

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 według Erica Meyera. Kolejna odsłona

Autor: Eric A. Meyer

Tłumaczenie: Wojciech Moch

ISBN: 83-7361-901-1

Tytuł oryginału: [More Eric Meyer on CSS](#)

Format: B5, stron: 270

[Przykłady na ftp: 1082 kB](#)



Technologia kaskadowych arkuszy stylów (CSS) stopniowo wypiera inne metody definiowania układu i wyglądu witryn WWW. Uniwersalność i elastyczność stylów CSS sprawiają, że zmiana kolorystyki, liternictwa i konstrukcji strony sprowadza się do zmodyfikowania arkusza definiującego wygląd określonych elementów. W książce „CSS według Erica Meyera. Sztuka projektowania stron WWW” autor – prawdziwy guru projektantów witryn WWW – przedstawił czytelnikom zasady poprawnego stosowania stylów w projektach. Nadszedł czas na wykorzystanie wiedzy w praktyce.

„CSS według Erica Meyera. Kolejna odsłona” to doskonała pozycja dla wszystkich, którzy chcą poznać technologię CSS na praktycznych przykładach. Czytając ją i wykonując opisane w niej projekty, przekonasz się, jakim ułatwieniem dla projektanta stron WWW są kaskadowe arkusze stylów. W kolejnych rozdziałach przeczytasz o konwertowaniu „starych” stron WWW na dokumenty oparte o CSS, wykorzystywaniu stylów do formatowania witryn WWW i projektowaniu z wykorzystaniem najnowocześniejszych technologii definiowania wyglądu i układu strony. Poznasz sposoby przygotowywania atrakcyjnych i uniwersalnych witryn WWW, które zostaną prawidłowo wyświetlone we wszystkich przeglądarkach i na różnych urządzeniach.

O autorze:

Eric A. Meyer zajmuje się projektowaniem witryn i zagadnieniami związanymi z siecią WWW od końca 1993 roku. Jest głównym konsultantem firmy Complex Spiral Consulting (www.complexspiral.com). Mieszka w Cleveland w stanie Ohio. Ukończył Uniwersytet Case Western Reserve, na którym przez pewien czas pełnił funkcję webmastera. Obecnie jest ekspertem współpracującym z grupą roboczą W3C CSS&FP. Był jednym z autorów i koordynatorów zestawu testów W3C CSS1. Często bierze udział w konferencjach, omawiając CSS, projektowanie stron WWW, standardy sieci WWW oraz przeglądarki WWW. Jest też autorem książek CSS według Erica Meyera, CSS. Kaskadowe arkusze stylów. Przewodnik encyklopedyczny, CSS 2.0 Programme's Reference, a także słynnych tabel porównania zgodności przeglądarek ze specyfikacją CSS.



SPIS TREŚCI

O Autorze	9
O Recenzentach technicznych	10
Przedmowa	11
Wprowadzenie	15
PROJEKT 1. Przekształcanie istniejącej strony	21
Cele projektu	22
Przygotowania	22
Kładziemy fundamenty	23
Konwertowanie dokumentu	24
Okrajanie dokumentu	25
Struktura szkieletowa	25
Przebudowa projektu	28
Style podstawowe	28
Składanie nagłówka	28
Wypełnianie paska nawigacyjnego	30
Style tytułu i podsumowania	33
Informacje z boku strony	35
Stylizowana recenzja	38
Wyróżnianie cytatu i rozbudowa projektu	40
Powrót do nagłówka	42
Ocena korzyści	46
Drobna uwaga	47
Warianty	48
PROJEKT 2. Stylizowanie kolekcji zdjęć	49
Cele projektu	50
Przygotowania	50
Kładziemy fundamenty	51
Tworzenie widoku arkusza prezentacji	52
Opływanie	52
Odstępy i centrowanie	53
Style i slajdy	55
Tworzenie widoku galerii	57
Usuwanie stylów slajdów	57
Tytuły, tytuły, tytuły	58
Czyszczenie	60
Wysokość i tabele	61

Tworzenie widoku katalogu	62
Znów opływanie	63
Ułożenie i wyrównanie	65
Poprawianie listy	67
Informacje do ramek	69
Warianty	71

PROJEKT 3. Stylizowanie raportu finansowego 73

Cele projektu	74
Przygotowania	74
Style dla ekranu	74
Kładziemy fundamenty	75
Style czcionek	76
Etykiety tabeli	78
Style zysków	80
Akcentowanie wartości ujemnych	81
Style podsumowania	82
Ostatnie poprawki	84
Style do wydruku	86
Zaczynamy	86
Zaznaczanie wierszy	87
Wiersz podsumowania	89
Warianty	91

PROJEKT 4. Pozycjonowanie w tle 93

Cel projektu	94
Przygotowania	94
Style o poranku	94
Zaczynamy	94
Style nagłówka	96
Czyszczenie	100
Style na plaży	102
Ocena struktury i stylów	102
Style tytułu	104
Style treści	105
Dodawanie tła	107
Wszystkie za i przeciw	109
Warianty	111

PROJEKT 5. Menu tworzone na podstawie list 113

Cele projektu	114
Przygotowania	114
Kładziemy fundamenty	114
Sprawdzanie kodu	115
Troszkę przestrzeni	116
Rozdzielenie	117
Styl ze strzałką	118
Stylizowanie łączy	120

Zamykanie łączy	124
Zmiany	124
Łącza w ramkach	125
Zawiesić na linie	126
Warianty	130

PROJEKT 6. Menu rozwijalne tworzone za pomocą CSS 133

Cele projektu	134
Przygotowania	134
Kładziemy fundamenty	134
Układanie menu	136
Planowanie	136
Pozycjonowanie menu podrzędnych	138
Style ładniejszych menu	139
Grupowanie łączy	140
Pozycje wskazywane	143
Specjalne style podrzędnych menu	144
Przygotowywanie menu rozwijalnego	145
Ostatnie poprawki	148
Zmiana układu menu	149
Zmiana orientacji	150
Wprowadzanie poprawek	152
Poprawki w menu niższych poziomów	154
Pod rozwagę	156
Warianty	157

PROJEKT 7. Tworzenie atrakcyjnych zakładek159

Cele projektu	160
Przygotowania	160
Kładziemy fundamenty	161
Stylizowanie łączy	164
Początkowe style	164
Poprawianie wyglądu zakładek	166
Dodawanie tekstury	170
Wskazywanie i ostatnie poprawki	171
Obciążony tekst i zmniejszony obszar aktywny	172
Tworzenie rzeczywistych zakładek	175
Wprowadzanie zmian	176
Zakładki	177
Wyróżnianie aktualnej zakładki	179
Końcowe uwagi	182
Warianty	184

PROJEKT 8. Stylizacja bloga187

Cele projektu	188
Przygotowania	188
Kładziemy fundamenty	188

Stylizowanie bloga	190
Zabawy z tytułem	190
Tytuł i data wpisu	194
Rozdzielanie i łączenie	195
Tekst i informacje	198
Ostatnie poprawki	202
Warianty	205

PROJEKT 9. Tworzenie strony domowej 207

Cele projektu	208
Przygotowania	208
Kładziemy fundamenty	208
Tworzenie projektu	210
Dwa obrazki za nagłówkiem	210
Przesunięcia i ramki	213
Zawartość i pasek boczny	214
Proste style paska bocznego	218
Kolejny listek	219
Łączy pasek bocznego	221
Zmiany ramek	223
Naturalne wyróżnienie	224
Ostatnie modyfikacje	225
Warianty	229

PROJEKT 10. Projektowanie ogrodu 231

Cele projektu	232
Przygotowania	233
Kładziemy fundamenty	233
Tworzenie projektu	235
Ziarna i sadzonki	236
Nagłówek strony	237
Dodajemy kwiat	238
Stylizowanie podsumowania	241
Stylizowanie właściwej zawartości dokumentu	244
A teraz stopka	248
Style paska bocznego	252
Dodajemy obrazki PNG	259
Przemyślenia	262
Warianty	264
Skorowidz	265

2

STYLIZOWANIE KOLEKCJI ZDJEĆ

Fotografie istnieją po to, żeby przypominać nam to, o czym zapominamy. Pod tym względem — a także pod wieloma innymi — są one przeciwnością obrazów. Obrazy są zapisem tego, co pamięta malarz. Każdy z nas zapomina o czymś innym, dlatego fotografia może mieć całkowicie różne znaczenie w zależności od tego, kto na nią patrzy.

— JOHN BERGER

CO PRAWDA, NIE KAŻDY UMIESZCZA swoje zdjęcia w sieci, jednak tego rodzaju kolekcje zdjęć stanowią wyjątkowo interesujące wyzwanie. Każde zdjęcie i związane z nim opisy tworzą niewielką, zwartą całość, która musi zostać ułożona z uwzględnieniem pozostałych zdjęć znajdujących się na stronie. Można powiedzieć, że takie kolekcje podobne są do portali, z tą różnicą, że w takim portalu każde pole prowadzi do dodatkowych informacji o zdjęciu, a nie do najnowszych wiadomości ze świata polityki lub sportu.

Kolekcje zdjęć podobne są również do znacznie częściej spotykanego układu strony; stanowią jakby katalogi produktów sprzedawanych w witrynach sklepów internetowych. Rzeczywiście, czasami sprzedawanym produktem mogą być same zdjęcia, i takie założenie przyjmujemy na czas tworzenia tego projektu.



CELE PROJEKTU

W tym projekcie będziemy sprawdzać różne metody prezentowania kolekcji zdjęć wystawionych na sprzedaż. Nasz klient przedstawił nam następujące wymagania:

- ◆ Musimy przygotować trzy metody prezentacji zdjęć: widok arkusza kontaktowego dla artysty, aby mógł sprawdzić, co jest dostępne i co pokazać innym, widok galerii, aby użytkownicy mogli zobaczyć, co jest oferowane i w końcu szczegółowy widok katalogowy pozwalający na dokonywanie zamówień.
- ◆ W galerii i w arkuszu kontaktowym na ekranie powinno pojawiać się jak najwięcej fotografii, ale w taki sposób, żeby niezależnie od wielkości okna nie było konieczne poziome przewijanie strony. W tym widoku można przedstawić tylko zdjęcie wraz z jego tytułem, jednak zdjęcia powinny układać się w postaci siatki.
- ◆ W widoku katalogu każda fotografia powinna być prezentowana w połączeniu z jej tytułem, numerem katalogowym i ceną. W tym widoku przewijanie strony nie jest żadnym problemem.
- ◆ Wszystkie trzy widoki powinny być sterowane tymi samymi znacznikami, ponieważ klient nie chce płacić za dynamiczne strony i w związku z tym chce, aby znaczniki tworzone były jednokrotnie.

W tym projekcie pracować będziemy tylko z układem kolekcji fotografii, i nie musimy zajmować się niczym ponad to. Zakładamy, że układ naszej strony zostanie umieszczony w jednej z kolumn większej strony, ale nie ma to żadnego wpływu na nasz projekt.

Ze względu na ograniczenia nałożone w projekcie, szczególnie w widokach galerii i arkusza kontaktowego, nie będziemy mogli wykorzystać tabel do ułożenia zdjęć na ekranie. Dlaczego? Wiąże się to z żądaniem, by jak najwięcej zdjęć umieścić w oknie przeglądarki.

W związku z tym w tych dwóch „zwarłych” widokach zamiast tabel musimy użyć funkcji opływania rysunków i związanych z nimi informacji. Opływanie umożliwi umieszczenie w jednym wierszu tylu obrazków, ile zmieści się w oknie przeglądarki. Oznacza to, że użytkownik otwierający przeglądarkę w rozdzielczości 800×600 pikseli zobaczy cztery obrazki w wierszu, a użytkownik działający w rozdzielczości 1280×1024 pikseli zobaczy ich sześć lub siedem. Na tego rodzaju „przepływ” dokumentu pozwala wyłącznie opływanie generowanego za pomocą CSS. Przy użyciu tabel nie ma szans na uzyskanie tego efektu. Dodatkowo, elementy możemy ustawić tak, żeby każdy z nich miał taką samą szerokość. W ten sposób sprawimy, że będą się one układały w postaci regularnej siatki.

PRZYGOTOWANIA

Proszę pobrać pliki związane z drugim projektem. Kto chce sam pracować z dokumentem, niech otworzy w swoim edytorze tekstowym plik *ch02proj.html*. To w tym pliku wprowadzać będziemy wszystkie zmiany dokonywane wraz z rozwojem naszego projektu.



KŁADZIEMY FUNDAMENTY

Na początku musimy przyjrzeć się znacznikom, z którymi przyjdzie nam pracować. Oto dwa pierwsze zestawy obrazów i informacji, które znajdziemy w dokumencie:

```
<div class="pic ls"><a href="orig/img01.jpg" class="tn"></a><ul>
<li class="title">The Ferrett's Daffodil</li>
<li class="catno">03F01</li>
<li class="price">$79.95</li>
</ul></div>
<div class="pic pt"><a href="orig/img02.jpg" class="tn"></a><ul>
<li class="title">At Lunch</li>
<li class="catno">03F02</li>
<li class="price">$59.95</li>
</ul></div>
```

Jak widać, w kodzie użyto wielu różnych klas, musimy więc sprawdzić, jakie mają one znaczenie. Na szczęście otrzymaliśmy też krótkie objaśnienie tych stylów.

- ◆ Klasa `pic` oznacza elementy `div` zawierające w sobie obrazki oraz związane z nimi informacje. Pozwala to na oddzielenie tego rodzaju elementów od pozostałych używanych na stronie.
- ◆ Klasa `ls` oznacza, że obejmuje ona obrazek o orientacji poziomej, natomiast klasa `pt` oznacza pionową orientację obrazka.
- ◆ Klasa `tn` oznacza łącza otaczającego miniaturkę obrazka.
- ◆ Klasa `title` oznacza tytuł obrazka, klasa `catno` to numer katalogowy, a klasa `price` (czyli cena)... no cóż, to jest dość oczywiste.

Jak się niedługo przekonamy, najważniejszymi klasami są `ls` i `pt`, choć wszystkie z nich przydadzą się nam w czasie prac. Skorzystamy z nich na przykład do ustalenia wysokości i szerokości obrazków, które nie są podane w samym kodzie HTML. Wiemy już, że miniaturki poziome mają szerokość 128 pikseli i wysokość 96 pikseli (miniaturki pionowe mają wielkość 96×128 pikseli), dlatego dane te będziemy musieli zapisać w regułach CSS. Na początku przygotujmy proste style dla ciała strony (ang. *body*) i stopki. Dla elementu `body` zdefiniujemy lekko beżowe tło i niewielkie marginesy. Często będziemy korzystać z opływania, a jednocześnie chcemy, żeby stopka pojawiała się za wszystkimi obrazkami, musimy więc wyczyścić te oznaczenia. Wynik tych działań przedstawiono na rysunku 2.1.

```
<style type="text/css">
body {background: #EED; margin: 1em;}
div#footer {clear: both; padding-top: 3em; font: 85% Verdana, sans-serif;}
</style>
```

Tak przygotowane style nie zmieniają się już w naszym projekcie, wobec czego nie będziemy już do nich wracać. Zabieramy się do właściwej pracy!



Zabawa ze znakami białymi

Proszę dokładnie przyjrzeć się tym znacznikom. Można zauważyć, że element `ul` został dosunięty do poprzedzającego go łącza, a zaraz za nim znajduje się znacznik zamykający element `div`. Taki zapis jest konieczny, aby uniknąć pewnych błędów ze znakami białymi, jakie ujawniają się w starszych wersjach Internet Explorera. Niestety, dla niektórych starszych przeglądarek znaki białe nadal mają duże znaczenie i dodanie ich do dokumentu lub usunięcie ich z dokumentu może powodować rozwiązanie niektórych zagadkowych problemów.

RYSUNEK 2.1.

Pierwsze kroki



TWORZENIE WIDOKU ARKUSZA PREZENTACJI

Na tym etapie możemy zobaczyć tylko samą strukturę dokumentu, czyli obrazki z dołączoną listą danych, tworzącą podpisy pod obrazkami. Na tym etapie chcemy przygotować „stronę prezentacji”, w której wszystkie obrazki poukładane będą jak w tabeli. W ten sposób umożliwimy przeglądanie dużej ilości obrazków.

Opływanie

Nie korzystamy z tabeli, dlatego dość oczywistym rozwiązaniem będzie włączenie opływania obrazków. Wiemy, że obrazki nie są szersze ani wyższe niż 128 pikseli, zatem wszystkim elementom `div` nadamy wielkość `128x128` pikseli, białe tło i czarną ramkę. A propos ramek, pozbędziemy się też niebieskiej ramki otaczającej same obrazki. Teraz strona wygląda tak, jak na rysunku 2.2.

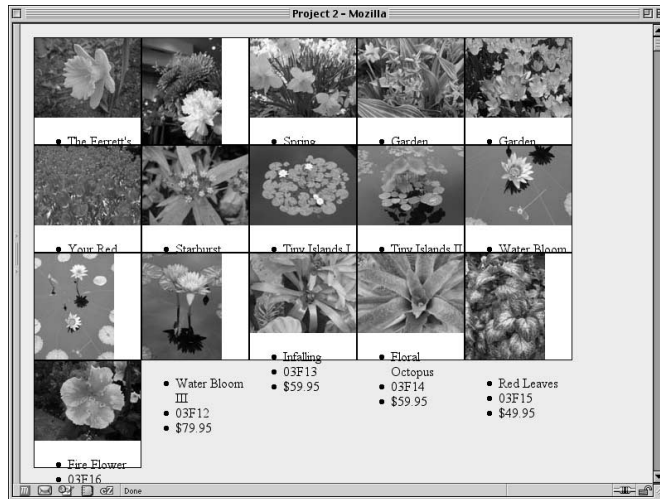
```
div#footer {clear: both; padding-top: 3em; font: 85% Verdana, sans-serif;}
div.pic {float: left; height: 128px; width: 128px;
background: white;
border: 1px solid;}
div.pic img {border: none;}
</style>
```

Efekt zaczyna powoli przypominać naszą stronę prezentacyjną, ale niestety od razu rzuca się w oczy dość poważny problem — listy pod obrazkami! Elementom `div` nadaliśmy odgórnie założoną wysokość, dlatego listy nie mogą się już w nich zmieścić, co dość skutecznie psuje wygląd naszej strony. Musimy się ich na razie pozbyć, wyłączymy więc możliwość ich wyświetlania.

Wypadki z ramkami

Nie wszystkie przeglądarki otaczają niebieską ramką obrazki będące łączami, choć część z nich tak postępuje.

W związku z tym dobrym rozwiązaniem jest nakazanie usuwania tego obramowania wszystkim przeglądarkom. Nie wpłynie to jednocześnie na działanie przeglądarek, które i tak tego nie robiły.



```
div.pic img {border: none;}
div.pic ul {display: none;}
</style>
```

W ten sposób informacje z list nie będą w ogóle wyświetlane. W dalszej części prac z powrotem włączymy wyświetlanie tych informacji, ale na razie po prostu się ich pozbędziemy.

Odstępy i centrowanie

Dzięki opływaniu obrazki tworzą w końcu regularną siatkę, ale nie sprawia ona najlepszego wrażenia. Spróbujmy rozsunąć nieco obrazki przez dodanie do elementu `div` niewielkich marginesów.

```
div.pic {float: left; height: 128px; width: 128px;
margin: 5px 3px; background: white;
border: 1px solid black;}
```

Marginesy elementu nie nakładają się na siebie, dlatego faktyczna odległość między dwoma obrazkami znajdującymi się w jednym wierszu wynosić będzie 6 pikseli ($3px + 3px$), a odstęp między wierszami będzie miał 10 pikseli. Oczywiście ustawienia te można dowolnie zmieniać w celu uzyskania innego wyglądu strony. Na tym etapie strona wygląda tak, jak na rysunku 2.3.

Całość wygląda coraz lepiej, ale same obrazki są dziwnie poukładane. Dosunięcie do górnej lub lewej krawędzi sprawia nie najlepsze wrażenie. Znacznie lepsze wrażenie sprawiałoby, gdyby były wycentrowane wewnątrz swojego pola.

Aby uzyskać taki efekt, musimy najpierw zdefiniować wielkości obrazków. Zrealizować to można dwoma prostymi (i bardzo podobnymi do siebie) regułami.

RYСУNEK 2.2.

Opływanie elementu `div` z rysunkami

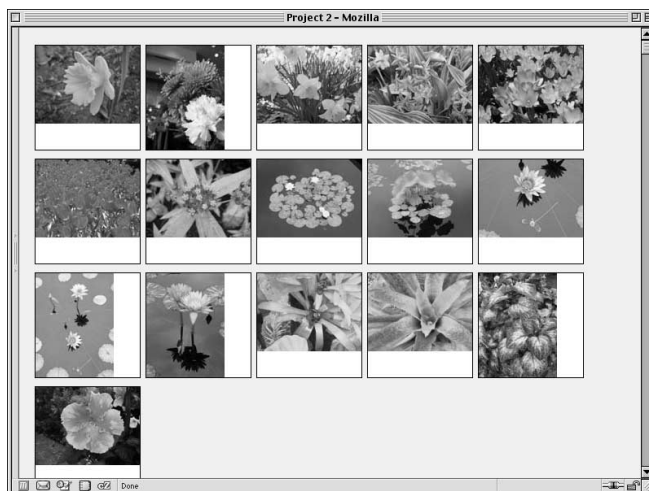


Uwaga na Explorera

W Internet Explorerze listy spowodują powiększenie wysokości elementów `div`, co przyczyni się do powstania zupełnie innego, ale również fatalnego układu strony. Jest to błąd tej serii przeglądarek znany już od wersji 5.0. Parametr `height` traktują one tak, jakby był parametrem `min-height`, którego (o ironio) nie obsługuje żadna wersja IE.

RYSUNEK 2.3.

Rozsuniecie miniatur



```
div.pic img {border: none;}
div.ls img {height: 96px; width: 128px;}
div.pt img {height: 128px; width: 96px;}
div.pic ul {display: none;}
```

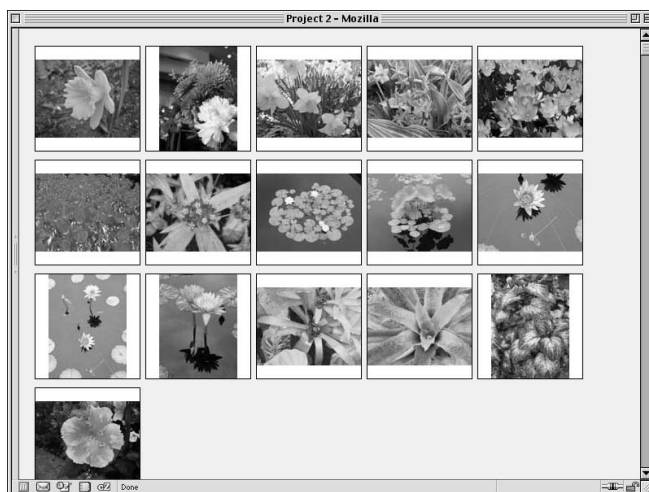
W ten sposób zapisaaliśmy tylko to, o czym sami wiedzieliśmy, ale przeglądarka nie miała pojęcia: obrazki poziome (ls) mają 96 pikseli wysokości i 128 pikseli szerokości, a obrazki pionowe (pt) mają odwrotne wymiary.

Pamiętamy, że elementom `div` nadaliśmy wielkość 128×128 pikseli. Aby teraz wycentrować obrazki wewnątrz elementów `div`, musimy do samych obrazków dodać odpowiednie marginesy. Różnica między 128 a 96 wynosi 32, a połowa z tego to 16. Wynika z tego, że obrazkom poziomym musimy dodać górny i dolny margines o wielkości 16px, a obrazkom pionowym podobnej wielkości margines lewy i prawy. Wyniki tych modyfikacji przedstawiono na rysunku 2.4.

```
div.ls img {height: 96px; width: 128px; margin: 16px 0;}
div.pt img {height: 128px; width: 96px; margin: 0 16px;}
```

RYSUNEK 2.4.

Centrowanie miniatur



Style i slajdy

Teraz układ strony wygląda już całkiem ładnie, ale nadal można go nieco poprawić. Do elementu `div` dodajmy niewielkie dopełnienia, żeby wokół obrazków powstało białe obramowanie. Pozostańmy przy naszej manierze stosowania potęg liczby dwa i do elementu `div` dodamy 16 pikseli dopełnienia.

```
div.pic {float: left; height: 128px; width: 128px;
padding: 16px; margin: 5px 3px; background: white;
border: 1px solid black;}
```

W tym momencie nasza strona zaczyna przypominać kolekcję slajdów 35 mm. Spróbujmy teraz pobawić się tym pomysłem. Najpierw przywróćmy ramki otaczające same obrazki, tak by przypominały one oprawę slajdu.

```
div.pic img {border: 1px solid;
border-color: #444 #AAA #AAA #444;}
```

Ta drobna zmiana spowoduje otoczenie obrazków ciemnoszarą ramką po lewej stronie i na górze, a jasnoszarą po prawej stronie i na dole. Daje to dość ładny efekt rzeczywistej ramki.

Jednak zabieg ten spowodował drobne rozchwianie ułożenia obrazków na stronie. Obrazki uzupełnione o ramki nie mają już swojej standardowej wielkości 128×96 pikseli lub odwrotnie, ale powiększyły się do wymiarów 130×98 pikseli, ponieważ ramki zostały dodane do szerokości i wysokości samego obrazka. Aby skorygować tę usterkę, musimy zmienić deklaracje wysokości i szerokości elementów `div`, a także ich dopełnienia.

```
div.pic {float: left; height: 130px; width: 130px;
padding: 15px; margin: 5px 3px; background: white;
border: 1px solid black;}
```

Za pomocą tych drobnych zmian odtworzyliśmy ułożenie obrazków, jakie mieliśmy wcześniej. Aby wykończyć tworzony efekt, musimy dodać zewnętrzne obramowanie, do czego doskonale nadawać się będą ramki samego elementu `div`. Musimy zastosować te same kolory, których użyliśmy w ramce obrazka, zmieniając tylko ich kolejność.

```
div.pic {float: left; height: 130px; width: 130px;
padding: 15px; margin: 5px 3px; background: white;
border: 1px solid; border-color: #AAA #444 #444 #AAA;}
```

Proszę zauważyć, że z deklaracji `border` usunęliśmy wartość `black`. Nie jest już ona potrzebna, ponieważ dodaliśmy deklarację `border-color`, co pozwoliło na ograniczenie wielkości pliku. Wyniki wprowadzonych zmian zaprezentowano na rysunku 2.5.

To wygląda już niemalże jak zbiór slajdów 35 mm, ale nadal można poprawić jego wygląd. Zamiast stosować kolory tła i ramek, możemy je całkowicie usunąć i wstawić w tło odpowiedni rysunek.

Jak? Doskonale znamy już wymiary elementów `div`, w których umieszczamy obrazki, 162×162 piksele. Na przykład slajdy poziome w osi poziomej mają następującą wielkość:

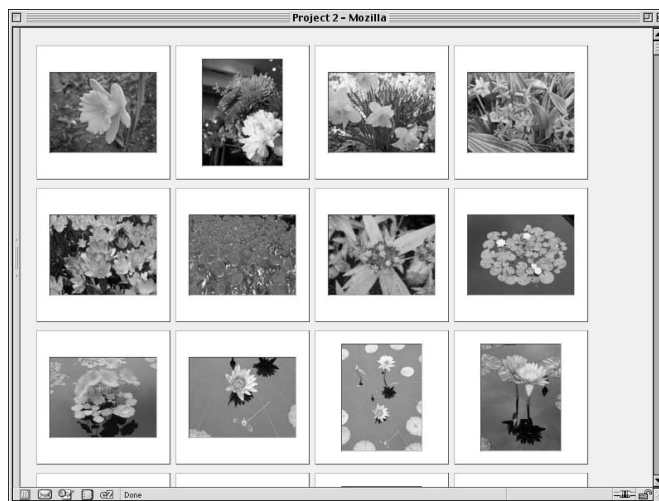
$$128\text{px szerokości elementu } \text{img} + 2\text{px ramki elementu } \text{img} + \\ 30\text{px dopełnienia elementu } \text{div} + 2\text{px ramki elementu } \text{div} = 162 \text{ piksele.}$$


Inna metoda

Zamiast przedstawionej techniki można też użyć jednego koloru ramki i zastosować styl ramki `inset`. Niestety, to rozwiązanie ma pewną wadę. Przeglądarki mogą dowolnie modyfikować kolory ramek typu `inset` (a także `outset`, `groove` i `ridge`). Jak można przewidzieć, każda z nich robi to nieco inaczej, dlatego w przypadkach, w których ważne jest cieniowanie takiej ramki, lepiej wykorzystać ramkę typu `solid` i ręcznie określić potrzebne kolory.

RYSUNEK 2.5.

Arkusz ze slajdami,
a w każdym razie coś
do niego podobnego



Mamy zamiar usunąć ramki z elementu `div`, dlatego można je odjąć z powyższego równania, co w rezultacie da nam wymiary 160×160 pikseli. Wiemy już, jak duże obrazki musimy przygotować, aby pasowały jako tło naszych elementów `div`. Trzeba pamiętać, że ze względu na dwa rodzaje obrazków (pionowe i poziome) potrzebować będziemy też dwóch rodzajów tła. Odpowiednie pliki z obrazami tła nazwiemy *frame-pt.gif* i *frame-ls.gif*. Nie jest przy tym ważne, czy powstaną one w wyniku skanowania rzeczywistej ramki slajdu, czy zostaną narysowane w Photoshopie. Potrzebny nam jest po prostu obrazek slajdu.

Po przygotowaniu potrzebnych nam obrazów slajdów, musimy dopisać je do stylów strony. Zaczniemy od usunięcia ze stylu elementu `div` definicji ramek i dodania do nich jednego z obrazków.

```
div.pic {float: left; height: 130px; width: 130px;
padding: 15px; margin: 5px 3px;
background: url(frame-ls.gif) center no-repeat;}
```

Efekt prezentuje się świetnie w przypadku obrazków poziomych, ale w przypadku pionowych tworzy nieco dziwaczne wrażenie. Musimy dla obrazków pionowych podmienić obraz tła, co wykonujemy tworząc krótką regułę.

```
div.pic {float: left; height: 130px; width: 130px;
padding: 15px; margin: 5px 3px;
background: url(frame-ls.gif) center no-repeat;}
div.pt {background-image: url(frame-pt.gif);}
div.pic img {border: 1px solid; border-color: #444 #AAA #AAA #444;}
```

Powyższy zapis spowoduje podmianę wartości obrazu wstawianego w tło, ale pozostałe parametry (`center no-repeat`) pozostawi bez zmian. W ten sposób deklaracja `background-image: url(frame-pt.gif)` jest równoważna deklaracji `background: url(frame-pt.gif) center no-repeat`. Wynik poprawek zobaczyć można na rysunku 2.6.

Naprawdę nieźle! Najlepsze w tym rozwiązaniu jest to, że tło „ramki” można później dowolnie zmieniać tylko przez aktualizację plików obrazków.

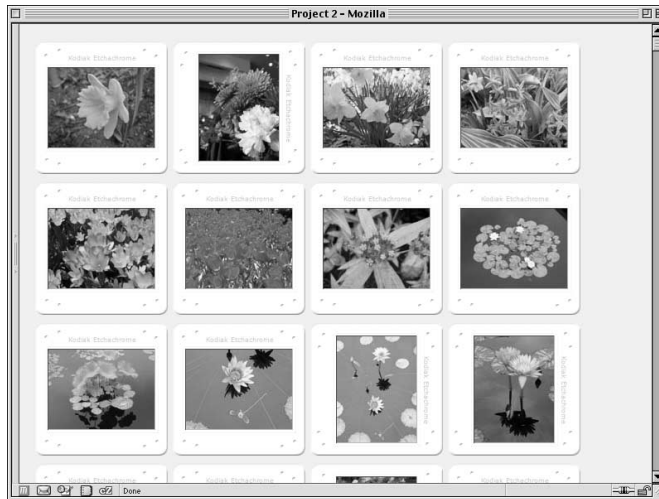
Na listingu 2.1 przedstawiam przygotowany do tej pory arkusz stylów „kolekcji slajdów”.



Gotowe obrazki ramek
slajdów znajdują się
w archiwum przygotowanym
na potrzeby tego projektu.
Pliki te noszą nazwy
frame-ls.gif i *frame-pt.gif*.



Teraz dobrym posunięciem
byłoby zapisanie naszego
projektu do osobnego pliku, bo
w kolejnym kroku usuniemy
z niego wiele stylów
i zaczniemy dodawać nowe.

**RYСУNEK 2.6.**

Teraz rzeczywiście wygląda to jak zbiór slajdów!

LISTING 2.1. Ukończony arkusz stylów „slajdów”

```
body {background: #EED; margin: 1em;}
div#footer {clear: both; padding-top: 3em;
font: 85% Verdana, sans-serif;}
div.pic {float: left; height: 130px; width: 130px;
padding: 15px; margin: 5px 3px;
background: url(frame-ls.gif) center no-repeat;}
div.pt {background-image: url(frame-pt.gif);}
div.pic img {border: 1px solid; border-color: #444 #AAA #AAA #444;}
div.ls img {height: 96px; width: 128px; margin: 16px 0;}
div.pt img {height: 128px; width: 96px; margin: 0 16px;}
div.pic ul {display: none;}
```

TWORZENIE WIDOKU GALERII

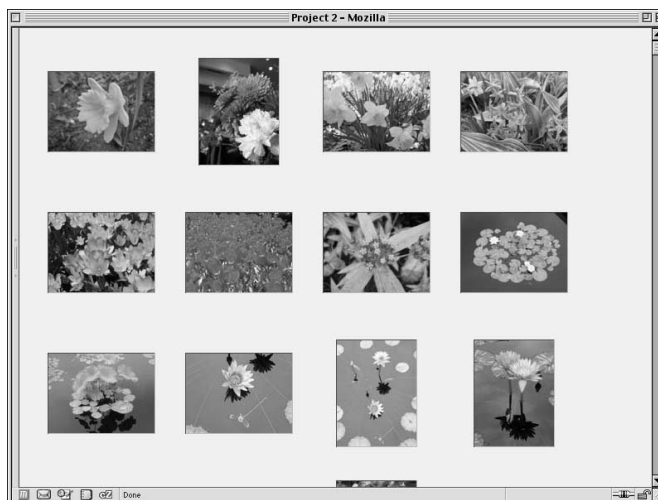
Poprzedni arkusz stylów tworzy bardzo ciekawy efekt wizualny, ale ma on pewną wadę — nie przedstawia nazw poszczególnych zdjęć. Z takiej formy najprawdopodobniej najbardziej zadowolony będzie sam artysta, który doskonale wie, jak nazywają się poszczególne zdjęcia, a chce tylko mieć możliwość przeglądania ich wszystkich. Niestety, układ ten nie będzie bardzo użyteczny dla osób odwiedzających taką stronę. Właśnie dlatego zamienimy teraz kolekcję slajdów w galerię zdjęć, z których każde podpisane będzie swoim tytułem.

Usuwanie stylów slajdów

Oczyszczanie pola do pracy zaczniemy od usuwania stylów, które wstawiają na stronie obrazki w tło elementu div (efekt widać na rysunku 2.7). Po ich usunięciu, nasz arkusz kurczy się do zestawu stylów, który prezentują na listingu 2.2.

RYSUNEK 2.7.

Usunięte ramki slajdów



LISTING 2.2. Okrojony arkusz stylów

```

body {background: #EED; margin: 1em;}
div#footer {clear: both; padding-top: 3em;
  font: 85% Verdana, sans-serif;}
div.pic {float: left; height: 130px; width: 130px;
  padding: 15px; margin: 5px 3px;}
div.pic img {border: 1px solid; border-color: #444 #AAA #AAA #444;}
div.ls img {height: 96px; width: 128px; margin: 16px 0;}
div.pt img {height: 128px; width: 96px; margin: 0 16px;}
div.pic ul {display: none;}

```

Taki zestaw stylów sprawia, że większość prac mamy już właściwie za sobą. Pozostaje tylko wyświetlić tytuły zdjęć.

Tytuły, tytuły, tytuły

Aby przedstawić same tytuły zdjęć, musimy wyłączyć ukrywanie elementów `ul`, a włączyć ukrywanie tych części listy, które w tym miejscu nie są nam potrzebne. Wyłączenie ukrywania jest niezwykle proste — wystarczy z reguły `div.pic ul` usunąć deklarację `display: none;`. Oczywiście w ten sposób pozostanie nam tylko pusty blok deklaracji:

```
div.pic ul {}
```

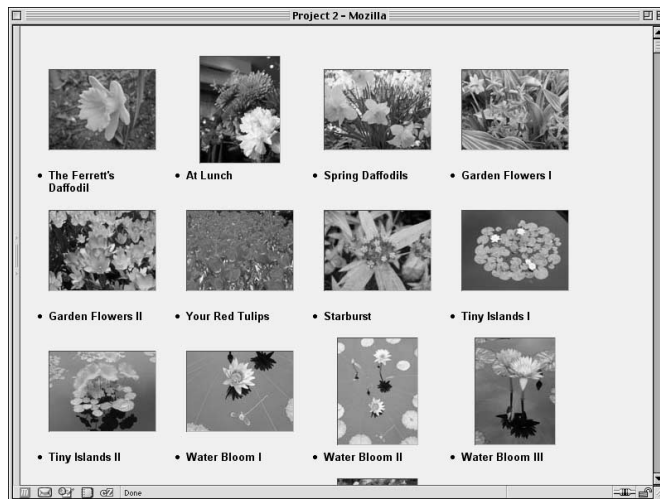
Jest to w pełni poprawny zapis, choć nie ma on żadnego sensu. W jego wyniku elementy będą wybierane tylko po to, żeby nie nadać im żadnego stylu, musimy więc wypełnić tę pustą przestrzeń. Elementom `ul` nadamy marginesy, dopełnienie, a także style czcionek.

```
div.pic ul {margin: 0.25em 0 0; padding: 0;
  font: bold small Arial, Verdana, sans-serif;}

```

Teraz listy znowu się pojawiają. Niestety, pojawią się w całości, a nam potrzebne są tylko tytuły. Wobec tego ukryjemy pozostałe elementy listy, co pozwoli nam na uzyskanie strony pokazanej na rysunku 2.8.

```
div.pic ul {margin: 0.25em 0 0; padding: 0;
  font: bold small Arial, Verdana, sans-serif;}
li.catno, li.price {display: none;}
```



RYSUNEK 2.8.

*Wyświetlamy tytuł,
ale ukrywamy
pozostałe dane*

Nieźle, ale wyraźnie widać, że projekt wymaga dalszych poprawek. Na początku musimy pozbyć się znaków wypunktowania, bo nie dość, że wyglądają paskudnie, to jeszcze rozpraszają uwagę. Można by je usunąć przez wprowadzenie właściwości `list-style`, ale możemy też zmienić po prostu styl elementu `li` tak, by nie był on traktowany jak element listy. Będzie to działało doskonale z tym tylko wyjątkiem, że Internet Explorer dla Windows mimo wszystko zachowa znak wypunktowania. Z tego względu trzeba będzie wprowadzić obie zmiany. Skoro jesteśmy już na tym etapie, to jednocześnie możemy wyrównać tekst.

```
div.pic ul {margin: 0.25em 0 0; padding: 0;
  font: bold small Arial, Verdana, sans-serif;}
li.title {display: block; list-style: none; text-align: center;}
li.catno, li.price {display: none;}
```

Teraz podniesiemy tytuły do góry, zbliżając je do obrazków. Jak pamiętamy, podczas tworzenia arkuszy stylów „slajdów” do obrazków dodawaliśmy marginesy. To właśnie te marginesy odsuwają tytuły od poziomych obrazków.

Proszę zwrócić uwagę, że mówiłem tylko o obrazkach **poziomych**. Pionowe obrazki nie mają po prostu żadnego marginesu górnego ani dolnego, dlatego tylko w ich przypadku musimy poprawić wartości marginesów. Na rysunku 2.8 zauważyć można jeszcze jedną ciekawą rzecz. Pierwsze linie tytułów obrazków w każdym wierszu są wyrównane. Jest to efekt, który warto zachować, dlatego nie będziemy usuwać marginesu dolnego w obrazkach poziomych, tylko przeniesiemy go w całości na górę.

```
div.ls img {height: 96px; width: 128px; margin: 32px 0 0;}
```


Na razie wygląda to nieźle, ale nie pomyśleliśmy o jeszcze jednej sprawie — wysokości elementu `div`. Nadal nadajemy im wysokość `130px` uzupełnioną o niewielkie dopełnienie, co w sumie oznacza, że podpisy mają szansę wysunięcia się poza ramy elementu. Aby to zobrazować, a jednocześnie nie zakłócać zbytnio toku naszych prac, dodajmy do elementu `div` delikatną ramkę, taką jak przedstawiona na rysunku 2.9.

```
div.pic {float: left; height: 130px; width: 130px;
padding: 15px; margin: 5px 3px; border: 1px dotted silver;}
```

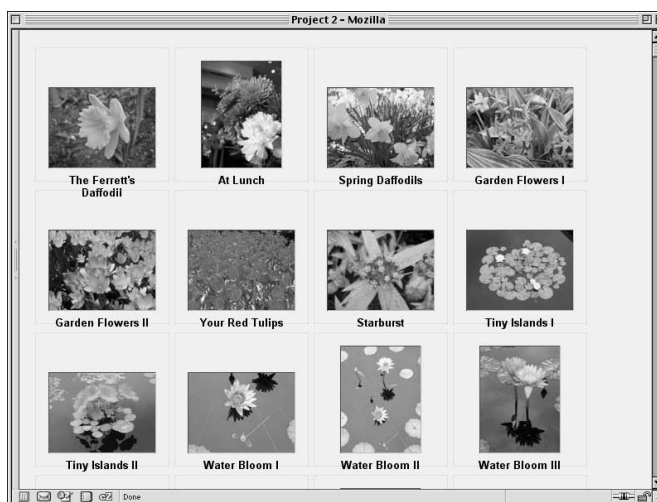
RYСУNEK 2.9.

*Już prawie gotowe,
jeszcze tylko kilka kroków*



Uwaga na Explorera

Tak jak poprzednio, błąd Internet Explorera dla Windows związany z parametrem `height` spowoduje, że generowana przez niego strona będzie wyglądała nieco inaczej niż pokazana na rysunku 2.9.



Czyszczenie

Na tym etapie wystarczy, że powiększymy wysokość elementu `div` o tyle, żeby zmieścił się w nim tytuł obrazu. Po co w ogóle się tym zajmować? Pierwszym powodem jest to, że taki układ strony nie będzie wyglądał dobrze w Internet Explorerze dla Windows. Drugim — fakt, że w przypadku gdy długi tytuł znajdzie się nad pionowym obrazkiem, istnieje prawdopodobieństwo, że będą się na siebie nakładać. Trzeci powód jest natury estetycznej. Pozostawienie elementów, które „wystają” poza ramy ich elementów nadrzędnych, wydaje się być przejawem braku staranności (przynajmniej na razie, w dalszej części projektu postaramy się wykorzystać to właśnie zachowanie).

Wobec tego usuniemy dolne i górne dopełnienie elementu `div` i odpowiednio poprawimy ich wysokość. Ach, nie można też zapomnieć o usunięciu srebrnej ramki.

```
div.pic {float: left; height: 190px; width: 130px;
padding: 0 15px; margin: 5px 3px;}
```

Przez wprowadzenie tej zmiany zakończyliśmy przygotowywanie arkusza stylów galerii, który można w całości zobaczyć na listingu 2.3, a efekt jego działania — na rysunku 2.10.



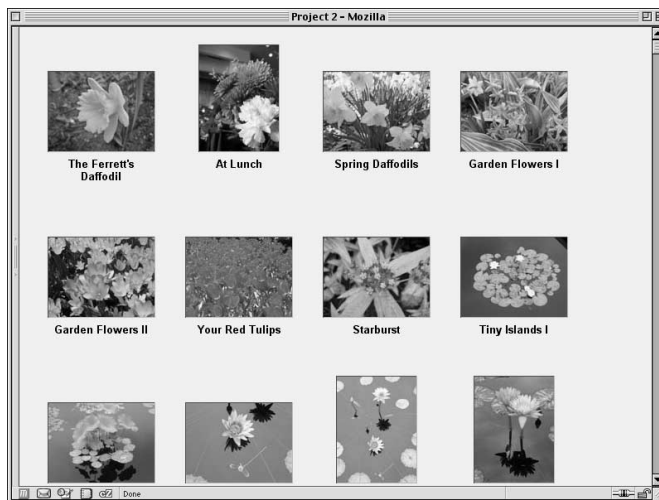
Teraz dobrym posunięciem byłoby zapisanie naszego projektu do osobnego pliku, bo w kolejnym podrozdziale rozpoczniemy prace od nowa, usuwając z niego większość stylów.

LISTING 2.3. Arkusz stylów „galerii”

```

body {background: #EED; margin: 1em;}
div#footer {clear: both; padding-top: 3em;
  font: 85% Verdana, sans-serif;}
div.pic {float: left; height: 190px; width: 130px;
  padding: 0 15px; margin: 5px 3px;}
div.pic img {border: 1px solid; border-color: #444 #AAA #AAA #444;}
div.ls img {height: 96px; width: 128px; margin: 32px 0 0;}
div.pt img {height: 128px; width: 96px; margin: 0 16px;}
div.pic ul {margin: 0.25em 0 0; padding: 0;
  font: bold small Arial, Verdana, sans-serif;}
li.title {display: block; text-align: center;}
li.catno, li.price {display: none;}

```

**RYSUNEK 2.10.**

*Galeria piękności
ogrodowych*

Wysokość i tabele

Wysokość 190 pikseli została wybrana w ten sposób, aby mieściły się również tytuły tak długie, że wymagałyby podzielenia na trzy wiersze. Niestety, w tym rozwiązaniu można się dopatrzeć pewnej wady. Co się stanie, w momencie gdy użytkownik powiększy czcionkę albo tytuł rozciągnie się na pięć lub sześć wierszy? W takim przypadku układ strony zostanie zniszczony, a my nie mamy sposobu na jego poprawienie.

Jest to nieuniknione ograniczenie elementów opływanych. Po prostu każdy taki element stanowi samotną wyspę; jego wymiary w ogóle nie są zależne od jakiegokolwiek innego elementu. Trzeba tutaj powiedzieć, że jedynymi elementami, które będą dostosowywały swoją wysokość do wysokości sąsiadujących z nimi elementów, są komórki tabeli.



Układ tabelaryczny bez tabel!

Istnieje jednak rozwiązanie alternatywne w stosunku do tabel i opływania. Można zastosować podany tu kod i do poszczególnych elementów `div` dołączyć wartości `display` opisujące tabele. Można by na przykład zadeklarować reguły `div.row` `{display: table-row;}` i `div.row div {display: table-cell;}`. W ten sposób utworzylibyśmy strukturę tabelaryczną składającą się z elementów nietabelarycznych, która działałaby w najnowszych przeglądarkach, oczywiście z wyjątkiem Internet Explorera.

To oznacza, że czasami tabela jest najlepszą metodą tworzenia układu galerii zdjęć. Jeżeli znajdziemy się w sytuacji, w której nie jest możliwe określenie długości ich podpisów, albo chcemy, żeby każdy obrazek umieszczony był w polu o takiej samej wielkości, jak pole największego obrazka w wierszu, to jesteśmy skazani na użycie tabel. Oczywiście, taką tabelę można stylizować za pomocą CSS i wielu technik opisywanych w tej książce.

Inna sytuacja, w której opłacalne jest stosowanie tabeli, to taka, gdy w jednym wierszu musimy umieścić określoną liczbę obrazków. Można oczywiście podzielić strukturę dokumentu za pomocą CSS. W ramach przykładu proszę przejrzeć poniższy kod.

```
<div class="wiersz">
  <div class="pic ls">...</div>
  <div class="pic pt">...</div>
  <div class="pic ls">...</div>
  <div class="pic ls">...</div>
</div>
```

Dzięki zgrupowaniu czterech obrazków zyskujemy pewność, że w wierszu zawsze będzie ich tylko cztery. Ale po co tak komplikować sprawę, skoro moglibyśmy po prostu użyć tabeli? Podany wyżej kod jest praktycznie odpowiednikiem kodu tabeli. Proszę spojrzeć:

```
<tr>
  <td class="pic ls">...</td>
  <td class="pic pt">...</td>
  <td class="pic ls">...</td>
  <td class="pic ls">...</td>
</tr>
```

Ten kod jest znacznie prostszy, wymaga wpisania mniejszej liczby znaków i pozwala łatwo ograniczyć liczbę obrazków wyświetlanych w jednym wierszu. Do tego celu nadają się właśnie tabele.

Proszę jednak pamiętać, że naszym nadrzędnym zamierzeniem było umożliwienie obrazkom „płynięcia” tak, żeby każdy wiersz składał się z jak największej liczby obrazków pozwalających wyświetlić się w oknie przeglądarki. Nie można tego osiągnąć za pomocą tabel, a tylko wykorzystując opływanie. Oznacza to, że jak zwykle należy odpowiednio wybierać narzędzia w zależności od stojącego przed nami zadania.

TWORZENIE WIDOKU KATALOGU

Dowiedzieliśmy się, jak należy tworzyć „pływające” tabele z obrazkami. Spróbujmy teraz nieco innego rozwiązania. Tym razem przygotujemy pionowy zestaw obrazków, obok których znajdować się będą informacje o tytule zdjęcia, numerze katalogowym i cenie. W tym celu będziemy musieli pozbyć się praktycznie wszystkich przygotowanych wcześniej stylów i zacząć prace od początku. Zatrzymamy tylko style ciała i stopki strony, co zaprezentowano na listingu 2.4. Praktycznie wróciliśmy do stanu, który mieliśmy na samym początku projektu (dla przypomnienia proszę spojrzeć na rysunek 2.1).

LISTING 2.4. *Zaczynamy od początku*

```
body {background: #EED; margin: 1em;}
div#footer {clear: both; padding-top: 3em;
font: 85% Verdana, sans-serif;}
```

Znów opływanie

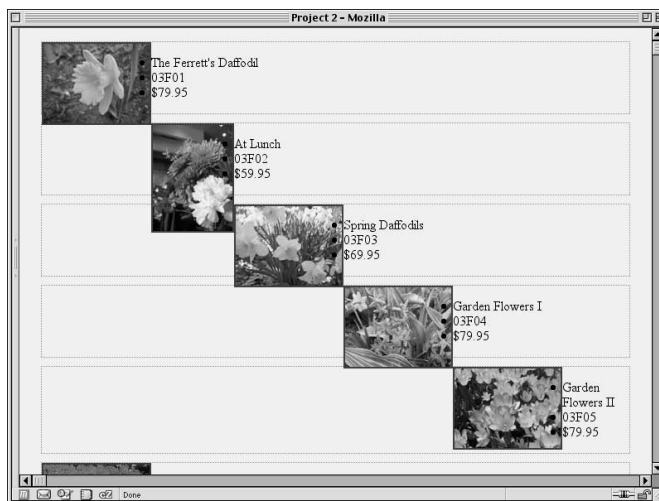
Aby ograniczyć długość strony tworzonej w takim układzie, umieścimy tytuł i inne informacje tekstowe obok obrazka, a nie pod nim. Umieszczenie tekstu obok obrazka jest bardzo proste; wystarczy włączyć jego opływanie. W tym jednak przypadku włączymy opływanie łącza zawierającego obrazek.

```
div#footer {clear: both; padding-top: 3em;
font: 85% Verdana, sans-serif;}
div.pic a.tn {float: left;}
</style>
```

Opływanie łącza powoduje, że razem z nim opływany jest też obrazek, podobnie jak było to w opływanym elemencie `div`. W obu tych przypadkach opływane są wszystkie elementy zawarte w elemencie opływanym.

Elementy `div` uzupełnimy też o odpowiednie style. Niewielki margines i ramka umożliwiająca nam śledzenie, jak postępują prace nad układem. Wynik tych zmian pokazano na rysunku 2.11.

```
div#footer {clear: both; padding-top: 3em;
font: 85% Verdana, sans-serif;}
div.pic {margin: 10px; padding: 0; border: 1px dotted gray;}
div.pic a.tn {float: left;}
```

**RYСУNEK 2.11.**

Jeżeli nie jest się ostrożnym, opływanie może być niebezpieczne

O rety! Jeżeli chcielibyśmy uzyskać taki efekt, to nie ma sprawy, ale nam w ogóle nie o to chodziło!

Co się stało? Dokładnie to, o co prosiłimy. Włączając opływanie elementów powodujemy usunięcie ich z normalnego przepływu dokumentu, co oznacza, że nie wpływają one na wysokość swoich elementów nadrzędnych. W efekcie wysokość każdego elementu `div` wyznaczana jest przez normalnie przepływający tekst, czyli listę informacyjną.

Co więcej, elementy opływane umieszczane są w miejscu, w którym znalazłyby się, gdyby normalnie uczestniczyły w przepływie tekstu i dopiero wtedy poddawane są opływowi. W tym przypadku opływane są z prawej strony, wobec czego druga, będąca łączem miniaturka umieszczana jest na szczycie swojego elementu `div`, po jej lewej stronie. Jednak przed osiągnięciem lewej krawędzi strony natyka się na inny element opływany, czyli pierwsze łącze. Zostaje więc zatrzymane i umieszczone po prawej stronie poprzedniego elementu opływanego. To samo dzieje się z trzecim, czwartym łączem i kolejnymi. Każde z nich zatrzymywane jest na prawej krawędzi poprzedniego obrazka.



Nie do końca rozumiem...

To wyjaśnienie może być trochę niejasne. Jeżeli tak jest, to proszę przeczytać je powoli kilka razy. W tym czasie proszę pomyśleć, co się stanie, jeżeli włączymy opływanie w serii obrazków, przy których nie ma żadnego tekstu. Mniej więcej to samo dzieje się tutaj, z tą różnicą, że tekst nieco bardziej komplikuje sprawę.

Sytuację tę można naprawić na kilka sposobów. Jeżeli chcielibyśmy, aby element `div` każdego obrazka „rozciągał się” wokół opływanego łącza, to można również włączyć opływanie samych elementów. W aktualnych przeglądarkach (oraz w specyfikacji CSS 2.1) elementy opływane rozciągną się tak, by objąć wszystkie zawarte w sobie opływane elementy podrzędne. Takie rozwiązanie nie jest nam tutaj potrzebne, ponieważ kropkowane ramki i tak zostaną później usunięte. Niezbędne są nam one tylko do celów diagnostycznych.

W związku z tym spróbujemy wyczyścić element `div`. Skoro elementy opływane są po prawej stronie, to możemy włączyć czyszczenie od lewej strony. Ustalimy też szerokość elementu `div`, ponieważ i tak później będziemy chcieli ją ograniczyć.

```
div.pic {margin: 10px; padding: 0; border: 1px dotted gray;
clear: left; width: 350px;}
```

Zapis ten spowoduje „zepchnięcie” elementu `div` w dół każdego elementu opływanego znajdującego się przed nim i z jego lewej strony. Wynik tej poprawki zobaczyć można na rysunku 2.12.

RYСУNEK 2.12.

Usuwanie problemów z opływaniem



Ułożenie i wyrównanie

Na rysunku 2.12 z pewnością zauważymy element wymagający poprawienia — pionowe obrazki nie są wyrównane do pozostałych. W tabeli taki obrazek najprawdopodobniej znalazłby się w osobnej komórce, w której moglibyśmy dosunąć go do prawej krawędzi. Niestety, tutaj nie możemy skorzystać z tego luksusu. Na szczęście możemy jeszcze pobawić się troszkę marginesami opływających łączy i za pomocą tej metody uzyskać pożądany efekt. Trzeba tylko zrównać jego szerokość z szerokością znajdującego się wewnątrz obrazka i tak ustalić lewy margines, żeby zrównać szerokość tych łączy z szerokością łączy zawierających obrazki poziome.

```
div.pic a.tn {float: left;}
div.pt a.tn {width: 96px; margin-left: 32px;}
</style>
```

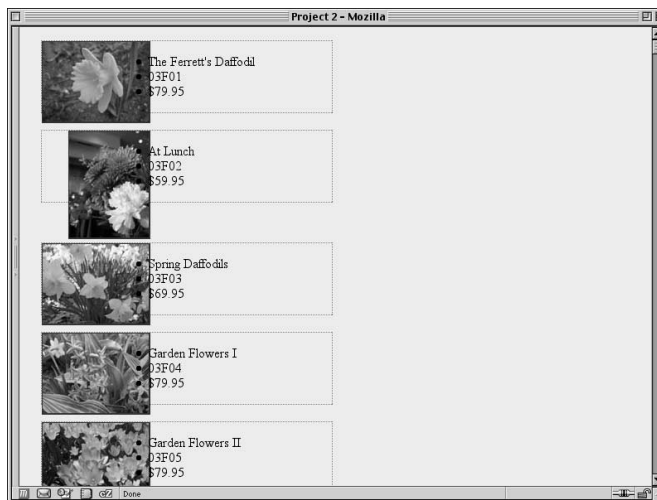
Skoro podaliśmy już szerokość łączy z obrazkami pionowymi, nic nie stoi na przeszkodzie, żeby podobnie określić szerokość łączy z obrazkami poziomymi. Z pewnością nie zaszkodzi, a w przyszłości może okazać się to przydatne.

```
div.pt a.tn {width: 96px; margin-left: 32px;}
div.ls a.tn {width: 128px;}
</style>
```

Usuńmy teraz te paskudne niebieskie obramowania rysunków i zastąpmy je czymś troszkę mniej jaskrawym.

```
div.ls a.tn {width: 128px;}
a.tn img {border: 1px solid #333; border-width: 1px 2px 2px 1px;}
</style>
```

Nieco grubsza dolna i prawa ramka wytworzy ledwie zauważalny „cienie” rzucany przez każdy rysunek. Można go zauważyć na niektórych obrazkach przedstawionych na rysunku 2.13.



RYSUNEK 2.13.

Wyrównanie rysunków i uzupełnienie ich o cienie tworzone przez ramki

Po dokładnym przyjrzeniu się rysunkowi 2.13 zauważyć można dwie rzeczy. Pierwszą jest to, że tekst i ramki obrazków lekko na siebie nachodzą. Wynika to stąd, że rysunki są teraz większe niż zawierające je łącza z powodu ramek, które dodaliśmy do rysunków. Teraz dobrze by było poprawić szerokości samych łączy. Spróbujmy nadać im szerokość trochę większą niż szerokość obrazków.

```
div.pt a.tn {width: 100px; margin-left: 32px;}
div.ls a.tn {width: 130px;}
```

Drugą rzeczą, jaką widać na rysunku 2.13, jest nakładanie się znaków wypunktowania na obrazki. Jest to wynik działania opływania elementów i umiejscawiania znaków wypunktowania. Co ciekawe, nie dzieje się tak we wszystkich przeglądarkach. Takie zachowanie mogłoby sprawiać nam problemy, gdybyśmy obok obrazków chcieli umieszczać listy ze znakami wypunktowania. Skoro chcemy i tak pozbyć się tych znaków, to właściwie nie ma to dla nas znaczenia.

```
a.tn img {border: 1px solid #333; border-width: 1px 2px 2px 1px;}
div.pic li {list-style: none;}
</style>
```

W ten sposób likwidujemy nakładanie się znaków wypunktowania na obrazki, ale nadal pozostawiamy teksty „przyklejone” do prawych krawędzi obrazków. Odsuniemy je od siebie przez zastosowanie kombinacji marginesów i dopełnienia. Dodamy też z lewej strony ramkę, dzięki której będziemy wiedzieć, gdzie znajduje się początek pola opisów. Wynik tych operacji zaprezentowano na rysunku 2.14.

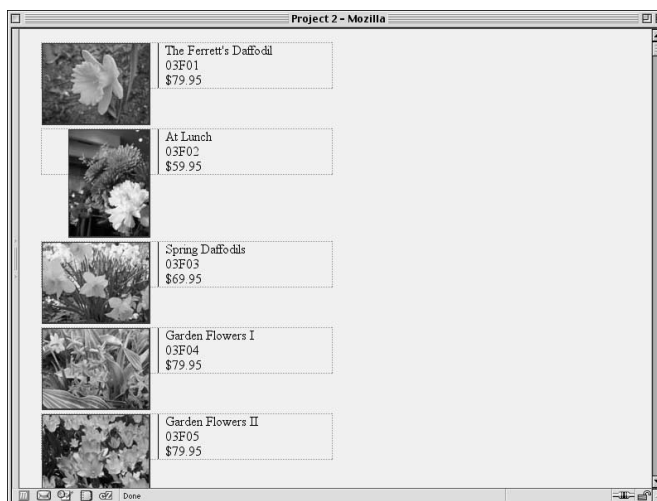
```
a.tn img {border: 1px solid #333; border-width: 1px 2px 2px 1px;}
div.pic ul {margin: 0 0 140px; padding: 0 0 0 0.5em;
border-left: 1px solid;}
div.pic li {list-style: none;}
```

Umieszczenie znaków wypunktowania

Znaki wypunktowania umiejscawiane są względem lewego brzegu opisów elementów listy. Niezależnie od tego, gdzie znajduje się ten brzeg, znaki wypunktowania umieszczane są nieco na lewo od niego. W naszym przypadku lewe krawędzie tekstów listy dosunięte są do prawej krawędzi opływanych obrazków, dlatego znaki wypunktowania trafiają na obrazki.

RYSUNEK 2.14.

Marginesy i dopełnienia ułatwiają oddzielenie tekstu od obrazków



I jak? Lewy margines o wielkości 140 pikseli tak naprawdę „wślizguje się” pod opływane łącza, opierając się na lewym brzegu elementu `div`. Po przyjrzeniu się drugiemu elementowi `div` na rysunku 2.14 dostrzec można, że marginesy te rozciągają się również po lewej stronie pionowego obrazka. W rzeczywistości dzieje się tak przy wszystkich rysunkach, ale jest to zauważalne jedynie w przypadku rysunków pionowych.

Poprawianie listy

Skoro udało się nam już wyrównać wszystkie łącza i obrazki, a towarzyszące im listy zostały nieco odsunięte od obrazków, możemy zająć się tekstami znajdującymi się na tych listach. Na początku musimy pozbyć się ramek, bo zaczynają troszkę przeszkadzać. W związku z tym konieczne jest zmodyfikowanie reguły `div.pic` tak, by wyglądała ona następująco:

```
div.pic {margin: 10px; padding: 0;
clear: left; width: 350px;}
```

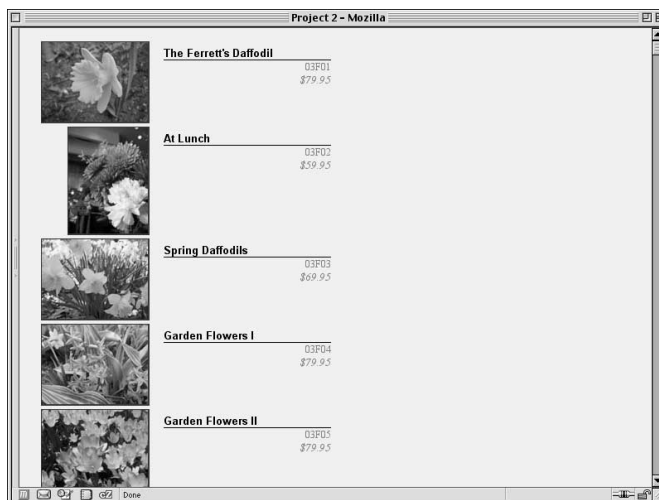
W podobny sposób należy zmodyfikować regułę `div.pic ul`. Powinna ona wyglądać tak:

```
div.pic ul {margin: 0 0 0 140px; padding: 0 0 0 0.5em;}
```

Teraz można zająć się wyróżnianiem tytułów. Powinny one nieźle wyglądać, jeżeli nadalibyśmy im pogrubioną czcionkę *sans-serif* i uzupełnili o dolną ramkę. Należałoby je też lekko obniżyć w stosunku do opływanych łączy (powiedzmy, o pół wartości `em`).

Zajmijmy się numerem katalogowym i ceną. Nie są one tak ważne, jak tytuł zdjęcia, dlatego stopimy je lekko z tłem i nadamy nieco ciemniejszy odcień koloru, jaki wykorzystaliśmy w tle. Poza tym wyrównamy je do prawej strony i wyróżnimy cenę przez wyświetlenie jej czcionką w kursywie. Efekty tych działań można podziwiać na rysunku 2.15.

```
div.pic li.title {font: bold small Arial, Verdana, sans-serif;
padding-top: 0.5em; border-bottom: 1px solid;}
div.pic li.catno {color: #776; text-align: right;}
div.pic li.price {color: #776; text-align: right;
font-style: italic;}
</style>
```



RYСУNEK 2.15.

Po dodaniu koloru i odpowiednim wyrównaniu, tekst wygląda o wiele lepiej

Właściwie można by na tym poprzestać, ale proponuję jeszcze troszkę popracować. Myślę, że gdyby numer katalogowy i cena znajdowały się obok siebie, w jednym wierszu, to wyglądałoby to wspaniale. Może nie wspaniale, ale bardziej elegancko na pewno. W każdym razie warto spróbować. Dodatkowo postaramy się osiągnąć taki efekt nie wprowadzając do układu strony żadnych dodatkowych elementów opływanych.

W celu realizacji tych zamierzeń będziemy musieli podciągnąć tekst zawierający cenę o jeden wiersz do góry. Jeżeli ten **wybieg** ma działać, musimy się upewnić, że we wszystkich przeglądarkach wysokość wiersza będzie identyczna i numer katalogowy nie będzie nam przeszkadzał. Ustalimy jednolitą wysokość wiersza przez zdefiniowanie parametru `line-height` i utworzenie zerowych marginesów i dopełnień.

```
div.pic li {list-style: none; font-size: small;
  line-height: 1.2em; margin: 0; padding: 0;}
```

Teraz trzeba przesunąć numer katalogowy. Dodamy do niego po prostu prawy margines o wielkości 4.5em.

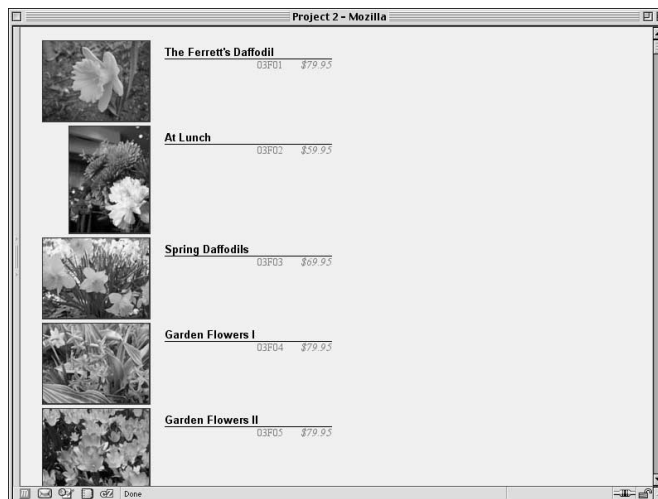
```
div.pic li.catno {color: #776; text-align: right;
  margin-right: 4.5em;}
```

Trzeba już tylko wsunąć cenę w miejsce, które dla niej przygotowaliśmy. Proszę sobie przypomnieć, jaką wartość nadaliśmy parametrowi `line-height`. Tę samą wartość musimy nadać górnemu marginesowi, by tekst znalazł się we właściwym miejscu. Wynik można zobaczyć na rysunku 2.16.

```
div.pic li.price {color: #776; text-align: right;
  font-style: italic; margin: -1.2em 0 0 0;}
```

RYСУNEK 2.16.

*Dzięki magii marginesów
można umieścić tekst
przy tekście*



Informacje do ramek

Teraz wykonamy kolejną operację mającą na celu poprawienie wyglądu strony. Umieścimy pionową ramkę po lewej stronie numeru katalogowego i kolejną między numerem katalogowym a ceną. Następnie wszystkie informacje tekstowe zamknijemy wewnątrz ładnie wyglądającej ramki.

Ramki ustawimy w potrzebnych nam miejscach przez nadanie klasom numeru katalogowego i ceny podobnych stylów. Podstawą postępowania jest nadanie im pewnej szerokości, a następnie odepchnięcie ich w prawo na żadaną odległość. W związku z tym dla klasy ceny musimy zapisać deklaracje:

```
div.pic li.price {color: #776; text-align: right;
font-style: italic; margin: -1.2em 0 0 auto;
width: 4em; border-left: 1px solid;}
```

Następnie w klasie numeru katalogowego zamieniamy parametr `margin-right` parametrem `margin` i nadajemy mu tę samą szerokość oraz ramkę, którą nadaliśmy klasie ceny. Efekty przedstawiono na rysunku 2.17.

```
div.pic li.catno {color: #776; text-align: right;
margin: 0 4.5em 0 auto;
width: 4em; border-left: 1px solid;}
```



W kolejnym kroku otoczmy wszystkie informacje tekstowe jednolitą ramką. Tutaj niestety będzie troszkę trudniej. Jeżeli nadalibyśmy po prostu elementom `ul` ramkę i tło, to IE6 dla Windows pogubi się całkowicie i samowolnie zacznie ukrywać te elementy. W takim razie, żeby nie narażać biedaczka na stresy, ukryjemy przed nim te „nieprzyzwoite” style.

```
div.pic ul {margin: 0 0 0 140px; padding: 0 0 0 0.5em;}
html>body div.pic ul {background: #CCB; border: 3px double #552;}
div.pic li {list-style: none; font-size: small;
line-height: 1.2em; margin: 0; padding: 0;}
```



Uwaga na Explorera

Właściwie cała dalsza część projektu nie będzie działała w Internet Explorerze 5.x dla Windows, ponieważ przeglądarka ta nie obsługuje automatycznych marginesów. Pierwsza część tego punktu zadziała w Internet Explorerze 6.0 pod warunkiem, że będzie on działał w trybie zgodności ze standardami.

RYСУNEK 2.17.

Ramki tworzone przez magię marginesów

Pierwsza część selektora, czyli zapis `html>body`, powoduje ukrycie reguły przed Internet Explorerem dla Windows. Jest to pełnoprawny selektor potomny opisany w specyfikacji CSS2, ale IE dla Windows nie jest w stanie go prawidłowo rozpoznać. Bardziej zaawansowane przeglądarki nie mają z tym problemów, więc zastosują odpowiednie style w czasie rysowania strony.

Podczas dodawania ramek musimy pamiętać o tym, żeby cenę przesunąć lekko w lewo, by nie stykała się ona z podwójną, zewnętrzną ramką. Wykonamy to wprowadzając niewielką poprawkę w prawym marginesie ceny.

```
div.pic li.price {color: #776; text-align: right;
font-style: italic; margin: -1.2em 3px 0 auto;
width: 4em; border-left: 1px solid;}
```

W końcu musimy przygotować odstępy między poszczególnymi pozycjami w widoku katalogu. Można wprowadzić je na dwa sposoby: przez dodanie dolnego marginesu do opływanych łączy albo do znajdujących się wewnątrz nich obrazków. Dodawanie górnych marginesów nic nam nie da, z powodu deklaracji `clear`, jaką wcześniej wprowadziliśmy do stylów. Aby uniknąć paskudnych błędów w Internet Explorerze, marginesy umieścimy na dole obrazków.

```
a.tn img {border: 1px solid #333; border-width: 1px 2px 2px 1px;
margin: 0 0 1em;}
```

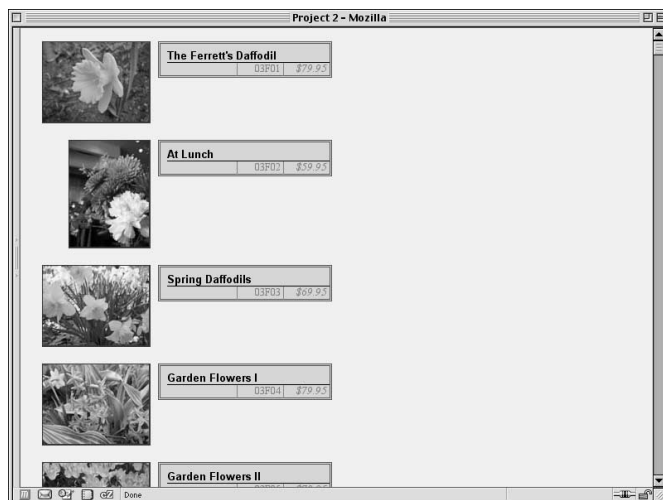
W ten sposób uzyskaliśmy arkusz stylów zaprezentowany na listingu 2.5, który generuje widok katalogu przedstawiony na rysunku 2.18.

LISTING 2.5. Kompletny arkusz stylów „katalogu”

```
body {background: #EED; margin: 1em;}
div#footer {clear: both; padding-top: 3em;
font: 85% Verdana, sans-serif;}
div.pic {margin: 10px; padding: 0;
clear: left; width: 350px;}
div.pic a.tn {float: left;}
div.pt a.tn {width: 100px; margin-left: 32px;}
div.ls a.tn {width: 132px;}
a.tn img {border: 1px solid #333; border-width: 1px 2px 2px 1px;
margin: 0 0 1em;}
div.pic ul {margin: 0 0 0 140px; padding: 0 0 0 0.5em;}
html>body div.pic ul {background: #CCB; border: 3px double #552;}
div.pic li {list-style: none; font-size: small;
line-height: 1.2em; margin: 0; padding: 0;}
div.pic li.title {font: bold small Arial, Verdana, sans-serif;
padding-top: 0.5em; border-bottom: 1px solid;}
div.pic li.catno {color: #776; text-align: right;
margin: 0 4.5em 0 auto;
width: 4em; border-left: 1px solid;}
div.pic li.price {color: #776; text-align: right;
font-style: italic; margin: -1.2em 3px 0 auto;
width: 4em; border-left: 1px solid;}
```

Reguły clear

Deklaracja `clear` powoduje powiększenie górnego marginesu czyszczonego elementu tak, by jego górne obramowanie znalazło się tuż pod dolnym obramowaniem poprzedniego elementu opływanego. W związku z tym próby ustawiania górnego marginesu elementu `div` zostałyby anulowane przez mechanizm deklaracji `clear`.



RYSUNEK 2.18.

Informacje
prezentowane są
w stylu katalogu

WARIANTY

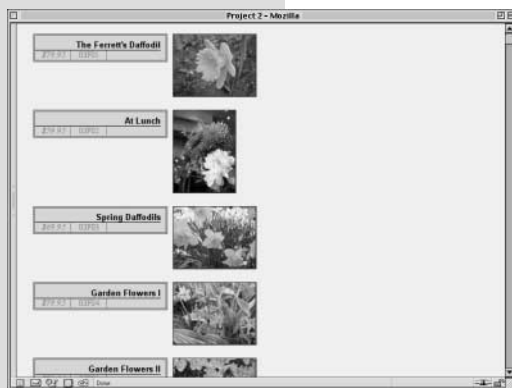
Proszę spróbować odtworzyć następujące zmiany w działaniu naszego projektu.

1. W arkuszu prezentacji należy umieścić tytuł obrazka zaraz pod nim, tak by wyglądał on jak zapisany na ramce slajdu. W tym celu konieczne będzie usunięcie dolnego marginesu rysunków bez jednoczesnego niszczenia całego układu. Proszę też pamiętać, że nie będzie to możliwe w przypadku rysunków pionowych, ponieważ CSS nie dysponuje możliwością obracania tekstu. Trzeba będzie zatem odpowiednio ograniczyć style. Proszę też pamiętać, że dłuższe tytuły mogą wypłynąć poza ramkę, wobec czego przydatną może okazać się właściwość `overflow`.





2. W widoku galerii do wyświetlanych elementów należy dołączyć numer katalogowy i cenę, ale umieścić je obok siebie, a nie jedno nad drugim. Powinno to rozszerzyć widok galerii bez jednoczesnej zmiany układu strony. Proszę pamiętać, że przydałoby się podnieść nieco wysokość pól obrazków.



3. W widoku katalogu należy spróbować odwrócić kolejność elementów, tak by obrazek znajdował się po prawej stronie, a informacje tekstowe po lewej. Wymaga to czegoś więcej niż tylko zmiany kierunku opływania. Konieczna będzie też modyfikacja układu elementów listy.