

Elementy `div` i style CSS w praktyce

Włodzimierz Gajda
<http://www.gajdaw.pl>

23 listopada 2005
ver. 0.2

Streszczenie

Artykuł omawia kulisy tworzenia strony internetowej wyłącznie w oparciu o elementy `div` oraz kaskadowe arkusze stylów. Jako przykład służy witryna <http://gwz.gajdaw.pl>, która udostępnia ćwiczenia omówione w książce zatytułowanej „*Gimp w zastosowaniach*”.

Spis treści

1. Zanim przystąpimy do kodowania	2
2. Krok 1: główny pojemnik	3
3. Krok 2: cztery główne obszary witryny	4
4. Krok 3: podział kolumn lewej i prawej	6
5. Krok 4: menu pionowe i poziome	6
6. Krok 5: nagłówek, stopka i treść strony	9
7. Uwagi końcowe	11

1. Zanim przystąpimy do kodowania

Przed rozpoczęciem pisania kodu HTML i CSS witryny należy ustalić główne założenia dotyczące serwisu. Zagadnieniami wymagającymi zdefiniowania są: docelowa grupa przeglądarek oraz szerokość układu witryny.

Przeglądarki internetowe dzielimy na trzy główne rodziny:

- pierwsza generacja: przeglądarki nie wspierające CSS,
- druga generacja: przeglądarki częściowo wspierające CSS,
- trzecia generacja: przeglądarki z pełnym wsparciem CSS.

Do pierwszej rodziny zaliczamy między innymi przeglądarki tekstowe oraz archaiczne wersje współczesnego oprogramowania, na przykład **Internet Explorer 4.0** czy **Netscape 4**. Oprogramowanie to interpretuje wybiórczo kilka podstawowych atrybutów CSS.

Przeglądarki drugiej generacji, należą do nich na przykład **Netscape 6** oraz **Internet Explorer 5**, interpretują dużą część CSS, jednak robią to w sposób na tyle niezgodny i niespójny, że przygotowywanie układu witryny w oparciu o style jest zadaniem trudnym.

Wreszcie najnowsza generacja przeglądarek, **Internet Explorer 6**, **Opera 7**, **Firefox** oraz **Netscape 8**, zawiera pełne wsparcie dla CSS, różniąc się jedynie w szczegółach.

Witryna „*Gimp w zastosowaniach*” jest przygotowana z myślą o najnowszych wersjach przeglądarek, które posiadają pełne wsparcie dla CSS. Pozwala to na korzystanie ze wszystkich atrybutów CSS dotyczących warstw oraz pudełek.

Drugim ważnym zagadnieniem jest szerokość układu witryny. Powszechnie stosowanymi rozwiązaniami są:

- układ o stałej szerokości, niezależnej od szerokości okna przeglądarki,
- układ o szerokości zależnej od szerokości okna przeglądarki,
- układ o szerokości zależnej od szerokości okna przeglądarki posiadający minimalną szerokość.

Witryna <http://gwz.gajdaw.pl> posiada stałą szerokość wynoszącą 760 pikseli. Szerokość ta pozwala na pełne wyświetlenie witryny na monitorach o rozdzielczości od 800×600 wzwyż. Owe 40 pikseli marginesu jest zarezerwowane na pionowy pasek przewijania, który jest wyświetlany na podstronach zawierających więcej tekstu. W ten sposób na witrynie nigdy nie pojawia się poziomy pasek przewijania uznawany za błąd w układzie witryny.

2. Krok 1: główny pojemnik

W pierwszym kroku w kodzie HTML umieszczamy sekcję `div`, która będzie pojemnikiem na całą zawartość strony. Sekcji tej nadajemy style:

```
#strona {  
    margin      : auto;  
    border      : 1px solid black;  
    padding     : 0px;  
    width       : 758px;  
}
```

Dzięki czemu będzie ona wyśrodkowana na stronie (właściwość `margin`) oraz będzie posiadała zadaną szerokość. Szerokość całego elementu `div` wynosi 760 pikseli, z czego 758 jest przeznaczone na zawartość, a 2 na obramowanie.

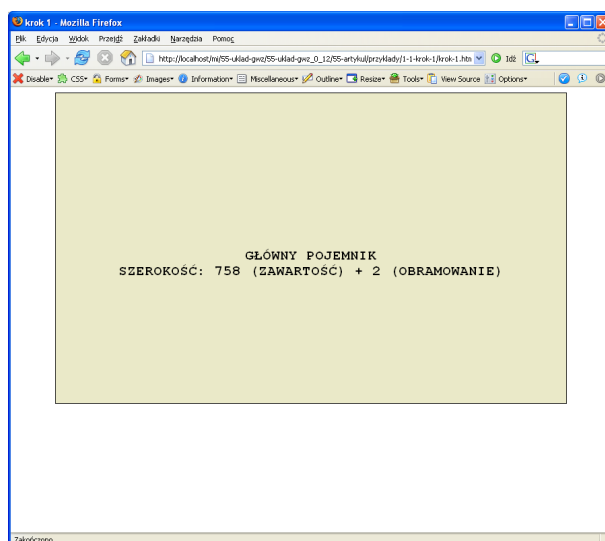
Cała zawartość witryny będzie umieszczona wewnątrz głównego pojemnika `div` o identyfikatorze `#strona`.

Style zapisujemy w pliku `style.css` i dołączamy do kodu strony w następujący sposób:

```
@import url(style.css);
```

dzięki czemu staną się one niedostępne dla przeglądarek pierwszej generacji. W ten sposób stare przeglądarki (pierwszej generacji) wyświetlą wprawdzie nieatrakcyjną, ale za to czytelną witrynę.

Pierwsza z ilustracji przedstawia wygląd witryny zawierającej jedynie główne pudełko `div#strona` po modyfikacji koloru tła.



Rysunek 1. Główny pojemnik

3. Krok 2: cztery główne obszary witryny

Pojemnik główny witryny dzielimy następnie na cztery podobszary: nagłówek, stopkę, lewą kolumnę oraz prawą kolumnę. Podział ten realizujemy umieszczając w głównym pojemniku cztery sekcje `div`, tak jak to przedstawiono na listingu 1.

```
<div id="strona">

    <div id="naglowek">
    </div>

    <div id="kolumnaLewa">
    </div>

    <div id="kolumnaPrawa">
    </div>

    <div id="stopka">
    </div>

</div>
```

Listing 1. Główny kontener po podzieleniu na cztery obszary

Szerokości nagłówka i stopki odpowiadają szerokości głównego pojemnika i wynoszą 758 pikseli. Natomiast szerokości kolumn lewej i prawej mają tak ustalone wartości, by łącznie tworzyły blok o szerokości 758 pikseli. Kolumna lewa nie posiada obramowania, a jej szerokość wynosi 180 pikseli. Natomiast kolumna prawa posiada lewą jednopikselową krawędź, więc jej szerokość musi wynosić 577 pikseli. Tak by w sumie:

$$180 + 1 + 577 = 758$$

Elementy `div` zawarte w głównym kontenerze przylegają ściśle do siebie nawzajem i do brzegów głównego pojemnika. Efekt ten osiągniemy za pomocą atrybutów `margin`, `padding` oraz `border` nadając im wartości:

```
margin   : 0px;
border   : none;
padding  : 0px;
```

Szerokości elementów ustalamy oczywiście atrybutem `width`.

Kolumny lewą i prawą umieszczamy z lewej i z prawej strony za pomocą atrybutów `float`:

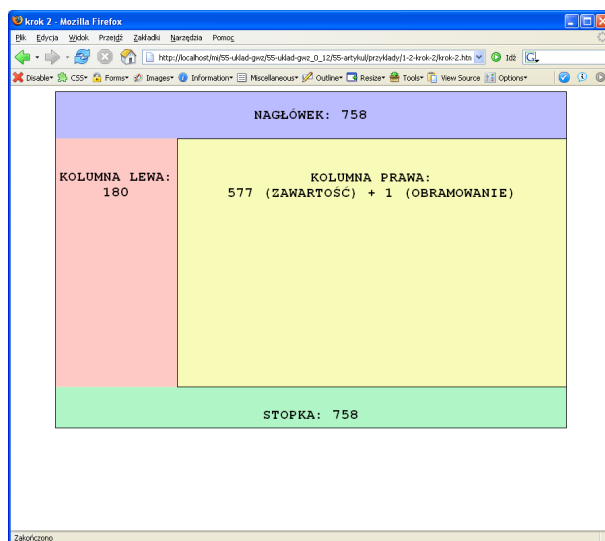
```
float : left;  
float : right;
```

Natomiast stopkę przesuwamy pod spód kolumn lewej i prawej atrybutem `clear`:

```
clear : both;
```

Otrzymany układ, z zaznaczeniem poszczególnych fragmentów i ich szerokości jest przedstawiony na ilustracji 2. Na tym etapie pracy warto sprawdzić, czy otrzymana witryna wygląda zgodnie z oczekiwaniami we wszystkich przeglądarkach, na jakie jest opracowywana. W naszym przypadku są to cztery przeglądarki: Internet Explorer 6, Opera 7, FireFox oraz Netscape 8. Po stwierdzeniu, że witryna wygląda dobrze należy jeszcze dokonać sprawdzenia faktycznych rozmiarów sekcji `div`. W tym celu modyfikujemy kolor tła wszystkich sekcji `div`, a następnie dokonujemy pomiaru na przykład za pomocą programu **Miarka**.

Jakiegolwiek niedociągnięcia zbagatelizowane na tym poziomie (w szczególności choćby jednopikselowa różnica szerokości którejkolwiek z sekcji `div`) będą powodowały zamieszanie w kolejnych etapach pracy. Z tym, że im później, tym trudniej będzie odnaleźć błąd w kodzie HTML czy CSS.



Rysunek 2. Nagłówek, stopka, lewa i prawa kolumna

4. Krok 3: podział kolumn lewej i prawej

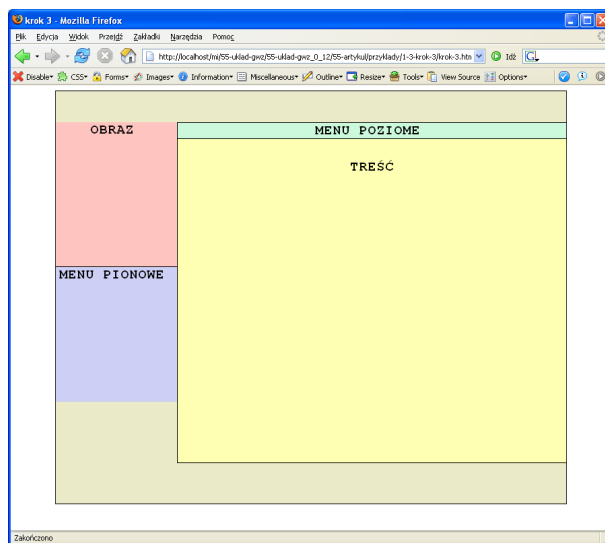
Następnie kolumny lewą i prawą dzielimy na mniejsze obszary. Tym razem podział jest poziomy.

W kolumnie lewej umieszczamy dwie sekcje `div`: jedną przeznaczoną na ilustrację, drugą — na menu pionowe. Sekcje te otrzymają identyfikatory `obrazOkladka` oraz `menuPion`. Obie one mają szerokość 180 pikseli i ściśle przylegają do brzegów sekcji nadrzędnej (czyli `div#kolumnaLewa`).

Podobnie, w kolumnie prawej umieszczamy dwie sekcje `div`. Pierwsza z nich, `div#menuPoziome`, jest przeznaczona na menu, zaś druga, `div#tresc` — na zawartość strony. Sekcje te mają łącznie szerokość 577 pikseli i przylegają ściśle do sekcji nadrzędnej `div#kolumnaPrawa`.

Otrzymamy kod HTML zawiera wszystkie elementy `div`, jakie będą stanowiły układ witryny. Kod ten jest zawarty na listingu 2.

Ilustracja 3 natomiast przedstawia wygląd otrzymanej witryny z zaznaczeniem poszczególnych jej obszarów. Ponownie dokonujemy sprawdzenia wyglądu witryny oraz rozmiarów poszczególnych sekcji.



Rysunek 3. Podział kolumny lewej i prawej

5. Krok 4: menu pionowe i poziome

Kolejnym etapem tworzenia witryny jest przygotowanie menu poziomego i pionowego oraz ilustracji w lewym górnym rogu witryny. Ilustracja jest wstawiona zwykłym elementem `img`, zaś menu tworzymy w oparciu o listy wypunktowane `ul` i odpowiednie arkusze stylów. Niezmiernie ważne jest, by otrzymane menu miało odpowiednią szerokość. Menu pionowe umieszczone w lewej kolumnie musi mieć sze-

```
<div id="strona">

    <div id="naglowek">
    </div>

    <div id="kolumnaLewa">

        <div id="obrazOkładka">
        </div>

        <div id="menuPion">
        </div>

    </div>

    <div id="kolumnaPrawa">

        <div id="menuPoziome">
        </div>

        <div id="tresc">
        </div>

    </div>

    <div id="stopka">
    </div>

</div>
```

Listing 2. Podział kolumn lewej i prawej na podobszary

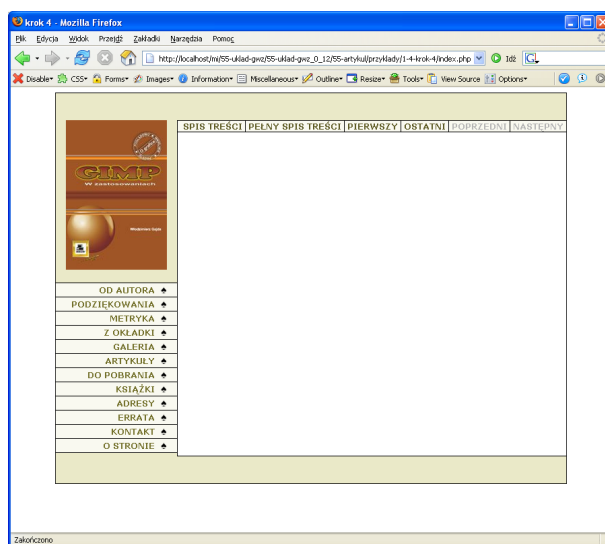
rokość 180 pikseli, natomiast menu poziome, umieszczone w kolumnie prawej ma szerokość 577 pikseli.

Najpierw przygotowujemy menu pionowe, a następnie menu poziome. Opracowując menu poziome natrafimy na pewien problem.

Wykonanie menu poziomego wymaga usunięcia białych znaków (w tym znaków złamania wiersza) pomiędzy elementami listy ul. Oczywiście możemy usunąć białe znaki z kodu HTML, jednak jest to bardzo niewygodne, gdyż utrudnia dalszą edycję strony. Problem ten rozwiązaliśmy stosując szablony **Smarty**, w szczególności funkcję `{strip}`. Kod menu poziomego umieszczamy w szablonie, który w całości jest objęty funkcją `{strip}`. Na początku pliku umieszczamy znacznik **Smarty** `{strip}`, a na końcu — `{/strip}`. W ten sposób otrzymamy witrynę, która w trybie edycji zawiera białe znaki (podział na wiersze i wcięcia ułatwiające pracę z kodem HTML). Natomiast po przesłaniu do przeglądarki, ta sama witryna jest już pozbawiona białych znaków.

Dla ułatwienia, oba menu przygotowujemy na osobnej pustej stronie, zawierającej ewentualnie jedną sekcję `div`. Gdy oba menu są gotowe i mają odpowiednią szerokość (wygląd i rozmiar sprawdzamy we wszystkich przeglądarkach!), wklejamy je w stronę z poprzedniego kroku, po czym dokonujemy sprawdzenia otrzymanej witryny.

Strona otrzymana po dodaniu menu poziomego, menu pionowego i obrazka jest widoczna na rysunku 4. Od tego kroku tworzona witryna będzie już stosowała szablony **Smarty**.



Rysunek 4. Strona po dodaniu menu poziomego i pionowego

6. Krok 5: nagłówek, stopka i treść strony

Kończąc pracę nad witryną, dodajemy treść nagłówka oraz stopkę strony. Elementy te wymagają jedynie stylów modyfikujących atrybuty czcionki. Ich rozmiar i położenie zostały już ustalone. W stopce witryny możemy umieścić hiperłącza do adresów stron internetowych <http://validator.w3.org/check?uri=referer> oraz <http://jigsaw.w3.org/css-validator/check/referer>. Odnośniki takie umożliwią szybkie sprawdzenie poprawności kodu HTML oraz CSS walidatorami W3C (oczywiście strona musi być uprzednio opublikowana w Internecie).

W miejscu przeznaczonym na treść witryny umieszczamy kilka akapitów tekstu *Lorem ipsum*. Do tego stosujemy funkcję `{include}` szablonów Smarty:

```
{include file="zawartosc.tpl"}
```

Dzięki temu umieszczenie odpowiedniej zawartości na stronie będzie się odbywało w pliku `zawartosc.tpl`, czyli bez jakichkolwiek modyfikacji układu witryny. Otrzymana witryna jest przedstawiona na rysunku 5.

Lorem ipsum to początkowe dwa wyrazy rozprawy napisanej w starożytności przez Cyserona. Rozprawa ta jest od ponad 500 lat stosowana przez drukarzy do sprawdzania krojów pism i układu graficznego drukowanych publikacji. Powodem tego jest fakt, że oglądanie tekstu w nieznanym języku pozwala się skupić na jego wizualnych aspektach. Tekst w znanym języku podświadomie zaczynamy czytać, co odrywa uwagę od jego graficznych cech.

W środowisku twórców stron internetowych tekst *Lorem ipsum* jest powszechnie stosowany jako wypełniacz szkieletów stron.

Więcej informacji jak również pełny tekst *Lorem ipsum* znajdziemy na stronach Wikipedii pod adresem http://pl.wikipedia.org/wiki/Lorem_ipsum.

Warto również zajrzeć na <http://www.lipsum.com> oraz <http://www.loremipsum.net>.

W ten sposób otrzymamy witrynę, w której układ strony jest oddzielony fizycznie od treści. Należy odpowiednio zmodyfikować adresy URL zawarte w znacznikach `` menu poziomego i pionowego oraz zorganizować „wlewanie” zawartości witryny w szablonie `zawartosc.tpl`.

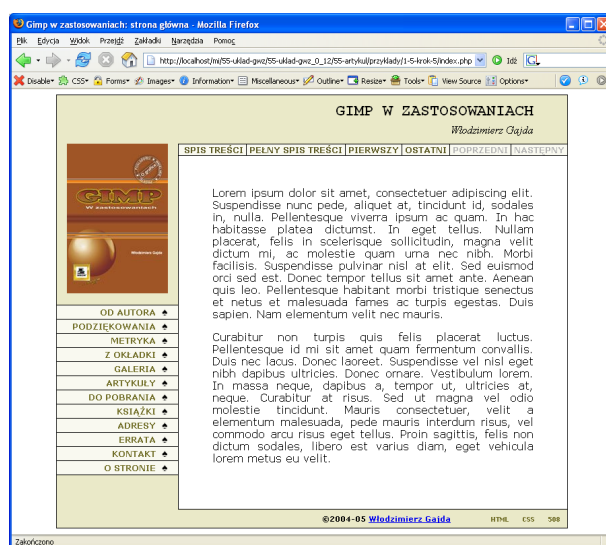
Oczywiście treść zawarta w sekcji `div#tresc` nie może nigdy przekroczyć maksymalnej dopuszczalnej szerokości, czyli 577 pikseli. Jeśli tak się stanie, układ witryny

będzie się rozjeżdżał. Do ustalenia stylów elementów zawartych w sekcji `div#tresc` stosujemy selektory typu potomek. Na przykład akapity zawarte w treści formatujemy umieszczając w pliku stylów wpisy:

```
#tresc p {  
    ...  
}
```

Stosowanie selektorów typu potomek z jednej strony nie będzie powodowało żadnych zmian w formatowaniu elementów odpowiedzialnych za układ witryny, a z drugiej — nie będzie wymagało umieszczania w kodzie HTML wszechobecnych identyfikatorów. Powyższy selektor `#tresc p` dotyczy jedynie elementów `p` zawartych wewnątrz sekcji `div` o identyfikatorze `tresc`:

```
<div id="tresc">  
    <p>  
        Lorem ipsum...  
    </p>  
</div>
```



Rysunek 5. Szablon strony wypełniony tekstem *Lorem ipsum*

7. Uwagi końcowe

Gotowa witryna jest przedstawiona na rysunku 6. Opracowując kod witryny warto od początku dbać o jego przejrzystość. Style dotyczące poszczególnych sekcji `div` komentujemy i umieszczamy w kolejności ułatwiającej późniejszą edycję. Na przykład:

- style ogólne,
- style dotyczące układu (tj. sekcji `div` z listingu 2),
- poszczególne fragmenty układu (np. menu poziome, menu pionowe),
- oraz pozostałe.

Porządkując style każdego z selektorów CSS warto stosować raz zdefiniowaną kolejność atrybutów. Możemy na przykład (taka kolejność jest stosowana w przykładach) na początek przenieść wpisy dotyczące pudełka i nadać im kolejność: `margin`, `border`, `padding`, `width`, pozostałe: np. `float`, `clear` (kolejność `margin`, `border`, `padding`, `width` jest posortowana od najbardziej zewnętrznego do najbardziej wewnętrznego atrybutu pudełka). Natomiast wpisy dotyczące kolorów umieszczamy jako ostatnie. W środkowej części, grupujemy pozostałe atrybuty (np. czcionki, czy atrybuty dotyczące tekstu).

Ponadto już na samym początku pracy nad witryną należy zadbać o kolejność sekcji `div` w kodzie HTML. Kolejność ta jest istotna dla przeglądarek nie interpretujących stylów. W omawianym przykładzie witryna będzie zawierała kolejno:

- nagłówek,
- obraz tytułowy,
- menu pionowe,
- menu poziome,
- treść,
- oraz stopkę.

Zmiana kolejności elementów `div` decydujących o układzie witryny w skończonym projekcie jest bardzo pracochłonna, gdyż pociąga za sobą właściwie nowy projekt układu.

Na zakończenie dodajmy, że nawet najnowsze wersje przeglądarek nie są do końca zgodne ze sobą i zawierają pewne błędy. W omawianym przykładzie błędy takie wystąpiły w dwóch miejscach. Po pierwsze przy wyznaczaniu minimalnej wysokości pojemnika `div#tresc` przeznaczonego na treść, po drugie w przypadku szerokości akapitów zawartych w treści.

Wysokość sekcji o identyfikatorze `tresc` ustalamy sztucznie na 400 pikseli:

```
#tresc {
    height : 400px;
}
```

Podany atrybut jest interpretowany przez przeglądarkę IE jako co najmniej 400 pikseli zaś przez trzy pozostałe jako dokładnie 400 pikseli. Ponieważ IE nie interpretuje selektorów typu dziecko, które zawierają znak `>`, zatem wszystkie przeglądarki poza IE informujemy jeszcze, że tak naprawdę wysokość sekcji `div#tresc` wynosi co najmniej 400 pikseli:

```
body>#strona>#kolumnaPrawa>#tresc {
    height      : auto;
    min-height  : 400px;
}
```

W taki oto skomplikowany sposób walczymy z trzema błędami IE: błędą interpretacją atrybutów `height` i `min-height` oraz brakiem interpretacji selektora dziecko.

Drugi problem, błędnie wyznaczaną szerokość akapitu, poprawiamy dodając wpis:

```
#tresc p {
    width : 470px;
}
```

który w przypadku pozostałych przeglądarek jest zbędny.

Niestety tak wykonana witryna nie wygląda poprawnie w przeglądarkach drugiej generacji.



Rysunek 6. Strona główna witryny <http://gwz.gajdaw.pl>

lp.	Przykład	Pobierz
1.	Główny pojemnik	krok-1.zip
2.	Nagłówek, stopka, kolumny lewa oraz prawa	krok-2.zip
3.	Podział kolumn lewej i prawej	krok-3.zip
4.1	Menu pionowe	krok-4-1.zip
4.2	Menu poziome	krok-4-2.zip
4.3	Witryna zawierająca menu poziome i pionowe	krok-4-3.zip
5.	Szablon witryny	krok-5.zip
6.	Witryna „ <i>Gimp w zastosowaniach</i> ” w Internecie	-

Tabela 1. Kolejne kroki powstawania witryny

lp.	Nazwa	Pobierz
1.	Miarka	miarka_0_10.zip

Tabela 2. Oprogramowanie wymienione w artykule