

ZWIĘKSZ WYDAJNOŚĆ SWOJEJ PRACY!

Apress®

HTML5 i CSS3 Zaawansowane wzorce projektowe

Michael Bowers, Dionysios Synodinos, Victor Sumner

Helion



Tytuł oryginału: Pro HTML5 and CSS3 Design Patterns

Tłumaczenie: Łukasz Piwko

ISBN: 978-83-246-4471-1

Original edition copyright © 2011 by Michael Bowers, Dionysios Synodinos, and Victor Sumner.
All rights reserved.

Polish edition copyright © 2012 by HELION S.A.
All rights reserved.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from the Publisher.

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiejkolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie książki na nośniku filmowym, magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi bądź towarowymi ich właścicieli.

Autor oraz Wydawnictwo HELION dołożyli wszelkich starań, by zawarte w tej książce informacje były kompletne i rzetelne. Nie biorą jednak żadnej odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub autorskich. Autor oraz Wydawnictwo HELION nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Wydawnictwo HELION
ul. Kościuszki 1c, 44-100 GLIWICE
tel. 32 231 22 19, 32 230 98 63
e-mail: helion@helion.pl
WWW: <http://helion.pl> (księgarnia internetowa, katalog książek)

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

<http://helion.pl/user/opinie/html5z>

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Printed in Poland.

- [Kup książkę](#)
- [Poleć książkę](#)
- [Oceń książkę](#)

- [Księgarnia internetowa](#)
- [Lubię to! » Nasza społeczność](#)

Spis treści

O autorach	13
O korektorze merytorycznym	15
Podziękowania	17
Wstęp	19
Adresaci	19
Innowacje	20
Konwencje	23
Korzystanie z tej książki	25
Jak podzielona jest książka	25
Kod źródłowy	27
Używanie kodu źródłowego	27
Kontakt z autorami	27
Rozdział 1. Wzorce projektowe, czyli koniec problemów z CSS	29
Wzorce projektowe — przepisy strukturalne	30
Używanie wzorców projektowych	30
Używanie arkusza stylów	35
Składnia CSS	36
Kaskadowość	40
Upraszczenie kaskady	42
Przydatne adresy	43
Często używane własności CSS	44
Własności i wartości CSS: ogólne	45
Własności i wartości CSS: treść	45
Własności i wartości CSS: układ	46
Własności i wartości CSS: specjalne	47
Selektory	47
Zapytania o media	48
Elastyczne jednostki miary	49
Stałe jednostki miary	50
Przeliczniki jednostek dla 96 dpi	50
Typowe rozmiary pisma przy 96 dpi	50

	Przejścia, animacje i przekształcenia dwuwymiarowe	51
	Rozwiązywanie problemów z CSS	52
	Znormalizowany arkusz stylów	53
Rozdział 2.	Wzorce projektowe HTML	57
	Streszczenie	57
	Struktura HTML	58
	Struktura HTML	60
	XHTML	64
	DOCTYPE	66
	DOCTYPE	67
	Elementy Nagłówka	69
	Arkusz Warunkowy	71
	Strukturalne Elementy Blokowe	73
	Terminalne Elementy Blokowe	75
	Wszechstronne Elementy Blokowe	77
	Elementy Śródliniowe	79
	Atrybuty Klasy i Identyfikatora	81
	Białe Znaki w HTML	83
Rozdział 3.	Selektory i dziedziczenie w CSS	87
	Streszczenie	87
	Selektory Typu, Klasy i Identyfikatora	88
	Selektory Pozycji i Grupowanie Selektorów	90
	Selektory Atrybutów	92
	Selektory Pseudoelementów	95
	Selektory Pseudoklas	97
	Selektor Podklasy	99
	Dziedziczenie	101
	Dziedziczenie Wizualne	103
Rozdział 4.	Modele polowe	105
	Streszczenie	105
	Rodzaje Pól	106
	Model Polowy	108
	Pole Śródliniowe	110
	Pole Śródliniowo-Blokowe	112
	Pole Blokowe	114
	Pole Tabeli	116
	Pole Pozycjonowane Bezwzględnie	118
	Pole Pływające	120
Rozdział 5.	Określanie wymiarów pól	123
	Streszczenie	123
	Szerokość	124
	Wysokość	126
	Konkretny Rozmiar	128
	Kurczliwy	130
	Rozciągnięty	132

Rozdział 6.	Własności modelu polowego	135
	Streszczenie	135
	Margines	136
	Obramowanie	138
	Dopełnienie	141
	Tło	143
	Przepelnienie	145
	Widoczność	147
	Łamanie Stron	149
Rozdział 7.	Modele pozycjonowania	151
	Streszczenie	151
	Modele Pozycjonowania	152
	Pozycjonowany	154
	Najbliższy Pozycjonowany Przodek	156
	Kontekst Stosowy	158
	Wewnętrzny	160
	Statyczny	162
	Pozycjonowanie Bezwzględne	164
	Pozycjonowanie Stałe	166
	Pozycjonowanie Względne	168
	Elementy Pływające	170
	Względny Element Pływający	172
Rozdział 8.	Pozycjonowanie: wcinanie, przesuwanie i wyrównywanie elementów	175
	Streszczenie	175
	Wcięcia	176
	Przesuwanie Elementów Statycznych	178
	Przesuwanie i Wcinanie Statycznych Tabel	180
	Przesuwanie Elementów Pływających	182
	Przesuwanie Elementów Bezwzględnych i Stałych	184
	Przesuwanie Elementów Względnych	186
	Wyrównywanie Statycznych Elementów Śródliniowych	188
	Wyrównywanie i Przesuwanie Statycznych Bloków	190
	Wyrównywanie i Przesuwanie Statycznych Tabel	192
	Wyrównywanie i Przesuwanie Elementów Bezwzględnych	194
	Środkowanie Elementów Bezwzględnych	196
	Wyrównywanie Zewnętrzne	198
Rozdział 9.	Pozycjonowanie — techniki zaawansowane	201
	Streszczenie	201
	Wyrównanie do Lewej	202
	Przesuwanie Elementów Wyrównanych do Lewej	204
	Wyrównanie do Prawej	206
	Przesuwanie Elementów Wyrównanych do Prawej	208
	Wyrównanie do Środka	210
	Przesuwanie Elementów Wyrównanych do Środka	212
	Wyrównanie do Góry	214
	Przesuwanie Elementów Wyrównanych do Góry	216

Wyrównanie do Dołu	218
Przesuwanie Elementów Wyrównanych do Dołu	220
Wyśrodkowanie Pionowe	222
Przesuwanie Elementów Wyśrodkowanych Pionowo	224
Rozdział 10. Formatowanie tekstu	227
Streszczenie	227
Font	228
Wyróżnienie	230
Ozdoby Tekstu	232
Cień Tekstu	234
Zastępowanie Tekstu Grafiką	236
Zastępowanie Tekstu Kanwą i Obiektami VML	238
Osadzanie Fontów	240
Niewidoczny Tekst	241
Tylko dla Czytników Ekranu	242
Rozdział 11. Stosowanie odstępów	245
Odstępy	246
Blokowy	248
Bez Zawijania	250
Zatrzymanie Białych Znaków	251
Kod	253
Dopełnienie Treści	255
Dystans Śródliniowy	257
Ozdoby Śródliniowe	259
Złamanie Wiersza	261
Śródliniowa Linia Pozioma	263
Rozdział 12. Wyrównywanie treści	265
Wcięcie Tekstu	266
Wysunięcie Tekstu	267
Wyrównanie Treści w Poziomie	269
Pionowe Wyrównanie Treści	271
Przesuwanie Treści w Pionie	273
Indeks Górny i Dolny	275
Zagnieżdżanie Kontekstów Wyrównywania	277
Zaawansowany Przykład Wyrównywania	279
Rozdział 13. Bloki	281
Streszczenie	281
Znaczenie Strukturalne	282
Struktura Wizualna	284
Sekcja	286
Listy	288
Punktory Graficzne	290
Śródliniowy	292
Scalanie Marginesów	294
Wsuwany	296
Pozioma Kreska	298

Odstęp między Blokami	300
Redukcja Odstępu między Blokami	301
Lewy Margines	303
Prawy Margines	305
Rozdział 14. Obrazy	309
Streszczenie	309
Grafika	310
Mapa Obrazkowa	312
Gradient	314
Półprzezroczysty	316
Tekst Zastępowany	318
Treść nad Obrazem	319
Treść nad Obrazem Tła	321
Sprite'y CSS	323
Sprite'y CSS, kontynuacja	325
Obraz z Prostym Cieniem	327
Cień Obrazu	329
Cień Obrazu, kontynuacja	331
Cień Obrazu, kontynuacja	332
Zaokrąglone Rogi	334
Zaokrąglone Rogi, kontynuacja	336
Przykładowy Obraz	338
Rozdział 15. Tabele	341
Streszczenie	341
Tabela	342
Grupowanie Wierszy i Kolumn	344
Selektory Tabel	346
Oddzielanie Krawędzi	348
Scalanie Krawędzi	350
Formatowanie Scalanych Krawędzi	351
Ukrywanie i Usuwanie Komórek	353
Ukrywanie i Usuwanie Wierszy i Komórek	355
Pionowe Wyrównanie Danych	357
Tabela w Paski	359
Tabelowy, Wierszowy i Komórkowy	361
Układ Tabeli	363
Rozdział 16. Układy kolumn tabel	365
Modele układu tabel	365
Korzystanie z układów kolumn	366
Streszczenie	366
Szerokość Kolumny	368
Kurczliwe Kolumny	370
Kolumny o Stałej Szerokości	372
Kolumny Proporcjonalne do Treści	374
Kolumny Proporcjonalne do Innych Kolumn	376
Kolumny o Rozmiarach Procentowych	378
Kolumny o Odwróconych Proporcjach	380
Jednakowe Kolumny o Rozmiarze Treści	382

Kolumny o Takich Samych Rozmiarach	384
Niewyrośnięte Kolumny	386
Sprężyste Kolumny	388
Mieszane Układy Kolumn	390
Rozdział 17. Układy stron	393
Streszczenie	393
Układy Płynne — Informacje Ogólne	394
Szerokość Zewnętrzna Pola	396
Projektowanie do wewnątrz, a projektowanie na zewnątrz	398
Pływająca Sekcja	400
Odstępy między Elementami Pływającymi	402
Układ Płynny	404
Naprzeciwległe Elementy Pływające	406
Formatowanie Zdarzeń	408
Sekcje Rozwijane	410
Menu z Zakładkami	413
Zakładki	417
Menu Rozwijane	421
Przycisk	425
Łączy Układowe	429
Układ Wielokolumnowy	431
Szablon	432
Przykładowy Układ	434
Rozdział 18. Inicjały	439
Streszczenie	439
Wyrównany Inicjał	440
Inicjał	442
Wysunięty Inicjał	444
Inicjał z Grafiką i Dopełnieniem	446
Pływający Inicjał	448
Pływający Inicjał z Grafiką	450
Inicjał na Marginesie	452
Graficzny Inicjał na Marginesie	453
Rozdział 19. Myśli przewodnie i cytaty	457
Streszczenie	457
Myśl Przewodnia Spływająca w Lewo	458
Myśl Przewodnia Spływająca w Prawo	460
Wyśrodkowana Myśl Przewodnia	462
Myśl Przewodnia na Lewym Marginesie	464
Myśl Przewodnia na Prawym Marginesie	466
Cytat Blokowy	468
Śródliniowy Cytat Blokowy	470
Cytat Śródliniowy	472

Rozdział 20. Alerty	475
Streszczenie	475
Alert w JavaScript	476
Alert w Chmurce	478
Wyskakujący Alert	479
Wyskakujący Alert	481
Alert	483
Alert Śródliniowy	485
Wysunięty Alert	487
Alert Graficzny	489
Wsuwany Alert	491
Pływający Alert	493
Alert na Lewym Marginesie	495
Alert na Prawym Marginesie	497
Sprawdzanie Poprawności Formularzy	499
Skorowidz	501

ROZDZIAŁ 7



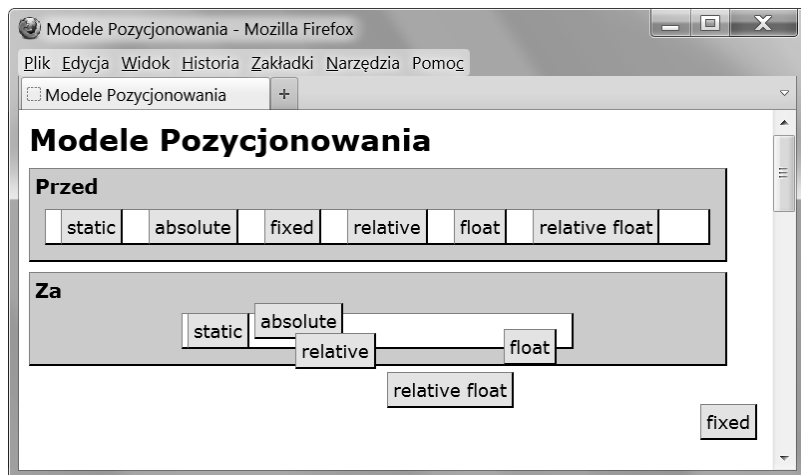
Modele pozycjonowania

Jest to pierwszy z trzech rozdziałów o pozycjonowaniu elementów na stronie. W tym rozdziale poznasz ogólne modele pozycjonowania CSS, w rozdziale 8. — nauczysz się wcinać, przesuwac oraz wyrównywać elementy, a w rozdziale 9. dowiesz się, jak korzystając z wiedzy zdobytej w dwóch poprzednich rozdziałach, tworzyć zaawansowane wzorce projektowe układu.

Streszczenie

- Wzorzec **Modele Pozycjonowania** zawiera opis i przykłady zastosowania sześciu modeli pozycjonowania elementów na stronie.
- We wzorcu **Pozycjonowany** objaśniamy zastosowania czterech ustawień własności `position`: `static`, `absolute`, `fixed` oraz `relative`.
- We wzorcu **Najbliższy Pozycjonowany Przodek** pokazujemy, jak ustawiać położenie bezwzględnie pozycjonowanych pól w odniesieniu do dowolnego przodka, a nie tylko ich rodzica.
- Na przykładzie wzorca **Kontekst Stosowy** pokazujemy, jak sprawić, aby pozycjonowane elementy znajdowały się nad lub pod innymi elementami pozycjonowanymi i statycznymi.
- Wzorzec **Wewnętrzny** zawiera objaśnienie, jak sprawić, aby treść śródliniowa była wyświetlana **wewnątrz** bloku, a nie na nim.
- We wzorcu **Statyczny** przedstawiamy podstawowe wiadomości na temat tzw. układu normalnego elementów.
- We wzorcu **Pozycjonowanie Bezwzględne** pokazujemy, jak wyjąć dowolny element z układu normalnego i określić jego położenie w odniesieniu do *wewnętrznej strony obramowania* jego najbliższego pozycjonowanego przodka.
- Wzorzec **Pozycjonowanie Stałe** pokazuje, jak wyjąć dowolny element z normalnego układu elementów i określić jego położenie w odniesieniu do *obszaru roboczego*.
- We wzorcu **Pozycjonowanie Względne** pokazujemy, jak za pomocą własności pozycjonowania względnego kontrolować rozmieszczenie elementów na osi pionowej oraz jak przesuwac pozycjonowane elementy, nie naruszając ich stanu ani położenia innych elementów.
- Na przykładzie wzorca **Elementy Pływające** pokazujemy, jak sprawić, aby wybrany element automatycznie ustawiał się przy lewej lub prawej krawędzi swojego rodzica oraz aby sąsiednie elementy znajdowały się po jednej z jego stron, po obu jego stronach albo pod nim.
- Wzorzec **Względny Element Pływający** zawiera objaśnienie technik względnego pozycjonowania elementów pływających.

Modele Pozycjonowania



HTML

```
<h1>Modele Pozycjonowania</h1>
<div class="section"><h2>Przed</h2>
  <p><span>static</span><span>absolute</span>
    <span>fixed</span><span>relative</span>
    <span>float</span><span>relative float</span></p></div>

<div class="section"><h2>Za</h2>
  <p class="static centered">
    <span class="static centered">static</span>
    <span class="absolute">absolute</span>
    <span class="fixed">fixed</span>
    <span class="relative">relative</span>
    <span class="float">float</span>
    <span class="relative float">relative float</span></p></div>
```

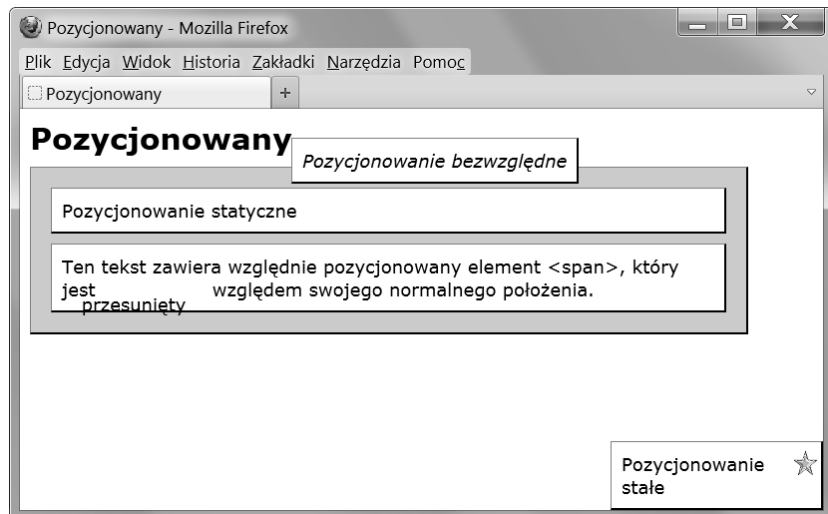
CSS

```
*.centered { width:380px; margin-left:auto; margin-right:auto; }
*.static { position:static; }
*.absolute { position:absolute; top:20px; left:215px; }
*.fixed { position:fixed; bottom:20px; right:5px; }
*.relative { position:relative; top:20px; left:30px; }
*.float { float:right; }
```

Modele Pozycjonowania

Wprowadzenie	To nie jest wzorzec projektowy, lecz wprowadzenie do technik pozycjonowania. W CSS elementy można pozycjonować na sześć sposobów: statycznie, bezwzględnie, na stałe, względnie, pływająco oraz względnie pływająco. Modele pozycjonowania są powiązane z sześcioma rodzajami pól, ale nie są im równoważne. Statycznie można pozycjonować elementy śródliniowe, śródliniowo-blokowe, blokowe i tabele. Modele pozycjonowania bezwzględnego i stałego mają zastosowanie do elementów pozycjonowanych bezwzględnie, które mogą być dowolnego typu. Model pozycjonowania pływającego ma zastosowanie do elementów pływających, które również mogą być dowolnego typu. Pozycjonowanie względne można stosować do wszystkich typów pól oprócz pól pozycjonowanych bezwzględnie. Model pozycjonowania względnego elementów pływających można stosować tylko do pól pływających. W każdym modelu pozycjonowania używa się kombinacji tych samych własności: display, width, height oraz margin. Mimo iż używa się tych samych własności, w każdym modelu działają one w inny sposób. Na przykład ustawienie width:auto powoduje rozciągnięcie bloku statycznego, ale obkurczenie do rozmiaru treści elementu pozycjonowanego bezwzględnie. Widać to w przedstawionym przykładzie, gdzie pierwszy akapit jest rozciągnięty, a bezwzględnie pozycjonowany element <code></code> jest obkurczony. W modelach pozycjonowania używa się także pewnych dodatkowych własności, których zastosowanie różni się w zależności od modelu. W pozycjonowaniu bezwzględnym i stałym położenie pola bezwzględnego kontroluje się za pomocą własności left, right, top, bottom oraz z-index. W pozycjonowaniu względnym używa się własności left, top i z-index, a w pozycjonowaniu elementów pływających — float i clear. Ponieważ podstawowy zestaw własności we wszystkich modelach jest taki sam, konkretne efekty pozycjonowania uzyskuje się poprzez zastosowanie odpowiedniej kombinacji typu elementu, rodzaju pola oraz wartości tych własności. W opisie każdego wzorca znajduje się szczegółowy opis warunków, jakie należy spełnić, aby uzyskać określony rodzaj pozycjonowania. Na przykład ustawienie własności width na konkretną wartość oraz własności margin-left i margin-right na auto powoduje wyśrodkowanie statycznego elementu blokowego, ale nie statycznego elementu śródliniowego. Aby natomiast wyśrodkować element pozycjonowany bezwzględnie, należy dodatkowo ustawić jego własności left i right na 0. Istnieje ponad 50 różnych kombinacji wzorców projektowych pozwalających otrzymać różne rodzaje układu elementów. Wzorze te zostały opisane w rozdziałach 7. – 9. Łatwo jest je sobie przyswoić, ponieważ na wszystkie składają się modele polowe, techniki określania rozmiarów, marginesy oraz własności pozycjonowania. Innymi słowy, wszystkie sześć modeli polowych (śródliniowy, śródliniowo-blokowy, blokowy, tabelowy, bezwzględny i pływający) można łączyć z trzema modelami określania rozmiaru elementów (konkretna wartość, rozciągnięcie i obkurczenie) oraz trzema typami marginesów (wcięcie, przesunięcie oraz wyrównanie). Dodatkowo wszystkie rodzaje pól oprócz bezwzględnych można pozycjonować względnie. Opisy modeli polowych, sposobów określania wymiarów pól oraz techniki definiowania marginesów znajdują się w rozdziałach 4. – 6. Modele pozycjonowania są opisane w tym rozdziale. Sposoby wcinania, przesuwania i wyrównywania elementów opisaliśmy w rozdziale 8. Natomiast w rozdziale 9. prezentujemy ponad 50 różnych układów utworzonych z kombinacji wzorców opisanych w rozdziałach 7. i 8.
Podobne	Pozycjonowany, Statyczny, Pozycjonowanie Bezwzględne, Pozycjonowanie Stałe, Pozycjonowanie Względne, Elementy Pływające, Względny Element Pływający

Pozycjonowany



HTML

```
<h1>Pozycjonowany</h1>
<div class="relative" id="canvas" >
  <p class="static">Pozycjonowanie statyczne</p>
  <p class="static">Ten tekst zawiera względnie pozycjonowany element <span>, który jest
    <span class="relative offset">przesunięty</span> względem swojego normalnego położenia.</p>
  <em class="absolute">Pozycjonowanie bezwzględne</em>
  
  <p class="fixed2">Pozycjonowanie stałe</p>
</div>
```

CSS

```
div,p,em { margin:10px; padding:10px; background-color:white;
  border-left:1px solid gray; border-right:2px solid black;
  border-top:1px solid gray; border-bottom:2px solid black; }

*.static { position:static; }
*.relative { position:relative; left:auto; top:auto; bottom:auto; right:auto; }
*.absolute { position:absolute; left:35%; top:-40px; }
*.fixed1 { position:fixed; z-index:20; right:5px; bottom:35px; }
*.fixed2 { position:fixed; z-index:10; right:0px; bottom:0;
  width:190px; margin:0;}

*.offset { bottom:-15px; left:-20px; }

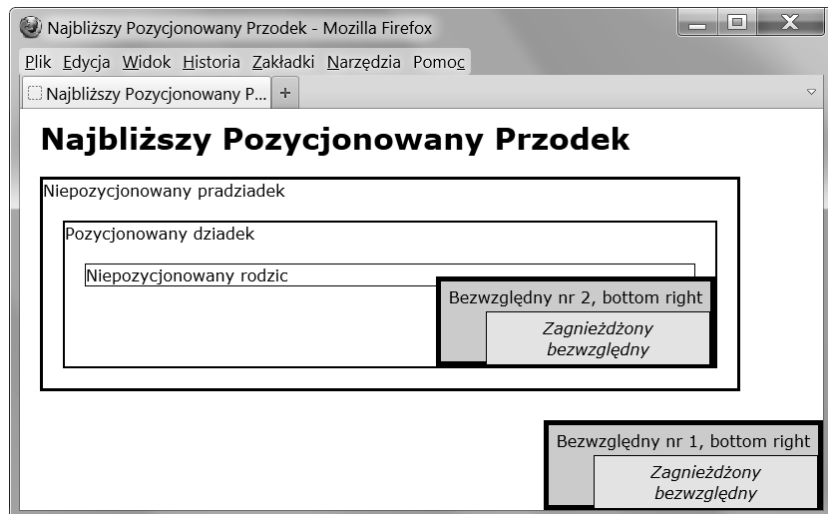
#canvas { background-color:gold; }

/* Mniej ważne style nie zostały pokazane. */
```

Pozycjonowany

Problem	Chcemy utworzyć element pozycjonowany, aby móc pozycjonować względem niego jego elementy potomne. Dodatkowo możemy chcieć przesunąć ten element względem jego aktualnego położenia, najbliższego pozycjonowanego przodka lub obszaru roboczego, przenieść go na osobną warstwę, wyjąć go z normalnego układu elementów albo zdefiniować, które elementy ma on zakrywać lub które mają zakrywać jego.
Rozwiązanie	Aby <i>anulować pozycjonowanie elementu</i>, należy mu zastosować ustawienie <code>position:static</code>. Wartość <code>static</code> jest domyślną wartością własności <code>position</code>. Aby przesunąć element względem jego normalnego położenia w układzie elementów na stronie, należy użyć ustawienia <code>position:relative</code>. Aby przesunąć element względem jego normalnego położenia w układzie elementów na stronie lub jego najbliższego pozycjonowanego przodka, należy użyć ustawienia <code>position:absolute</code>. Aby określić położenie elementu w odniesieniu do obszaru roboczego, należy mu zdefiniować ustawienie <code>position:fixed</code>. Za pomocą własności <code>left</code> można przesunąć lewy bok elementu względem lewej krawędzi jego pola odniesienia. Dodatnie wartości powodują przesunięcie w prawo, a ujemne — w lewo. Za pomocą własności <code>right</code> można przesunąć prawy bok elementu względem prawej krawędzi jego pola odniesienia. Dodatnie wartości powodują przesunięcie w lewo, a ujemne — w prawo. Za pomocą własności <code>top</code> można przesunąć górny bok elementu względem górnej krawędzi jego pola odniesienia. Dodatnie wartości powodują przesunięcie w dół, a ujemne — do góry. Za pomocą własności <code>bottom</code> można przesunąć dolny bok elementu względem dolnej krawędzi jego pola odniesienia. Dodatnie wartości powodują przesunięcie do góry, a ujemne — w dół. Za pomocą własności <code>z-index</code> można określić pozycję elementu na osi pionowej. Im większa wartość, tym bliżej frontu znajduje się element. Elementy można przesuwac względem ich aktualnego położenia za pomocą własności <code>margin</code>.
Wzorzec	<pre>SELEKTOR { position:ABSOLUTE_FIXED_RELATIVE; z-index:+WARTOŚĆ; left:±WARTOŚĆ; right:±WARTOŚĆ; margin-left:±WARTOŚĆ; margin-right:±WARTOŚĆ; top:±WARTOŚĆ; bottom:±WARTOŚĆ; margin-top:±WARTOŚĆ; margin-bottom:±WARTOŚĆ; }</pre>
Zastosowanie	Ten wzorzec projektowy ma zastosowanie do wszystkich elementów
Ograniczenia	Pozycjonowanie stałe (<code>position:fixed</code>) nie działa w przeglądarce Internet Explorer 6, ale działa już we wszystkich nowszych wersjach tego programu
Przykład	Elementowi <code><div></code> zdefiniowaliśmy pozycjonowanie <code>position:relative</code>, aby zamienić go w element pozycjonowany. Element jest pozycjonowany, gdy ma przypisane jedno z następujących ustawień: <code>position:relative</code>, <code>position:absolute</code> oraz <code>position:fixed</code>. Elementy pływające można pozycjonować przy użyciu techniki <code>position:relative</code>. Element pozycjonowany jest punktem odniesienia dla swoich najbliższych pozycjonowanych bezwzględnie potomków. W kodzie źródłowym rysunek gwiazdki znajduje się przed ostatnim akapitem. Normalnie akapit ten byłby wyświetlony na tej gwiazdce, ale dzięki nadaniu jej wyższego numeru <code>z-index</code> znajduje się ona na wierzchu.
Podobne	Najbliższy Pozycjonowany Przodek, Statyczny, Pozycjonowanie Bezwzględne, Pozycjonowanie Stałe, Pozycjonowanie Względne, Elementy Pływające

Najbliższy Pozycjonowany Przodek



HTML

```
<h1>Najbliższy Pozycjonowany Przodek</h1>
```

```
<div class="static ggp">Niepozycjonowany pradiadek
  <div class="absolute sized bottom-right box1">Bezwzględny nr 1, bottom right
    <em class="absolute offset box2">Zagnieżdżony bezwzględny</em></div>
  <div class="relative gp">Pozycjonowany dziadek
    <div class="static parent">Niepozycjonowany rodzic
      <span class="absolute sized bottom-right box1">Bezwzględny nr 2, bottom right
        <em class="absolute offset box2">Zagnieżdżony bezwzględny</em></span>
      </div></div></div>
</body>
```

CSS

```
*.static { position:static; }
*.relative { position:relative; }
*.absolute { position:absolute; }

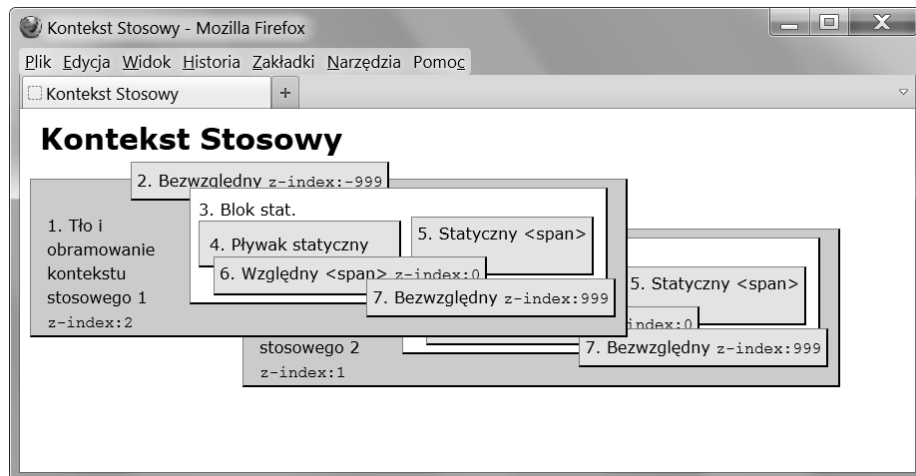
*.sized { width:230px; height:70px; }
*.bottom-right { bottom:0; right:0; }
*.offset { left:45px; top:30px; }
```

```
/* Mniej ważne style nie zostały pokazane. */
```

Najbliższy Pozycjonowany Przodek

Problem	Chcemy wypozycjonować element, aby móc w odniesieniu do niego pozycjonować inne elementy. Element taki jest najbliższym pozycjonowanym przodkiem swoich potomków.
Rozwiązanie	Aby element był pozycjonowany, należy mu przypisać jedno z następujących ustawień: <code>position: relative</code>, <code>position: absolute</code> lub <code>position: fixed</code>. Położenie elementów pozycjonowanych jest wyznaczane względem położenia ich najbliższego pozycjonowanego przodka. Dzięki temu można wyjąć elementy z normalnego układu elementów na stronie i przenieść je w dowolne inne miejsce. Zwróć uwagę, jak w przedstawionym przykładzie bezwzględnie pozycjonowany element <code></code> (Bezwzględny nr 2) został wyjęty z niepozycjonowanego elementu rodzica i ustawiony w dolnym prawym rogu (<code>bottom right</code>) pozycjonowanego dziadka, który jest jego najbliższym pozycjonowanym przodkiem. Jeśli pozycjonowany element nie ma pozycjonowanego przodka, jego położenie ustala się w odniesieniu do elementu <code><body></code>, co oznacza, że element ten jest pozycjonowany domyślnie. Zwróć uwagę, jak w przedstawionym przykładzie bezwzględnie pozycjonowany element <code><div></code> (Bezwzględny nr 1) został wyjęty ze swojego niepozycjonowanego rodzica i ustawiony w prawym dolnym rogu elementu <code><body></code>. Wyznaczenie położenia elementów pozycjonowanych w odniesieniu do ich najbliższych pozycjonowanych przodków pozwala na tworzenie zamkniętych struktur układu. Strukturę taką można przenosić w różne miejsca, a rozmieszczenie jej zawartości — zarówno elementów pozycjonowanych, jak i niepozycjonowanych — zawsze pozostanie bez zmian. Zwróć uwagę, jak w przedstawionym przykładzie bezwzględnie pozycjonowane elementy <code></code> zostały rozmieszczone w odniesieniu do ich najbliższych pozycjonowanych przodków, którzy z kolei są przeniesieni do dolnego prawego rogu ich najbliższych pozycjonowanych przodków.
Wzorzec	<pre>SELEKTOR { position: relative; } lub SELEKTOR { position: absolute; } lub SELEKTOR { position: fixed; }</pre>
Zastosowanie	Ten wzorzec projektowy ma zastosowanie do wszystkich elementów.
Ograniczenia	Elementy mogą być pozycjonowane tylko w odniesieniu do elementów, w których rzeczywiście się znajdują, tzn. nie ma możliwości pozycjonowania wg dowolnego elementu. Byłoby to bardzo przydatne, ale na razie tylko rzeczywisty przodek może stanowić kontekst pozycjonowania elementu pozycjonowanego.
Zalety	Zamknięte struktury pozycjonowanych elementów można zagnieżdżać na dowolną głębokość. Jest to bardzo przydatne przy tworzeniu projektów do wielokrotnego użytku.
Wady	Pozycjonowanie to bardzo przydatna technika, ale jej największą wadą jest to, że elementy pozycjonowane muszą mieć konkretnie określone wymiary. Tak zaprojektowane układy mogą źle się prezentować przy innych ustawieniach rozmiaru pisma lub na wyświetlaczach innej wielkości, niż przewidział projektant.
Wskazówka	Zastosowanie ustawienia <code>position: relative</code> jest bardzo dobrym sposobem na utworzenie pozycjonowanego przodka, ponieważ nie powoduje ono wyjęcia pozycjonowanego elementu z normalnego układu elementów na stronie. Dzięki temu można w projektach używać mieszanki elementów pozycjonowanych normalnie i bezwzględnie.
Podobne	Pozycjonowany, Kontekst Stosowy, Wewnętrzny, Pozycjonowanie Bezwzględne, Pozycjonowanie Stałe, Pozycjonowanie Względne, Względny Element Pływający

Kontekst Stosowy



HTML

```
<h1>Kontekst Stosowy</h1>
<div class="stacking-context1 box">
  <div class="caption" >1. Tło i obramowanie kontekstu stosowego 1
    <br /><code>z-index:2</code></div>
  <span class="level2 box">2. Bezwzględny <code>z-index:-999</code></span>
  <div class="level3 box">3. Blok stat.<br />
    <span class="level4 box">4. Pływak statyczny</span>
    <span class="level5 box">5. Statyczny &lt;span></span></span><br /><br /><p class="clear"></p>
    <span class="level6 box">6. Względny &lt;span>&gt;</span> <code>z-index:0</code></span>
    <span class="level7 box">7. Bezwzględny <code>z-index:999</code></span>
  </div>
</div>

<div class="stacking-context2 box"> <!-- Dalszy kod jest taki sam, jak powyżej -->
```

CSS

```
*.stacking-context1 { z-index:2; position:absolute; left:10px; top:70px; }
*.stacking-context2 { z-index:1; position:absolute; left:223px; top:120px; }

*.level2 { z-index:-999; position:absolute; }
*.level3 { position:static; }
*.level4 { float:left; }
*.level5 { position:static; }
*.level6 { z-index:0; position:relative; }
*.level7 { z-index:999; position:absolute; }
```

/ Mniej ważne style nie zostały pokazane. */*

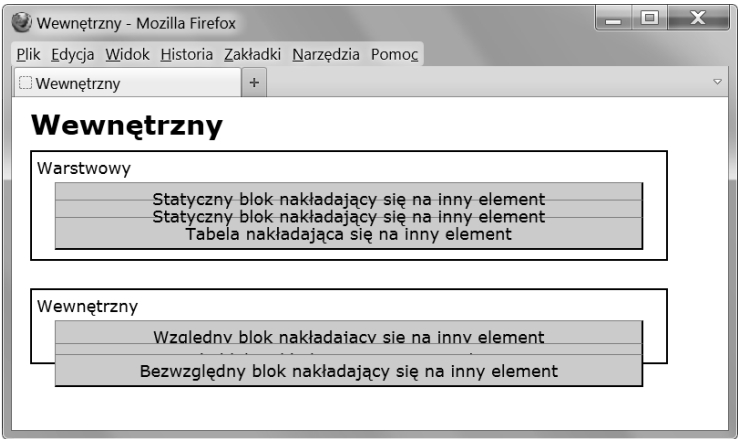
Kontekst Stosowy

Inne nazwy	kolejność na stosie, poziom stosu, z-index, warstwy, kolejność rysowania, kolejność na osi z
Problem	Chcemy mieć wpływ na kolejność rozmieszczenia elementów na osi pionowej
Rozwiązanie	W języku CSS istnieje możliwość kontrolowania kolejności elementów względem powierzchni ekranu. Elementy statyczne są ustawiane od dołu do góry w takiej kolejności, w jakiej znajdują się w kodzie źródłowym dokumentu. O kolejności elementów pozycjonowanych decyduje wartość ich własności z-index — im wyższa, tym bliżej ekranu znajduje się dany element. Elementy pozycjonowane, których własność z-index ma wartość ujemną, są umieszczane pod elementami statycznymi i niepozycjonowanymi elementami pływającymi. Numerowanie elementów we własności z-index nie musi być ciągle. Domyślnie własność ta ma wartość auto. Pozycjonowany element mający przypisaną własność z-index o liczbowej wartości stanowi samodzielny kontekst stosowy, w którym prezentowane są <i>wszystkie</i> jego elementy potomne (statyczne, pływające i pozycjonowane). Kontekst stosowy <i>nie</i> jest tworzony, gdy własność z-index zostanie ustawiona na auto lub zostanie zdefiniowana dla niepozycjonowanego elementu. Następujące wartości powodują powstanie kontekstu stosowego: z-index:0, z-index:-1 oraz z-index:9999. Każdy kontekst stosowy jest niezależną zamkniętą strukturą, do której nie mogą przenikać elementy <i>przodki</i> ani <i>siostrzane</i>. Każdy lokalny kontekst stosowy jest przypisany do <i>wewnętrznego poziomu</i> stosowego 0 i jego elementy potomne są ustawiane na stosie <i>względem tego poziomu</i>. Jest to bardzo ważne. Własność z-index nie jest globalna, lecz odnosi się do najbliższego pozycjonowanego przodka, któremu została przypisana własność z-index z wartością liczbową. Główny kontekst stosowy reprezentuje element <code><html></code>. Kontekst stosowy jest prezentowany na ekranie warstwowo w następujący sposób (od najniżej położonej warstwy): - Tło i obramowanie elementu wyznaczającego kontekst stosowy. - Pozycjonowane elementy potomne mające przypisaną własność z-index o ujemnej wartości. - Potomne niepozycjonowane elementy blokowe. - Potomne niepozycjonowane elementy pływające. - Potomne niepozycjonowane elementy śródliniowe. - Potomne elementy pozycjonowane z ustawieniami z-index:auto i z-index:0. - Potomne elementy pozycjonowane z własnością z-index o dodatniej wartości. Elementy opisane w punktach 2, 6 i 7 rekurencyjnie tworzą konteksty stosowe, ponieważ każdy pozycjonowany element mający własność z-index o wartości liczbowej tworzy lokalny kontekst stosowy. Zanim zostanie wyświetlona zawartość elementu, przeglądarka najpierw wyświetla jego pole z kolorem i obrazem oraz obramowaniem. Dopiero później następuje wydruk treści na wcześniej przygotowanym polu.
Wzorzec	<code>SELEKTOR { z-index:±WARTOŚĆ; position:ABSOLUTE_FIXED_RELATIVE; }</code>
Zastosowanie	Ten wzorzec projektowy ma zastosowanie do wszystkich elementów.
Ograniczenia	Przeglądarka Firefox miesza punkty 1 i 2, przez co ujemne potomne konteksty stosowe są w niej umieszczane pod tłem i obramowaniem kontekstu nadrzędnego! W nowszych wersjach programu błąd ten jest już poprawiony.

Kontekst Stosowy — ciąg dalszy

Przykład	W przykładzie zostało przedstawionych wszystkie siedem poziomów stosowych w dwóch kontekstach. Zwróć uwagę, jak poziomy stosowe odnoszą się do każdego z kontekstów stosowych.
Podobne	Pozycjonowany, Najbliższy Pozycjonowany Przodek, Pozycjonowanie Bezwzględne, Pozycjonowanie Względne, Względny Element Pływający

Wewnętrzny



HTML

```
<h1>Wewnętrzny</h1>
<div>Warstwowy
  <p class="static">Stacyjny blok nakładający się na inny element</p>
  <p class="static overlap">Statyczny blok nakładający się na inny element</p>
  <table class="static overlap"><tr><td>Tabela nakładająca się na inny element</td></tr></table>
</div>

<div>Wewnętrzny
  <p class="relative">Względny blok nakładający się na inny element</p>
  <p class="fixed">Stały blok nakładający się na inny element</p>
  <p class="absolute">Bezwzględny blok nakładający się na inny element</p></div>
```

CSS

```
*.static { position:static; }
*.overlap { margin-top:-22px; }

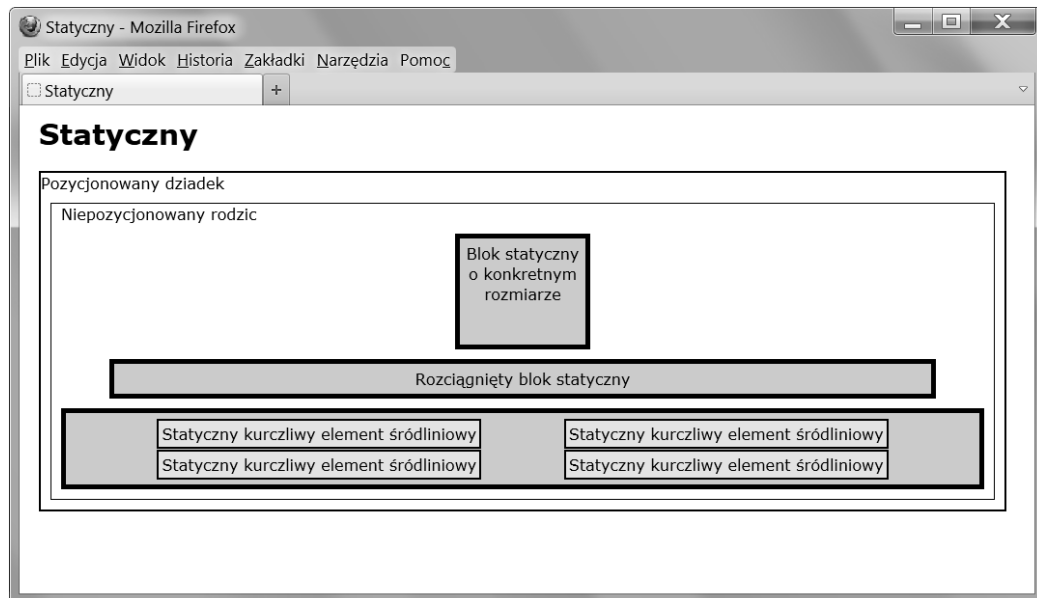
*.relative { position:relative; }
*.fixed { position:fixed; margin-top:-16px; }
*.absolute { position:absolute; top:65px; }

/* Mniej ważne style nie zostały pokazane. */
```

Wewnętrzny

Inne nazwy	hasLayout, grupowanie
Problem	Chcemy, aby treść bloku była rysowana <i>wewnątrz</i> niego, a nie <i>na</i> nim. Innymi słowy, chcemy, aby tekst i pozostała treść śródliniowa były rysowane jako część bloku, do którego należą, dzięki czemu jeśli na element ten zostanie nałożony jakiś inny element, to przesłoni go wraz z tą treścią. Problem polega na tym, że przeglądarki statyczną treść <i>śródliniową</i> rysują na osobnej warstwie nad tłem statycznych elementów <i>blokowych</i>. Gdy dwa statyczne elementy blokowe nałożą się na siebie, tło jednego przesłoni tło drugiego, ale z treścią śródliniową już tak się nie stanie! Spójrz na pierwszą część rysunku, gdzie widać, że obramowanie i tło elementów znajdujących się pod spodem są zasłonięte, natomiast tekst nie. To samo zobaczymy we wszystkich najważniejszych przeglądarkach, ponieważ zgodnie z zasadami kontekstu stosowego najpierw rysowane są tła i obramowania <i>wszystkich</i> bloków, a dopiero potem ich elementy śródliniowe i treść. Z tego powodu tła i obramowania bloków zostają umieszczone na warstwie znajdującej się pod elementami pływającymi i treścią śródliniową. Może Ci się to wydawać sprzeczne z intuicją, ponieważ zazwyczaj myślimy, że treść śródliniowa znajduje się <i>wewnątrz</i> zawierających ją bloków, a nie <i>na</i> nich. Jednak elementy śródliniowe powinny znajdować się <i>na</i> blokach, ponieważ treść śródliniowa może w przypadku przepełnienia wystawać poza blok.
Rozwiązanie	Elementy pozycjonowane są niepodzielne, tzn. między ich tło, statycznych potomków oraz treść śródliniową nie da się niczego włożyć. Zwróć uwagę, jak w drugiej części przykładowego rysunku sąsiadujące ze sobą bloki nakładają się na siebie i w wyniku tego przesłaniania jest także treść śródliniowa. Jest tak dzięki temu, że elementy te są pozycjonowane, a zasady kontekstu stosowego wymagają, aby elementy pozycjonowane (względnie, bezwzględnie lub stale) były niepodzielne. Także bloki mające zdefiniowane ustawienie <code>overflow: scroll</code> są niepodzielne, ponieważ ich treść znajduje się w dającym się przewijać obszarze.
Wzorzec	<code>SELEKTOR { position:RELATIVE_ABSOLUTE_FIXED; }</code>
Zastosowanie	Ten wzorzec projektowy ma zastosowanie do wszystkich elementów.
Ograniczenia	W starszych przeglądarkach ustawienie własności <code>overflow</code> nie zawsze powoduje, że element staje się niepodzielny. Na przykład ustawienie <code>overflow: hidden</code> sprawia, że element jest niepodzielny w przeglądarkach Firefox 2.0 i Internet Explorer 7, ale nie w Internet Explorerze 6. Bloki mające ustawienie <code>overflow: scroll</code> są niepodzielne we wszystkich przeglądarkach oprócz Internet Explorera 6. W nowszych przeglądarkach zastosowanie własności <code>overflow</code> zawsze sprawia, że element staje się niepodzielny. Tabele i bloki o konkretnie ustalonym rozmiarze są przez przeglądarkę Internet Explorer 7 traktowane jako niepodzielne. W innych przeglądarkach tak nie jest. Powodem tego jest to, że w Internet Explorerze do wersji 7 używana jest specjalna wewnętrzna własność DOM o nazwie <code>hasLayout</code>, która ma wartość prawdziwą, gdy element ma układ. Jeśli element ma układ, to jest rysowany niepodzielnie w swoim własnym oknie i we własnym kontekście układu. Wszystkie jego elementy potomne są rysowane wewnątrz jego prostokątnego pola. Element taki nie może się obkurczać do rozmiaru treści, a zewnętrzne elementy pływające nie mają wpływu na znajdującą się w nim treść śródliniową.
Wskazówka	Przeglądarka Internet Explorer zawiera błędy, które można naprawić poprzez włączenie własności <code>hasLayout</code> . Aby to zrobić, można użyć niestandardowej własności <code>zoom: 1</code> , ale trzeba pamiętać, że to spowoduje, iż arkusz stylów nie przejdzie poprawnie walidacji.
Podobne	Pozycjonowany, Statyczny, Pozycjonowanie Bezwzględne, Pozycjonowanie Względne, Elementy Pływające

Statyczny



HTML

```
<h1>Statyczny</h1>
<div class="gp">Pozycjonowany dziadek
  <div class="parent">Niepozycjonowany rodzic
    <div id="zs" class="box">Blok statyczny o konkretnym rozmiarze </div>
    <div id="ss" class="box">Rozciągnięty blok statyczny</div>
    <div class="box">
      <span>Statyczny kurczliwy element śródliniowy</span>
      <span>Statyczny kurczliwy element śródliniowy</span>
      <span>Statyczny kurczliwy element śródliniowy</span>
      <span>Statyczny kurczliwy element śródliniowy</span>
    </div>
  </div>
</div>
```

CSS

```
span { position:static; margin:40px; line-height:32px;
padding:3px; border:2px solid black; background-color:yellow; }

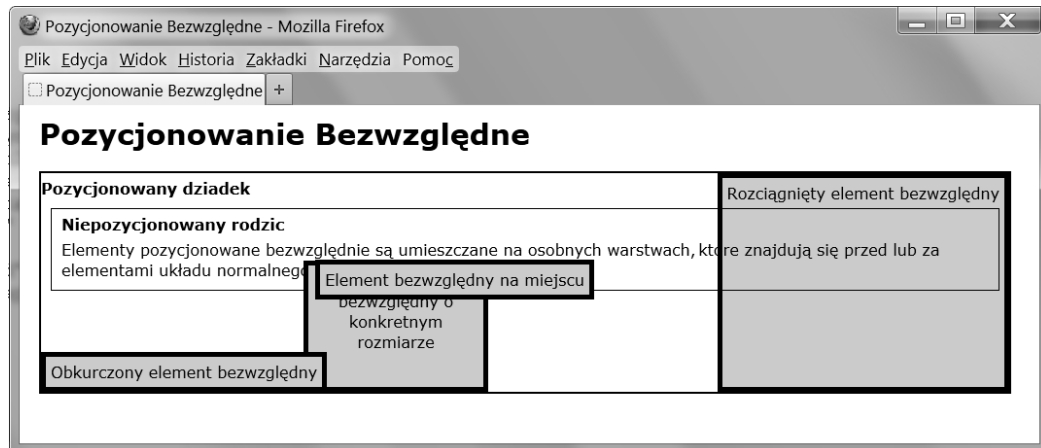
#zs { position:static; width:120px; height:100px; margin:10px auto; }

#ss { position:static; width:auto; height:auto; margin:10px 50px; }
```

Statyczny

Problem	Chcemy, aby elementy automatycznie ustawiały się w rzędach i blokach i płynnie dostosowywały swój układ do rozmiaru okna przeglądarki.
Rozwiązanie	Aby umieścić element w normalnym układzie elementów na stronie, można mu przypisać ustawienie <code>position:static</code> . Ponieważ jest ono domyślne, to standardowo wszystkie elementy są pozycjonowane w układzie normalnym. Układ normalny składa się z zagnieżdżonych bloków, które są rysowane jeden pod drugim w obszarze roboczym przeglądarki. Wewnątrz każdego bloku może znajdować się dowolna liczba innych elementów blokowych i śródliniowych. W linii mogą znajdować się tekst i obiekty, które są rysowane w poziomie jeden obok drugiego. Początek elementu statycznego wyznaczają ustawienia rozmiaru, dopełnienia, obramowania i marginesów poprzedniego elementu statycznego.
Wzorce	Śródliniowy element statyczny <pre>SELEKTOR_ŚRÓDLINIOWY { position:static; line-height:±WARTOŚĆ; margin-left:±WARTOŚĆ; margin-right:±WARTOŚĆ; }</pre> Blokowy element statyczny <pre>SELEKTOR_BLOKOWY { position:static; width:+WARTOŚĆ; height:+WARTOŚĆ; margin-left:±WARTOŚĆ; margin-right:±WARTOŚĆ; margin-top:±WARTOŚĆ; margin-bottom:±WARTOŚĆ; }</pre>
Zastosowanie	Ten wzorec projektowy ma zastosowanie do wszystkich elementów.
Przykład	Wszystkie elementy w przedstawionym przykładzie są statyczne. Elementy blokowe ustawiają się w linii pionowej. Wszystkie elementy blokowe oprócz jednego, który ma konkretnie ustawiony rozmiar, są rozciągnięte na szerokość kontenera, a ich rozmiar jest równy szerokości rodzica pomniejszonej o ich marginesy i dopełnienie rodzica. Górny margines spycha statyczny element blokowy w dół, a dolny margines powoduje zepchnięcie w dół następnego statycznego elementu blokowego. Przylegające pionowe marginesy ulegają scaleniu i stosowany jest większy z nich. W tym przykładzie wszystkie marginesy mają ustawione marginesy górny i dolny na 10 pikseli. Dzięki scalaniu elementy te dzieli odstęp o wysokości tylko 10 pikseli. Błokom statycznym można określić konkretny rozmiar za pomocą własności <code>height</code> i <code>width</code>. Jeśli lewy lub prawy margines zostanie ustawiony na <code>auto</code>, to będzie się automatycznie dostosowywał rozmiarem do tych ustawień. Ustawienie lewego i prawego marginesu statycznego elementu blokowego na <code>auto</code> powoduje ustawienie tego elementu na środek kontenera, jak widać na przykładzie pierwszego przedstawionego bloku. Statyczne elementy <i>śródliniowe</i> przedstawione w tym przykładzie mają marginesy lewy i prawy ustawione na 40 pikseli. Marginesy te rozsuwają elementy śródliniowe i nie ulegają scaleniu. Jeśli zawartość elementu śródliniowego przekracza szerokość kontenera, następuje przeniesienie tej treści do następnego wiersza. W elementach śródliniowych marginesy górne i dolne są ignorowane, ponieważ o wysokości linii decyduje w nich własność <code>line-height</code>.
Podobne	Pozycjonowanie Bezzględne, Pozycjonowanie Stałe, Pozycjonowanie Względne; Konkretny Rozmiar, Rozciągnięty, Kurczliwy (rozdział 5.)

Pozycjonowanie Bezwzględne



HTML

```
<h1>Pozycjonowanie Bezwzględne</h1>
<div class="gp"><h2>Pozycjonowany dziadek</h2>
  <div class="parent"><h2>Niepozycjonowany rodzic</h2>
    Elementy pozycjonowane bezwzględnie są umieszczane na osobnych warstwach, które znajdują się przed lub za elementami układu normalnego.
    <span id="in-place" class="box">Element bezwzględny na miejscu</span>
    <span id="sized" class="box">Element bezwzględny o konkretnym rozmiarze</span>
    <p id="stretched" class="box">Rozciągnięty element bezwzględny </p>
    <p id="shrinkwrapped" class="box">Obkurczony element bezwzględny</p> </div></div>
```

CSS

```
#in-place { position:absolute; z-index:1; }

#shrinkwrapped { position:absolute; z-index:0;
  width:auto; left:0; bottom:0; margin:0; }

#sized { position:absolute; z-index:auto;
  width:170px; height:115px; bottom:0; left:270px; margin:0; }

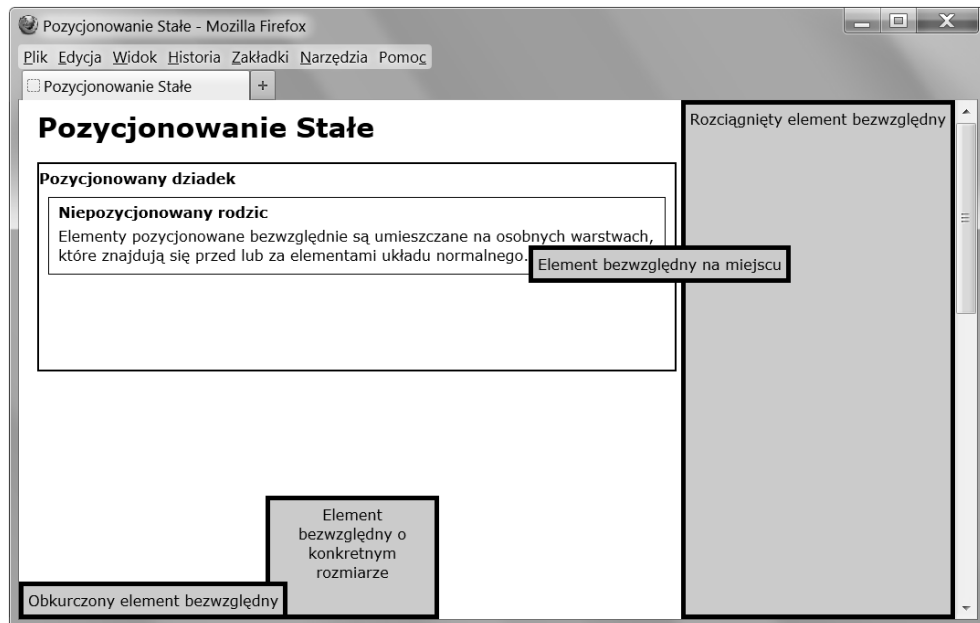
#stretched { position:absolute; z-index:-1;
  height:auto; right:0; top:0; bottom:0; margin:0; }

/* Mniej ważne style nie zostały pokazane. */
```

Pozycjonowanie Bezwzględne

Problem	Chcemy wyjąć element z normalnego układu elementów na stronie i umieścić go na osobnej warstwie. Ponadto chcemy określić położenie tego elementu w odniesieniu do wewnętrznej krawędzi obramowania jego najbliższego pozycjonowanego przodka albo w miejscu, w którym znajdowałby się, gdyby pozostał w układzie normalnym. Chcemy, aby położenie tego elementu nie miało wpływu na położenie innych elementów.
Rozwiązanie	Aby wypozycjonować <i>dowolny</i> element bezwzględnie, należy użyć ustawienia <code>position:absolute</code> . Do określania wymiarów takiego elementu służą własności <code>width</code> i <code>height</code> . Wartości procentowe odnoszą się do najbliższego pozycjonowanego przodka, a nie rodzica. Aby pozycjonowany bezwzględnie element <i>wyrównać</i> do jednej z krawędzi jego najbliższego pozycjonowanego przodka, należy ustawić na zero własność <code>left</code> , <code>right</code> , <code>top</code> lub <code>bottom</code> . Ustawienie wszystkich tych własności na <code>auto</code> powoduje umieszczenie elementu w tym samym miejscu, w którym znajdowałby się, gdyby był w układzie normalnym. Aby <i>przesunąć</i> element względem boków jego najbliższego pozycjonowanego przodka, można użyć własności <code>margin-left</code> , <code>margin-right</code> , <code>margin-top</code> oraz <code>margin-bottom</code> . Za pomocą własności <code>z-index</code> można kontrolować kolejność elementów na osi pionowej. Im wyższa wartość, tym „bliżej” użytkownika znajduje się dany element. Aby element przodek był pozycjonowany, musi mieć zdefiniowane ustawienie <code>position:relative</code> , <code>position:absolute</code> lub <code>position:fixed</code> . Jeśli żaden z przodków elementu nie jest pozycjonowany, położenie tego elementu będzie ustalane w odniesieniu do elementu <code><body></code> .
Wzorce	<pre>SELEKTOR { position:absolute; z-index:WARTOŚĆ; width:±WARTOŚĆ; left:±WARTOŚĆ; margin-left:±WARTOŚĆ; right:±WARTOŚĆ; margin-right:±WARTOŚĆ; height:±WARTOŚĆ; top:±WARTOŚĆ; margin-top:±WARTOŚĆ; bottom:±WARTOŚĆ; margin-bottom:±WARTOŚĆ; }</pre> plus <pre>SELEKTOR_PRZODKA { position:relative; }</pre> lub <pre>SELEKTOR_PRZODKA { position:absolute; }</pre> lub <pre>SELEKTOR_PRZODKA { position:fixed; }</pre>
Zastosowanie	Bezwzględnie można pozycjonować każdy element.
Ograniczenia	Przeglądarka Internet Explorer 6 obkurcza rozciągnięte elementy pozycjonowane bezwzględnie. Przeglądarka Internet Explorer do wersji 7 włącznie nie ustawia na środku elementów pozycjonowanych bezwzględnie.
Zalety	Pozycjonowanie bezwzględne pozwala precyzyjnie ustalić położenie elementu względem jego najbliższego pozycjonowanego przodka. Elementom takim można określić konkretny rozmiar albo ustawić, aby się kurczyły lub rozciągały. Element pozycjonowany bezwzględnie jest rysowany nad elementami układu normalnego na warstwie, która pod wieloma względami przypomina pole blokowe. Elementy pozycjonowane bezwzględnie nie mogą pływać. Inne elementy nie mają wpływu na ich położenie ani też elementy te nie mają wpływu na położenie innych. Dlatego mogą one zasłaniać inne elementy i same być zasłaniające. Jeśli wszystkie elementy dzieci danego elementu są pozycjonowane bezwzględnie, wysokość tego elementu zostaje zredukowana do zera (chyba że zostanie celowo inaczej ustawiona), ponieważ nie ma on dzieci w układzie normalnym.
Wady	Układy zaprojektowane przy użyciu pozycjonowania bezwzględnego źle się skalują na urządzeniach o ekranach i rozmiarze pisma, które znacznie się różnią od parametrów urządzenia, na którym opracowywany był projekt.
Podobne	Przesunięcie Elementu Bezwzględnego (rozdział 8.), Pozycjonowanie Stałe, Konkretny Rozmiar, Kurcziwy, Rozciągnięty (rozdział 5.)

Pozycjonowanie Stałe



HTML

```
<h1>Pozycjonowanie Stałe</h1>
<div class="gp"><h2>Pozycjonowany dziadek</h2>
  <div class="parent"><h2>Niepozycjonowany rodzic</h2>
    Elementy pozycjonowane bezwzględnie są umieszczane na osobnych warstwach, które znajdują się
    przed lub za elementami układu normalnego.
    <span id="in-place"      class="box">Element bezwzględny na miejscu</span>
    <span id="sized"        class="box">Element bezwzględny o konkretnym rozmiarze</span>
    <p id="stretched"      class="box">Rozciągnięty element bezwzględny</p>
    <p id="shrinkwrapped"  class="box">Obkurczony element bezwzględny</p> </div></div>
```

CSS

```
*.gp { position:relative; z-index:1; }

#in-place { position:fixed; z-index:1; }

#shrinkwrapped { position:fixed; z-index:0;
  width:auto; left:0; bottom:0; margin:0; }

#sized { position:fixed; z-index:auto;
  width:170px; height:115px; bottom:0; left:270px; margin:0; }

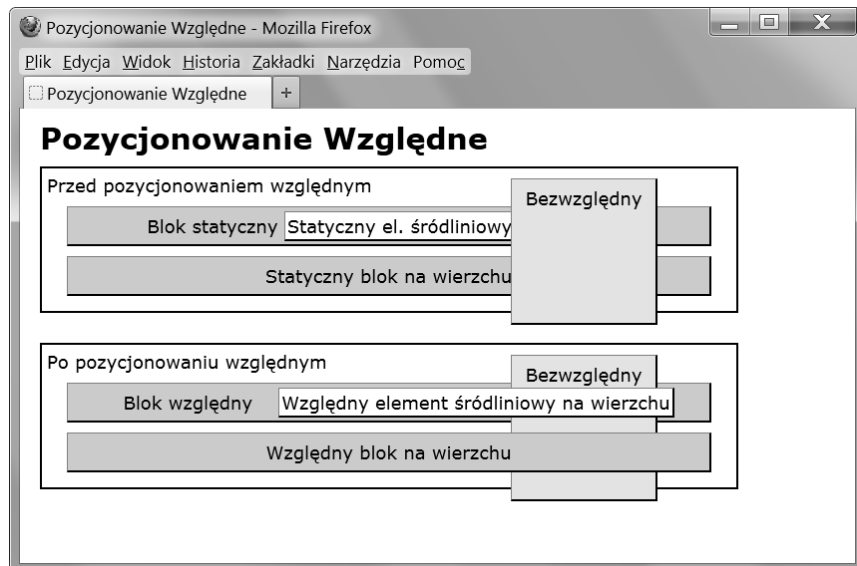
#stretched { position:fixed; z-index:-1;
  height:auto; right:0; top:0; bottom:0; margin:0; }

/* Mniej ważne style nie zostały pokazane. */
```

Pozycjonowanie Stałe

Problem	Chcemy umieścić element na osobnej warstwie przytwierdzonej do obszaru roboczego lub umiejscowionej w tym samym miejscu, w którym znalazłby się, gdyby był w układzie normalnym. Ponadto chcemy, aby podczas przewijania strony element ten pozostawał cały czas w tym samym miejscu. Jest to tzw. element stały albo element o stałym położeniu.
Rozwiązanie	Aby określić stałe położenie elementu, należy mu zdefiniować ustawienie <code>position:fixed</code> . Ustawienie to działa podobnie do <code>absolute</code> , z tym wyjątkiem, że pozycjonowanie elementu odbywa się zawsze względem obszaru roboczego, a nie najbliższego pozycjonowanego przodka oraz pozycjonowany w ten sposób element nie przewija się wraz z resztą strony. Jeśli element pozycjonowany na stałe zostanie umieszczony w tym samym miejscu, w którym znajdowałby się w układzie normalnym, to nadal nie będzie on przewijany wraz z resztą strony.
Wzorzec	<pre>SELEKTOR { position:fixed; z-index:WARTOŚĆ; width:±WARTOŚĆ; left:±WARTOŚĆ; margin-left:±WARTOŚĆ; right:±WARTOŚĆ; margin-right:±WARTOŚĆ; height:±WARTOŚĆ; top:±WARTOŚĆ; margin-top:±WARTOŚĆ; bottom:±WARTOŚĆ; margin-bottom:WARTOŚĆ; }</pre>
Zastosowanie	Ten wzorzec projektowy ma zastosowanie do wszystkich elementów.
Ograniczenia	Przeglądarka Internet Explorer 6 elementy pozycjonowane na stałe prezentuje jako bezwzględne. W nowszych wersjach programu błąd ten jest już naprawiony.
Zalety	Pozycjonowanie stałe pozwala precyzyjnie kontrolować położenie elementów w odniesieniu do obszaru roboczego. Elementy te nie są przewijane wraz z resztą strony. Doskonale nadają się do przechowywania elementów sterujących, takich jak menu, paski narzędzi, przyciski itp.
Wady	Układy zaprojektowane przy użyciu pozycjonowania stałego źle się skalują na urządzeniach o ekranach i rozmiarze pisma, które znacznie odbiegają parametrami od urządzenia, na którym opracowywany był projekt.
Przykład	W tym przykładzie zostały użyte te same elementy, co we wzorcu Pozycjonowanie Bezwzględne, tylko zmieniono im pozycjonowanie na stałe. Zwróć uwagę, że mimo iż zawartość okna przeglądarki została przewinięta, elementy te nadal są widoczne w całości. Zauważ także, że położenie tych elementów zostało określone względem obszaru roboczego, a nie w odniesieniu do ich pozycjonowanego przodka. Element oznaczony jako Element bezwzględny na miejscu znajduje się w tym samym miejscu, w którym znajdowałby się w układzie normalnym, ale również nie poddaje się przewijaniu. Gdyby element ten został początkowo narysowany w miejscu niewidocznym na ekranie, to nie pojawiłby się na widoku nawet po przewinięciu strony. Zauważ, że elementy pozycjonowane na stałe w tym przykładzie są rozmieszczone dokładnie tak samo, jak w przykładzie do wzorca Pozycjonowanie Bezwzględne. Element bezwzględny na miejscu znajduje się przed elementem o konkretnym rozmiarze, ponieważ jego własność <code>z-index</code> ma wartość 1, a wartość <code>z-index</code> tego drugiego elementu jest ustawiona na <code>auto</code>. Rozciągnięty element pozycjonowany bezwzględnie znajduje się pod pozycjonowanym dziadkiem, ponieważ ma własność <code>z-index</code> o wartości -1, a dziadek — 1. Dzięki temu, że tło elementu dziadka jest przezroczyste, widać znajdujący się pod nim rozciągnięty element pozycjonowany bezwzględnie.
Podobne	Pozycjonowanie Bezwzględne; Konkretny Rozmiar, Kurczliwy, Rozciągnięty (rozdział 5.)

Pozycjonowanie Względne



HTML

```
<h1>Pozycjonowanie Względne</h1>
<div class="relative">Przed pozycjonowaniem względnym
  <p class="static" >Blok statyczny
    <span class="static ontop" >Statyczny el. śródliniowy na wierzchu</span></p>
  <p class="static ontop" >Statyczny blok na wierzchu</p>
  <p class="absolute" >Bezwzględny</p></div>

<div class="relative">Po pozycjonowaniu względnym
  <p class="relative" >Blok względny
    <span class="relative ontop offset" >Względny element śródliniowy na wierzchu</span></p>
  <p class="relative ontop" >Względny blok na wierzchu</p>
  <p class="absolute" >Bezwzględny</p></div>
```

CSS

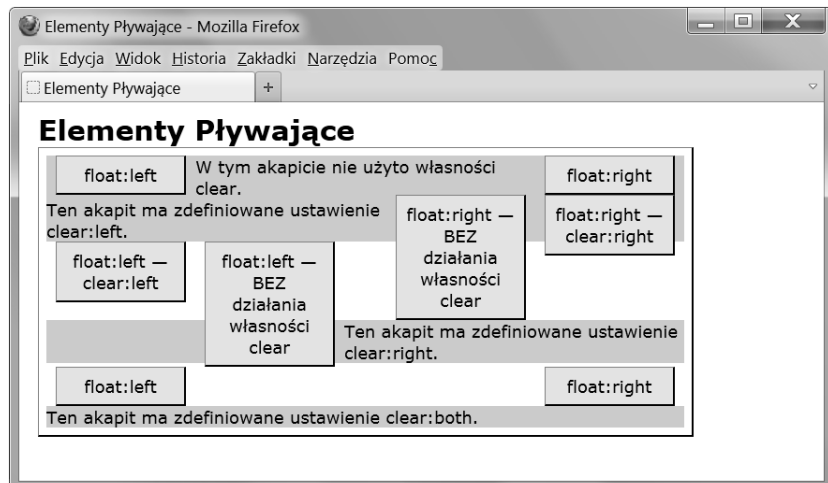
```
*.ontop { z-index:1; }
*.static { position:static; }
*.relative { position:relative; }
*.absolute { position:absolute; z-index:auto; }
*.offset { left:20px; top:auto; }
```

/ Mniej ważne style nie zostały pokazane. */*

Pozycjonowanie Względne

Problemy	Chcemy kontrolować kolejność na osi pionowej elementów pływających i należących do układu normalnego. Trudność polega na tym, że do tego typu elementów własność <code>z-index</code> nie ma zastosowania. Możliwość kontroli kolejności elementów na osi pionowej jest ważna, gdy pozycjonowane elementy nachodzą na elementy pływające lub statyczne. Chcemy wypozycjonować element, aby stał się najbliższym pozycjonowanym przodkiem innego elementu. Chcemy przesunąć element, nie usuwając jego miejsca z normalnego układu elementów na stronie. Nie chcemy zmieniać kształtu tego elementu w układzie normalnym. Nie chcemy także, aby jego przesunięcie powodowało zmianę położenia innych elementów.
Rozwiązania	Aby móc kontrolować miejsce elementu na osi pionowej, można zastosować dla niego pozycjonowanie względne za pomocą własności <code>position: relative</code>, a następnie użyć własności <code>z-index</code>. Pozycjonowanie względne nie powoduje wyjęcia elementu z układu normalnego ani zmiany jego kształtu w tym układzie. Jeśli na przykład dany element śródliniowy zajmuje jeden lub więcej wierszy, to po zastosowaniu dla niego pozycjonowania względnego zachowa on te cechy. Porównaj to z pozycjonowaniem bezwzględnym, które powoduje zamianę elementu śródliniowego w pole bezwzględne i rozmieszczenie jego treści w bezwzględnym polu blokowym, co może zmienić układ rozmieszczenia w nim treści. Element pozycjonowany względnie można przesunąć względem jego pierwotnego położenia w układzie normalnym za pomocą własności <code>left</code> i <code>top</code>. Nie powoduje to zmiany położenia innych elementów. Własności <code>left</code> i <code>top</code> mają domyślną wartość <code>auto</code>, która powoduje, że elementy te są standardowo umieszczane na swojej normalnej pozycji. Ustawienie <code>position: relative</code> można zdefiniować dowolnemu elementowi, aby móc w odniesieniu do niego pozycjonować bezwzględnie znajdujące się w nim elementy — szczegółowe informacje na ten temat znajdują się w opisie wzorca Najbliższy Pozycjonowany Przodek wcześniej w tym rozdziale. Za pomocą kombinacji własności <code>position: relative</code>, <code>left</code> oraz <code>top</code> można przesunąć element — szczegółowe informacje na ten temat znajdują się w opisie wzorca Przesunięcie Elementu Względego w rozdziale 8. Za pomocą ustawienia <code>position: relative</code> można także kontrolować kolejność na osi pionowej elementów pływających — szczegóły w opisie wzorca Względny Element Pływający w tym rozdziale.
Wzorzec	<pre>SELEKTOR { position: relative; z-index: +WARTOŚĆ; left: auto; top: auto; }</pre>
Zastosowanie	Ten wzorzec projektowy ma zastosowanie do wszystkich elementów.
Ograniczenia	Ze względu na sposób implementacji własności <code>hasLayout</code> w przeglądarce Internet Explorer do wersji 7, <i>względnie pozycjonowane elementy śródliniowe i pozycjonowane elementy blokowe</i> nie mogą być ustawiane nad elementami znajdującymi się poza ich blokiem. Z tego powodu przeglądarka Internet Explorer 7 nie narysuje śródliniowego względnie pozycjonowanego elementu <code></code> przed bezwzględnie pozycjonowanym akapitem. Jest tak dlatego, że element <code></code> znajduje się dosłownie wewnątrz akapitu. Problem ten nie występuje w pozostałych najważniejszych przeglądarkach ani w Internet Explorerze, gdy blok nadrzędny nie ma układu, a więc gdy jest on np. blokiem statycznym. W nowszych niż omawiane wersjach Internet Explorera błąd ten został naprawiony.
Podobne	Pozycjonowany, Najbliższy Pozycjonowany Przodek, Kontekst Stosowy, Wewnętrzny, Względny Element Pływający; Przesunięcie Elementu Względego (rozdział 8.)

Elementy Pływające



HTML

```
<h1>Elementy Pływające</h1>
<div>
  <div class="float left clear-left" >float:left </div>
  <div class="float right clear-right">float:right</div>

  <p class="clear-none" >W tym akapicie nie użyto własności clear.
    <span class="float right clear-right">float:right – clear:right</span>
    <span class="float right clear-none" >float:right – BEZ działania własności clear</span></p>

  <p class="clear-left" >Ten akapit ma zdefiniowane ustawienie clear:left.</p>
  <div class="float left clear-left" >float:left – clear:left</div>
  <div class="float left clear-none" >float:left – BEZ działania własności clear</div>

  <p class="clear-right">Ten akapit ma zdefiniowane ustawienie clear:right.
    <span class="float left clear-left" >float:left </span>
    <span class="float right clear-right">float:right</span></p>

  <p class="clear-both">Ten akapit ma zdefiniowane ustawienie clear:both.</p>
</div>
```

CSS

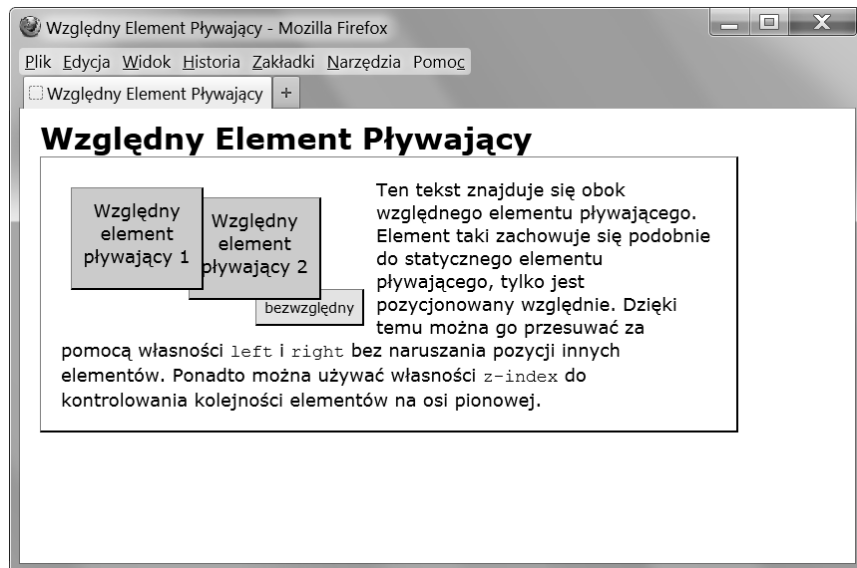
```
*.float { margin:0px 10px; width:120px; background-color:yellow; color:black; }
*.left { float:left; }
*.right { float:right; }
*.clear-left { clear:left; }
*.clear-right { clear:right; }
*.clear-both { clear:both; }
*.clear-none { clear:none; }
```

/ Mniej ważne style nie zostały pokazane. */*

Elementy Pływające

Problem	Chcemy wyjąć element z normalnego układu elementów na stronie i wyświetlić go po lewej lub prawej stronie jego rodzica. Chcemy, aby element ten był rysowany jako blok i wyrównany do wewnętrznej krawędzi dopełnienia kontenera. Ponadto chcemy, aby górna krawędź pozycjonowanego elementu była wyrównana z linią, z której element ten został wyjęty. Dodatkowo chcemy też mieć kontrolę nad tym, czy inne elementy pływające i niepływające mają znajdować się obok pozycjonowanego elementu, czy pod nim.
Rozwiązanie	Aby wyjąć element z normalnego układu i umieścić go przy wewnętrznej krawędzi obszaru dopełnienia jego kontenera, należy zdefiniować mu ustawienie <code>float:left</code> lub <code>float:right</code>. Istnieje też ustawienie <code>float:none</code>, które anuluje wcześniej zdefiniowane ustawienie pływania. Elementy pływające są umieszczane na osobnej warstwie, która znajduje się nad tłem elementów blokowych i obok treści śródliniowej należącej do układu normalnego. Jeśli element spływa w lewo, to opływająca go treść znajduje się po jego prawej stronie, a jeśli element spływa w prawo — treść ta umieszczona zostaje po jego lewej stronie. Elementy pływające nie zmieniają położenia pól blokowych — a jedynie znajdujących się w nich treści śródliniowej. Elementy pływające zmieniają położenie innych elementów pływających i można je ustawiać jeden obok drugiego, zaczynając od lewej lub prawej strony. Elementy pływające mogą też spychać inne elementy pływające i treść śródliniową w dół. Marginesy powodują przesunięcie elementu pływającego względem jego kontenera i innych elementów pływających. Elementy pływające nie nakładają się na inne elementy pływające ani treść (chyba że zostanie im przypisany ujemny margines). Aby wybrany blok lub element pływający przenieść pod elementy pływające znajdujące się po jego lewej stronie, należy mu zdefiniować ustawienie <code>clear:left</code>. Aby wybrany blok lub element pływający przenieść pod elementy pływające znajdujące się po jego prawej stronie, należy mu zdefiniować ustawienie <code>clear:right</code>. Natomiast ustawienie <code>clear:both</code> działa obustronnie.
Wzorce	<pre>SELEKTOR { float:none; } SELEKTOR { float:left; } SELEKTOR { float:right; } SELEKTOR { clear:none; } SELEKTOR { clear:left; } SELEKTOR { clear:right; } SELEKTOR { clear:both; }</pre>
Zastosowanie	Każdy element może być pływający. Własność <code>clear</code> działa na tabele, bloki i elementy pływające, natomiast <i>nie</i> działa na elementy śródliniowe, pozycjonowane bezwzględnie oraz pozycjonowane na stałe.
Wskazówki	Jeśli trzeba sprawdzić, jakie będzie pionowe położenie elementu pływającego, najlepiej jest zamienić w pływający <i>element blokowy</i> . Przeglądarki górną krawędź pływających <i>bloków</i> umieszczają dokładnie w tym samym miejscu, w którym by się ona znajdowała, gdyby dany element nie był pływakiem. Natomiast położenie <i>elementów śródliniowych</i> jest zależne od tego, gdzie znajdowałyby się w obrębie <i>linii</i> , gdyby nie były pływakami. Jeśli na początku linii, to górna krawędź pływaka jest wyrównywana z <i>górną</i> krawędzią linii, a w pozostałych przypadkach — z <i>dolną</i> .
Przykład	Przykład zawiera osiem elementów pływających: cztery elementy <code></code> i cztery elementy <code><div></code> . Cztery akapity demonstrują działanie własności <code>clear</code> . Jeśli element nie ma zdefiniowanej własności <code>clear</code> dla strony, w którą spływa, to ustawia się obok innych elementów spływających w tę samą stronę. Jeśli natomiast ma tę własność zdefiniowaną, to zostaje przeniesiony pod te elementy.
Podobne	Statyczny, Pozycjonowanie Bezwzględne, Pozycjonowanie Stałe

Względny Element Pływający



HTML

```
<h1>Względny Element Pływający</h1>
```

```
<div class="parent">
  <div class="relative1 float" >Względny element pływający 1</div>
  <div class="relative2 float" >Względny element pływający 2</div>
```

```
<p>Ten tekst znajduje się obok względnego elementu pływającego. Element taki zachowuje się
podobnie do statycznego elementu pływającego, tylko jest pozycjonowany względnie. Dzięki temu
można go przesunąć za pomocą własności <code>left</code> i <code>right</code> bez naruszania
pozycji innych elementów. Ponadto można używać własności <code>z-index</code> do kontrolowania
kolejności elementów na osi pionowej.
```

```
<span class="absolute">bezwzględny</span> </p> </div>
```

CSS

```
*.parent    { position:relative; padding:20px; }

*.relative1 { position:relative; z-index:3; top:10px; left:10px; }
*.relative2 { position:relative; z-index:2; top:20px; left:-30px; }

*.float     { float:left; width:110px; height:80px;
              margin-right:25px; margin-bottom:40px; }

*.absolute  { position:absolute; z-index:1; top:132px; left:215px; }
/* Mniej ważne style nie zostały pokazane */
```

Względny Element Pływający

Problem	Chcemy przesunąć element pływający względem jego aktualnego położenia, nie zmieniając pozycji żadnego innego elementu i treści śródliniowej. Ponadto chcemy mieć wpływ na rozmieszczenie elementów pływających na osi poziomej względem innych elementów pływających i elementów pozycjonowanych.
Rozwiązanie	Można zastosować ustawienie <code>position:relative</code> , aby względnie wypozycjonować element pływający. Względnie pozycjonowany element pływający pozostaje w normalnym układzie elementów pływających i można go przesunąć za pomocą własności <code>left</code> i <code>top</code> . Względnie pozycjonowany element pływający jest rysowany na pozycjonowanej warstwie, dzięki czemu za pomocą własności <code>z-index</code> można kontrolować jego miejsce na osi pionowej względem innych elementów pływających i pozycjonowanych. Ponieważ element pływający jest pozycjonowany, można określać w odniesieniu do niego położenie jego bezwzględnie pozycjonowanych potomków.
Wzorzec	<pre>SELEKTOR { position:relative; left:±WARTOŚĆ; right:±WARTOŚĆ; z-index:±WARTOŚĆ; float:LEFT_RIGHT; width:±WARTOŚĆ; height:±WARTOŚĆ; margin:±WARTOŚĆ; }</pre>
Zastosowanie	Ten wzorzec ma zastosowanie do wszystkich elementów.
Zalety	W tym wzorcu za pomocą własności <code>margin</code> można ustawiać położenie treści śródliniowej względem elementu pływającego. Za pomocą własności <code>top</code> i <code>left</code> można ustawiać położenie elementu pływającego, nie zmieniając położenia treści śródliniowej. Daje to duże możliwości pozycjonowania elementów pływających. Gdyby nie ten wzorzec, jedyną możliwością kontroli kolejności na osi poziomej elementów pływających i innych elementów pozycjonowanych byłaby zmiana ich kolejności w kodzie źródłowym.
Wskazówka	Do elementów pływających można stosować tylko ustawienia <code>position:relative</code> i <code>position:static</code> . Efekt przypisania im własności <code>position:absolute</code> lub <code>position:fixed</code> jest nieprzewidywalny i w każdej przeglądarce może być inny. Na przykład niektóre wersje przeglądarki Firefox ustawiają własność <code>float</code> na <code>none</code> i rysują element jako pozycjonowany bezwzględnie, a przeglądarka Internet Explorer 7 częściowo zachowuje ustawienia pływania, a częściowo stosuje ustawienia pozycjonowania.
Przykład	Przedstawiony przykład zawiera dwa względne elementy pływające, statyczny akapit i bezwzględnie pozycjonowany element <code></code> . Za pomocą własności <code>top</code> i <code>left</code> każdy z elementów pływających został przesunięty względem swojej pierwotnej pozycji i nie miało to wpływu na położenie sąsiadującej z nim treści śródliniowej. Za pomocą własności <code>z-index</code> określiliśmy położenie elementów na osi pionowej w kolejności odwrotnej niż kolejność ich zapisu w kodzie źródłowym.
Podobne	Pozycjonowany, Statyczny, Pozycjonowanie Bezwzględne, Pozycjonowanie Stałe, Pozycjonowanie Względne, Elementy Pływające

Skorowidz

!important, dyrektywa, 40
 85
.css, rozszerzenie, 35
:first-child, 92
:nth-child, 92
:nth-last-child, 92
@font-face, własność, 240, 241
@import, 41
@important, 53
@keyframes, 51
<!DOCTYPE>, 68
<a>, element, 81
<abbr>, element, 81
<address>, element, 76, 77
<area>, element, 313
<article>, element, 75
<bdo>, element, 81
<blockquote>, element, 76, 469, 471
<body>, element, 63

, element, 81, 262
<button>, element, 81
<caption>, element, 74, 76, 77
<cite>, element, 81, 469, 473
<code>, element, 80, 81, 254
<col>, element, 74, 345
<colgroup>, element, 74, 345
<dd>, element, 74, 78, 79
, element, 60
<dfn>, element, 81
<div>, element, 74, 78, 83
<dl>, element, 74
<dt>, element, 74, 76
, element, 80, 81
<form>, element, 78, 79
<frame>, element, 60
<frameset>, element, 60
<h1>, element, 76

<head>, element, 63
<hr />, element, 299
<html>, element, 63
<iframe>, element, 60
, element, 81, 311
alt, atrybut, 311
<input>, element, 81
<ins>, element, 60
<kbd>, element, 81
, element, 78
<link>, element, 35, 41
<map>, element, 313
<noframe>, element, 60
<noscript>, element, 78, 79
<object>, element, 81
, element, 74, 289
<p>, element, 76
<pre>, element, 60, 85, 252
<q>, element, 60
<samp>, element, 81
<section>, element, 75, 287
<select>, element, 81
, element, 80, 81, 83
, element, 80, 81
<style>, element, 35, 36, 41
<sub>, element, 81, 276
<sup>, element, 81, 276
<table>, element, 74, 343
<tbody>, element, 74, 345
<td>, element, 74, 78, 79, 343
<textarea>, element, 81
<tfoot>, element, 74, 345
<th>, element, 74, 78, 79, 343
<thead>, element, 74, 345
<tr>, element, 74, 343
, element, 74, 289
<var>, element, 81

A

akapity, tworzenie, 76
 Alert Graficzny, wzorzec, 475, 489, 490
 Alert na Lewym Marginesie, wzorzec, 475, 495, 496
 Alert na Prawym Marginesie, wzorzec, 475, 497, 498
 Alert Śródliniowy, wzorzec, 475, 485, 486
 Alert w Chmurce, wzorzec, 475, 478, 479
 Alert w JavaScript, wzorzec, 475, 476, 477
 alert(), 476
 Alert, wzorzec, 475, 483, 484
 alerty, 475
 almost-standards mode, *Patrz* tryb standardowy
 Arkusz Warunkowy, wzorzec, 57, 71, 72
 arkusze stylów, 35
 <link>, element, 41
 <style>, element, 35, 36, 41
 animacje, 51
 białe znaki, 37, 38
 deklaracje, 36
 jednostki miary, 49, 50
 kaskadowość, 40, 41, 42
 media, 48
 problemy, 52
 przejścia, 51
 przekształcenia dwuwymiarowe, 51, 52
 reguły stylistyczne, 36
 selektory, 36, 47
 składnia, 36
 style, atrybut, 35, 36
 wartości, 36, 38, 45, 46, 47
 warunkowe, 72
 własności, 36, 44, 45, 46, 47
 zewnątrzny plik, 35
 znormalizowane, 53
 Atrybuty Klasy i Identyfikatora, wzorzec, 58, 81, 82

B

background, własność, 109, 144
 background-attachment, własność, 144
 background-color, własność, 104, 144, 231
 background-image, własność, 104, 144, 231, 233
 background-position, własność, 144, 231, 233
 background-repeat, własność, 144, 231, 233
 Bez Zawijania, wzorzec, 245, 250, 251
 białe znaki, 84, 85, 252, 253
 Białe Znaki w HTML, wzorzec, 58, 83, 84
 bloki
 terminalne, 76
 wszechstronne, 78
 blokowe, elementy, 64
 Blokowy, wzorzec, 245, 248, 249
 border, własność, 109, 111, 113, 115, 117, 119, 138, 233
 border-collapse, własność, 117, 343, 349, 351

border-radius, własność, 139
 bottom, własność, 119, 155
 box model, *Patrz* model polowy
 box-shadow, własność, 140

C

children(), 410
 Cień Obrazu, wzorzec, 309, 329, 330, 331, 332, 333
 Cień Tekstu, wzorzec, 227, 234, 235
 class, atrybut, 82
 click(), 409
 closest(), 409
 cm, 50
 color, własność, 231
 CSS
 animacje, 51
 białe znaki, 37, 38
 deklaracje, 36
 dziedziczenie, 102
 jednostki miary, 49, 50
 kaskadowość, 40, 41, 42
 media, 48
 polimorfizm, 29
 pozycjonowanie, 153
 problemy, 52
 profile, 29
 przejścia, 51
 przekształcenia dwuwymiarowe, 51, 52
 reguły stylistyczne, 36
 selektory, 36, 47
 składnia, 36
 tło, 104
 wartości, 36, 38, 45, 46, 47
 własności, 36, 44, 45, 46, 47
 Cufón, technika, 238
 Cytat Blokowy, wzorzec, 457, 468, 469
 Cytat Śródliniowy, wzorzec, 457, 472, 473
 cytaty, 457
 tworzenie, 76

D

dane relacyjne, 284
 display, własność, 105, 107
 run-in, 297
 DOCTYPE, 66, 67
 DOCTYPE, wzorzec, 57, 67
 dolny, indeks, 276
 dopełnienie, 141, 142
 Dopełnienie Treści, wzorzec, 245, 255, 256
 Dopełnienie, wzorzec, 135, 141, 142
 Dostępność Tabel, wzorzec, 341
 DTD, 67

Dystans Śródliniowy, wzorzec, 245, 257, 258
 Dziedziczenie Wizualne, wzorzec, 87, 103, 104
 Dziedziczenie, wzorzec, 87, 101, 102

E

elementy

- blokowe, 64
- dopełnienie, 141, 142
- dotyczące układu, 80, 81
- marginisy, 136, 137
- obramowanie, 138, 139
- plywające, 121, 129, 137, 171, 173, 183
- pozycjonowanie, 153
- przepełnienie, 146
- semantyczne, 80, 81
- statyczne, 159
- strukturalne, 63
- szerokość, 129
- śródliniowe, 64, 80
- tło, 144
- ukrywanie, 148
- wysokość, 129
- względne, 187
- zastępowane, 80, 81

Elementy Nagłówek, wzorzec, 57, 69, 70
 Elementy Pływające, wzorzec, 151, 170, 171
 Elementy Śródliniowe, wzorzec, 58, 79, 80
 em, 49
 ex, 49

F

filter-shadow, własność, 235
 first-letter, selektor, 269, 443
 float, własność, 121, 171
 Font, wzorzec, 227, 228, 229
 font-color, własność, 229
 font-family, własność, 229
 font-size, własność, 229
 font-style, własność, 229
 font-variant, własność, 229
 font-weight, własność, 229
 fonty, 229

- osadzanie, 240, 241
- rodzina, 229

Formatowanie Scalanych Krawędzi, wzorzec, 341, 351, 352, 353
 Formatowanie Zdarzeń, wzorzec, 393, 408, 409, 410
 formularze, 79

- sprawdzanie poprawności, 500

G

GIF, 310, 311
 górny, indeks, 276
 Gradient, wzorzec, 309, 314, 315
 Graficzny Inicjał na Marginesie, wzorzec, 439, 453, 454
 Grafika, wzorzec, 309, 310, 311
 grupowanie selektorów, 91
 Grupowanie Wierszy i Kolumn, wzorzec, 341, 344, 345, 346

H

height, własność, 109, 111, 113, 115, 117, 119, 127, 129

- 100%, 127
- auto, 127

 hierarchie, 283
 hiperłącza, wybór na podstawie stanu, 98
 HTML, 65

- elementy blokowe, 64
- elementy strukturalne, 63
- elementy śródliniowe, 64
- ramki, 60
- struktura, 58, 60, 63

I

id, atrybut, 82
 identyfikatory, 82, 83, 89

- tworzenie, 82

 in, 50
 Indeks Górny i Dolny, wzorzec, 265, 275, 276
 Inicjał na Marginesie, wzorzec, 439, 452, 453
 Inicjał z Grafiką i Dopełnieniem, wzorzec, 439, 446, 447
 Inicjał, wzorzec, 439, 442, 443
 inicjały, 439

J

Jednakowe Kolumny o Rozmiarze Treści, wzorzec, 367, 382, 383
 jednostki miary, 49, 50

- przeliczniki dla 96 dpi, 50

 JPG, 310, 311
 jQuery, 409, 410

K

kaskadowość, 40, 41, 42
 klasy, 82, 89

- tworzenie, 82

 Kod, wzorzec, 245, 253, 254
 Kolumny o Odwróconych Proporcjach, wzorzec, 367, 380, 381

Kolumny o Rozmiarach Procentowych, wzorzec, 367, 378, 379
 Kolumny o Stałej Szerokości, wzorzec, 366, 372, 373
 Kolumny o Takich Samych Rozmiarach, wzorzec, 367, 384, 385
 Kolumny Proporcjonalne do Innych Kolumn, wzorzec, 367, 376, 377
 Kolumny Proporcjonalne do Treści, wzorzec, 367, 374, 375
 Konkretny Rozmiar, wzorzec, 123, 128, 129
 kontekst formatowania śródliniowego, 111
 kontekst stosowy, 159
 Kontekst Stosowy, wzorzec, 151, 158, 159
 kontekst wyrównania, 272
 zagnieżdżanie, 278
 kontrolki, 80, 81
 Kurcziwe Kolumny, wzorzec, 366, 370, 371
 Kurcziwy, wzorzec, 123, 130, 131
 kursywa, 229

L

lang(), 98
 large, 51
 left, własność, 119, 155
 letter-spacing, własność, 247
 Lewy Margines, wzorzec, 281, 303, 304
 line-height, własność, 111, 231, 247
 list-style-image, własność, 289
 list-style-position, własność, 289
 list-style-type, własność, 289
 listy, 284
 definicji, 74
 elementy, 78
 nieuporządkowane, 74, 284
 numerowane, 289
 punktory, 289
 słownikowe, 284
 uporządkowane, 74
 wypunktowane, 289
 Listy, wzorzec, 281, 288, 289, 290

Ł

Łamanie Stron, wzorzec, 135, 149, 150
 łączy, 430
 wybór na podstawie stanu, 98
 Łączy Układowe, wzorzec, 394, 429, 430, 431

M

Mapa Obrazkowa, wzorzec, 309, 312, 313
 margin, własność, 109, 111, 113, 117, 119, 136, 137
 auto, 137
 Margines, wzorzec, 135, 136, 137

marginesy, 136, 137
 maska gradientowa, 315
 max-height, własność, 131
 max-width, własność, 131
 media, 48
 właściwości, 49
 medium, 50
 Menu Rozwijane, wzorzec, 394, 421, 422, 423, 424
 Menu z Zakładkami, wzorzec, 393, 413, 414, 415, 416
 miary, jednostki, 49, 50
 przeliczniki dla 96 dpi, 50
 Mieszane Układy Kolumn, wzorzec, 367, 390, 391
 MIME, 67
 mm, 50
 model połowy, 105
 Model Połowy, wzorzec, 105, 108, 109
 Modele Pozycjonowania, wzorzec, 151, 152, 153
 mouseout(), 420
 mouseover(), 420
 myśl przewodnia, 457
 Myśl Przewodnia na Lewym Marginesie, wzorzec, 457, 464, 465
 Myśl Przewodnia na Prawym Marginesie, wzorzec, 457, 466, 467
 Myśl Przewodnia Spływająca w Lewo, wzorzec, 457, 458, 459
 Myśl Przewodnia Spływająca w Prawo, wzorzec, 457, 460, 461

N

nagłówki, tworzenie, 76
 Najbliższy Pozycjonowany Przodek, wzorzec, 151, 156, 157
 Naprzeciwnie Elementy Pływające, wzorzec, 393, 406, 407
 nawigacja okruszkowa, 430
 Niewidoczny Tekst, wzorzec, 227, 241, 242
 Niewyrośnięte Kolumny, wzorzec, 367, 386, 387
 not(), 410
 nowy wiersz, 84

O

obramowanie, 138, 139
 Obramowanie, wzorzec, 135, 138, 140
 Obraz z Prostym Cieniem, wzorzec, 327, 328
 obrazy
 cień, 328, 330
 formaty, 310, 311
 sprite, 324, 325, 327
 wstawianie, 311
 wymiary, 311

Oddzielanie Krawędzi, wzorzec, 341, 348, 349
 Odstęp między Blokami, wzorzec, 281, 300, 301
 Odstępy Między Elementami Pływającymi, wzorzec, 393, 402, 403
 Odstępy, wzorzec, 245, 246, 247
 onclick, zdarzenie, 477
 Osadzanie Fontów, wzorzec, 227, 240
 overflow, własność, 109, 111, 117, 146, 161
 Ozdoby Śródliniowe, wzorzec, 245, 259, 260
 Ozdoby Tekstu, wzorzec, 227, 232, 233

P

padding, własność, 109, 111, 113, 115, 119, 141
 page-break-after, własność, 150
 page-break-before, własność, 150
 pc, 50
 Pionowe Wyrównanie Danych, wzorzec, 341, 357, 358
 Pionowe Wyrównanie Treści, wzorzec, 265, 271, 272
 Pływająca Sekcja, wzorzec, 393, 400, 401, 402
 Pływający Alert, wzorzec, 475, 493, 494
 Pływający Inicjał z Grafiką, wzorzec, 439, 450, 451
 Pływający Inicjał, wzorzec, 439, 448, 449
 PNG, 310
 pogrubienie, 229
 Pole Bezwzględne, wzorzec, 105
 Pole Blokowe, wzorzec, 105, 114, 115
 Pole Blokowo-Śródliniowe, wzorzec, 105
 Pole Pływające, wzorzec, 105, 120, 121
 Pole Pozycjonowane Bezwzględnie, wzorzec, 118, 119
 Pole Śródliniowe, wzorzec, 105, 110, 111
 Pole Śródliniowo-Blokowe, wzorzec, 112, 113
 Pole Tabeli, wzorzec, 105, 116, 117
 pole wewnętrzne, 397
 polimorfizm CSS, 29
 position, własność, 155
 absolute, 155, 165
 fixed, 155, 167
 relative, 155, 169
 static, 155
 powrót karetki, 84
 Pozioma Kreska, wzorzec, 281, 298, 299
 pozycjonowanie, 153
 bezwzględne, 165
 stałe, 167
 względne, 169
 Pozycjonowanie Bezwzględne, wzorzec, 151, 164, 165
 Pozycjonowanie Stałe, wzorzec, 151, 166, 167
 Pozycjonowanie Względne, wzorzec, 151, 168, 169
 Pozycjonowany, wzorzec, 151, 154, 155
 Półprzezroczysty, wzorzec, 309, 316, 317
 Prawy Margines, wzorzec, 281, 305, 306
 Projektowanie do wewnątrz, a projektowanie na zewnątrz, wzorzec, 398, 399
 Prostý Cień pod Obrazem, wzorzec, 309

przepelnienie, 146
 Przepelnienie, wzorzec, 135, 145, 146
 Przesuwanie Elementów Bezwzględnych i Stałych, wzorzec, 175, 184, 185
 Przesuwanie Elementów Pływających, wzorzec, 175, 182, 183
 Przesuwanie Elementów Statycznych, wzorzec, 175, 178, 179
 Przesuwanie Elementów Wyrównanych do Dołu, wzorzec, 202, 220, 221
 Przesuwanie Elementów Wyrównanych do Góry, wzorzec, 201, 216, 217
 Przesuwanie Elementów Wyrównanych do Lewej, wzorzec, 201, 204, 205
 Przesuwanie Elementów Wyrównanych do Prawej, wzorzec, 201, 208, 209
 Przesuwanie Elementów Wyrównanych do Środka, wzorzec, 201, 212, 213
 Przesuwanie Elementów Wyśrodkowanych Pionowo, wzorzec, 202, 224, 225
 Przesuwanie Elementów Względnych, wzorzec, 175, 186, 187
 Przesuwanie i Wcinanie Statycznych Tabel, wzorzec, 175, 180, 181
 Przesuwanie Treści w Pionie, wzorzec, 273, 274
 Przycisk, wzorzec, 394, 425, 426, 427, 428
 Przykładowy Obraz, wzorzec, 309, 338, 339
 Przykładowy Układ, wzorzec, 394, 434, 435
 pt, 50
 Punktory Graficzne, wzorzec, 281, 290, 291, 292
 px, 50

Q

quirks mode, *Patrz* tryb zgodności wstecznej

R

Redukcja Odstępu między Blokami, wzorzec, 281, 301, 302
 right, własność, 119, 155
 Rodzaje Pól, wzorzec, 105, 106, 107
 rodzina fontów, 229
 Rozciągnięty, wzorzec, 123, 132, 133
 ryalizacja reguła, 40

S

Scalanie Krawędzi, wzorzec, 341, 350, 351
 Scalanie Marginesów, wzorzec, 281, 294, 295
 Sekcja, wzorzec, 281, 286, 287
 sekcje, 287
 Sekcje Rozwijane, wzorzec, 393, 410, 411, 412, 413
 Selektor Podklasy, wzorzec, 87, 99, 100

selektory, 47, 89
 dziecka, 91
 elementu siostrzanego, 92
 first-letter, 96
 first-line, 96
 grupowanie, 91
 identyfikatora, 89
 istnienia atrybutu, 93
 klasy, 89
 pierwszego dziecka, 92
 potomka, 91
 słowa atrybutu, 94
 typu, 89
 uniwersalny, 90
 wartości atrybutu, 94
 Selektory Atrybutów, wzorzec, 87, 92, 93, 94
 Selektory Pozycji i Grupowanie Selektorów,
 wzorzec, 87, 90, 91
 Selektory Pseudoelementów, wzorzec, 87, 95, 96
 Selektory Pseudoklas, wzorzec, 87, 97, 98
 Selektory Tabel, wzorzec, 341, 346, 347
 Selektory Typu, Klasy i Identyfikatora, wzorzec, 87,
 88, 89
 small, 50
 spacja, 84
 niełamliwa, 85
 Sprawdzanie Poprawności Formularzy, wzorzec,
 475, 499, 500
 Sprężyste Kolumny, wzorzec, 367, 388, 389
 Sprite'y CSS, wzorzec, 309, 323, 324, 325, 326
 Statyczny, wzorzec, 162, 163
 strict, 68
 strona, czas otwierania, 55
 Struktura HTML, wzorzec, 57, 63, 64
 Struktura Wizualna, wzorzec, 281, 284, 285
 Strukturalne Elementy Blokowe, wzorzec, 57, 73, 74
 strukturalne, elementy, 63
 style, atrybut, 35, 36
 Szablon, wzorzec, 394, 432, 433
 Szerokość Kolumny, wzorzec, 366, 368, 369
 Szerokość Zewnętrzna Pola, wzorzec, 393, 396, 397
 Szerokość, wzorzec, 123, 124, 125

Ś

Środkowanie Elementów Bezwzględnych, wzorzec,
 176, 196, 197
 Śródliniowa Linia Pozioma, wzorzec, 245, 263, 264
 śródliniowe, elementy, 64
 Śródliniowy Cytat Blokowy, wzorzec, 457, 470, 471
 Śródliniowy, wzorzec, 281, 292, 293

T

Tabela w Paski, wzorzec, 341, 359, 360
 Tabela, wzorzec, 341, 342, 343
 tabele, 74, 193, 343
 grupowanie elementów, 345
 grupy kolumn, 74
 height, własność, 129
 komórki, 74, 79
 modele układu, 365
 nagłówki, 74
 naprzemienne kolory wierszy, 360
 podpis, 74
 stopka, 74
 tło, 345
 ukrywanie komórek, 354
 usuwanie komórek, 354
 usuwanie wierszy, 356
 width, własność, 129
 wiersze, 74
 Tabelowy, Wierszowy i Komórkowy, wzorzec, 341,
 361, 362
 table-layout, własność, 117, 343
 tabulator, 84
 tekst
 cień, 234, 235
 elementy dekoracyjne, 260
 kolor, 229
 kursywa, 229
 nadkreślenie, 233
 podkreślenie, 233
 pogrubienie, 229
 przekreślenie, 233
 rozmiar, 229
 ukrywanie, 241
 wcięcie, 267
 wyjustowanie, 270
 wyrównanie, 270
 wyrównanie do lewej, 203
 wyrównanie do prawej, 207
 wyrównanie do środka, 211
 wyróżnienie, 231
 wysunięcie, 268
 wysunięcie pierwszej litery, 269
 zastępowanie grafiką, 237
 zawijanie wierszy, 251
 Tekst Zastępowany, wzorzec, 309, 318, 319
 Terminalne Elementy Blokowe, wzorzec, 57, 75, 76
 text-align, własność, 270
 text-decoration, własność, 233
 text-indent, własność, 247, 267, 268
 ukrywanie tekstu, 241

text-shadow, własność, 234
 Safari, przeglądarka, 234
 text-transform, własność, 229
 tło, 144
 Tło, wzorzec, 135, 143, 144
 toggleClass(), 410
 top, własność, 119
 transitional, 68
 treść mieszana, 78
 Treść nad Obrazem Tła, wzorzec, 309, 321, 322
 Treść nad Obrazem, wzorzec, 309, 319, 320
 tryb
 dziwactw, *Patrz* tryb zgodności wstecznej
 standardowy, 68
 zgodności wstecznej, 68
 zgodności ze standardami, 68
 Tylko dla Czytników Ekranu, wzorzec, 227, 242, 243

U

układ normalny, 163
 Układ Płynny, wzorzec, 393, 404, 405
 Układ Tabeli, wzorzec, 341, 363, 364
 Układ Wielokolumnowy, wzorzec, 394, 431, 432
 Układy Płynne - Informacje Ogólne, wzorzec, 393, 394, 395
 Ukrywanie i Usuwanie Komórek, wzorzec, 341, 353, 354, 355
 Ukrywanie i Usuwanie Wierszy i Komórek, wzorzec, 341, 355, 356, 357

V

vertical-align, własność, 272, 274, 358
 visibility, własność, 109, 148, 354

W

wcięcie, 176, 177
 Wcięcie, wzorzec, 175, 176, 177
 Wcięcie Tekstu, wzorzec, 265, 266, 267
 webkit-validation-bubble-message, własność, 500
 Wewnętrzny, wzorzec, 151, 160, 161
 white-space, własność, 251, 252, 253
 Widoczność, wzorzec, 135, 147, 148
 width, własność, 29, 109, 111, 113, 115, 117, 119, 125, 129
 100%, 125
 auto, 125
 wiersz
 wcięcie, 267
 wysunięcie, 268
 złamanie, 81, 262

word-spacing, własność, 247
 Wsuwany Alert, wzorzec, 475, 491, 492
 Wsuwany, wzorzec, 281, 296, 297
 Wszechstronne Elementy Blokowe, wzorzec, 57, 77, 78
 wyrównanie
 do dołu, 219
 do góry, 215
 do lewej, 203
 do prawej, 207
 do środka, 211, 223
 Wyrównanie do Dołu, wzorzec, 201, 218, 219
 Wyrównanie do Góry, wzorzec, 201, 214, 215
 Wyrównanie do Lewej, wzorzec, 201, 202, 203
 Wyrównanie do Prawej, wzorzec, 201, 206, 207
 Wyrównanie do Środka, wzorzec, 201, 210, 211
 Wyrównanie Treści w Poziomie, wzorzec, 265, 269, 270
 Wyrównanie Zewnętrzne, wzorzec, 176, 198, 199
 Wyrównany Inicjał, wzorzec, 439, 440, 441
 Wyrównywanie i Przesuwanie Elementów
 Bezwzględnych, wzorzec, 176, 194, 195
 Wyrównywanie i Przesuwanie Statycznych Bloków,
 wzorzec, 175, 190, 191
 Wyrównywanie i Przesuwanie Statycznych Tabel,
 wzorzec, 175, 192, 193
 Wyrównywanie Statycznych Elementów
 Śródliniowych, wzorzec, 175, 188, 189
 Wyróżnienie, wzorzec, 227, 230, 231
 Wyskakujący Alert, wzorzec, 475, 479, 480, 481, 482
 Wysokość, wzorzec, 123, 126, 127
 Wysunięcie Tekstu, wzorzec, 265, 267, 268
 Wysunięty Alert, wzorzec, 475, 487, 488
 Wysunięty Inicjał, wzorzec, 439, 444, 445
 Wyśrodkowana Myśl Przewodnia, wzorzec, 457, 462, 463
 Wyśrodkowanie Pionowe, wzorzec, 202, 222, 223
 Względny Element Pływający, wzorzec, 151, 172, 173
 wzorce projektowe, 30

X

XHTML, 64, 65, 66, 67, 68
 XHTML, wzorzec, 57, 65
 x-large, 51
 x-moz-errormessage, atrybut, 500
 XQUERY, 68
 XSLT, 68
 x-small, 50
 xx-large, 51
 xx-small, 50

Z

- Zaawansowany Przykład Wyrównywania, wzorzec, 265, 279, 280
- Zagnieżdżanie Kontekstów Wyrównywania, wzorzec, 265, 277, 278
- Zakładki, wzorzec, 394, 417, 418, 419, 420, 421
- Zaokrąglone Rogi, wzorzec, 309, 334, 335, 336, 337
- Zastępowanie Tekstu Grafiką, wzorzec, 227, 236, 237
- Zastępowanie Tekstu Kanwą i Obiektami VML, wzorzec, 227, 238, 239
- Zatrzymanie Białych Znaków, wzorzec, 245, 251, 252
- zbiory, 283
- zdarzenia, 409
- z-index, własność, 119, 155, 159
- złamanie wiersza, 81
- Złamanie Wiersza, wzorzec, 245, 261, 262
- Znaczenie Strukturalne, wzorzec, 281, 282, 283, 284

PROGRAM PARTNERSKI

GRUPY WYDAWNICZEJ HELION



1. ZAREJESTRUJ SIĘ
2. PREZENTUJ KSIĄŻKI
3. ZBIERAJ PROWIZJĘ

Zmień swoją stronę WWW
w działający bankomat!

Dowiedz się więcej i dołącz już dzisiaj!

<http://program-partnerski.helion.pl>

GRUPA WYDAWNICZA

 **Helion SA**

Oszalaniające możliwości HTML5 pozwoliły projektantom na tworzenie zaawansowanych stron oraz aplikacji internetowych. Jednak wszędzie tam, gdzie realizowane są rozbudowane projekty, zaczynają się problemy ze złożonością struktury tworzonego rozwiązania. W typowych językach programowania zostało to już zauważone dawno temu, a antidotum na te bolączki stały się wzorce projektowe. Teraz przyszedł czas na HTML — najwyższa pora poznać najlepsze rozwiązania typowych problemów!

Książka *HTML5 i CSS3. Zaawansowane wzorce projektowe* zawiera opis ponad 350 wzorców projektowych w językach HTML5 i CSS3, przeznaczonych do budowy nowoczesnych stron internetowych. Każdy wzorec można stosować w połączeniu z innymi, co daje nieskończoną liczbę możliwości udoskonalania produktów. Dzięki przedstawionym tu technikom nie trzeba stosować żadnych sztuczek i można uniknąć niekończącego się testowania aplikacji w różnych przeglądarkach internetowych. Autorzy szczegółowo opisują wszystkie przydatne własności CSS3 oraz pokazują, jak połączyć je z kodem HTML5, aby uzyskać praktyczne rezultaty. Z tą książką nauczysz się tworzyć płynne układy stron, inicjały, uwagi na marginesie, cytaty i alerty. Ponadto poznasz wiele nowych technik, takich jak zaokrąglanie rogów i sprawdzanie danych wprowadzanych do formularzy.

Dzięki tej książce:

- poznasz najlepsze techniki tworzenia przejrzystego kodu HTML i CSS
- rozwiążesz typowe problemy z tworzeniem układów stron
- bezbłędnie sformatujesz tekst
- stworzysz kod tani w utrzymaniu!

Apress®

Nr katalogowy: 11716

Księgarnia internetowa:
<http://helion.pl>

Zamówienia telefoniczne:
0 801 339900
0 601 339900

helion.pl
księgarnia
internetowa

Sprawdź najnowsze promocje:
• <http://helion.pl/promocje>
Książki najchętniej czytane:
• <http://helion.pl/bestsellery>
Zamów informacje o nowościach:
• <http://helion.pl/novosci>



Helion

Helion SA
ul. Kościuszki 1c, 44-100 Gliwice
tel.: 32 230 98 63
e-mail: helion@helion.pl
<http://helion.pl>

sięgnij po WIĘCEJ



KOD KORZYŚCI

Cena 79,00 zł

978-83-246-4471-1



9 788324 644711

Informatyka w najlepszym wydaniu