

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

CSS. Nieoficjalny podręcznik

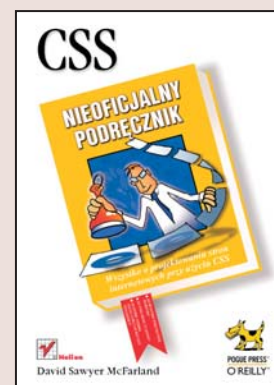
Autor: David Sawyer McFarland

Tłumaczenie: Łukasz Piwko, Marcin Rogóż

ISBN: 978-83-246-1117-1

Tytuł oryginału: [CSS: The Missing Manual](#)

Format: B5, stron: 488



Wszystko o projektowaniu stron internetowych przy użyciu CSS

- Jak przygotować atrakcyjną stronę internetową?
- Jak szybko zmienić styl witryny?
- Jak zapewnić poprawną obsługę stylów CSS we wszystkich przeglądarkach?

Obecnie w internecie coraz większe znaczenie ma forma. Witryny muszą swoim wyglądem sprawiać, że chce się na nich pozostać – w przeciwnym wypadku użytkownicy wybiorą jedną spośród niezliczonych konkurencyjnych stron. Doskonałym narzędziem do nadawania atrakcyjnego i spójnego stylu witrynom są arkusze CSS, które pozwalają dowolnie definiować wszystkie elementy stron internetowych i błyskawicznie zmieniać ich wygląd. Są przy tym łatwe do nauczenia się oraz stosowania i pozwalają na uzyskiwanie ciekawych efektów wizualnych.

„CSS. Nieoficjalny podręcznik” to wszechstronny przegląd możliwości i zastosowań tej technologii. Dzięki tej książce dowiesz się, jak przygotować poprawny arkusz stylów i dołączyć go do strony, oraz poznasz sposoby formatowania elementów kodu HTML. Nauczysz się projektować profesjonalne i wygodne w użyciu strony WWW oraz menu do nawigacji po witrynach. Przeczytasz także o tym, jak przygotowywać strony do wydruku, zapewniać poprawne wyświetlanie stylów we wszystkich przeglądarkach i dbać o wiele innych szczegółów, które zapewnią Twoim witrynom wiernych użytkowników.

- Pisanie kodu HTML pod kątem stylów CSS
- Budowa stylów i arkuszy CSS
- Dziedziczenie i kaskadowość stylów
- Dołączanie stylów CSS do stron WWW
- Formatowanie elementów stron WWW
- Tworzenie atrakcyjnych menu
- Zarządzanie układem stron za pomocą stylów CSS
- Przygotowywanie stron do wydruku
- Obsługa stylów CSS w różnych przeglądarkach

Wykorzystaj pełnię możliwości CSS i spraw, że użytkownicy zakochają się w Twoich witrynach od pierwszego wejrzenia



Spis treści

Nieoficjalna czołówka	11
Wstęp	15
Część I Podstawy CSS	27
Rozdział 1. Przystosowanie kodu HTML do pracy z CSS	29
HTML kiedyś i teraz	29
HTML kiedyś — aby dobrze wyglądało	30
HTML teraz — szkielet dla CSS	31
Pisanie HTML-a z myślą o CSS	32
Pamiętaj o strukturze	32
Dwa nowe znaczniki HTML do nauczenia	33
O czym trzeba zapomnieć	34
Podstawowe wskazówki	37
Znaczenie deklaracji typu dokumentu	38
Rozdział 2. Tworzenie stylów i arkuszy stylów	41
Anatomia stylu	41
Zrozumieć arkusze stylów	44
Styl wewnętrzny czy zewnętrzny	44
Wewnętrzne arkusze stylów	45
Style zewnętrzne	46
Dołączanie arkusza stylów przy użyciu znacznika HTML	47
Dołączanie arkuszy stylów za pomocą kodu CSS	48
Kurs: tworzenie pierwszego stylu	49
Tworzenie stylu wpisanego	50
Tworzenie wewnętrznych arkuszy stylów	51
Tworzenie zewnętrznego arkusza stylów	53

Rozdział 3. Podstawy selektorów — do czego odnoszą się style	57
Selektory znaczników — style dla całej strony	57
Selektory klas — precyzyjna kontrola	59
Selektor ID — unikalne elementy strony	62
Stylizowanie znaczników zagnieżdżonych	63
Drzewo rodzinne HTML	64
Tworzenie selektorów potomka	65
Nadawanie stylów grupom znaczników	67
Grupowanie selektorów	67
Selektor uniwersalny	68
Pseudoklasy i pseudoelementy	68
Style odnośników	69
Więcej pseudoklas i pseudoelementów	69
Zaawansowane selektory	72
Selektor dziecka	73
Selektor brata	74
Selektor atrybutu	75
Kurs: Selektory	76
Tworzenie selektora grupowego	78
Tworzenie i stosowanie selektora klasy	79
Tworzenie i stosowanie selektora identyfikatora	81
Tworzenie selektora potomka	82
Rozdział 4. Oszczędzanie czasu dzięki dziedziczeniu	85
Czym jest dziedziczenie?	85
Jak dziedziczenie upraszcza arkusze stylów	87
Granice dziedziczenia	87
Kurs: dziedziczenie	89
Prosty przykład: dziedziczenie jednopoziomowe	89
Wykorzystanie dziedziczenia do zmiany stylu całej strony	90
Kiedy dziedziczenie nie działa	92
Rozdział 5. Zarządzanie wieloma stylami: kaskada	95
Kaskadowość stylów	96
Style dziedziczone mogą się grupować	96
Najbliższy przodek bierze górę	97
Górę bierze styl bezpośredni	98
Jeden znacznik wiele stylów	98
Precyzja: który styl weźmie górę	100
Remis: wygrywa ostatni	101
Kontrolowanie kaskady	103
Zmienianie precyzji	103
Wybiórcze przesłanianie	103

Kurs: Kaskadowość w akcji	105
Tworzenie stylu mieszanego	105
Łączenie kaskady i dziedziczenia	107
Rozwiązywanie konfliktów	108
Część II Stosowanie CSS	111
Rozdział 6. Formatowanie tekstu	113
Formatowanie tekstu	113
Wybór kroju czcionki	115
Kolorowanie tekstu	117
Zmiana rozmiaru pisma	118
Stosowanie pikseli	118
Stosowanie słów kluczowych, procentów i jednostki em	119
Formatowanie słów i liter	122
Pogrubienie i kursywa	123
Zamiana tekstu na wielkie litery	123
Dekorowanie tekstu	124
Odstęp między wyrazami i literami	125
Formatowanie całych akapitów	127
Zmienianie odstępu między wierszami	127
Wyrównywanie tekstu	128
Wcinanie pierwszego wiersza i usuwanie marginesów	130
Formatowanie pierwszej litery lub pierwszego wiersza akapitu	131
Stylizowanie list	132
Typy list	133
Pozycjonowanie punktatorów i numerów	134
Punktory graficzne	135
Kurs: formatowanie tekstu	137
Ustawienia strony	137
Formatowanie nagłówków i akapitów	139
Formatowanie list	141
Wykańczanie projektu	142
Rozdział 7. Marginesy, dopełnienie i obramowanie	145
Istota modelu blokowego	145
Marginesy i dopełnienie	147
Zapis skrótowy marginesów i dopełnienia	148
Konflikty marginesów	149
Likwidowanie odstępu za pomocą marginesów ujemnych	150
Prezentacja elementów śródliniowych i blokowych	152

Ramki	153
Skrócony zapis właściwości ramek	155
Formatowanie poszczególnych krawędzi	155
Kolorowanie tła	157
Określanie wysokości i szerokości	157
Obliczanie rzeczywistych wymiarów bloku	158
„Kontrolowanie wycieków” za pomocą właściwości overflow	160
Naprawianie modelu blokowego w IE 5	162
Umieszczaj treść na elementach pływających	165
Tła i ramki a elementy pływające	167
Wstrzymać pływanie	168
Kurs: marginesy, tła i ramki	170
Ustawianie marginesów	170
Ustawianie odstępów wokół znaczników	172
Wyróżnianie tekstu za pomocą tła i ramek	172
Tworzenie paska bocznego	175
Naprawianie błędów przeglądarki	178
O krok dalej	180

Rozdział 8. Umieszczanie grafiki na stronach WWW 181

CSS i znacznik 	181
Obrazy tła	182
Sterowanie powtarzaniem	186
Pozycjonowanie obrazu tła	188
Słowa kluczowe	188
Dokładne wartości	190
Procenty	190
Umieszczanie obrazu na sztywno	192
Skrócona właściwość background	194
Kurs: Tworzenie galerii zdjęć	195
Tworzenie ramki obrazu	195
Dodawanie podpisu	197
Tworzenie galerii fotografii	200
Dodawanie cieni	204
Kurs: Używanie obrazów tła	207
Umieszczanie obrazu w tle strony	207
Zastępowanie ramek grafiką	211
Używanie grafiki w listach punktowanych	212
Zaokrąglanie rogów paska bocznego	214
Tworzenie zewnętrznego arkusza stylów	217

Rozdział 9. Upiększanie systemu nawigacji	219
Wybieranie odnośników do stylizacji	219
Poznaj stany odnośników	219
Wybieranie określonych odnośników	221
Stylizowanie odnośników	222
Podkreślanie odnośników	222
Tworzenie przycisku	224
Używanie grafiki	226
Tworzenie pasków nawigacji	228
Używanie list nienumerowanych	228
Pionowe paski nawigacji	229
Poziome paski nawigacji	232
Zaawansowane techniki pracy z odnośnikami	236
Duże, klikalne przyciski	237
CSS	239
Drzwi przesuwne	240
Kurs: Stylizowanie odnośników	242
Podstawy formatowania odnośników	242
Dodawanie obrazu tła do odnośnika	245
Wyróżnianie odnośników zewnętrznych	246
Oznaczanie odwiedzonych stron	248
Tworzenie pionowego paska nawigacji	248
Dodawanie efektu rollover i tworzenie odnośników „Jesteś tutaj”	252
Poprawianie błędów IE	255
Z pionowego w poziomy	256
Rozdział 10. Formatowanie tabel i formularzy	261
Właściwy sposób używania tabel	261
Stylizowanie tabel	264
Dodawanie dopełnienia	264
Ustawianie wyrównania w pionie i w poziomie	264
Tworzenie ramek	266
Stylizowanie wierszy i kolumn	268
Stylizowanie formularzy	269
HTML elementów formularza	271
Rozmieszczanie elementów formularza za pomocą CSS	273
Kurs: Stylizowanie tabeli	275
Kurs: Stylizowanie formularza	279

Część III Tworzenie układu strony za pomocą CSS285

Rozdział 11. Tworzenie układów opartych na elementach pływających287

Jak działa układ w CSS?	287
Dzielny znacznik <div>	288
Typy układów stron WWW	289
Podstawy układów opartych na elementach pływających	291
Stosowanie elementów pływających w układach	295
Używanie właściwości float dla wszystkich kolumn	296
Elementy pływające wewnątrz elementów pływających	298
Pozycjonowanie elementów za pomocą ujemnych marginesów	298
Rozwiązywanie problemów z elementami pływającymi	304
Czyszczenie i zawieranie elementów pływających	304
Tworzenie kolumn o pełnej wysokości	308
Zapobieganie upadaniu elementów pływających	311
Rozwiązywanie problemów z Internet Explorerem	313
Podwójny margines	313
Trzypikselowa luka	315
Inne problemy w IE	318
Kurs: Układy wielokolumnowe	319
Strukturyzowanie HTML-a	319
Tworzenie stylów układu	320
Dodawanie kolejnej kolumny	322
Dodawanie „fałszywej kolumny”	323
Ustawianie stałej szerokości	325
Kurs: Układ z ujemnym marginesem	326
Wyśrodkowywanie układu	327
Pływające kolumny	330
Końcowe poprawki	333

Rozdział 12. Pozycjonowanie elementów na stronie WWW 335

Jak działają właściwości pozycjonujące?	336
Ustawianie wartości pozycjonujących	338
Gdy pozycjonowanie bezwzględne jest względne	342
Kiedy (i gdzie) używać pozycjonowania względnego?	343
Stos elementów	346
Ukrywanie fragmentów strony	347
Użyteczne strategie pozycjonowania	347
Pozycjonowanie wewnątrz elementu	349
Wyłamywanie elementu poza blok	350
Używanie pozycjonowania CSS dla układu strony	351
Użycie stałego pozycjonowania do tworzenia ramek za pomocą stylów CSS	356

Kurs: Pozycjonowanie elementów strony	360
Wzbogacanie banera strony	360
Dodawanie podpisu do zdjęcia	364
Rozmieszczanie głównych elementów strony	366
Część IV Zaawansowany CSS	373
Rozdział 13. CSS dla strony przeznaczonej do wydruku	375
Jak działają arkusze stylów dla mediów?	375
Jak dodawać arkusze stylów przeznaczone dla mediów?	377
Określanie typu medium dla zewnętrznego arkusza stylów	378
Określanie typu medium w arkuszu stylów	378
Tworzenie stylów dla wydruku	379
Używanie !important do przesłonięcia stylów ekranowych	379
Zmiana stylów tekstu	381
Stylizowanie tła dla wydruków	382
Ukrywanie niepotrzebnych obszarów strony	384
Wstawianie podziałów stron w wydrukach	385
Kurs: Tworzenie arkusza stylów przeznaczonego dla wydruków	387
Usuwanie niepotrzebnych elementów strony	387
Usuwanie tła i dostosowywanie układu	389
Zmiana formatowania tekstu	390
Wyświetlanie logo	391
Wyświetlanie URL	392
Rozdział 14. Dobre nawyki w CSS	393
Wstawianie komentarzy	393
Porządkowanie stylów i arkuszy stylów	394
Jasno nazywaj style	395
Używanie kilku klas dla zaoszczędzenia czasu	396
Uporządkuj style za pomocą grupowania	398
Korzystanie z wielu arkuszy stylów	399
Usuwanie przeszkadzających stylów przeglądarki	401
Wykorzystanie selektorów potomków	404
Dzielenie stron na sekcje	405
Zidentyfikuj ciało	406
Obsługa błędów Internet Explorera	409
Najpierw projektuj dla nowoczesnych przeglądarek	409
Oddziel kod CSS dla IE za pomocą komentarzy warunkowych	410

Dodatki	413
Dodatek A Zestawienie właściwości CSS	415
Dodatek B CSS w Dreamweaverze 8	445
Dodatek C Zasoby CSS	469
Skorowidz	477

2

ROZDZIAŁ

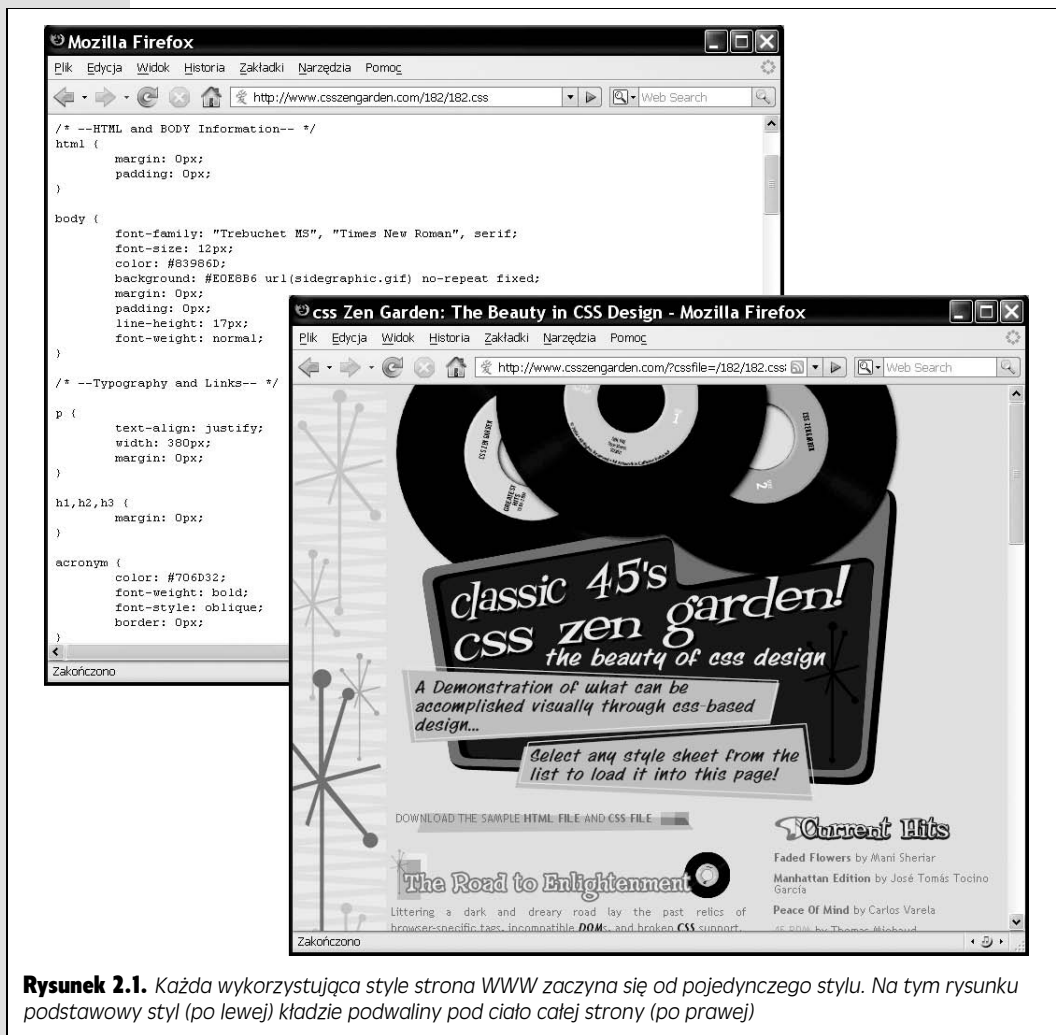
Tworzenie stylów i arkuszy stylów

Nawet najbardziej skomplikowane i najpiękniejsze strony, jak ta na rysunku 2.1, są tworzone od pojedynczego stylu. W miarę dodawania następnych stylów i arkuszy stylów strona rozwija się aż do uzyskania pełnego efektu, który inspiruje innych projektantów i zachwyca odwiedzających. Zarówno nowicjusze, jak i doświadczeni programiści muszą przestrzegać kilku zasad dotyczących tworzenia stylów i arkuszy stylów. Rozdział ten zaczniemy właśnie od podstawowych zasad ich tworzenia i używania.

Wskazówka: Niektórzy szybciej się uczą, robiąc coś, zamiast czytając. Osoby, które wolą najpierw spróbować coś zrobić, a dopiero potem wrócić i przeczytać, co dokładnie zrobiły, mogą przejść do kursu na stronie 49.

Anatomia stylu

Pojedynczy styl definiujący wygląd jednego elementu na stronie jest bardzo prosty. W zasadzie to tylko reguła informująca przeglądarkę o tym, jak ma formatować dany element na stronie — zmień kolor tekstu nagłówek na niebieski, narysuj czerwoną obwódkę wokół obrazu albo stwórz pasek boczny o szerokości 150 pikseli do przechowywania listy odnośników. Gdyby style mogły mówić, to przeglądarka usłyszałaby coś w tym rodzaju: „Hej przeglądarko! To ma wyglądać dokładnie tak”. Styl składa się z dwóch zasadniczych części: nazwy elementu strony, który przeglądarka ma sformatować (*selektora*), i rzeczywistych instrukcji formatowania (*bloku deklaracji*). Na przykład selektor może być nagłówkiem, akapitem tekstu, obrazem itd. Blok deklaracji może zamienić kolor tego tekstu na niebieski, dodać czerwoną obwódkę wokół akapitu, umieścić zdjęcie na środku strony — ilość możliwości jest nieskończona.



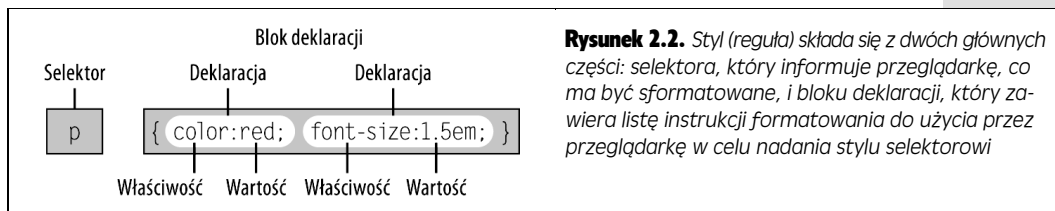
Rysunek 2.1. Każda wykorzystująca style strona WWW zaczyna się od pojedynczego stylu. Na tym rysunku podstawowy styl (po lewej) kładzie podwaliny pod całość strony (po prawej)

Uwaga: Specjaliści często, idąc za przykładem W3C, style CSS nazywają regułami. W tej książce terminy **styl** i **reguła** stosowane są zamiennie.

Oczywiście style nie mogą porozumiewać się w żadnym ludzkim języku, jak pokazuje piękny przykład z poprzedniego akapitu. Posługują się własnym językiem. Przykładowo, aby ustawić standardowy kolor i rozmiar pisma we wszystkich akapitach na stronie, można napisać:

```
p { color: red; font-size: 1.5em; }
```

Powyższy styl mówi: „Spraw, aby tekst we wszystkich akapitach — oznaczonych znacznikiem <p> — był czerwony i żeby rozmiar czcionki wynosił 1.5 em” (em to jednostka miary, która bazuje na normalnym rozmiarze tekstu w przeglądarce; więcej na ten temat w rozdziale 6.). Jak widać na rysunku 2.2, nawet tak prosty styl jak ten przykładowy składa się z kilku elementów:



Rysunek 2.2. Styl (reguła) składa się z dwóch głównych części: selektora, który informuje przeglądarkę, co ma być sformatowane, i bloku deklaracji, który zawiera listę instrukcji formatowania do użycia przez przeglądarkę w celu nadania stylu selektorowi

- **Selektor** — jak pisałem wcześniej, selektor informuje przeglądarkę o tym, który element lub które elementy na stronie mają mieć nadany styl — np. nagłówek, akapit, obrazek czy odnośnik. Na rysunku 2.2 selektor (p) odnosi się do znacznika <p>. Dzięki temu wszystkie przeglądarki będą formatowały zawartość znaczników <p> zgodnie z wytycznymi zdefiniowanymi w tym stylu. Dzięki dużej liczbie selektorów dostępnych w CSS i przy odrobinie inwencji twórczej można zapanować nad formatowaniem stron (następny rozdział opisuje selektory szczegółowo).
- **Blok deklaracji** — kod znajdujący się po selektorze definiuje wszystkie opcje formatowania, które chcemy zastosować do selektora. Blok otwiera lewy nawias klamrowy { i zamyka prawy }.
- **Deklaracja** — pomiędzy otwierającym a zamykającym bloku deklaracji nawiasem klamrowym znajduje się jedna lub więcej *deklaracji* lub instrukcji formatowania. Każda deklaracja składa się z dwóch części: *właściwości* i jej *wartości* oraz kończy się znakiem średnika.
- **Właściwość** — w CSS dostępny jest cały szereg opcji formatowania zwanych *właściwościami*. Właściwość to pojedyncze słowo, lub kilka słów oddzielonych myślnikami, powodujące określony efekt stylizujący. Większość właściwości ma nazwy składające się ze zwykłych angielskich słów, jak font-size (rozmiar pisma), margin-top (górny margines) czy background-color (kolor tła). Przykładowo właściwość background-color — nietrudno się domyślić — ustawia kolor tła. Mnóstwa właściwości nauczymy się w trakcie czytania tej książki.

Wskazówka: W dodatku A znajduje się poręczny spis właściwości CSS.

- **Wartość** — nareszcie możemy wykazać się swoim geniuszem, przypisując *wartości* do właściwości CSS i w ten sposób zmieniając kolor tła na przykład na niebieski, czerwony, fioletowy lub zielony. Jak wyjaśnię w kolejnych rozdziałach, różne właściwości CSS wymagają stosowania różnych, określonych typów wartości — określających kolor (na przykład red albo #FF0000), jednostki miary (na przykład 18px, 2in czy 5em), adres URL (na przykład images/background.gif) — lub konkretnych słów kluczowych (na przykład top, center, bottom).

Style nie muszą znajdować się w jednym wierszu, jak na rysunku 2.2. Wiele stylów składa się z wielu właściwości formatujących i aby były one bardziej czytelne, można je rozbić na kilka wierszy. Przykładowo selektor i otwierający nawias

klamrowy można umieścić w pierwszym wierszu, każdą deklarację w oddzielnym, a zamykający nawias klamrowy w ostatnim wierszu, jak poniżej:

```
p {
    color: red;
    font-size: 1.5em;
}
```

Pomocne jest też wcinanie właściwości przy użyciu pojedynczego tabulatora albo kilku spacji, co wizualnie oddziela deklaracje od selektora i ułatwia orientację w tym, co jest czym. Na koniec jeszcze można dodać pojedynczą spację pomiędzy dwukropkiem a wartością właściwości, co również wpływa dodatnio na czytelność kodu. W rzeczywistości można zastosować dowolną liczbę spacji. Można więc napisać `color:red`, `color: red`, a także `color: red`.

Zrozumieć arkusze stylów

Oczywiście jeden styl nie przekształci strony w dzieło sztuki. Można za jego pomocą zmienić kolor akapitów na czerwony, ale aby nadać stronie wspaniały wygląd, trzeba użyć wielu stylów. Ich zbiór tworzy *arkusz stylów*. Arkusz stylów może być *wewnętrzny* lub *zewnętrzny* w zależności od tego, gdzie został umieszczony — w kodzie samej strony internetowej czy w oddzielnym pliku do niej dołączonym.

Styl wewnętrzny czy zewnętrzny

W większości przypadków najlepszym wyborem są zewnętrzne arkusze stylów, ponieważ ułatwiają one tworzenie i przyspieszają aktualizację całej witryny. Wszystkie style znajdują się w jednym pliku. Wystarczy tylko jeden wiersz kodu, aby taki plik ze stylami dołączyć do strony HTML i całkowicie zmienić jej wygląd. Wygląd całej witryny można zmienić, edytując tylko jeden plik — arkusz stylów.

Po stronie użytkownika zewnętrzne arkusze stylów pomagają w szybszym otwieraniu stron. Przy użyciu zewnętrznego arkusza stylów strony zawierają tylko podstawowy kod HTML — żadnych „bajtożernych” tabel czy znaczników `` ani też kodu stylów wewnętrznych. Ponadto gdy przeglądarka pobierze zewnętrzny arkusz stylów, to przechowuje go na dysku komputera użytkownika (w niewidocznym folderze o nazwie *cache*), zapewniając sobie do niego szybki dostęp. Kiedy użytkownik przejdzie na inną stronę witryny, na której wykorzystano ten sam zewnętrzny arkusz stylów, przeglądarka nie musi pobierać go ponownie, tylko używa tego, który zapisała w pamięci podręcznej (*cache*). Daje to znaczne oszczędności, jeśli chodzi o czas pobierania strony.

Uwaga: Podczas pracy nad stroną i podglądania postępu w przeglądarce pamięć podręczna może zwrócić się przeciwko webmasterowi. Rozwiązanie tego problemu znajduje się w ramce na stronie 49.

Wewnętrzne arkusze stylów

Wewnętrzny arkusz stylów to zbiór stylów stanowiący część kodu strony. Znajduje się zawsze pomiędzy otwierającym a zamykającym znacznikiem `<style>` w nagłówku strony (`<head>`). Poniżej znajduje się przykład:

```
<style type="text/css">
h1 {
  color: #FF7643;
  font-face: Arial;
}
p {
  color: red;
  font-size: 1.5em;
}
</style>
</head>
<body>
```

/ Dalej znajduje się reszta strony...*/*

Uwaga: Znacznik `<style>` można umieścić po znaczniku `<title>` w nagłówku strony, ale większość webmasterów umieszcza go bezpośrednio przed zamykającym znacznikiem `</head>`, jak w tym przykładzie.

Znacznik `<style>` należy do kodu HTML, a nie CSS. Jego przeznaczeniem jest jednak informować przeglądarkę, że kod w nim zawarty to CSS, a nie HTML. Tworzenie wewnętrznych arkuszy stylów polega na wpisaniu jednego lub większej liczby stylów do znacznika `<style>`.

Wewnętrzne arkusze stylów łatwo dodaje się do stron i wywołują one natychmiastowy efekt wizualny. Nie są jednak najbardziej efektywnym rozwiązaniem przy tworzeniu witryny składającej się z wielu podstron. Jednym z powodów jest to, że taki arkusz trzeba skopiować i wkleić na każdą stronę witryny, co jest bardzo czasochłonne i dodaje do stron zużywający transfer kod.

Jednak wewnętrzne arkusze stylów sprawiają najwięcej problemów, gdy chce się zmienić wygląd całej witryny. Przykładowo chcemy zmienić wygląd nagłówka `<h1>`, który pierwotnie miał być pisany dużymi zielonymi i pogrubionymi literami, a teraz chcemy, aby jednak był pisany małymi niebieskimi literami i używał kroju Courier. Stosując wewnętrzne arkusze stylów, trzeba by było edytować *każdą* stronę witryny. Kto ma tyle czasu? Na szczęście jest na to proste rozwiązanie — zewnętrzne arkusze stylów.

Uwaga: Istnieje też możliwość (aczkolwiek nie zaleca się tego) dodania stylów do konkretnego znacznika HTML bez użycia arkusza stylów. Jak tego dokonać, stosując tak zwany styl wpisany, można przeczytać w kursie na stronie 50.

Style zewnętrzne

Zewnętrzny arkusz stylów to nic innego jak zwykły plik tekstowy zawierający wszystkie reguły CSS. Nigdy nie zawiera kodu HTML, więc nie należy w nim umieszczać znacznika `<style>`, i zawsze ma rozszerzenie `.css`. Można mu nadać dowolną nazwę, ale opłaca się stosować nazwy opisowe. Na przykład arkusz stylów dotyczących całej witryny może mieć nazwę `global.css`, a arkusz stylów tylko dla formularza może nosić nazwę `form.css`.

ABY ZYSKAĆ NA CZASIE

Sprawdzaj poprawność tworzonego kodu CSS

Podobnie jak za pomocą walidatora W3C upewniamy się, czy tworzone przez nas strony są w pełni poprawne (patrz ramka na stronie 36), tak samo powinno się dbać o kod CSS. W3C udostępnia narzędzie internetowe do sprawdzania kodu arkuszy stylów pod adresem <http://jigsaw.w3.org/cssvalidator/>. Działa ono podobnie jak walidator HTML — można podać adres URL strony do sprawdzenia (albo adres do samego zewnętrznego pliku CSS), wysłać plik do walidatora lub skopiować i wkleić kod do formularza, a następnie wysłać go do sprawdzenia.

Podczas pisania kodu CSS można łatwo zrobić literówkę, a jeden mały błąd może obrócić cały skrupulatnie

zaplanowany projekt wniwecz. Jeśli oparta na arkuszach stylów strona nie wygląda tak, jak zostało zaplanowane, to powodem może być prosty błąd w kodzie CSS. Walidator CSS konsorcjum W3C jest pierwszym etapem na drodze do usuwania usterek związanych z projektem.

W przeglądarce Firefox możliwe jest szybkie sprawdzenie tego, co powoduje problemy. Wystarczy w niej otworzyć stronę wykorzystującą arkusz stylów, który chcemy sprawdzić, i z menu *Narzędzia* wybrać polecenie *Konsola błędów*. Następnie w zakładce *Błędy* można znaleźć wszystkie miejsca w kodzie CSS, których przeglądarka nie rozumie.

Wskazówka: Aby, mając stronę z wewnętrznym arkuszem stylów, przerobić ją na stronę z arkuszem zewnętrznym, wystarczy wyciąć cały kod znajdujący się w znaczniku `<style>` (bez samego znacznika). Następnie należy utworzyć plik tekstowy i wkleić do niego pobrany kod CSS. Plik można zapisać pod dowolną nazwą z rozszerzeniem `.css` — na przykład `global.css` — i dołączyć go do strony przy użyciu jednej z technik opisanych poniżej.

Jeśli mamy już utworzony arkusz stylów, trzeba go dołączyć do strony, dla której został przeznaczony. Można to zrobić za pomocą znacznika HTML `<link>` lub dyrektywy CSS `@import` — robi ona w zasadzie to samo, co znacznik `<link>`. Wszystkie dostępne w tej chwili przeglądarki traktują obie te techniki równorzędnie i jako że każda z nich dołącza arkusze stylów do strony, wybór konkretnej jest kwestią preferencji (z jednym wyjątkiem opisanym poniżej).

Uwaga: Dyrektywa `@import` potrafi robić jedną rzecz, której nie potrafi znacznik `<link>`: dołączać zewnętrzne arkusze stylów do innych zewnętrznych arkuszy stylów. Ta zaawansowana technika została opisana na stronie 399.

WARSZTAT NAPRAWCZY

Zbyt stara na @import

Zewnętrzne arkusze stylów można dołączać do stron za pomocą znacznika HTML `<link>` albo przy użyciu dyrektywy CSS `@import`. Jedyna sytuacja, w której ma znaczenie wybór jednej z tych dwóch technik, to taka, gdy użytkownicy korzystają z jednej starej przeglądarki — mówiąc ściślej, chodzi o Netscape Navigator 4.

Ta bardzo stara przeglądarka tak słabo obsługuje style, że strony w niej oglądane mogą być całkowicie nieczytelne.

Navigator 4 nie rozpoznaje dyrektywy `@import` i ignoruje arkusze dołączane za jej pomocą. Fakt ten można także obrócić na własną korzyść. Jeśli zawsze używa się dyrektywy `@import`, to Navigator 4 nie będzie próbował wyświetlić strony przy użyciu dołączonych w ten sposób stylów. Strony nie będą wyglądały aż tak dobrze w tej przeglądarce, ale przynajmniej będą czytelne.

Dołączanie arkusza stylów przy użyciu znacznika HTML

Jedną z metod dodawania zewnętrznych arkuszy stylów do stron jest użycie znacznika HTML `<link>`. Jego wygląd różni się nieznacznie w zależności od tego, czy jest on stosowany na stronie HTML, czy XHTML. Poniższy przykład dotyczy strony w HTML-u:

```
<link rel="stylesheet" type="text/css" href="css/global.css">
```

A ten strony w XHTML-u:

```
<link rel="stylesheet" type="text/css" href="css/global.css" />
```

Jedyna różnica widoczna jest na końcu znacznika. Znacznik `<link>` to element pusty, ponieważ ma tylko znacznik **otwierający** i nie istnieje odpowiadający mu znacznik zamykający `</link>`. W XHTML-u konieczne jest dodanie zamykającego ukośnika (`/>`), aby oznaczyć, że znacznik w tym miejscu się kończy. W HTML-u ukośnik ten nie jest wymagany.

Pozostałe części znacznika `<link>` są takie same zarówno w HTML-u, jak i XHTML-u. Wymaga on podania trzech atrybutów:

- **rel** — oznacza typ łącza — w tym przypadku jest to łącze do arkusza stylów,
- **type** — informuje przeglądarkę, jakiego typu danych może się spodziewać — plik tekstowy zawierający kod CSS,
- **href** — wskazuje lokalizację zewnętrznego pliku arkusza stylów. Wartością tego atrybutu jest adres URL i działa on podobnie jak atrybut `src` używany przy dodawaniu obrazów do strony lub atrybut `href` wykorzystywany przy tworzeniu łącza do innych stron.

Wskazówka: Do każdej strony można dołączyć wiele zewnętrznych arkuszy stylów poprzez użycie wielu znaczników `<style>`, z których każdy wskazuje na inny plik. Technika ta jest doskonałym sposobem na organizację stylów CSS, o czym przekonamy się w rozdziale 14. (strona 399).

Dołączanie arkuszy stylów za pomocą kodu CSS

CSS ma wbudowany mechanizm dołączania zewnętrznych arkuszy stylów — dyrektywę `@import`. Stosuje się ją wewnątrz znacznika HTML `<style>` w ten sposób:

```
<style type="text/css">
@import url(css/global.css);
</style>
```

W przeciwieństwie do znacznika HTML `<link>` dyrektywa `@import` należy do języka CSS i ma pewne właściwości niespotykane w HTML-u:

- Aby utworzyć łącze do zewnętrznego pliku CSS, trzeba użyć `url`, a nie atrybutu `href`, oraz adres umieścić w nawiasach okrągłych. W tym przykładzie `css/global.css` jest ścieżką do zewnętrznego pliku CSS. Cudzysłowy otaczające adres URL są opcjonalne, w związku z czym poprawny jest zarówno zapis `url(css/global.css)`, jak i `url("css/global.css")`.

Uwaga: Poza tym, że cudzysłowy otaczające adres URL w CSS stanowią kilka dodatkowych znaków w pliku, to jeszcze Internet Explorer dla komputerów Mac ma problemy ze zrozumieniem tak prezentowanych adresów.

- Podobnie jak przy użyciu znacznika `<link>`, za pomocą dyrektywy `@import` można dodawać wiele zewnętrznych arkuszy stylów do jednego pliku HTML:

```
<style type="text/css">
@import url(css/global.css);
@import url(css/forms.css);
</style>
```

- Po dyrektywach `@import` można dodawać zwykłe arkusze stylów, jakie na przykład mają odnosić się tylko do tej jednej strony, która jednak korzysta także ze stylów używanych w całej witrynie.

Uwaga: Na temat tego, jak reguły oddziałują między sobą i jak tworzyć reguły przesłaniające inne reguły, piszę na stronie 102. Można nawet utworzyć zewnętrzny arkusz stylów, który zawiera tylko dyrektywy `@import` dołączające *inne* zewnętrzne arkusze stylów. Technika ta jest często wykorzystywana do organizacji arkuszy (patrz strona 399).

Poniżej znajduje się przykład:

```
<style type="text/css">
@import url(css/global.css);
@import url(css/forms.css);
p { color:red; }
</style>
```

W zasadzie wszystkie dyrektywy `@import` powinny znajdować się przed regułami CSS, ale jeśli się o tym zapomni, nic złego się nie stanie. Przeglądarki powinny ignorować wszystkie arkusze stylów importowane po regułach CSS, ale żadna z dostępnych obecnie przeglądarek nie przestrzega tej zasady.

Kurs: tworzenie pierwszego stylu

Reszta tego rozdziału została poświęcona podstawom dodawania stylów wpisanych, pisania reguł CSS i tworzenia wewnętrznych oraz zewnętrznych arkuszy stylów. Pracując z tą książką, będziemy tworzyć fikcyjną witrynę o nazwie *CosmoFarmer.com*, opracowując różne jej aspekty projektowe i w końcu tworząc całe składające się na nią strony. Aby zacząć, trzeba pobrać pliki kursu dostępne na poświęconej tej książce stronie pod adresem www.sawmac.com/css/. Należy kliknąć odnośnik prowadzący do archiwum ZIP zawierającego pliki kursu (na stronie można znaleźć także szczegółowe informacje na temat rozpakowywania archiwów ZIP). Pliki do każdego rozdziału znajdują się w oddzielnych folderach, których nazwy zostały nadane według wzoru *r02*, *r03* itd.

Następnie trzeba uruchomić swój ulubiony edytor stron internetowych — bez względu na to, czy jest to prosty edytor tekstu, jak Notatnik albo TextEdit, czy też wizualne środowisko, jak Dreamweaver lub Microsoft Expression Web Designer (informacje na temat wyboru edytora znajdują się na stronie 21).

WARSZTAT NAPRAWCZY

Nie daj się złapać na pamięć podręczną

Pamięć podręczna przeglądarki (cache) to doskonały sposób na przyspieszenie surfowania po sieci. Za każdym razem, gdy do pamięci tej zapisywane są jakieś często używane pliki — jak zewnętrzne arkusze stylów lub obrazy — zapisywane są cenne chwile z podróży po względnie powolnych autostradach internetu. Kiedy przeglądarka potrzebuje ponownie tego samego pliku, to zamiast jeszcze raz go pobierać, może przejść od razu do tego, co jest nowe, czyli wcześniej niewyświetlanych części strony czy nowych obrazów.

Jednak to, co jest dobre dla odwiedzających stronę, nie musi być dobre dla webmastera ją tworzącego. Jako że przeglądarki zapisują pobrane zewnętrzne pliki CSS w pamięci podręcznej, można łatwo wpaść w pułapkę podczas pracy nad stroną. Wyobraźmy sobie, że pracujemy nad stroną, na której wykorzystujemy zewnętrzny arkusz stylów, i podglądamy efekt naszych działań w przeglądarce. Coś jest nie tak, więc wracamy do edytora i wprowadzamy korekty w pliku CSS. Kiedy ponownie ładujemy stronę do przeglądarki, zmiany, których dokonaliśmy, nie są widoczne! Właśnie daliśmy się złapać na pamięć podręczną. Kiedy odświeżamy stronę, przeglądarka nie zawsze ponownie pobiera zewnętrzny arkusz stylów, w związku z czym możemy nie zobaczyć najnowszej i najlepszej wersji swojej strony.

Są dwa sposoby ominięcia tej przeszkody: wyłączyć pamięć podręczną lub zmusić przeglądarkę, aby pobierała ponownie wszystko za każdym razem.

Większość przeglądarek można zmusić do przeładowania (co powoduje również ponowne załadowanie wszystkich dołączonych plików), wciskając przycisk *Ctrl* (⌘) i przycisk odświeżania strony w przeglądarce. Kombinacja klawiszy *Ctrl+F5* działa w Internet Explorerze dla Windowsa. W Firefoksie należy użyć klawiszy *Ctrl+Shift+R* (⌘+Shift+R).

Pamięć podręczną można także wyłączyć całkowicie. W Internet Explorerze należy kolejno wybrać zakładki *Narzędzia/Opcje Internetowe/Ogólne/Historia przeglądania/Ustawienia*. W sekcji *Tymczasowe pliki internetowe* należy kliknąć *Ustawienia* i upewnić się, że opcja *Sprawdź, czy są nowsze wersje przechowywanych stron* jest ustawiona na *Za każdym razem, gdy odwiedzam tę stronę*. Klikamy dwa razy przycisk *OK*, aby zamknąć okno opcji internetowych. W Firefoksie należy przejść do *Narzędzia/Opcje* (Windows) lub *Firefox/Preferences* (Mac), aby otworzyć okno opcji tej przeglądarki. Następnie klikamy przycisk *Prywatność* i odznaczamy opcję *Pamiętaj pobrane pliki*.

Osoby używające przeglądarki Safari dla komputerów Mac powinny pobrać darmowy dodatek Safari Enhancer dostępny na stronie www.versiontracker.com/dyn/moreinfo/macosx/17776.

Wyłączenie pamięci podręcznej przeglądarki może drastycznie spowolnić przeglądanie stron internetowych. Lepiej zatem ją włączyć po skończeniu edycji plików CSS.

Uwaga: Osoby używające programu Dreamweaver mogą zajrzeć do dodatku B, w którym wyjaśniam, jak tworzyć style i arkusze stylów za pomocą tego programu. Dreamweaver, podobnie jak wiele innych programów do edycji HTML-a, pozwala także na pracę z czystym kodem HTML po naciśnięciu przycisku *Code View*. Wypróbuj go w tym kursie.

Tworzenie stylu wpisanego

Wpisując reguły CSS (jak te opisane na stronie 41) bezpośrednio w kodzie HTML strony, tworzy się style *wpisane*. Ten rodzaj stylów nie ma żadnych zalet związanych z oszczędnością czasu i transferu i dlatego profesjonaliści używają go niezmiernie rzadko. Jeśli jednak zajdzie konieczność zmiany wyglądu jednego elementu tylko na jednej stronie, to poratowanie się stylem wpisanym może mieć sens. Jeśli już to zrobimy, to chcemy, aby taki styl działał poprawnie. Ważne jest, aby poprawnie umieścić go w znaczniku, który chcemy sformatować. Poniżej znajduje się przykład pokazujący dokładnie, jak tego dokonać:

1. W wybranym edytorze otwórz plik *r02/basic.html*.

Ten prosty, ale elegancki plik XHTML zawiera kilka różnych nagłówków, jeden akapit, listę nienumerowaną oraz informację o prawach autorskich w znaczniku `<address>`. Zaczniemy od utworzenia stylu wpisanego dla znacznika `<h1>`.

2. Kliknij wewnątrz otwierającego znacznika `<h1>` i wpisz `style="color: red;"`.

Znacznik powinien wyglądać tak:

```
<h1 style="color: red;">
```

Atrybut `style` należy do języka HTML, a nie CSS, dlatego używamy po nim znaku równości i cały kod CSS umieszczamy w cudzysłowach. Tylko to, co znajduje się w cudzysłowie, jest kodem CSS. W tym przypadku dodaliśmy właściwość o nazwie `color`, która zmienia kolor tekstu, i nadaliśmy jej wartość `red` (więcej na temat kolorowania tekstu znajduje się na stronie 117). Dwukropek oddziela nazwę właściwości od jej wartości. Następnie obejrzymy wynik w przeglądarce.

3. Otwórz stronę *basic.html* w przeglądarce.

Na przykład wybierając z menu *Plik* polecenie *Otwórz plik*, zależnie od przeglądarki. Wiele edytorów HTML-a udostępnia także funkcję podglądu strony w przeglądarce (ang. *Preview In Browser*), która umożliwi otwarcie strony w wybranej przeglądarce po naciśnięciu odpowiedniej kombinacji klawiszy albo wybraniu opcji menu. Warto sprawdzić, czy używany przez nas edytor posiada taką oszczędzającą czas funkcję.

Gdy strona zostanie wyświetlona w przeglądarce, zobaczymy, że nagłówek będzie miał kolor czerwony. Style wpisane mogą składać się z większej liczby reguł CSS. Dodamy jeszcze jedną.

4. Wróć do edytora HTML-a, kliknij w miejscu znajdującym się bezpośrednio po średniku za słowem `red` i wpisz `font-size: 4em;`.

Średnik oddziela dwie różne reguły. Znacznik `<h1>` powinien wyglądać następująco:

```
<h1 style="color: red;font-size: 4em;">
```

5. Obejrzyj stronę w przeglądarce. Kliknij na przykład przycisk odświeżania przeglądarki (tylko nie zapomnij uprzednio zapisać pliku XHTML).

Teraz nagłówek jest bardzo duży. Przekonaliśmy się też, jak bardzo pracochłonne jest tworzenie stylów wpisywanych. Sprawienie, aby wszystkie nagłówki `<h1>` w witrynie wyglądały jak ten, mogłoby zabrać cały dzień — choćby na samo wpisywanie — i spowodowałoby powstanie ogromnych ilości kodu.

6. Wróć do edytora i usuń całą zawartość atrybutu `style`, przywracając znacznik nagłówka do postaci `<h1>`.

Teraz utworzymy wewnętrzny arkusz stylów.

Tworzenie wewnętrznych arkuszy stylów

Lepszym podejściem od stosowania wpisywanych arkuszy stylów jest użycie arkuszy, które zawierają wiele reguł sterujących wyglądem wielu elementów na stronie. W tym podrozdziale utworzymy arkusz stylów, który będzie zmieniał wygląd wszystkich nagłówków najwyższego rzędu za jednym zamachem. Ta prosta reguła automatycznie formatuje wszystkie znaczniki `<h1>` na stronie.

1. W pliku *basic.html* kliknij bezpośrednio po znaczniku `</title>`. Następnie naciśnij *Enter* i wpisz `<style type="text/css">`.

Kod HTML powinien teraz wyglądać tak (dodany fragment jest wyróżniony):

```
<title>Podstawowa strona internetowa</title>
<style type="text/css">
</head>
```

Otwierający znacznik `<style>` oznacza początek arkusza stylów. Po nim powinien znajdować się selektor CSS oznaczający początek pierwszego stylu.

2. Naciśnij klawisz *Enter* i wpisz `h1 {`.

`h1` oznacza znacznik, do którego przeglądarka ma zastosować następujący po nim styl.

Ten dziwnie powykręcany znak znajdujący się po `h1` to nawias klamrowy i oznacza on początek właściwości tego stylu. Innymi słowy, mówi: „Po mnie zaczyna się prawdziwa zabawa”.

3. Naciśnij klawisz *Enter*, aby przejść do nowego wiersza, naciśnij klawisz *Tab* i wpisz `color: red`.

Wpisać tę samą właściwość, co w przypadku stylu wpisanego (`color`), i jej wartość ustawiliśmy na `red`. Jak już wspominałem na stronie 43, ostatni średnik oznacza koniec deklaracji właściwości.

Uwaga: Z technicznego punktu widzenia właściwość nie musi znajdować się w nowym wierszu, ale jest to dobry zwyczaj. Gdy każda właściwość znajduje się w osobnym wierszu, łatwiej jest szybko przejrzeć cały arkusz stylów i sprawdzić, z jakich właściwości się on składa. Inną pomocną techniką jest stosowanie tabulacji (albo kilku spacji). Dzięki wcięciom można łatwo odróżnić wszystkie reguły za jednym spojrzeniem, ponieważ selektory ustawione są wtedy w jednej linii po lewej stronie, a właściwości są nieco przesunięte w prawo.

4. Naciśnij ponownie *Enter* i wpisz cztery dodatkowe właściwości, jak poniżej:

```
font-size: 2em;  
font-family: Arial, sans-serif;  
margin: 0;  
border-bottom: 2px dashed black;
```

Każda z tych właściwości dodaje inny efekt wizualny do nagłówka. Pierwsze dwie ustawiają rozmiar i krój pisma, trzecia usuwa przestrzeń otaczającą nagłówki, a ostatnia dodaje pod nim linię. Wszystkie te właściwości zostały szczegółowo omówione w części drugiej.

Praca nad tym stylem jest już zakończona, więc trzeba zaznaczyć jego koniec.

5. Naciśnij klawisz *Enter* i w ostatnim wierszu napisz tylko zamykający nawias klamrowy `}`.

Nawias ten jest odpowiednikiem nawiasu otwierającego, który wpisaliśmy w punkcie 2., jego zadaniem jest poinformować przeglądarkę, że „tutaj kończy się ta reguła CSS”. Jedyne, co pozostało do zrobienia, to zamknąć arkusz stylów znacznikiem zamykającym.

6. Naciśnij klawisz *Enter* i wpisz `</style>`.

Zamykający znacznik `</style>` oznacza koniec arkusza stylów. Gratulacje — właśnie ukończyłeś tworzenie wewnętrznego arkusza stylów. Kod na stronie powinien teraz wyglądać tak:

```
<title>Podstawowa strona internetowa</title>  
<style type="text/css">  
h1 {  
  color: red;  
  font-size: 2em;  
  font-family: Arial, sans-serif;  
  margin: 0;  
  border-bottom: 2px dashed black;  
}  
</style>  
</head>
```

7. Obejrzyj stronę w przeglądarce.

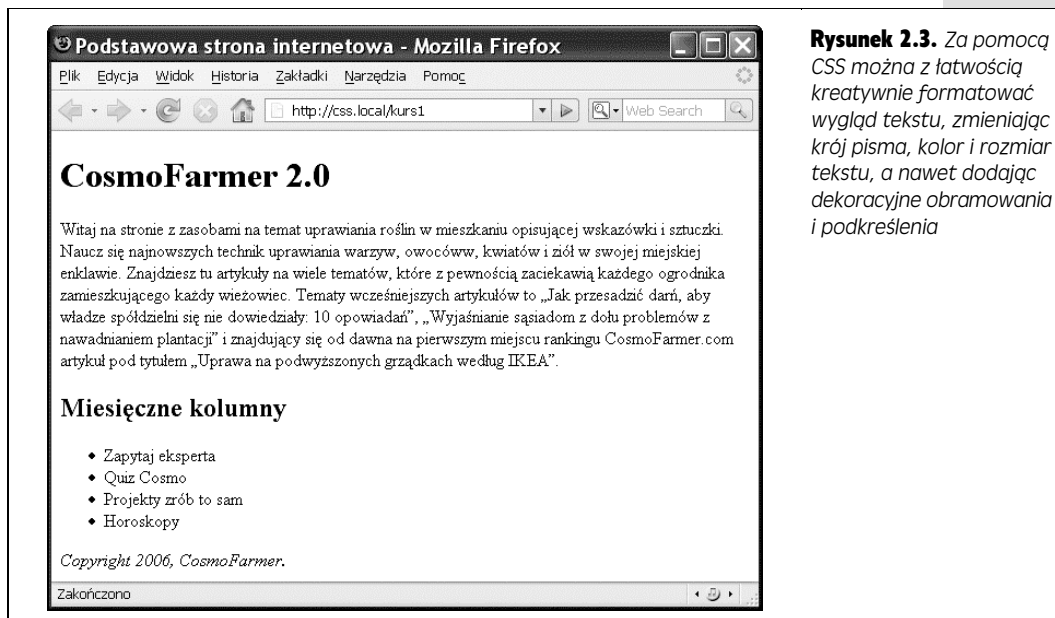
Stronę można obejrzeć w przeglądarce, otwierając ją tak, jak opisano w punkcie 3. na stronie 50, lub, jeśli jest cały czas otwarta, tylko ją odświeżając. W każdym przypadku strona ta powinna wyglądać tak, jak na rysunku 2.3.

Następnie dodamy jeszcze jeden styl.

Uwaga: Nigdy nie zapominaj dodać zamykającego znacznika `</style>` na końcu wewnętrznego arkusza stylów. Jego brak spowoduje, że przeglądarka wyświetli w oknie kod CSS, a po nim w ogóle niesformatowaną treść strony.

8. Wróć do edytora i kliknij za zamykającym nawiasem klamrowym stylu nagłówka `h1`, naciśnij klawisz *Enter* i dodaj poniższą regułę:

```
p {  
  color: #003366;  
  font-size: .9em;  
  line-height: 150%;  
  margin-top: 0;  
}
```



Rysunek 2.3. Za pomocą CSS można z łatwością kreatywnie formatować wygląd tekstu, zmieniając krój pisma, kolor i rozmiar tekstu, a nawet dodając dekoracyjne obramowania i podkreślenia

Reguła ta formatuje wszystkie akapity na stronie. Nie przejmuj się zbyt w tej chwili tym, co oznaczają wszystkie użyte w niej właściwości. W dalszych rozdziałach opisuję je szczegółowo.

9. Obejrzyj stronę w przeglądarce.

Strona zaczyna nabierać kształtu. Widać, w jakim kierunku stylistycznym będzie podążać. Aby ją ukończyć, trzeba jeszcze dodać style formatujące nagłówki drugiego rzędu, listę nienumerowaną i informację o prawach autorskich.

Proces, przez który przed chwilą przeszliśmy, to coś w rodzaju CSS w pigułce. Zaczęliśmy od strony HTML, dodaliśmy arkusz stylów i utworzyliśmy reguły CSS poprawiające wygląd naszej strony. W następnej części tego kursu nauczymy się pracować bardziej efektywnie, wykorzystując zewnętrzne arkusze stylów.

Tworzenie zewnętrznego arkusza stylów

Jako że wewnętrzne arkusze stylów grupują wszystkie style w nagłówku strony, są one o wiele łatwiejsze do tworzenia i utrzymania niż wcześniej tworzone style wpisane. Ponadto styl wewnętrzny pozwala na modyfikację wyglądu dowolnej liczby wystąpień znacznika na stronie (na przykład `<h1>`) przy wpisaniu tylko jednej prostej reguły. Jednak zewnętrzny arkusz stylów jest jeszcze lepszy — pozwala przechowywać w jednym pliku wszystkie style dotyczące całej witryny. Edycja jednego stylu w takim zewnętrznym arkuszu wywołuje efekt na wszystkich stronach witryny. W tym podrozdziale style utworzone wcześniej przeniesiemy do zewnętrznego pliku.

1. W swoim edytorze utwórz nowy plik i zapisz go pod nazwą *global.css* w tym samym folderze, w jakim znajduje się strona, nad którą pracujemy.

Pliki zewnętrznych arkuszy stylów mają rozszerzenie *.css*. Nazwa pliku *global.css* oznacza, że zawarte w nim style mają zastosowanie do wszystkich stron witryny (można oczywiście użyć dowolnej nazwy, pod warunkiem że zachowane zostanie rozszerzenie *.css*).

Zacznijmy od dodania nowego stylu do arkusza stylów.

2. Wpisz poniższą regułę do pliku *global.css*:

```
body {  
    background-image: url(images/bg.jpg);  
    background-repeat: repeat-x;  
}
```

Reguła ta ma zastosowanie do znacznika `<body>`, czyli tego, który zawiera całą treść widoczną w oknie przeglądarki, i ustawia tło na stronie. W przeciwieństwie do podobnej właściwości w HTML-u, właściwość CSS `background-image` umożliwia prezentację grafiki na wiele różnych sposobów — w tym przypadku obrazek będzie powtarzany w poziomie przez całą szerokość strony. Więcej na temat właściwości tła znajduje się na stronie 182.

Zamiast pisać od nowa wszystkie utworzone wcześniej style, można je skopiować z poprzedniego podrozdziału i wkleić do tego arkusza.

3. Otwórz plik *basic.html*, na którym pracowaliśmy wcześniej, i skopiuj całą zawartość znacznika `<style>` (nie kopiuj samych znaczników `<style>`).

Skopiuj style dokładnie w taki sam sposób, jak kopiuje się zwykły tekst. Na przykład z menu *Edycja* wybierz polecenie *Kopiuj* lub naciśnij klawisze `Ctrl+C` (⌘+C).

4. Do pustego pliku *global.css* wklej skopiowane style, wybierając polecenie *Wklej* z menu *Edycja* lub naciskając `CTRL+V` (⌘+V).

Zewnętrzne arkusze stylów nigdy nie zawierają kodu HTML — dlatego nie skopiowaliśmy znacznika `<style>`.

5. Zapisz i zamknij plik *global.css*.

Teraz pozostało tylko wyczyścić stary plik i dołączyć do niego nowy arkusz stylów.

6. Wróć do pliku *basic.html* i usuń znaczniki `<style>` i wszystkie wpisane wcześniej reguły CSS.

Style te nie są już potrzebne, ponieważ znajdują się w zewnętrznym pliku, który mamy dołączyć.

7. W miejscu, gdzie były style (pomiędzy zamykającym znacznikiem `</title>` a zamykającym `</head>`), wpisz poniższy kod:

```
<link href="global.css" rel="stylesheet" type="text/css" />
```

Znacznik `<link>` to jeden ze sposobów dołączania arkuszy stylów do strony. Innym jest użycie dyrektywy `@import`, o czym pisałem na stronie 48. Znacznik ten określa lokalizację zewnętrznego pliku arkusza stylów (na temat dwóch innych atrybutów — `rel` i `type` — można przeczytać na stronie 47).

Uwaga: W tym przykładzie plik arkusza stylów znajduje się w tym samym folderze co strona. Gdyby był w innym folderze, trzeba by było wskazać jego lokalizację za pomocą ścieżki dostępu definiowanej względem dokumentu lub folderu głównego. Czynności są takie same, jak przy tworzeniu odnośnika do innej strony (krótki kurs dotyczący odnośników względnych można znaleźć na stronie www.communitymx.com/content/article.cfm?cid=230AD).

8. Zapisz plik i obejrzyj go w przeglądarce.

Strona powinna wyglądać tak samo jak w punkcie 9. na stronie 53 — z dodatkiem fajnej grafiki biegnącej wzdłuż górnej krawędzi (za którą odpowiedzialny jest kod CSS dodany w punkcie 2.). Reguły CSS w tym zewnętrznym arkuszu stylów są takie same, jak te w wewnętrznym — znajdują się tylko w innym miejscu. Aby zademonstrować, jak przydatne jest umieszczanie arkuszy stylów w plikach zewnętrznych, dołączymy nasz plik do innej strony internetowej.

Uwaga: Jeśli na stronie nie jest widoczne żadne formatowanie (na przykład nagłówek CosmoFarmer nie jest czerwony), to prawdopodobnie został źle wpisany kod w punkcie 6. albo plik arkusza stylów znajduje się w innym folderze niż plik *basic.html*. W takim przypadku wystarczy plik *global.css* przenieść do tego samego folderu.

9. Otwórz plik *r02/linked_page.html*.

Ta prezentująca temat dnia strona witryny *CosmoFarmer.com* zawiera znaczniki HTML (h1, h2, p itd.) takie same jak strona, nad którą pracowaliśmy uprzednio.

10. Kliknij za zamykającym znacznikiem `</title>` i naciśnij *Enter*.

Utworzymy teraz odnośnik do arkusza stylów.

11. Wprowadź taki sam znacznik `<link>` jak w punkcie 6. powyżej.

Kod strony powinien wyglądać tak (kod dopiero wpisany jest wyróżniony):

```
<title>Explaining Irrigation Problems To Your Downstairs  
Neighbors</title>  
<link href="global.css" rel="stylesheet" type="text/css" />  
</head>
```

12. Zapisz stronę i obejrzyj ją w przeglądarce.

Gotowe! Wystarczył jeden wiersz kodu, aby z miejsca zmienić wygląd strony. W celu zademonstrowania, jak łatwo modyfikuje się style zewnętrzne, zmodyfikujemy jeden z nich, a jeden dodamy.

13. Otwórz plik *global.css* i dodaj deklarację CSS `margin-left: 25px` na końcu stylu dla znacznika `<p>`.

Kod powinien wyglądać tak (dodany kod jest wyróżniony):

```
p {  
  color: #003366;  
  font-size: .9em;  
  line-height: 150%;  
  margin-top: 0;  
  margin-left: 25px;  
}
```

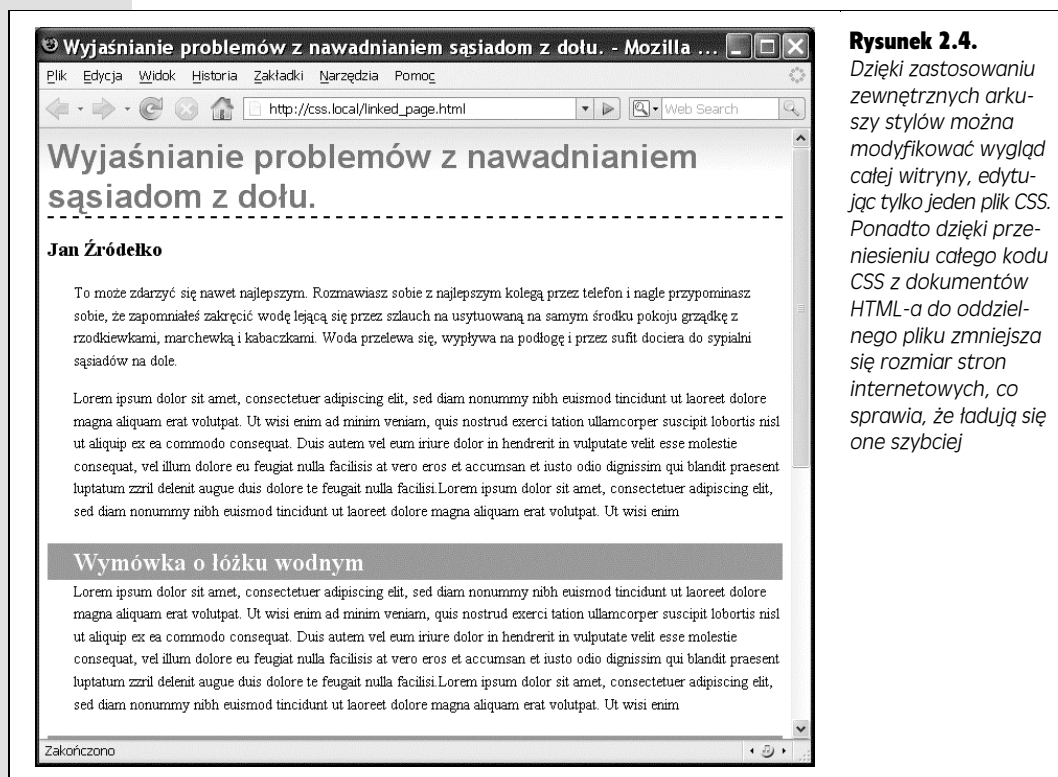
Na koniec utworzymy nową regułę dla znacznika `<h2>`.

14. Kliknij za nawiasem klamrowym zamykającym styl znacznika <p>, naciśnij **Enter** i wpisz następującą regułę:

```
h2 {  
  color: #FFFFCC;  
  margin-bottom: 0;  
  padding: 5px 0px 3px 25px;  
  background-color: #999999;  
}
```

Z większością z powyższych właściwości CSS już się zetknęliśmy. Na temat dopełnienia i marginesów piszę w rozdziale 7.

15. Zapisz plik *global.css* i obejrzyj w przeglądarce strony *basic.html* i *linked_page.html*. Rysunek 2.4 przedstawia ukończoną stronę *linked_page.html*.



Należy zwrócić uwagę, że wygląd obu stron zmienił się dzięki prostym modyfikacjom dokonany w pliku CSS. Wyobraźmy sobie teraz, że mamy tysiąc takich stron. Co za moc!