gdyby projektanci mogli definiować rozmiar czcionki w wartościach pikselowych, a użytkownicy mogli dostosować go zgodnie ze swoimi wymaganiami. Jednak nie ma możliwości skalowania tekstu w IE/Win, dlatego rozwiązanie to nie jest optymalne. Poznajmy więc inne sposoby definiowania rozmiaru tekstu. Na koniec ustalimy rozwiązanie, w którym rezygnujemy z precyzji zapewnianej przez wartości pikselowe, ale które będzie posiadało ważną zaletę — skalowalność we wszystkich przeglądarkach.



## UWAGA

Z technicznego punktu widzenia piksel to względna jednostka miary, która zależy od rozdzielczości urządzenia, na którym jest wyświetlana bądź drukowana.

Na przykład rzeczywista wielkość piksela wyświetlanego na monitorze różni się od wielkości piksela wydrukowanego.

Więcej informacji na ten temat można znaleźć na stronie [www.w3.org/](http://www.w3.org/)

➡ [TR/REC-CSS2/syntax](#).

➡ [html#pixel-units](#).

## Wszystkie możliwości

Oprócz stosowania wartości pikselowych istnieje kilkanaście innych metod definiowania rozmiaru tekstu przy użyciu stylów CSS — warto znać wszystkie.

Jednostki, które stosujemy, podając wartość dla właściwości `font-size` w arkuszu CSS, można podzielić na cztery grupy.

### JEDNOSTKI DŁUGOŚCI

Jednostki długości mogą być *względne* lub *bezwzględne*. Do względnych jednostek zaliczamy:

- `em` — określenie wielkości względem rozmiaru czcionki elementu nadrzędnego (rodzica),
- `ex` — określenie wielkości względem wysokości litery `x` danej czcionki,
- `px` — określenie wielkości względem rozdzielczości urządzenia (najczęściej stosowana jednostka).



Bezwzględne jednostki długości używane są przede wszystkim w druku, jednak stosuje się je także wtedy, gdy znane są fizyczne wymiary i właściwości przeglądarki lub urządzenia:

- in — cale (ang. *inches*),
- cm — centymetry,
- mm — milimetry,
- pt — punkty,
- pc — pica.

## SŁOWA KLUCZOWE — WZGLĘDNE

Jeśli chcemy określić względny rozmiar tekstu przy użyciu słów kluczowych, do wyboru mamy tylko dwa słowa:

- larger,
- smaller.

W ten sposób określimy wielkość czcionki względem aktualnej wielkości czcionki elementu rodzica. Konsorcjum W3C (ang. *World Wide Web Consortium*) — organizacja zajmująca się standaryzacją internetu — wyjaśnia, że jeśli wielkość czcionki elementu rodzica określono za pomocą słowa kluczowego medium, użycie słowa kluczowego larger zwiększy rozmiar czcionki bieżącego elementu do wielkości large. Więcej informacji na ten temat można znaleźć pod adresem [www.w3.org/TR/CSS21/fonts.html#font-size-props](http://www.w3.org/TR/CSS21/fonts.html#font-size-props).

O tych wartościach należy myśleć jak o odpowiednikach elementów `<small>` i `<big>` z języka HTML. Określają one zmianę wielkości tekstu względem wielkości zdefiniowanej gdzieś wyżej w drzewie dokumentu.

## PROCENTY

Wartości procentowe określają rozmiar tekstu względem rozmiaru tekstu elementu zawierającego bieżący element (elementu rodzica). Użycie wartości 120% zwiększy uprzednio zdefiniowaną wielkość o 20%, bez względu na to, jaka jednostka była stosowana (jeżeli jakkolwiek była stosowana). W naszym kuloodpornym rozwiązaniu do określenia wielkości tekstu wykorzystamy procenty w połączeniu ze słowami kluczowymi.

## SŁOWA KLUCZOWE — BEZWZGLĘDNE

Słowa kluczowe definiujące rozmiar w sposób bezwzględny korzystają z tabeli rozmiarów czcionek przeglądarki lub urządzenia. Oto siedem możliwych wielkości:

- xx-small,
- x-small,
- small,
- medium,
- large,
- x-large,
- xx-large.

Konsorcjum W3C sugeruje twórcom przeglądarek, by współczynnik skalowania pomiędzy poszczególnymi słowami kluczowymi wynosił 1,5 oraz zaleca, by skala pozostała bez zmian, bez względu na fizyczną wielkość tekstu po wyświetleniu. Czyli mówiąc prostym językiem — wielkość `large` jest półtora raza większa od wartości `medium`, a wielkość `small` jest półtora raza mniejsza od wielkości `medium`.

Prosta składnia to tylko jedna z wielu zalet stosowania bezwzględnych słów kluczowych, które bardziej szczegółowo omówimy w następnym podrozdziale.

Znamy już wszystkie jednostki, które można stosować we właściwości `font-size`, dlatego możemy przejść do omówienia strategii, która zapewni elastyczną podstawę i da użytkownikom możliwość sterowania wielkością tekstu.

## Rozwiązanie kuloodporne

Podzielę się teraz rozwiązaniem, które skutecznie stosuję we własnych projektach, łącząc bezwzględne słowa kluczowe z wartościami procentowymi. Użycie słów kluczowych przy definiowaniu wielkości tekstu we właściwości `font-size` rozwiązuje problem pokazany wcześniej na przykładzie witryny *eyeglasses.com* (dotyczący milionów stron). Gdy nie są stosowane wartości pikselowe, wszystkie przeglądarki i urządzenia potrafią zmienić rozmiar wyświetlanego na stronie tekstu. Przyjrzyjmy się, w jaki sposób należy stosować słowa kluczowe.

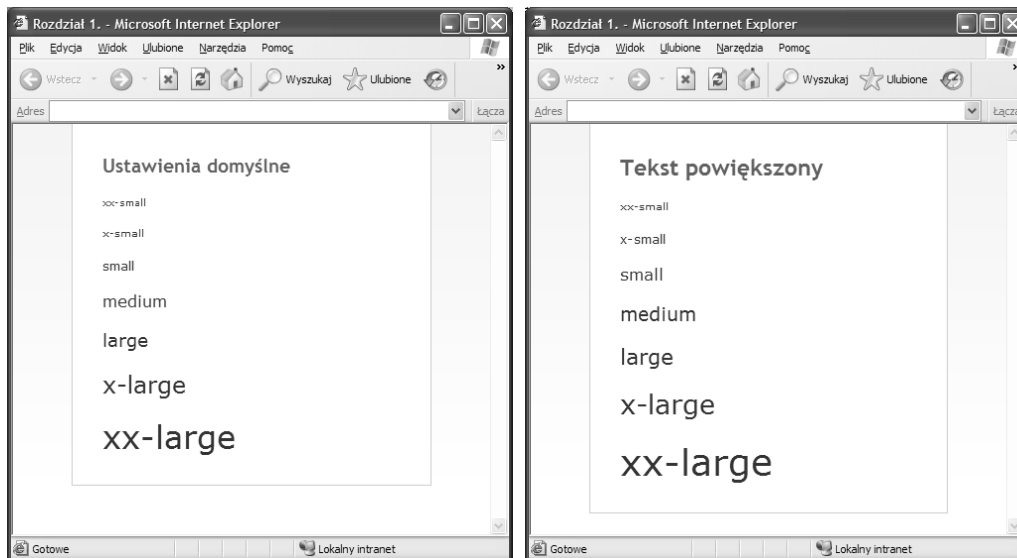
## SŁOWA KLUCZOWE — WYJAŚNIENIE

Jak już wcześniej wspomniałem, rozmiar czcionki możemy określić przy użyciu siedmiu słów kluczowych: `xx-small`, `x-small`, `small`, `medium`, `large`, `x-large`, `xx-large`. Oto przykład, w jaki sposób możemy użyć jednej z tych wartości, aby ustawić wielkość czcionki dla elementu `<body>`:

```
body {
    font-size: small;
}
```

Słowo kluczowe definiuje rozmiar tekstu w zależności od ustawień przeglądarki. Innymi słowy, jeżeli użytkownik ustawił w opcjach przeglądarki wyświetlanie tekstu większego lub mniejszego od wartości domyślnej, to słowo kluczowe definiuje wielkość tekstu według ustawionego w taki sposób rozmiaru podstawowego. Współczynnik skalowania pomiędzy poszczególnymi słowami kluczowymi pozostaje stały, bez względu na to, jaką wielkość podstawową ustalono.

Na rysunku 1.3 każde słowo kluczowe napisane jest tekstem innej wielkości. Skala wielkości zależy od ustawionego w przeglądarce domyślnego rozmiaru czcionki. Do wykonania zrzutów użyłem przeglądarki Internet Explorer.



**Rysunek 1.3.** Porównanie wielkości tekstu uzyskiwanej dzięki różnym słowom kluczowym. Na rysunku po prawej widoczna jest ta sama strona po zwiększeniu w przeglądarce rozmiaru tekstu

Jak widać, słowo `small` ma różną wielkość w zależności od ustawionej w przeglądarce wielkości podstawowej. Jednak różnica pomiędzy wielkością `small` a kolejną wartością w górę lub w dół jest stała.

## REZYGNACJA Z PIKSELOWEJ PRECYZJI

Największą trudnością do przewycięzenia dla każdego projektanta korzystającego ze słów kluczowych (lub innych jednostek niebędących wartościami pikselowymi) jest to, że tekst może mieć odrobinę różną wielkość w zależności od przeglądarki, systemu operacyjnego oraz indywidualnych ustawień.

Przykładowa wielkość `small` może być różna, gdy tekst jest wyświetlany w różnych przeglądarkach na różnych systemach operacyjnych, nawet jeśli we wszystkich przeglądarkach domyślnie jest ustawiony średni rozmiar czcionki.

Zasadę *piksel to piksel* można uznać za prawo ogólnie obowiązujące w internecie. Jednak słowa kluczowe to tylko *wskazówki* dla przeglądarki w kwestii tego, jak duży lub mały ma być wyświetlany tekst. Projekty (i ich użytkownicy) mogą wiele zyskać, jeśli ich autor jest w stanie zaakceptować takie małe wariacje wielkości, wynikające ze stosowania słów kluczowych, które zwiększają elastyczność projektu.

## DWIE PRZESZKODY DO POKONANIA

W dalszej części tego rozdziału przedstawię kilka rozwiązań, mówiących w jaki sposób można uniknąć tych drobnych różnic wielkości tekstu w różnych przeglądarkach, gdy rozmiar czcionki został zdefiniowany przy użyciu słów kluczowych. Jednak najpierw muszę omówić dwie ważne rozbieżności występujące w przeglądarkach Netscape 4.x oraz IE5.

## NETSCAPE 4.X

Nawet przy domyślnych ustawieniach (średni rozmiar tekstu), przeglądarka Netscape 4 ma tendencję do wyświetlania tekstu tak małego, że nie jest on czytelny na ekranie. Ze względu na niebezpieczeństwo, iż nasz tekst będzie całkowicie nieczytelny, gdy korzystamy ze stylów CSS do ustawienia wielkości tekstu na stronach wyświetlanych w przeglądarce Netscape 4.x, lepiej nie stosować słów kluczowych. Informacje wyjaśniające dlaczego nie jest to wskazane rozwiązanie można znaleźć na stronie [www.alistapart.com/articles/sizematters](http://www.alistapart.com/articles/sizematters). Jednak nie będziemy się zbytnio przejmować kłopotami

związanymi z przeglądarką Netscape 4.x, ponieważ w naszych przykładach i tak zawsze będziemy ukrywać style przed tą starą aplikacją. Mamy więc jedną przeszkodę mniej.

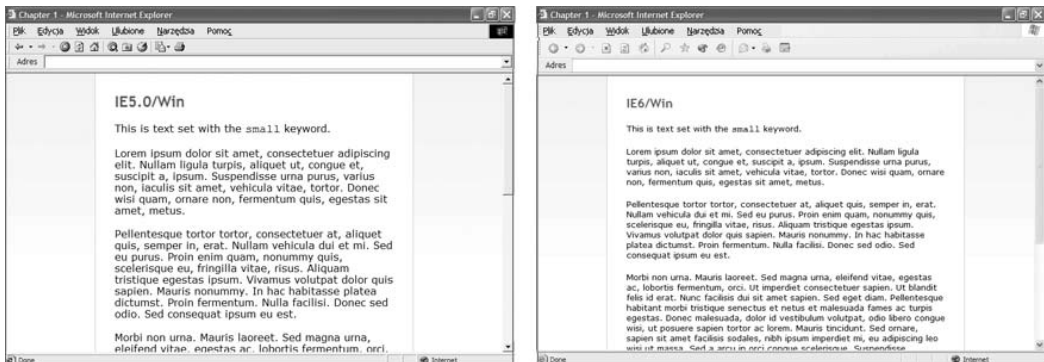
## IE5 DLA WINDOWS

Druga przeszkoda jest trochę bardziej skomplikowana, ponieważ nie ma możliwości całkowitego ukrycia stylów przed przeglądarką IE5/Win (a przeglądarka ta mimo wszystko nadal cieszy się ogromnym powodzeniem — przynajmniej w czasie pisania tej książki). Przeszkoda ta wygląda na oczywistą pomyłkę programistów, którzy stworzyli IE5/Win.

Z powodów znanych tylko firmie Microsoft, w IE5 rozmiar tekstu, którego wielkość została zdefiniowana przy użyciu słów kluczowych jest o jeden stopień *większy* niż obowiązujące standardy przeglądarek. Na przykład, jeśli zadeklarujemy

```
body {
    font-size: small;
}
```

i wyświetlimy stronę przy domyślnych ustawieniach przeglądarki, tekst w IE5/Win będzie miał taką samą wielkość, jak tekst o rozmiarze medium wyświetlony w innych przeglądarkach. W przeglądarce IE5/Win zaprojektowane przez nas strony nie wyglądają tak, jak chcieliśmy. Na rysunku 1.4 widoczna jest różnica wielkości tekstu (o rozmiarze small) w przeglądarkach IE5 i IE6.



**Rysunek 1.4.** Porównanie to ilustruje, że tekst o rozmiarze small w przeglądarce IE5 jest o jeden stopień większy niż w przeglądarce IE6/Win oraz w innych powszechnie stosowanych przeglądarkach

Aby ominąć ten problem i sprawić, by została zachowana jedna wielkość tekstu przy wyświetlaniu tej samej strony w różnych przeglądarkach, zastosujemy pewną sztuczkę, dzięki której dla przeglądarki IE5/Win można zastosować inną regułę CSS niż dla pozostałych przeglądarek. Dla stron wyświetlanych w IE5/Win zadeklarujemy (przy użyciu słów kluczowych) rozmiar tekstu o jeden stopień mniejszy niż dla pozostałych przeglądarek.

## SZTUCZKA DLA IE5/WIN

Dla przeglądarki IE5/Win podamy rozmiar tekstu o jeden stopień mniejszy przy użyciu sztuczki wymyślonej przez Tanteka Çelika (Box Model Hack — [www.tantek.com/CSS/Examples/boxmodelhack.html](http://www.tantek.com/CSS/Examples/boxmodelhack.html)), która polega na tym, że oszukujemy IE5/Win, mówiąc, że deklaracja stylu CSS już się zakończyła. Dzięki tej sztuczce możemy właściwości `font-size` najpierw przypisać mniejszą wartość, która zostanie użyta w IE5/Win, a następnie nadpisać ją wartością przeznaczoną dla innych przeglądarek.

Reguły CSS potrzebne do zadeklarowania rozmiaru `small` dla tekstu elementu `<body>` wyglądają następująco:

```
body {
    font-size: x-small; /* dla IE5/Win */
    voice-family: "\}\"";
    voice-family: inherit;
    font-size: small; /* dla innych przeglądarek */
}
html>body { /* bądźmy mili dla Operry */
    font-size: small;
}
```

Jak widać w powyższym kodzie, podstawowy rozmiar czcionki ustawiony został przy użyciu słowa kluczowego `small`. Jednak wcześniej, ze względu na przeglądarkę IE5/Win, deklarujemy wartość o jeden stopień niższą niż zamierzona, a następnie oszukujemy tę przeglądarkę, że deklaracja stylu została zakończona. Osiągamy to za pomocą następującej deklaracji:

```
voice-family: "\}\"";
```

Tylko przeglądarka IE5/Win rozpoznaje ten nawias zamykający, który sugeruje, że deklaracja została zakończona. Dla wszystkich pozostałych przeglądarek wartość `x-small` nadpisujemy poprzez zadeklarowanie wartości `small`. Dzięki tak zdefiniowanym stylom CSS uzyskujemy w miarę jednolitą wielkość tekstu podstawowego we wszystkich przeglądarkach.

Możemy także omówić drugą deklarację, w której została powtórzona reguła `font-size: small`. Reguła ta — określana jako *bądźmy mili dla Opery* — powoduje, że Opera (oraz inne przeglądarki) na pewno nie zignoruje reguły, która znajduje się za sztucznym zakończeniem deklaracji dla IE5. Jeśli korzystamy ze sztuczki Box Model Hack, zawsze powinniśmy umieścić tę deklarację. Dzięki niej wszystkie zgodne przeglądarki poprawnie nadadzą zadeklarowane style i będą mogły przejść do odczytywania następczej deklaracji w arkuszu stylów.

## UPROSZCZONA WERSJA SZTUCZKI BOX MODEL HACK

Opracowana przez Edwardsona Tana uproszczona sztuczka Box Model Hack (SMBH — ang. *Simplified Box Model Hack*) umożliwia osiągnięcie tego samego efektu (podanie innych wartości dla przeglądarki IE5/Win), przy mniejszej ilości kodu. Łatwiejsza do zrozumienia jest też chyba użyta w tej wersji składnia.

Po zastosowaniu SMBH nasz poprzedni przykład wygląda następująco:

```
body {
    font-size: small;
}
* html body {
    font-size: x-small; /* dla IE5/Win */
font-size: small; /* dla innych wersji IE */
}
```

Arkuszy rozpoczynamy od normalnej deklaracji stylów dla elementu `<body>` i umieszczamy w nim regułę deklarującą dla tekstu wielkość `small`. Następnie, używając selektora `* html`, wykorzystujemy dziurę w przeglądarce IE, która umożliwia zdefiniowanie deklaracji tylko dla przeglądarek IE. Przeglądarka IE5/Win ignoruje lewy ukośnik we własności `font-size`, co umożliwia nam zadeklarowanie wartości `x-small` dla IE5/Win i nadpisanie jej wartością `small` dla pozostałych wersji przeglądarki IE.

W ten sposób uzyskujemy rezultat taki sam, jak w poprzednim przykładzie. Jednak składnia użyta w tym rozwiązaniu jest dużo bardziej zrozumiała. Bardziej szczegółowe informacje na temat sztuczki SBMH można znaleźć na stronie <http://www.info.com.ph/~etan/w3pantheon/style/modifiedsbmh.html>.

Oczywiście pojawiają się głosy, że stosowanie sztuczek typu Box Model Hack czy SBMH jest zbyt uciążliwe i nie warto ich stosować tylko po to, by poprawić rozmiar tekstu. Jeśli jednak zależy nam na

spójnym wyglądzie strony we wszystkich przeglądarkach, warto zapłacić taką cenę.

Istnieją metody strategicznego zbierania wszystkich trików CSS i umieszczania ich oddzielnie od schludnych i zgodnych arkuszy stylów. Kilka takich metod poznamy w rozdziale 9.

## Dlaczego rozwiązanie to jest kuloodporne

Po pokonaniu wszystkich przeszkód mamy w kuloodporny sposób zdefiniowaną wielkość tekstu podstawowego. Zamiast deklarować rozmiar czcionki w pikselach (tak jak ma to miejsce w przypadku strony *eyeglasses.com* oraz wielu innych witryn), w deklaracji dla elementu `<body>` używamy słowa kluczowego `small`, aby ustawić domyślną wielkość tekstu na stronie. Dzięki stosowaniu słów kluczowych, w każdej przeglądarce i urządzeniu (włącznie z IE/Win) będzie istniała możliwość zmiany rozmiaru tekstu przez użytkownika.

Takie rozwiązanie jest *niezbędne* dla ludzi z wadą wzroku — i to nie tylko dla osób szukających nowych okularów, ale także dla osób szukających przepisów, robiących zakupy czy czytających artykuły w internecie. Mimo że ta mała zmiana okaże się prawie niezauważalna dla przeciętnego użytkownika strony, będzie ogromnym ułatwieniem dla tych, którzy dopiero po jej wprowadzeniu będą mogli z tej strony skorzystać.

## Elastyczna baza — i co dalej?

Na razie dowiedzieliśmy się, w jaki sposób należy definiować rozmiar czcionki dla całej strony. Oczywiście chcemy, by niektóre elementy projektu wyróżniały się inną wielkością czcionki — na przykład nagłówki, listy, podpisy, nawigacja czy tabele. Tekst każdego z tych elementów potencjalnie może być innej wielkości niż podstawowy tekst na stronie. Jaki jest jednak najlepszy sposób deklarowania rozmiaru tekstu innych elementów (mniejszego lub większego od wielkości bazowej)?

Polecam metodę, która okazała się dla mnie najlepszym rozwiązaniem — jest zrozumiała i dzięki niej można łatwo modyfikować rozmiar istniejącego tekstu oraz dodawać nowe deklaracje, nawet już po opublikowaniu witryny.



### WSKAZÓWKI

Więcej informacji na temat stosowania słów kluczowych i wspomnianych tutaj sztuczek można znaleźć w szczegółowym artykule Todda Fahrnera w magazynie *A List Apart* ([www.alistapart.com/articles/sizematters](http://www.alistapart.com/articles/sizematters)).

Jeśli *nie przeszkadza* nam, że w przeglądarce IE5/Win wyświetlany tekst jest o jeden stopień większy, możemy z powodzeniem pominąć opisane w tym podrozdziale przeszkody. Nie musimy ich pokonywać, możemy je ominąć.





## UWAGA

Aby zachować spójność wyglądu w przeglądarce IE5/Win, tutaj także można skorzystać z opisywanej wcześniej sztuczki `Box Model Hack`.

## USTAW I ZAPOMNIJ

Podczas definiowania rozmiaru tekstu, tylko w jednym miejscu w naszym arkuszu stylów użyjemy wartości bezwzględnej. Słowo kluczowe wielkości bezwzględnej zastosujemy tylko w deklaracji dla elementu `<body>`, tak jak omawialiśmy to wcześniej w tym rozdziale:

```
body {
    font-size: small;
}
```

Jak widać na rysunku 1.5, wielkość 12 pikseli można porównać z wielkością `small`. Oczywiście różnice te będą odrobinę inne w różnych przeglądarkach, ponieważ każdy program może mieć zdefiniowaną inną wielkość dla poszczególnych słów kluczowych. Jednak w większości przypadków użycie słowa kluczowego `small` przy definiowaniu podstawowego rozmiaru tekstu jest równoznaczne z ustawieniem wartości 12px (przy założeniu, że użytkownik nie zmienił domyślnych ustawień przeglądarki). Takie porównanie jest bardzo przydatne dla projektantów dopiero zaczynających stosować słowa kluczowe.

Tekst wielkości `small`

Tekst wielkości 12px

**Rysunek 1.5.** Porównanie tekstu o wielkości 12px z tekstem, którego rozmiar ustawiono za pomocą słowa kluczowego `small`. Przy domyślnych ustawieniach przeglądarki (średni rozmiar tekstu) otrzymany wynik jest bardzo podobny

150%  
↑  
120%  
↑  
`font-size: small;`  
↓  
90%  
↓  
80%

**Rysunek 1.6.** Przy użyciu procentów rozmiar tekstu możemy zarówno zwiększyć, jak i zmniejszyć w stosunku do podstawowej wielkości

## PROCENTOWA ZMIANA WARTOŚCI BAZOWEJ

Podstawową wielkość tekstu ustawioną za pomocą słowa kluczowego `small` będziemy modyfikować, korzystając z wartości procentowych. Stosowanie wartości procentowych jest prostą drogą do wizualizacji pożądanego rozmiaru elementu. Na rysunku 1.6 możemy zauważyć, jak wartości procentowe zwiększają lub zmniejszają rozmiar czcionki w stosunku do podstawowej wielkości tekstu.

Jeśli na przykład chcemy, by element `<h1>` był odrobinę większy niż tekst o wielkości `small`, wystarczy zadeklarować:

```
body {
    font-size: small;
}
h1 {
    font-size: 150%;
}
```

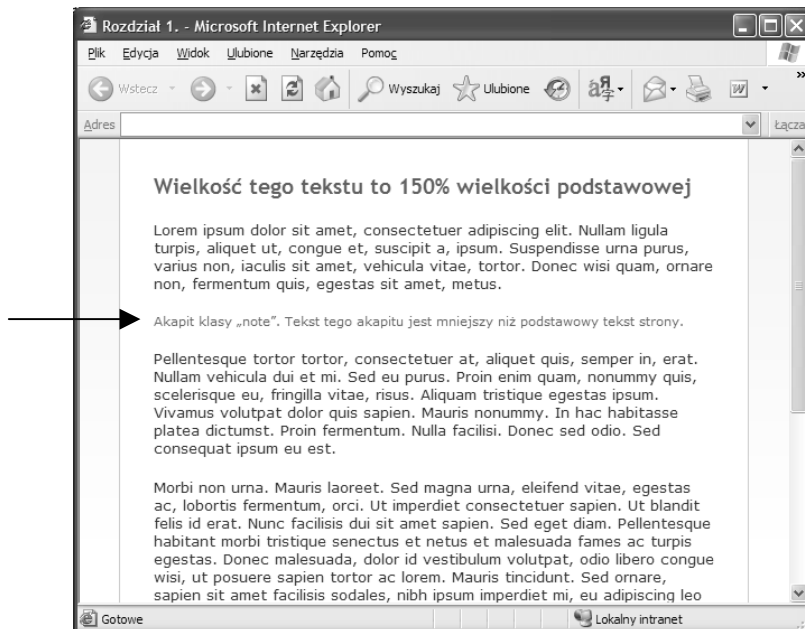
Elementy `<h1>` zawsze znajdują się *wewnątrz* elementu `<body>`, dlatego ich tekst na pewno będzie półtora raza większy niż tekst wielkości `small` (rysunek 1.7).



**Rysunek 1.7.** Wielkość tekstu elementu `<h1>` została ustawiona na 150% wielkości `small`

Podobnie, może być wskazane, by tekst niektórych akapitów był mniejszy niż podstawowy tekst strony, czyli w naszym przypadku miał wielkość mniejszą niż `small`. W tym celu tworzymy klasę `note` i dodajemy ją do krótkiego akapitu (rysunek 1.8).

```
<p class="note">Akapit klasy „note”. Tekst tego akapitu  
jest mniejszy niż podstawowy tekst strony.</p>
```



**Rysunek 1.8.** Wielkość tekstu klasy note to 85% wartości wielkości ustawionej przez słowo kluczowe small (akapit klasy note został dodatkowo wyróżniony szarym kolorem)

Tekst tego akapitu ma być nieznacznie mniejszy niż domyślna wielkość tekstu, dlatego dodajemy dla tej klasy deklarację, w której użyta jest wartość procentowa mniejsza niż 100%.

```
.note {
    font-size: 85%;
}
```

Wartości procentowe możemy przypisać dowolnemu elementowi strony, by zwiększyć lub zmniejszyć wielkość jego tekstu w stosunku do wielkości tekstu elementu <body>.

Uproszczony arkusz stylów, w którym rozmiar tekstu poszczególnych elementów określony jest przy użyciu wartości procentowych wygląda następująco:

```
body {
    font-size: small;
}
h1 {
    font-size: 150%;
}
```

```
h2 {
    font-size: 130%;
}
h3 {
    font-size: 120%;
}
ul li {
    font-size: 90%;
}
.note {
    font-size: 85%;
}
```

Zadeklarowaliśmy stopniowo malejące wielkości dla trzech typów nagłówków, pomniejszyliśmy tekst listy wypunktowanej oraz utworzyliśmy klasę `note` — możemy ją zastosować do każdego elementu, którego tekst ma być mniejszy od wielkości domyślnej. Jest to podstawowy przykład, jednak doskonale ilustruje on nasze dwuetapowe rozwiązanie — najpierw należy ustawić **podstawową wielkość tekstu przy użyciu słowa kluczowego**, a następnie **za pomocą wartości procentowych zwiększyć lub zmniejszyć wielkość tekstu pozostałych elementów**. Wystarczy zmienić wartość podstawową, a reszta rozmiarów dostosuje się samoistnie.

Zaletą stosowania wartości podstawowych w połączeniu z wartościami procentowymi jest to, że w celu zmiany podstawowej wielkości tekstu wystarczy zmienić jedną regułę w deklaracji stylów dla elementu `<body>`. Wielkości pozostałych elementów, które zostały zdefiniowane przy użyciu wartości procentowych dostosowują się do zmiany wielkości podstawowej. Zmiana rozmiaru czcionki z wartości `small` na `large` spowoduje zmianę wielkości tekstu wszystkich elementów. Rozwiązanie to jest bardzo poręczne, nawet jeśli w pewnym momencie zdecydujemy się na zdefiniowanie wielkości podstawowej w innych jednostkach.

Z tych samych powodów rozwiązanie to doskonale się sprawdza w przypadku użytkowników, którzy lubią korzystać z własnych arkuszy stylów, nadpisujących wartości zdefiniowane w arkuszach strony internetowej. Wystarczy, że użytkownik zdefiniuje jedną regułę modyfikującą podstawowy rozmiar tekstu, a rozmiar tekstu wszystkich elementów proporcjonalnie się zmieni.

## Stosowanie słów kluczowych i wartości procentowych

Aby pełniej wykorzystać zalety pracy ze słowami kluczowymi i wartościami procentowymi, należy pamiętać o kilku kwestiach. Informacje zawarte w tym podrozdziale powinny pomóc we właściwym dostosowaniu wartości procentowych. Wspomnę także o kilku najważniejszych aspektach, o których należy pamiętać przy zagnieźdzeniu wartości procentowych.

### USTAWIENIE POŚREDNIEJ WARTOŚCI BAZOWEJ

Bardzo przydatna sztuczka polega na dostosowaniu wielkości tekstu przy użyciu słowa kluczowego i elementu `<div>`, w którym umieszcza się całą zawartość strony. Często przy tworzeniu projektów opartych na stylach CSS z różnych powodów stosuje się główny element `<div>` zawierający wszystkie pozostałe elementy (więcej informacji na ten temat można znaleźć w rozdziale 8.). Element ten możemy także wykorzystać, by przy użyciu wartości procentowych dostosować do naszych wymagań wielkość tekstu zdefiniowaną słowami kluczowymi.

Wyobraźmy sobie, że pracujemy z następującą strukturą:

```
<body>

  <div id="container">
    <h1>To jest tytuł</h1>
    <p>Tekst o domyślnej podstawowej wielkości.</p>
  </div>

</body>
```

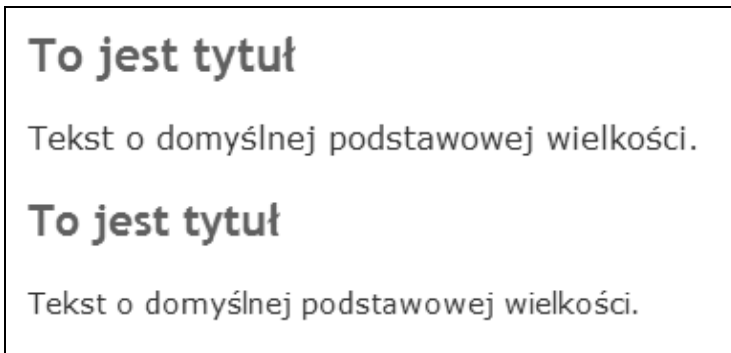
Tak jak robiliśmy to we wcześniejszych przykładach, podstawową wielkość tekstu możemy ustawić przy użyciu słów kluczowych, a wielkość nagłówka `<h1>` możemy ustawić za pomocą wielkości procentowych:

```
body {
  font-size: small;
}
h1 {
  font-size: 150%;
}
```

Jeśli jednak stwierdzimy, że wartość `small` jest zbyt duża dla naszego projektu (gdy wyświetlamy stronę przy domyślnych ustawieniach przeglądarki), możemy ją zmienić na nieco mniejszą. Wystarczy przy użyciu procentów zadeklarować mniejszą wielkość dla obiektu o identyfikatorze `#container`, w którym umieścimy wszystkie elementy tekstowe strony.

```
body {
    font-size: small;
}
h1 {
    font-size: 150%;
}
#container {
    font-size: 95%;
}
```

Na rysunku 1.9 widać, że zmniejszyliśmy trochę domyślny rozmiar czcionki. Wprowadzone zmiany dotyczą także tytułu.



**Rysunek 1.9.** Porównanie tekstu wielkości `small` (górna część) z tekstem zmniejszonym poprzez ustawienie dla zawierającego go pojemnika wielkości 95% (dolna część)

Dla tekstu w dolnej części rysunku, który w kodzie znajduje się w elemencie `<div>` o identyfikatorze `#container`, dodaliśmy regułę `font-size: 95%`. Tekst w górnej części rysunku ma wielkość ustaloną za pomocą słowa kluczowego `small`. Różnica w wyświetlanej wielkości jest niewielka.

Zaletą tego rozwiązania jest precyzyjne dostosowanie rozmiaru tekstu wtedy, gdy chcemy uzyskać wielkość pośrednią między wielkościami nadawanymi przez dwa kolejne słowa kluczowe. Korzystając

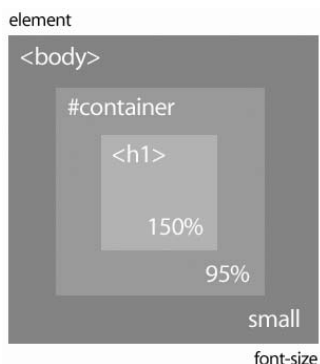
z istniejącego elementu `<div>`, w którym znajdują się wszystkie elementy strony, możemy dowolnie zmieniać rozmiar tekstu i w ten sposób zdefiniować własny rozmiar podstawowy.

Jednak najlepiej jest nie zagłębiać się zbytnio w szczegóły przy definiowaniu wielkości tekstu, ponieważ w różnych przeglądarkach i różnych systemach operacyjnych zawsze będą widoczne pewne różnice. Najważniejsze jest, że użytkownicy będą mieli możliwość zmiany wielkości tekstu przy użyciu narzędzi wbudowanych w przeglądarki.

## ZAGNIEŹDZAJĄC PROCENTY, MUSIMY ZACHOWAĆ OSTROŻNOŚĆ

Przy zagnieżdżaniu wartości procentowych należy zachować ostrożność. W poprzednim przykładzie podstawowa wielkość tekstu była zdefiniowana przy użyciu wartości `small`, a następnie została zmniejszona do 95% oryginalnej wielkości poprzez dopisanie odpowiedniej deklaracji do pojemnika `<div id="container">`. Wszystko może się trochę skomplikować, gdy zaczniemy stosować wartości procentowe do innych elementów znajdujących się wewnątrz elementu `#container`.

Na przykład wcześniej zadeklarowaliśmy, że wielkość elementu `<h1>` będzie wynosić 150% wielkości podstawowej. Elementy `<h1>` znajdują się niżej w drzewie dokumentu — poniżej obiektu `#container`. Dlatego ich wielkość to nie 150% wielkości podstawowej, lecz 150% z wartości 95% z wielkości `small`. Na rysunku 1.10 przedstawiono zależność pomiędzy zagnieżdżaniem elementów w dokumencie a przypisanymi do nich wartościami procentowymi.



**Rysunek 1.10.**  
Sposób wyliczania wartości procentowych w przypadku wielokrotnego zagnieżdżenia elementów

Faktycznie, możemy być trochę zdezorientowani. Być może pomocną będzie zamiana słowa kluczowego `small` na wartość numeryczną.

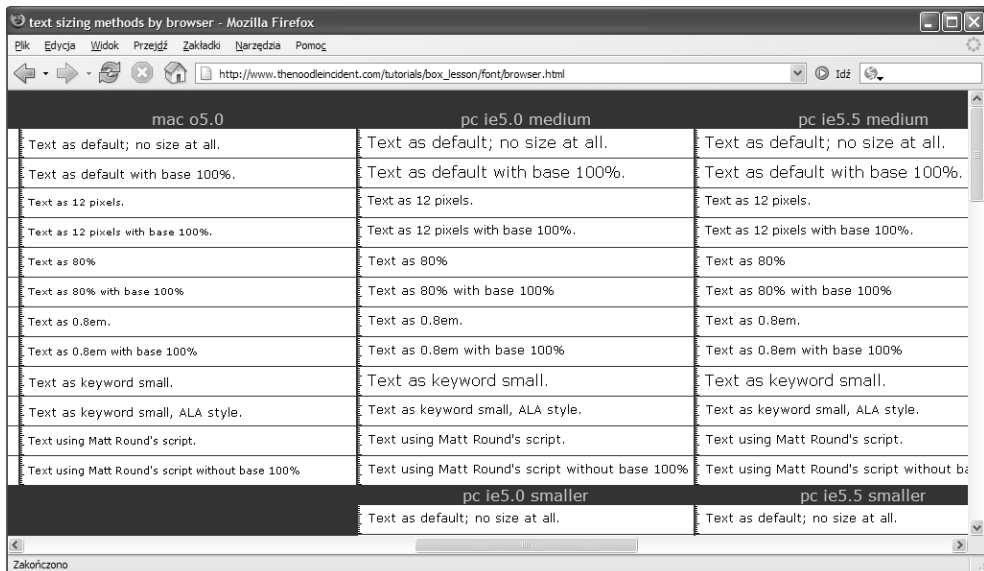
## WSKAZÓWKI

Należy zachować szczególną ostrożność przy zagnieżdżaniu wartości procentowych mniejszych niż 100% — zmniejszenie przez użytkownika domyślnej wielkości tekstu w przeglądarce może spowodować, że tekst będzie zbyt mały i stanie się nieczytelny.

Innymi słowy, jeżeli projektant przy tworzeniu strony sugerował się ustawieniami domyślnymi i użył małych wartości procentowych (takich, że przy ustawieniu średniego rozmiaru czcionki w przeglądarce tekst nadal był czytelny), możliwe jest, że strona będzie nieczytelna, gdy użytkownik wybierze w przeglądarce mniejszy rozmiar czcionki.

Czytelność tekstu na stronie należy przetestować przy różnych ustawieniach rozmiaru tekstu w różnych przeglądarkach, niezależnie od używanego połączenia słów kluczowych i procentów. Rada ta jest dobra bez względu na to, z jakiej metody ustawienia wielkości tekstu korzystamy.

Bardzo ciekawy jest test Owena Briggsa, porównujący wielkości tekstu w różnych przeglądarkach, na różnych systemach i przy zastosowaniu różnych systemów definiowania rozmiarów tekstu ([www.thenoodleincident.com/tutorials/box\\_lesson/font/method.html](http://www.thenoodleincident.com/tutorials/box_lesson/font/method.html)). Na stronie testu Briggs umieścił 264 rzuty ekranowe, dzięki czemu w jednym miejscu można porównać wszystkie uzyskane wyniki (rysunek 1.11).



**Rysunek 1.11.** Test Owena Briggsa dotyczący wyświetlanej wielkości tekstu to cenne źródło informacji, zawierające 264 rzuty ekranowe, które przedstawiają prawie każdą możliwą kombinację ustawień



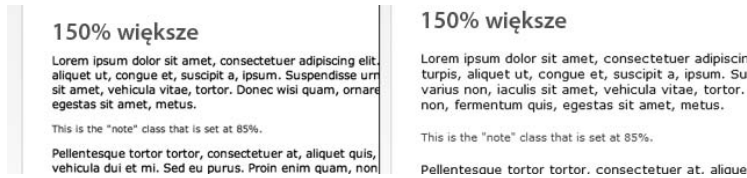
Jeżeli wielkość tekstu dla elementu `<body>` wynosi 10 (zamiast `small`), to wszystko wewnątrz elementu `#container` będzie miało wielkość 9,5 (czyli 95% z 10). Nagłówek `<h1>` znajduje się wewnątrz elementu `#container`, a rozmiar czcionki ustawiony dla tych nagłówków wynosi 150%. Wartość 150% z pierwszej wielkości podstawowej wynosiłaby 15, jednak ponieważ element `<h1>` znajduje się w elemencie `#container`, wartość ta będzie wynosić 14,25 (150% z 9,5). Obliczenia stają się coraz bardziej skomplikowane wraz z kolejnymi poziomami zagnieżdżeń. Morał z tego jest taki, że podczas deklarowania nowych wartości procentowych należy pamiętać o ustawionych już wartościach. Osobiście przy projektowaniu stron rzadko stosuję zagnieżdżanie na więcej niż dwu poziomach. Dzięki zrozumieniu tego, w jaki sposób zagnieżdżanie wpływa na wielkość wartości procentowych, możemy uniknąć rozczarowania, gdy uzyskane wielkości tekstu różnią się od zamierzonych.

## SPÓJNOŚĆ DZIĘKI WARTOŚCIOM PROCENTOWYM

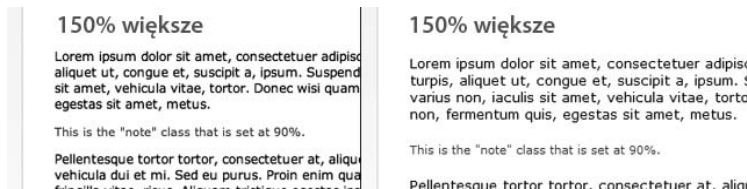
Eksperymentowanie z wartościami procentowymi może prowadzić do bardziej spójnego wyglądu strony wyświetlanego w różnych przeglądarkach na różnych platformach. Przy stosowaniu wartości procentowych dobrze jest poeksperymentować z różnymi wartościami, zwłaszcza jeśli stosujemy małe wielkości czcionki. Testy możemy wykonywać przy założeniu, że większość internautów przegląda strony, mając w przeglądarce włączony domyślny rozmiar tekstu (choć założenie to jest błędne — użytkownik może zmienić domyślne ustawienia).

Przykładowy test przeprowadzimy na używanej już wcześniej klasie `note`. Możemy porównać wielkość tekstu dla wartości 85% i 90% we właściwości `font-size`. Niektóre przeglądarki rozróżniają wartości procentowe będące wielokrotnością liczby 10 (70, 80, 90 itp.), a inne rozróżniają bardziej specyficzne wartości, które kończą się cyfrą 5 (75, 85, 95 itp.).

Porównajmy strony widoczne na rysunkach 1.12 oraz 1.13. Po lewej widoczne są strony wyświetlone w przeglądarce Safari, a po prawej strony wyświetlone w przeglądarce IE6/Win. W przeglądarce IE6/Win różnica pomiędzy wielkością 85% (na górze) i 90% (na dole) jest prawie niezauważalna. Natomiast w przeglądarce Safari widać znaczną różnicę. W tym szczególnym przypadku sugerowałbym użyć wielkości 90%, ponieważ wielkość tekstu dwóch popularnych przeglądarek prawie się nie różni.



**Rysunek 1.12.** Porównanie wielkości tekstu klasy note w przeglądarce Safari z tym samym tekstem w przeglądarce IE6/Win. Wielkość tekstu została zmniejszona do 85%



**Rysunek 1.13.** Porównanie wielkości tekstu klasy note w przeglądarce Safari z tym samym tekstem w przeglądarce IE6/Win. Wielkość tekstu została zmniejszona do 90%

Może dzielimy tutaj włos na czworo, ale zachęcamy do eksperymentowania podczas dobierania różnych wartości procentowych dla właściwości `font-size`. Zmiana wartości o 5% może prowadzić do pojawienia się zauważalnej różnicy wielkości, która może wpłynąć na spójność wyglądu projektu w różnych przeglądarkach.

## Podsumowanie

Najważniejszą informacją, jaką należy zapamiętać po przeczytaniu tego rozdziału jest to, że **użytkownikowi należy umożliwić zmianę rozmiaru tekstu**. Podzieliłem się swoimi rozwiązaniami, wykorzystującymi słowa kluczowe oraz wartości procentowe. Jednak każda strona internetowa jest inna i ma swoje własne wymagania. Najlepiej by było, gdybyśmy we wszystkich tworzonych projektach mogli uwzględnić opisaną w tym rozdziale elastyczność.

Poniżej zamieściłem kilka uwag wartych zapamiętania.

- Użytkownicy IE/Win nie mogą zmieniać wielkości tekstu, gdy jego rozmiar zdefiniowany jest w jednostkach pikselowych.
- Stosowanie słów kluczowych to łatwa, zrozumiała metoda definiowania wielkości czcionki, która umożliwia zmianę rozmiaru tekstu przez użytkownika.

- Stosowanie wartości procentowych ułatwia szybką zmianę wielkości całego tekstu strony poprzez zmianę jednej deklaracji lub użycie zdefiniowanego przez użytkownika arkusza stylów.
- Stosowanie wartości procentowych w połączeniu ze słowami kluczowymi daje możliwość bardziej precyzyjnego określenia wielkości.

Dobra wiadomość jest taka, że rozdział ten jest chyba najmniej interesujący z całej książki. Jednak jest on solidną — a raczej elastyczną — podstawą dla wszystkich kolejnych przykładów. W dalszych rozdziałach przyjmuję założenie, że użytkownik może skalować tekst na stronie w dowolnym kierunku. Z taką wiedzą możemy zacząć w kreatywny sposób stosować style CSS i przebudować wiele tradycyjnych komponentów stron tak, by stały się elastyczne.