

## Kaskadowe arkusze stylu

Dodał Administrator  
piątek, 23 lipiec 2010 09:20

---

Wprowadzenie przez konsorcjum W3C mechanizmu kaskadowych arkuszy stylu (ang. Cascade Style Sheets — CSS) zakończyło epokę niezgodnych ze sobą rozszerzeń języka HTML, wprowadzanych przez wszystkich liczących się producentów przeglądarek WWW, dając twórcom stron internetowych eleganckie i spójne narzędzie umożliwiające precyzyjne określenie wyglądu strony.

Cóż takiego specjalnego jest w kaskadowych arkuszach stylu? Przede wszystkim ich stosowanie jest niezwykle jednolite — prawie każdy element języka XHTML możesz rozbudować o atrybut style, wewnątrz którego możesz stosować ściśle określony zestaw atrybutów stylu. Arkusze stylu pozwalają też ujednoczyć wygląd serwisu internetowego oraz ograniczyć ilość danych pobieranych z Internetu podczas oglądania stron WWW. Jeśli przygotowujesz prezentacje internetowe i przewidujesz możliwość ich drukowania, zainteresować Cię może opcja stworzenia dwóch odrębnych arkuszy stylu dla tego samego dokumentu: jednego zoptymalizowanego pod kątem ekranu komputera, a drugiego — pod kątem drukarki.

Kaskadowe arkusze stylu możesz wykorzystać na trzy sposoby, stosując:

-  
styl wpisany,

-  
styl osadzony,

-  
zewnątrzne arkusze stylu.  
**Styl wpisany**

Ten sposób wykorzystania kaskadowych arkuszy stylu jest Ci już doskonale znany. Polega on na rozbudowywaniu wybranego elementu języka XHTML, którego wygląd chcesz zmienić, o atrybut style, którego wartością jest definicja stylu:

```
<span style="color: blue; background-color: yellow;">Niebieski
```

```
tekst na żółtym tle</span>
```

Zaletą tej metody jest jej bezpośredniość — możesz zmienić wygląd dowolnego, najmniejszego choćby elementu strony całkowicie niezależnie od wyglądu pozostałych elementów. Z tej zalety wynika również największa wada tej metody — zmiana wyglądu większej liczby elementów wymaga wprowadzania olbrzymiej ilości kodu.

Stosowanie stylu wpisanego ma sens w momencie, gdy chcesz zmienić wygląd kilku wyrazów lub linijek tekstu. Zanim jednak zastosujesz styl wpisany, zastanów się, czy nie warto zastosować stylu osadzonego. W prezentowanych dotychczas przykładach stosowałem w nadmiarze styl wpisany, gdyż jest on najprostszą formą kaskadowych arkuszy stylu. Jednak w większości przypadków zastosowanie stylu osadzonego pozwoliłoby znacznie ograniczyć ilość kodu składającego się na przykład.

Stosując styl wpisany, pamiętaj, że zalecane jest ograniczanie jego użycia. Kolejne edycje standardu XHTML mogą wręcz całkowicie usunąć obsługę stylu wpisanego, jeszcze silniej oddzielając strukturę logiczną dokumentu od opisu formy jego prezentacji. **Styl osadzony**

Styl osadzony jest najchętniej stosowanym elementem kaskadowych arkuszy stylu. Definicję stylu osadzonego tworzy się w całkowitym oderwaniu od rzeczywistego elementu — określamy rodzaj czcionki, kolor tekstu i szerokość marginesu, mając na myśli albo pojedynczy, konkretny element strony, albo też całą klasę elementów. Definicja stylu może określać wygląd wszystkich wystąpień wybranych elementów języka XHTML lub też wyłącznie

ich wydzielonych podklas.

Raz stworzona definicja stylu osadzonego może być wykorzystana później do zmiany wyglądu dowolnej liczby elementów strony WWW. Co więcej, jeśli nagle zapragniesz zmienić nieco tę definicję, automatycznie zmieni się wygląd wszystkich elementów, których wygląd ta definicja określa! Nie muszę mówić, jak wielkim jest to udogodnieniem.

Trudno się dziwić, że w zasadzie wygląd każdej większej strony WWW opisany jest właśnie za pomocą osadzonego arkusza stylu. Zmniejsza się w ten sposób rozmiar pliku HTML (definicja stylu umieszczona jest tylko w jednym miejscu kodu), zaś ewentualne zmiany wyglądu strony nie wymagają wprowadzania poprawek w dziesiątkach lub setkach miejsc kodu. **Styl dołączony**

Zewnętrzny, dołączony arkusz stylu nie różni się zasadniczo od osadzonego; rozszerza tylko jego uniwersalność, przenosząc definicje stylów z wnętrza kodu jednej strony WWW do osobnego pliku, który może być wykorzystany w dziesiątkach, setkach lub nawet tysiącach stron składających się na serwis internetowy.

Zewnętrzny arkusz stylu jest dla serwisu internetowego tym, czym styl osadzony dla pojedynczej strony — umożliwia scentralizowanie definicji stylu i uniknięcie konieczności wprowadzania poprawek w wielu plikach przy każdej najdrobniejszej zmianie Twojego projektu. Zewnętrzny arkusz stylu jeszcze bardziej ogranicza ilość danych, które czytelnik musi pobrać z Sieci — raz pobrany plik arkusza stylu pozostaje w pamięci podręcznej przeglądarki i jest dostępny natychmiast dla każdej kolejnej strony Twojego serwisu. **Dlaczego „kaskadowe”?**

Do wyjaśnienia pozostała jeszcze tylko jedna kwestia — dlaczego kaskadowe arkusze stylu nazywają się „kaskadowymi”? Wynika to ze sposobu ich działania; prawie każdy element podrzędny (a więc na przykład element `<span>` zawarty wewnątrz elementu `<p>`) dziedziczy atrybuty stylu elementu nadrzędnego, a wybrane z nich może samodzielnie nadpisywać. W efekcie powstaje swoista kaskada definicji stylu, której wyższe stopnie narzucają wygląd stopniom niższym, a te z kolei mogą anulować wybrane elementy definicji i zastąpić je własnymi, które z kolei spróbują narzucić własnym elementom podrzędnym.

Ponadto wygląd elementów może być precyzowany na kilku poziomach logicznych. Nic nie stoi na przeszkodzie, by określić ogólny wygląd jakiegoś elementu języka XHTML w dołączonym, zewnętrznym arkuszu stylu, zmodyfikować tę definicję w arkuszu osadzonym, a w przypadku wybranych elementów jeszcze wprowadzić drobne poprawki za pomocą stylu wpisanego.

Trzecim powodem, dla którego stosuje się słowo „kaskadowy”, jest możliwość określania wyglądu elementu w zależności od tego, w jakim miejscu w strukturze dokumentu znajduje się dany element strony. Inny styl można przypisać akapitom tekstu znajdującym się bezpośrednio w treści strony, a inny — umieszczonym w kontenerze o konkretnym identyfikatorze. Dzięki temu treść nie musi być rozszerzana o zbędne identyfikatory określające funkcję danego elementu. **Zmiana czcionki, kolorów tekstu i tła oraz marginesów**

### tekstu

O tym, jak zmieniać czcionkę, którą złożony jest tekst, kolor tej czcionki, kolor tła tekstu oraz szerokość marginesów elementu języka XHTML dowiedziałeś się już w części tego kursu poświęconej formatowaniu akapitów i bloków tekstu. Tutaj omówię te zagadnienia jeszcze raz w wielkim skrócie, by przypomnieć Ci sposób stosowania stylu wpisanego.

Za zmianę kroju pisma, którym rysowany jest tekst zapisany wewnątrz elementu, odpowiada atrybut stylu `font-family`. W jego wnętrzu podaje się listę sugerowanych krojów pisma w kolejności od preferowanego aż po najbardziej ogólny. Jeżeli nazwa kroju zawiera znaki odstępu, należy zamknąć ją w pojedynczych znakach cudzysłowu:

```
font-family: 'Trebuchet MS',Arial,sans-serif;
```

## Kaskadowe arkusze stylu

Dodał Administrator  
piątek, 23 lipiec 2010 09:20

---

Nazwy ogólnych krojów pisma to:

-

serif — czcionka szeryfowa, na przykład Times New Roman [\[RS1\]](#) ,

-

sans-serif — czcionka bezszeryfowa, na przykład Arial [\[RS2\]](#) ,

-

cursive oraz fantasy — ozdobne czcionki pochyle (rzadko używane, gdyż każda przeglądarka może podstawić w ich miejsce całkowicie inne kroje pisma — często w ogóle nieprzystające do zamysłu autora strony),

-

mono — czcionka nieproporcjonalna (o stałej szerokości wszystkich znaków), używana najczęściej do zapisywania tekstu programów komputerowych, tworzenia prymitywnych tabel lub wyróżniania poleceń programów; przykładem takiej czcionki jest Courier New [\[RS3\]](#) .

Za rozmiar czcionki odpowiada atrybut stylu font-size. Rozmiar najczęściej podaje się w punktach drukarskich (pt; każdy punkt to  $\frac{1}{72}$  część cala):

font-size: 12pt;

Atrybut stylu color pozwoli Ci zdefiniować kolor wyświetlanego tekstu. Podana w nim wartość może być zestandaryzowaną nazwą barwy, opisem natężenia składowych RGB zapisanym szesnastkowo (#RGB lub #RRGGBB) lub opisem natężenia zapisanym w eleganckiej postaci rgb(r,g,b):

color: blue;

color: #A05;

color: #0015A0;

color: rgb(100, 0, 255);

Atrybut stylu background-color określa kolor tła tekstu lub elementu. Oznaczenia koloru można podawać we wszystkich formach opisanych powyżej:

background-color: yellow;

Za pomocą atrybutu stylu margin możesz zdefiniować szerokość zewnętrznych marginesów akapitu tekstu lub elementu języka XHTML. Wartość atrybutu stylu margin składa się z czterech elementów oddzielonych znakami spacji i podanych w następującej kolejności:

- szerokość górnego marginesu,
- szerokość prawego marginesu,
- szerokość dolnego marginesu,
- szerokość lewego marginesu.

margin: 1cm 1cm 1cm 1cm;

Jeżeli chciałbyś wyzerować szerokości marginesów (usunąć je całkiem), możesz też użyć skróconej postaci:

margin: 0;

## Kaskadowe arkusze stylu

Dodał Administrator  
piątek, 23 lipiec 2010 09:20

---

Za wewnętrzny margines (zawierający w sobie kolor tła elementu) odpowiada z kolei atrybut stylu padding, którego zasady stosowania są identyczne:

```
padding: 5pt 1cm 10px 0;
```

```
padding: 0;
```

Kompletny kod znacznika otwierającego element `<p>` wraz z definicją stylu może wyglądać więc mniej więcej następująco:

```
<p style="font-family: 'Trebuchet MS',Arial,sans-serif; font-size: 12pt;
```

```
color: blue; background-color: yellow; margin: 0; padding: 5pt 5pt 5pt 5pt;">
```

---

[\[RS1\]](#)[Skład]: pozostawić czcionkę „times-opodobną”

[\[RS2\]](#)[Skład]: pozostawić czcionkę „helvetica-podobną”

[\[RS3\]](#)[Skład]: pozostawić czcionkę „couriero-podobną” **Zmiana wyglądu elementu języka**

## XHTML

Najprostsze zastosowanie stylu osadzonego polega na zmianie wyglądu wszystkich elementów jednego typu obecnych na stronie. Za jednym zamachem możesz zmienić wygląd wszystkich akapitów tekstu, przeddefiniowując styl elementu `<p>`, lub wszystkich komórek tabeli, zmieniając definicję stylu elementu `<td>`. Oczywiście wygląd tych elementów strony, których styl zmieniłeś bezpośrednio — na przykład korzystając ze stylu wpisanego — nie ulegnie zmianie.

Możliwość zmiany wyglądu wszystkich elementów strony przyda Ci się choćby po to, by jak najmniejszym nakładem pracy uczynić widocznym tekst strony, której kolor tła został zmieniony ze standardowego — białego — na ciemnoniebieski lub czarny.

Utworzenie bloku definiującego style osadzone polega na stworzeniu wewnątrz nagłówka kodu strony (elementu `<head>`) nowego elementu `style` o następującej postaci:

```
<style type="text/css">
```

```
...
```

```
</style>
```

W miejscu kropek wstawiane będą poszczególne deklaracje stylu. Wszystkie one będą miały następującą, ogólną postać:

```
nazwastyłu {
```

```
  atrybut1: wartosc1;
```

```
  atrybut2: wartosc2;
```

```
...
```

## Kaskadowe arkusze stylu

Dodał Administrator  
piątek, 23 lipiec 2010 09:20

---

```
    atrybutN: wartoscN;
```

```
}
```

Czas zastosować tę wiedzę w praktyce. Zmieńmy najpierw kolor tła strony (elementu <body>) na niebieski, na razie używając znanego Ci już doskonale mechanizmu stylu wpisanego (rysunek 7.1):

```
<?xml version="1.0" encoding="utf-8"?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pl" lang="pl">

<head>

<title>Style CSS</title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

</head>

<body style="background-color: blue;">

<p>Akapit 1</p>

<p>Akapit 2</p>

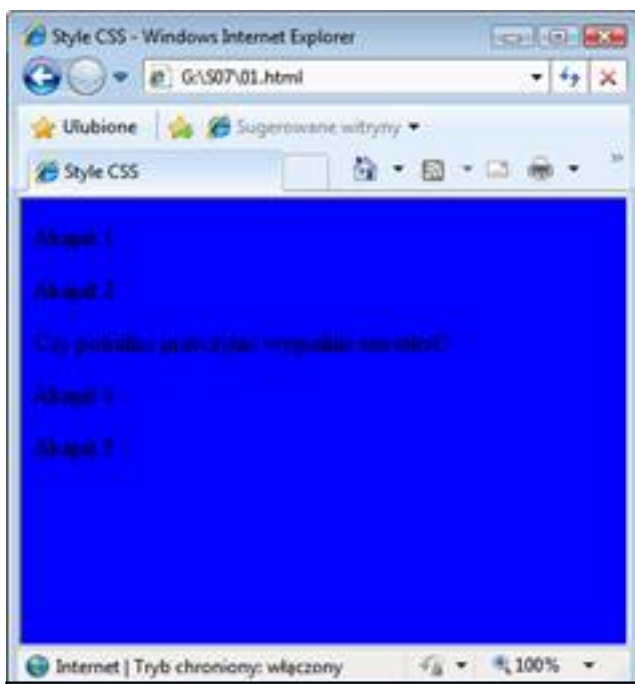
<p>Czy potrafisz przeczytać wygodnie ten tekst?</p>

<p>Akapit 4</p>

<p>Akapit 5</p>

</body>

</html>
```



## Kaskadowe arkusze stylu

Dodał Administrator  
piątek, 23 lipiec 2010 09:20

---

Rysunek 7.1. Niebieskie tło strony wymaga zmiany koloru czcionki — trudno jednak wymagać od autora strony WWW, by rę

Trudno powiedzieć, aby tekst był czytelny. Prezentowałby się o wiele lepiej, gdyby był zapisany kolorem białym lub żółtym. Gdybyś jednak chciał zmieniać kolor każdego akapitu, czekałoby Cię poprawianie deklaracji stylu w pięciu miejscach — a co począć z większymi stronami lub częstszymi zmianami kolorystyki?

Deklaracja stylu osadzonego pozwala za jednym zamachem zmienić atrybuty stylu wszystkich elementów języka XHTML danego typu. Wystarczy dodać do pliku (wewnątrz elementu <head>) następujący fragment:

```
<style type="text/css">

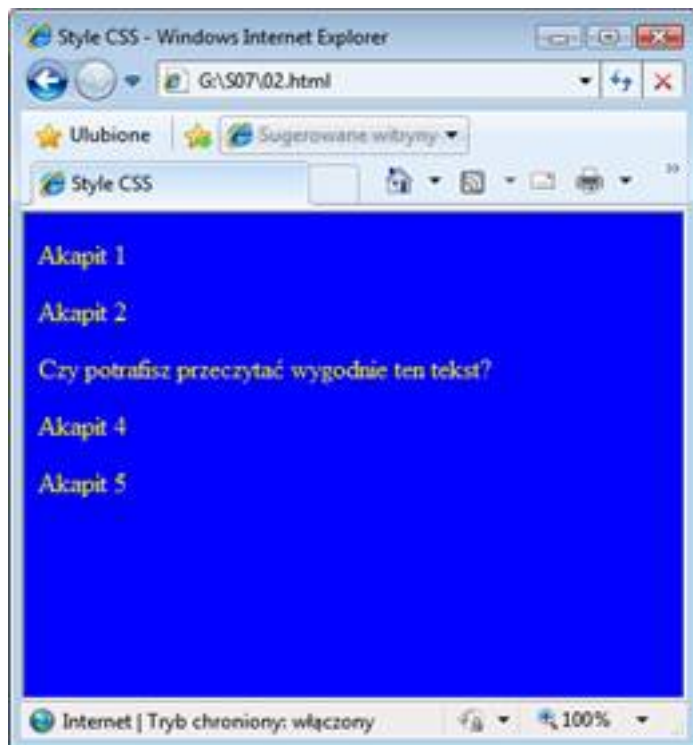
p {

  color: yellow;

}

</style>
```

a wszystkie akapity tekstu zmieniają swój kolor na żółty — dokładnie tak, jakbyś znacznik otwierający każdego z nich zapisał nie jako <p>, a <p style="color: yellow;">. Oczywiście osadzenie stylu elementu <p> na początku kodu strony umożliwia Ci teraz natychmiastową zmianę koloru tekstu na biały, turkusowy lub dowolny inny — możesz eksperymentować, za każdym razem zmieniając definicję koloru w jednym tylko miejscu (rysunek 7.2)!



Rysunek 7.2. Użycie definicji stylu osadzonego umożliwiło zmianę wyglądu wszystkich akapitów tekstu naraz

## Kaskadowe arkusze stylu

Dodał Administrator  
piątek, 23 lipiec 2010 09:20

---

Skoro określiliśmy kolor tekstu, warto rozszerzyć definicję stylu również na element <body>, dodatkowo zmieniając jeszcze dalej wygląd akapitów (rysunek 7.3). Oto przykład:

```
<?xml version="1.0" encoding="utf-8"?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pl" lang="pl">

<head>

<title>Style CSS</title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<style type="text/css"><!--

body {

background-color: blue;

}

p {

color: yellow;

font-family: Arial,sans-serif;

font-size: 11pt;

}

--></style>

</head>

<body>

<p>Akapit 1</p>

<p>Akapit 2</p>

<p>Czy potrafisz przeczytać wygodnie ten tekst?</p>

<p>Akapit 4</p>

<p>Akapit 5</p>

</body>

</html>
```



Rysunek 7.3. Strona wygląda podobnie, jednak tym razem definicja wyglądu poszczególnych elementów została umieszczona

Pamiętaj: zawsze powinieneś wykorzystywać osadzone definicje stylu w jak największym stopniu. Dzięki nim definicja wyglądu strony WWW zapisana jest w jednym miejscu i może podlegać prostym zmianom, a właściwy kod XHTML określa strukturę dokumentu, a nie jego wygląd. Ze stylu wpisanego korzystaj tylko tam, gdzie koniecznie zachodzi potrzeba zmiany wyglądu jednego tylko elementu strony. **Definiowanie klas wyglądu**

### elementu języka XHTML

Choć zmiana wyglądu wszystkich elementów danego typu — na przykład wszystkich akapitów tekstu na stronie — jest bardzo przydatna, częściej pojawia się potrzeba wpływania na wygląd wybranych tylko elementów strony. Na szczęście mechanizm kaskadowych arkuszy stylu pozwala zdefiniować tak zwane klasy stylu, które domyślnie nie są używane, a których zastosowanie wymaga dodania tylko kilku znaków do kodu elementu, którego dana klasa stylu ma dotyczyć. Nie traci się w ten sposób zalet wynikających z centralizacji definicji stylu, a zyskuje swobodę wyboru.

Załóżmy, że na swojej stronie — poświęconej, powiedzmy, motoryzacji — chcesz cytować fragmenty obcych tekstów, każdy z fragmentów wyróżniając kursywą i szerszymi marginesami. Dodatkowo inną czcionką warto wyróżnić nazwy tytułów czasopism oraz marek samochodów (rysunek 7.4). Aby za każdym razem nie wprowadzać wpisanej definicji stylu odróżniającej elementy `<p>` stanowiące cytaty od elementów `<p>` tworzących Twój własny tekst, należy dokonać podziału: normalne elementy `<p>` będą tworzyły akapity własnego tekstu, a elementy `<p>` klasy cytaty — cytaty. Z kolei elementy `<span>` klasy czasopismo będą służyły do zapisywania tytułów, a klasy marka — marek samochodów. Oto przykład:

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pl" lang="pl">
```

```
<head>
```

```
<title>Style CSS</title>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

```
<style type="text/css">
```

```
p {
```

```
font-size: 11pt;
```

```
line-height: 135%;
```

```
}
```

```
p.cytat {
```

```
font-style: italic;
```



## Kaskadowe arkusze stylu

Dodał Administrator  
piątek, 23 lipiec 2010 09:20

---

```
margin: 0 20pt 0 20pt;

}

span.tytul {

font-weight: bold;

font-style: normal;

color: brown;

}

span.marka {

color: blue;

font-weight: bold;

}

</style>

</head>

<body>

<p>Co się dzieje ostatnimi czasy na rynku motoryzacyjnym?

Miesięcznik <span class="tytul">Twoja Bryka</span> zadał

to pytanie swojemu korespondentowi, Arnoldowi Mogunckiemu,

który napisał:</p>

<p class="cytat">Najnowszy model <span class="marka">Poloneza</span>

może sporo namieszać na rynku motoryzacyjnym. Jego silnik

o mocy 82 KM pozwala właścicielowi pozostawić w tyle

wszystkie „maluchy”. Nawet <span class="marka">FIAT</span> 125p

z silnikiem 1300 cm<sup>3</sup> dał się pokonać!</p>

<p>Czy <span class="marka">Polonezy</span> pozostaną

niedoścignione? W <span class="tytul">Wysokich Szprychach</span>

czytamy:</p>

<p class="cytat">Chodzą słuchy, że nowy, wzmocniony silnik

1500 cm<sup>3</sup> montowany w najnowszych
```

## Kaskadowe arkusze stylu

Dodał Administrator  
piątek, 23 lipiec 2010 09:20

Polonezach ma zostać przeniesiony

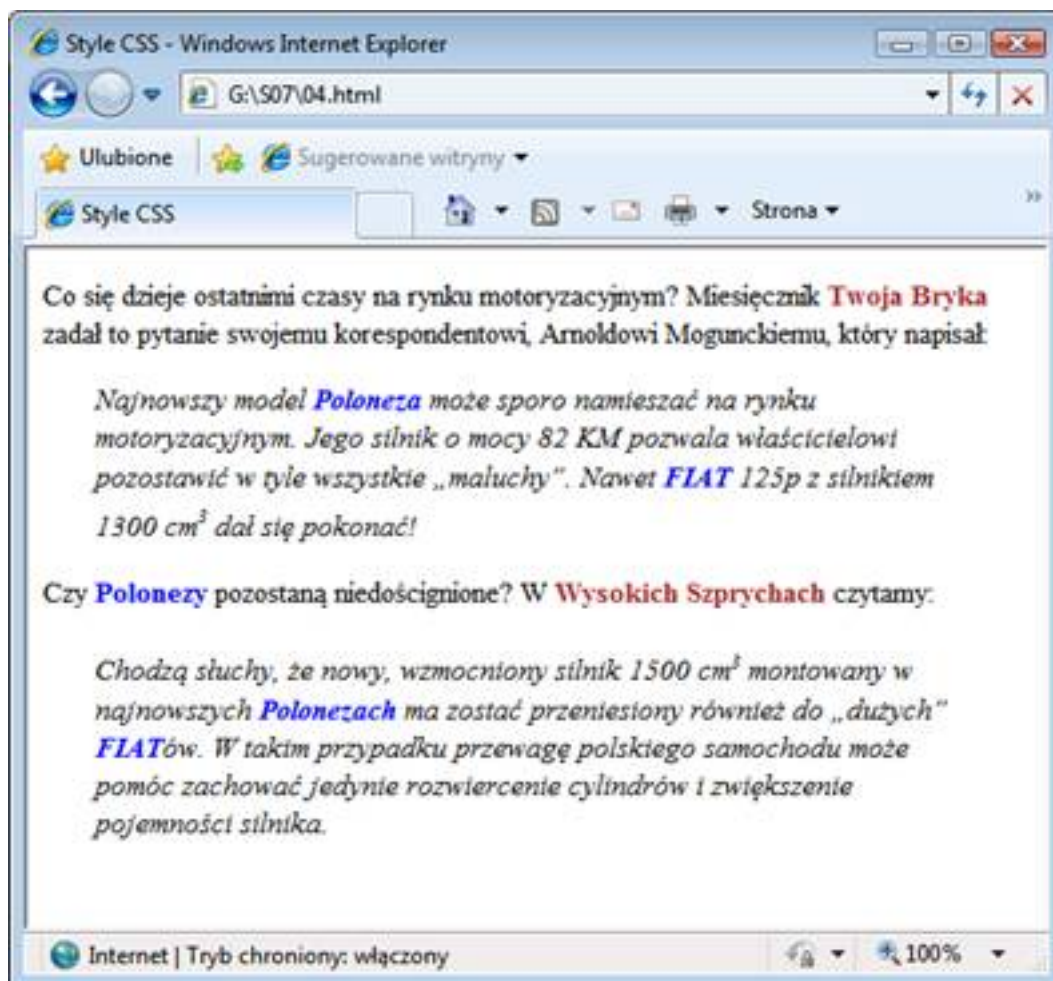
również do „dużych” FIATów.

W takim przypadku przewagę polskiego samochodu może pomóc zachować

jedynie rozwiercenie cylindrów i zwiększenie pojemności silnika.

</body>

</html>



Rysunek 7.4. Strona WWW korzystająca z możliwości tworzenia klas poszczególnych elementów, różniących się między sobą

Jak widać, definicja klasy stylu elementu wyróżnia się dodaniem do nazwy elementu dowolnie wybranej nazwy klasy, oddzielonej kropką. Na przykład aby założyć klasę cytata elementu `<p>` należy utworzyć wewnątrz elementu `<style>` następującą definicję:

```
p.cytat {
```

```
...
```

## Kaskadowe arkusze stylu

Dodał Administrator  
piątek, 23 lipiec 2010 09:20

---

}

Nazwa klasy stylu powinna składać się ze znaków alfabetu łacińskiego oraz cyfr i nie powinna zaczynać się od cyfry. Stosowanie polskich znaków diakrytycznych jest zabronione.

Aby skorzystać z utworzonej klasy, wystarczy dodać do elementu atrybut class i podać w cudzysłowie nazwę założonej wcześniej klasy, na przykład:

```
<p class="cytat">...</p>
```

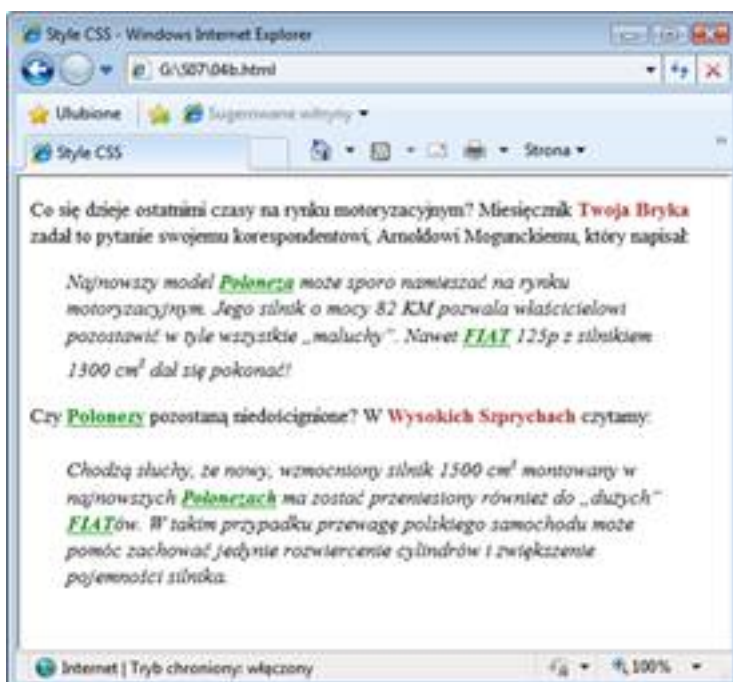
Wszystkie klasy dziedziczą atrybuty stylu elementu podstawowego. Widać to w powyższym przykładzie: mimo iż dla klasy cytat elementu <p> nie zadeklarowaliśmy rozmiaru czcionki, fakt, że dla samego elementu <p> została podana deklaracja font-size: 11pt;, wystarczył, by również cytaty były zapisane czcionką o rozmiarze 11 punktów.

Generalnie podstawowej klasie elementu (czyli samemu elementowi) powinniśmy nadawać wszystkie cechy wspólne klas pochodnych, by nigdy nie dopuszczać do duplikowania wpisów. Klasy powinny zawierać tylko deklaracje odróżniające je od klasy podstawowej.

Jakie są zalety tworzenia klas? Pomyśl sobie, że nie użyłeś ich, używając w powyższym przykładzie za każdym razem stylu wpisanego przy modyfikowaniu wyglądu tytułów i marek. Gdybyś teraz chciał zmienić wygląd marek samochodów, by były podkreślone i wypisane zieloną czcionką, czekałoby Cię poprawianie deklaracji w kilkunastu, kilkudziesięciu lub kilkuset miejscach! Używając klas, wystarczy wprowadzić poprawkę w deklaracji klasy stylu:

```
span.marka {  
  
color: green;  
  
font-weight: bold;  
  
text-decoration: underline;  
  
}
```

a wszystkie miejsca, w których otoczyłeś tekst elementami <span> klasy marka, natychmiast po odświeżeniu strony zmienią swój wygląd (rysunek 7.5).



Rysunek 7.5. Wystarczy drobna zmiana definicji stylu wybranej klasy elementów języka, by wygląd całej strony zmienił się di

Zauważ też, że tworzenie klas stylów stanowi kolejny krok w kierunku rozdzielenia logicznej struktury dokumentu (strony WWW) od jego wyglądu. Nawet gdybyś kiedyś stwierdził, że nazwy marek samochodów nie wymagają specjalnego podświetlenia, pozostaną wyróżnione w kodzie i będziesz miał potencjalną możliwość ich wyróżnienia. Poza tym tworząc klasy elementów, unikasz umieszczania w kodzie strony (i w zapisywanym tekście) jakichkolwiek informacji o wyglądzie.

Przy okazji tego przykładu poznałeś kilka nowych atrybutów stylu elementów. Pierwszym z nich jest `line-height`, w którym podaje się odstęp międzywierszowy tekstu. Możesz podawać go w dowolnych jednostkach, jednak najczęściej używa się skali względnej (%), zmieniającej odstępów równomiernie w zależności od rozmiaru czcionki. Pamiętaj jednak, że mała czcionka wymaga szerokich odstępów między wierszami, podczas gdy duża pozostaje czytelna, nawet gdy kolejne wiersze prawie stykają się ze sobą.

Drugi z atrybutów to `font-style`. Wartość `normal` oznacza zwykły tekst, a `italic` — kursywę. Atrybut ten zastępuje element `<i>`, pozwalając Ci nie używać go w ogóle, jeżeli cały tekst zapisany w konkretnych elementach ma być wyróżniony kursywą.

Trzeci to `font-weight`, który może przybierać wiele wartości, jednak dwie najczęściej używane (i często jedyne dające widoczne na ekranie efekty) to `normal` i `bold`. Wartość `normal` oznacza tekst normalnie zapisany, zaś `bold` — pogrubiony. Jak poprzednio, atrybut ten zastępuje element `<b>`, umożliwiając automatyczne pogrubianie całych fragmentów tekstu bez używania znaczników `<b>` i `</b>`.

Ostatnim jest atrybut `text-decoration`. Może on przyjmować następujące wartości:

- `none` — brak dekoracji tekstu,
- `underline` — podkreślenie,
- `overline` — nakreślenie,
- `line-through` — przekreślenie.

Tego atrybutu używaj bardzo rozważnie. Podkreślony tekst może kojarzyć się z odnośnikami, zaś przekreślenia czynią go nieczytelnym. Najlepiej w ogóle zrezygnować z tego typu dekoracji, rezerwując ją dla odnośników. Podany powyżej przykład zamiany stylu tekstu reprezentującego markę samochodu jest właśnie przykładem, jak nie używać podkreślenia. **Definiowanie niezależnych klas stylu**

Klasy stylu związane z elementami języka XHTML pozwalają określić wygląd podzbioru elementów danego typu. Czasem jednak przydaje się możliwość określenia wyglądu szerszej grupy elementów, niekoniecznie jednolitych pod względem typu. W takich przypadkach mogą się okazać pomocne niezależne klasy stylu.

Na przykład poniższy kod strony WWW definiuje niezależną klasę o nazwie `niebieski`, nadającą dowolnemu elementowi `niebieski` kolor (rysunek 7.6). Zauważ, że w deklaracji stylu nie pojawia się nazwa elementu języka XHTML, a od razu tylko kropka i nazwa klasy:

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pl" lang="pl">
```

## Kaskadowe arkusze stylu

Dodał Administrator  
piątek, 23 lipiec 2010 09:20

---

```
<head>

<title>Style CSS</title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<style type="text/css"><!--
th {
background-color: gray;
color: white;
}
.niebieski {
color: blue;
}
--></style>

</head>

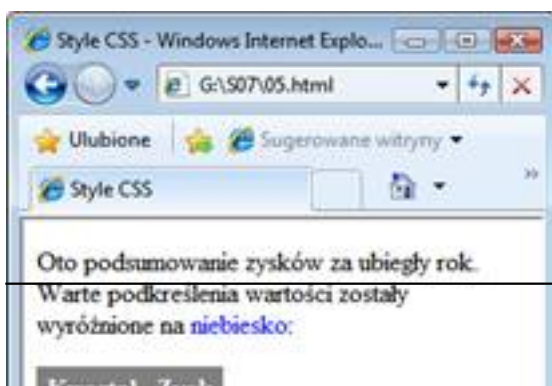
<body>

<p>Oto podsumowanie zysków za&nbsp;ubiegły&nbsp;rok. Warte podkreślenia
wartości zostały wyróżnione na&nbsp;<span class="niebieski">niebiesko</span>.</p>

<table cellpadding="4" cellspacing="0">
<tr><th>Kwartał</th><th>Zysk</th></tr>
<tr><th>pierwszy</th><td>140&nbsp;zł</td></tr>
<tr><th>drugi</th><td>150&nbsp;zł</td></tr>
<tr><th>trzeci</th><td class="niebieski">200&nbsp;zł</td></tr>
<tr><th>czwarty</th><td>190&nbsp;zł</td></tr>
</table>

</body>

</html>
```



Rysunek 7.6. Klasa stylu niebieski narzuca niebieski kolor tekstowi dowolnego elementu języka XHTML

Mimo wygody unikaj nadużywania uniwersalnych, niezwiązanych z żadnym konkretnym elementem języka XHTML klas stylów. Tak naprawdę przydają się one niezwykle rzadko, a w większości przypadków wygodniejsze jest stworzenie kilku specjalizowanych klas stylów — po jednej dla każdego typu elementu — niż dostosowywanie różnych elementów do jednej, niezależnej klasy. **Zmiana wyglądu odnośników**

Prawdopodobnie nie miałbyś problemów ze zmianą wyglądu elementu `<a>`, odpowiadającego za odnośniki umieszczane na stronie. Problem może pojawić się dopiero, gdy uświadomisz sobie, że odnośniki tak naprawdę występują w trzech odmianach:

- odnośniki normalne, prowadzące do stron, których jeszcze nie odwiedziłeś,
- odnośniki wskazane (podświetlone) myszą,
- odnośniki znane, prowadzące do odwiedzonych już kiedyś stron.

Ta różnorodność nie zwiększa na szczęście poziomu trudności zadania. Zmiana wyglądu odnośników wymaga tylko troszkę więcej kodu niż zmiana wyglądu innych elementów języka XHTML; w zamian otrzymujesz niezwykle ciekawe możliwości, na przykład dynamiczne podświetlanie odnośników wskazywanych myszą.

Odmiany elementu `<a>` przypominają nieco klasy stylu, różnią się jednak dwiema podstawowymi cechami:

- nazwy odmian są z góry ściśle określone,
- nazwy te zapisuje się nie po kropce, a po dwukropku.

Odmiana podstawowa nie nosi żadnej nazwy. Odpowiada jej po prostu element `<a>`:

```
a {  
...  
}
```

Za odnośniki wskazane myszą odpowiada już jednak odmiana `hover`, zapisywana w następujący sposób:

```
a: hover {  
...  
}
```

Z kolei odnośniki odwiedzone symbolizuje odmiana `visited`:

```
a: visited {  
...  
}
```

Poniższy przykład prezentuje wszystkie trzy odmiany odnośników (rysunek 7.7). Zwykłym odnośnikom usunąłem podkreślenie, nadając im kolor zielony. Odnośniki już odwiedzone rysowane są na szaro. Odnośniki wskazywane

## Kaskadowe arkusze stylu

Dodał Administrator  
piątek, 23 lipiec 2010 09:20

---

myszą otrzymują z powrotem podkreślenie i stają się czerwone. Oto kod:

```
<?xml version="1.0" encoding="utf-8"?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pl" lang="pl">

<head>

<title>Style CSS</title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<style type="text/css"><!--

a {

    color: green;

    text-decoration: none;

}

a:visited {

    color: gray;

}

a:hover {

    color: red;

    text-decoration: underline;

}

--></style>

</head>

<body>

<p><a href="http://www.nieodwiedzona.pl/">Odkaz do nieodwiedzanej strony</a></p>

<p><a href="http://www.grush.one.pl/">Odkaz do odwiedzanej strony</a></p>

<p><a href="http://www.onet.pl/">Odkaz wskazany myszą</a></p>

</body>

</html>
```



Rysunek 7.7. Przykład definiowania wyglądu wszystkich trzech odmian odnośników

Na koniec dwie uwagi. Przede wszystkim bądź ostrożny z usuwaniem podkreślenia z odnośników. Użytkownicy są przyzwyczajeni, że odnośniki mają podkreślenie, i usunięcie go może niepotrzebnie utrudnić nawigację. Przyjęła się zasada, że podkreślenia mogą być pozbawione odnośniki znajdujące się w menu nawigacyjnym lub tylko pomocnicze, nieistotne dla nawigacji i możliwe do przeoczenia. Te najważniejsze, umieszczane w tekście, muszą być podkreślone.

Drugim problemem jest kolorystyka odnośników. Dobierz ją tak, by niezależnie od koloru tła odnośnik był widoczny, i nie zmieniaj jej dowolnie w każdym miejscu. Drobne zmiany kolorystyki dopuszczalne są tylko w nagłówku lub stopce strony, gdzie ważniejszy jest wygląd odnośnika niż jego użyteczność.

## Scentralizowane sterowanie wyglądem indywidualnych elementów strony WWW

Na początku tego rozdziału wspomniałem, że nowe wersje standardu XHTML mogą zlikwidować możliwość stosowania wpisanych definicji stylu. Usunie to możliwość dowolnego zmieniania wyglądu poszczególnych akapitów tekstu, wyrazów czy komórek tabeli przez wpisanie definicji do atrybutu style elementu.

Nie oznacza to jednak, że osoby odpowiedzialne za ustalanie standardu chcą odebrać twórcom serwisów WWW możliwość sterowania wyglądem pojedynczych elementów składowych stron. Celem jest przesunięcie miejsca określenia wyglądu elementu z samego elementu do — osadzonego lub dołączonego — arkusza stylu. Spowoduje to jeszcze silniejsze odseparowanie struktury i treści dokumentu od opisu jego wyglądu.

Już dzisiaj możesz korzystać z takiego sposobu ustalania wyglądu elementów. Wystarczy — za pomocą atrybutu id — nadać wybranym elementom strony unikatowe identyfikatory, a następnie stworzyć definicję stylu odpowiadającą danemu wystąpieniu elementu. Oczywiście ma tutaj zastosowanie kaskadowość: wyróżniony element w pierwszej kolejności otrzyma styl wynikający z ogólnej definicji obejmującej wszystkie elementy jego typu; szczegółowa specyfikacja odpowiadająca temu jednemu elementowi musi tylko zmieniać te cechy wyglądu, którymi ma się on różnić od innych.

Na przykład by wśród poniższych czterech akapitów tekstu:

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum gravida, nisl sed blandit tincidunt, tortor turpis posuere orci, at suscipit justo orci quis arcu.</p>
```

```
<p>Proin pretium aliquam consequat. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Sed malesuada consectetur tincidunt. Nunc in posuere nulla.</p>
```

```
<p>Quisque convallis aliquet nibh ac laoreet. Sed id ante at nisl fermentum aliquam vitae quis libero. Vivamus imperdiet felis vel quam bibendum blandit. Sed placerat, dui ut auctor tincidunt, purus tortor euismod leo, lobortis hendrerit eros magna vitae mi.</p>
```

```
<p>Donec dui magna, tempus et condimentum non, eleifend ut dui. Curabitur dapibus, massa vitae dapibus auctor, arcu felis ornare nulla, id dapibus enim velit eu odio.</p>
```

wyróżnić (przez zmianę wyglądu) pierwszy i ostatni, należy nadać im unikatowe identyfikatory:

```
<p id="akapit_pierwszy">Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum gravida, nisl sed blandit tincidunt, tortor turpis posuere orci, at suscipit justo orci quis arcu.</p>
```



## Kaskadowe arkusze stylu

Dodał Administrator  
piątek, 23 lipiec 2010 09:20

---

...

<p id="akapit\_ostatni">Donec dui magna, tempus et condimentum non, eleifend ut dui. Curabitur dapibus, massa vitae dapibus auctor, arcu felis ornare nulla, id dapibus enim velit eu odio.</p>

Sama w sobie zmiana ta nie wpływa w żaden sposób na wygląd strony (co najwyżej zwiększa rozmiar pliku o kilkanaście bajtów), wprowadza jednak infrastrukturę umożliwiającą odwoływanie się do wybranych akapitów. Wykorzystamy to, dodając do osadzonej definicji stylu wpisy definiujące wygląd tych dwóch elementów (rysunek 7.8):

```
#akapit_pierwszy {
```

```
font-size: 13pt;
```

```
color: blue;
```

```
}
```

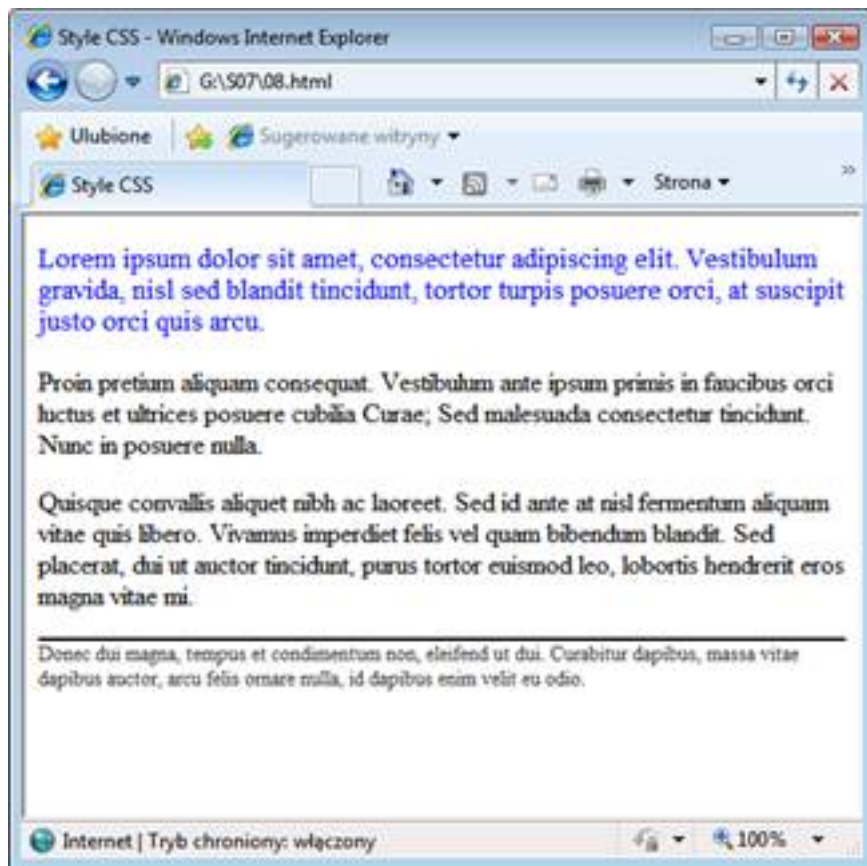
```
#akapit_ostatni {
```

```
font-size: 9pt;
```

```
border-top: 2px solid black;
```

```
color: #333;
```

```
}
```



Rysunek 7.8. Określanie wyglądu pojedynczych elementów strony, wybranych za pomocą unikatowych identyfikatorów

### Wskazówka

Pamiętaj: identyfikatory elementów mają być unikatowe. O ile zatem możliwe jest stosowanie zapisu `#pierwszy_akapit`, by określić jednoznacznie, że chodzi o akapit tekstu oznaczony identyfikatorem `pierwszy_akapit`, nie ma to sensu: i tak żaden inny element strony nie może dysponować takim identyfikatorem.

## Uzależnianie wyglądu elementów od ich pozycji w hierarchii dokumentu

Często takie same składniki logiczne strony WWW — akapity tekstu, komórki tabeli, nagłówki rozdziałów — muszą wyglądać inaczej w zależności od tego, w którym obszarze strony są używane. Oczywiście nic nie stoi na przeszkodzie, by wykorzystać mechanizm klas stylu i narzucać elementom znajdującym się w różnych miejscach strony klasę stylu. Problem pojawi się jednak w momencie przenoszenia informacji z jednego miejsca strony w inne: przeniesiona treść, zamiast przyjmować wygląd zgodny z miejscem, w którym się znalazła, będzie kopiowana wraz z wyglądem, wymagając ręcznych poprawek (zmiany nazwy klasy stylu).

Kaskadowe arkusze stylu CSS oferują jednak doskonałe rozwiązanie. Definicja stylu każdego elementu może zawierać w sobie informacje o tym, pod jakim warunkiem ma obowiązywać. Tym warunkiem najczęściej jest obecność wewnątrz jakiegoś innego elementu (tak zwanego kontenera).

Załóżmy, że chcemy przygotować stronę WWW zawierającą dwa umieszczone obok siebie łamy tekstu. Każdy z nich ma zawierać kilka akapitów tekstu, jednak wygląd tekstu w łamach ma być zróżnicowany. Zaczniemy od stworzenia dwóch kontenerów `<div>`, opatrzenia ich indywidualnymi, unikatowymi identyfikatorami (`lewy_panel` oraz `prawy_panel`) i wstawienia do ich wnętrza akapitów tekstu (rysunek 7.9):

```
<div id="lewy_panel">
```

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum gravida, nisl sed blandit tincidunt, tortor turpis posuere orci, at suscipit justo orci quis arcu.</p>
```

```
<p>Proin pretium aliquam consequat. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Sed malesuada consectetur tincidunt. Nunc in posuere nulla.</p>
```

```
</div>
```

```
<div id="prawy_panel">
```

```
<p>Quisque convallis aliquet nibh ac laoreet. Sed id ante at nisl fermentum aliquam vitae quis libero. Vivamus imperdiet felis vel quam bibendum blandit. Sed placerat, dui ut auctor tincidunt, purus tortor euismod leo, lobortis hendrerit eros magna vitae mi.</p>
```

```
<p id="akapit_ostatni">Donec dui magna, tempus et condimentum non, eleifend ut dui. Curabitur dapibus, massa vitae dapibus auctor, arcu felis ornare nulla, id dapibus enim velit eu odio.</p>
```

```
</div>
```



## Kaskadowe arkusze stylu

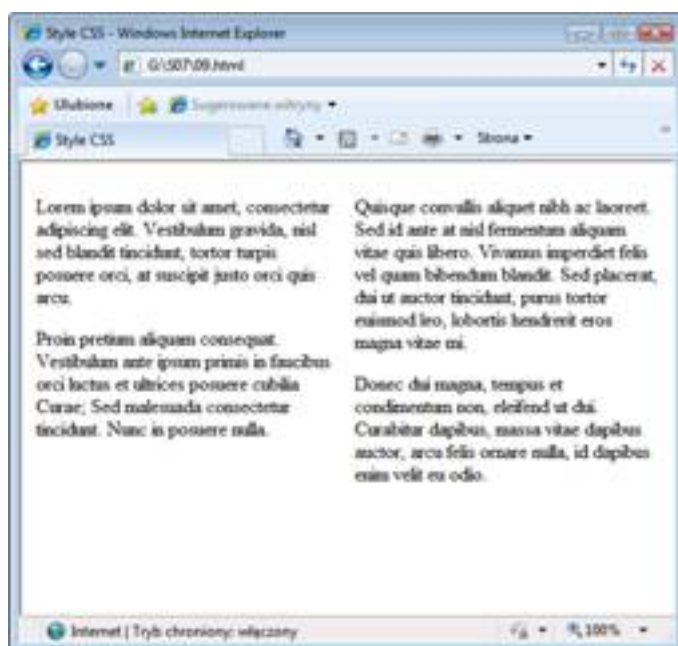
Dodał Administrator  
piątek, 23 lipiec 2010 09:20

---

Rysunek 7.9. Strona WWW na wczesnym etapie tworzenia: treść jest już na swoim miejscu, teraz wystarczy nałożyć warstwę

Jak na razie podział strony na dwa łamy w ogóle nie jest widoczny. Nic dziwnego: sam element <div> tworzy jedynie blok mogący zawierać inne elementy strony, nie różni się jednak zbyt wyglądem od zwykłego akapitu tekstu. Dodajmy zatem definicję stylu modyfikującą wygląd tych dwóch „pudełek” tak, aby leżały obok siebie (rysunek 7.10):

```
#lewy_panel {  
  
width: 48%;  
float: left;  
padding: 3pt;  
  
}  
  
#prawy_panel {  
  
width: 48%;  
float: right;  
padding: 3pt;  
  
}
```



Rysunek 7.10. Wystarczyło wprowadzenie osadzonej definicji stylu, by pudełka przeznaczone na treść zostały umieszczone

### Wskazówka

Szerokość obydwu kontenerów została ustalona na 48% szerokości okna przeglądarki, gdyż (niestety!) opis szerokości dotyczy treści kontenera, a nie jego zewnętrznych wymiarów. Ponieważ każde z „pudełek” musi

## Kaskadowe arkusze stylu

Dodał Administrator  
piątek, 23 lipiec 2010 09:20

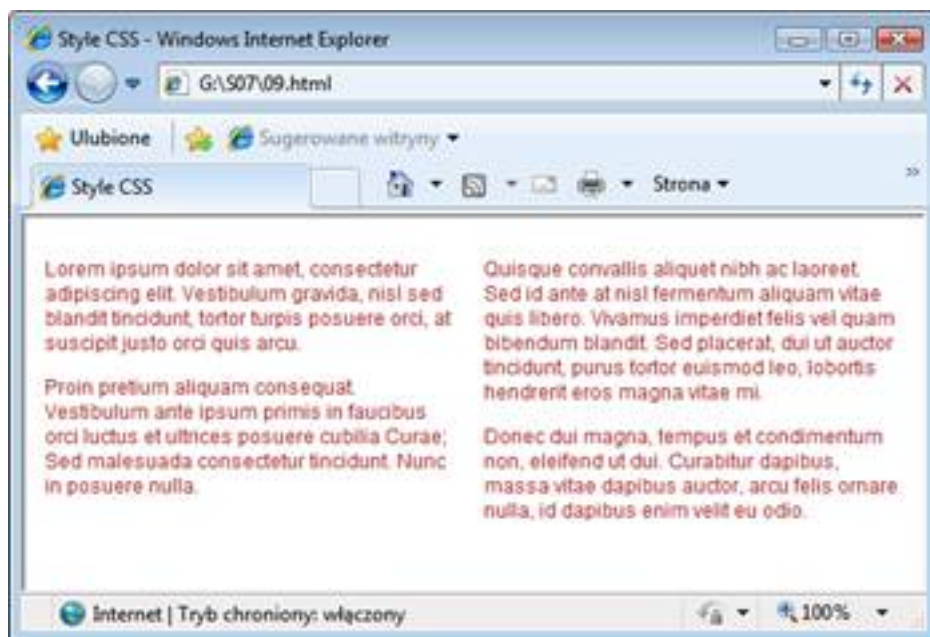
dysponować marginesem wewnętrznym zapobiegającym stykaniu się treści (padding), realna szerokość jest większa niż narzucone 48% (a nie może przekroczyć 50%, gdyż wtedy kontenery nie mogłyby znajdować się obok siebie).

### Wskazówka

Do określenia wyglądu kontenerów został zastosowany mechanizm indywidualnego narzucania stylu. Jak widać, pozwala on uprościć kod strony: nie trzeba zaciemniać go skomplikowanymi formułkami CSS, wystarczy wstawić unikatowy identyfikator elementu.

Zwykła definicja stylu elementu <p> spowoduje — oczywiście — zmianę wyglądu akapitów tekstu w obydwu kontenerach jednocześnie (rysunek 7.11):

```
p {  
  
font-family: sans-serif;  
  
font-size: 9pt;  
  
color: brown;  
  
}
```



Rysunek 7.11. Zmiana wyglądu tekstu niezależnie od jego położenia jest prosta: wystarczy zmodyfikować wygląd wszystkich

Aby zmiana dotyczyła tylko akapitów znajdujących się w jednym lub drugim kontenerze, musimy uzupełnić tę definicję informacją o elemencie, w którym powinien znajdować się akapit, by przyjął wygląd opisany daną formułką. Robi się to, umieszczając identyfikator elementu nadrzędnego przed nazwą elementu, którego wygląd zmieniamy, i rozdzielając obydwie nazwy znakiem odstępu. Taki zapis nosi nazwę selektora:

```
p {  
  
font-family: sans-serif;
```

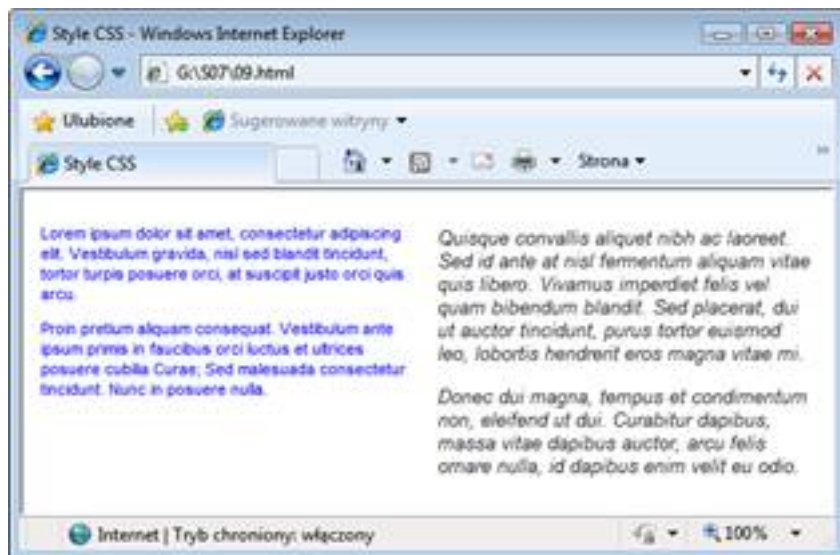
## Kaskadowe arkusze stylu

Dodał Administrator  
piątek, 23 lipiec 2010 09:20

---

```
font-size: 9pt;  
  
color: brown;  
  
}  
  
#lewy_panel p {  
  
font-size: 8pt;  
  
color: blue;  
  
}  
  
#prawy_panel p {  
  
font-size: 10pt;  
  
font-style: italic;  
  
color: #222;  
  
}
```

Pierwsza z powyższych trzech definicji określa cechy wspólne dla wszystkich akapitów tekstu. Druga definiuje wygląd zawartości wszystkich elementów <p> znajdujących się wewnątrz dowolnego elementu o unikatowym identyfikatorze #lewy\_panel. Trzecia — analogicznie — określa wygląd akapitów tekstu znajdujących się we wnętrzu elementu o unikatowym identyfikatorze #prawy\_panel (rysunek 7.12).



Rysunek 7.12. Dzięki selektorom wybierającym definicję stylu w zależności od tego, w którym z dwóch pudełek znajduje się

Widać wyraźnie, że od momentu nadania kontenerom unikatowych identyfikatorów kod XHTML strony nie zmienił się ani trochę. Wszystkie modyfikacje wyglądu strony były wprowadzane wyłącznie za pomocą mniej lub bardziej skomplikowanych formuł CSS. Gdy definicję stylu umieści się w odrębnym, dołączonym za pomocą elementu

<link> pliku i zastosuje w wielu stronach składowych serwisu, uzyska się niezmiernie oszczędny, zwięzły, a jednocześnie automatycznie ustalający wygląd tekstu kod.

### Wskazówka

Równie dobrze można uzależniać wygląd elementów strony od ich istnienia we wnętrzu elementu innego typu, a nie elementu o konkretnym identyfikatorze. Na przykład selektor `p span` spowoduje ustalenie wyglądu wszystkich elementów `<span>` znajdujących się wewnątrz dowolnych akapitów tekstu `<p>`. **Tworzenie**

## zewnętrznego arkusza stylu

Osadzone kaskadowe arkusze stylu, które przed chwilą nauczyłeś się wykorzystywać, znakomicie upraszczają ustalanie wyglądu strony. Jeśli obecny wygląd tworzonej przez Ciebie strony nie satysfakcjonuje Cię, wystarczy drobna zmiana definicji stylu znajdujących się wewnątrz obszaru elementu `<style>`, by wszystkie elementy przybrały nowy wygląd.

Sytuacja komplikuje się w momencie rozpoczynania prac nad serwisem internetowym składającym się z dziesiątków lub setek stron WWW. Nie dość, że osadzone arkusze stylu mogą stać się Twoją zimą, gdy przyjdzie Ci wprowadzać identyczne zmiany w szeregu plików XHTML, to jeszcze znajdująca się w każdym pliku kopia arkusza stylu będzie musiała być przesyłana przez Sieć wraz z każdą stroną oglądaną przez czytelnika Twojego serwisu.

Panaceum na te problemy stanowi mechanizm zewnętrznego arkusza stylu. By go stworzyć, wystarczy skopiować zawartość elementu `<style>` do odrębnego pliku i poinformować o tym przeglądarkę za pomocą odpowiedniego elementu języka XHTML. Od tego momentu wydzielonego w ten sposób arkusza stylu możesz używać z poziomu każdej Twojej strony WWW, a zmiany wprowadzane w arkuszu automatycznie będą obowiązywać na każdej stronie. **Zewnętrzny arkusz stylu**

Wróćmy na chwilę do przykładu prezentującego zastosowanie klas stylu powiązanych z elementami. Przygotowaliśmy tam cztery osobne klasy stylu: dwie dla elementu `<p>` i dwie dla elementu `<span>`. Aż prosi się, aby wydzielić je w osobnym pliku i skorzystać z nich na kilku innych stronach WWW. Aby to uczynić, stwórz nowy, pusty na razie zbiór i umieść w nim całą deklarację stylu, ignorując oczywiście znaczniki XHTML:

```
p {  
  
font-size: 11pt;  
  
line-height: 135%;  
  
}  
  
p.cytat {  
  
font-style: italic;  
  
margin: 0 20pt 0 20pt;  
  
}  
  
span.tytul {  
  
font-weight: bold;  
  
font-style: normal;
```



## Kaskadowe arkusze stylu

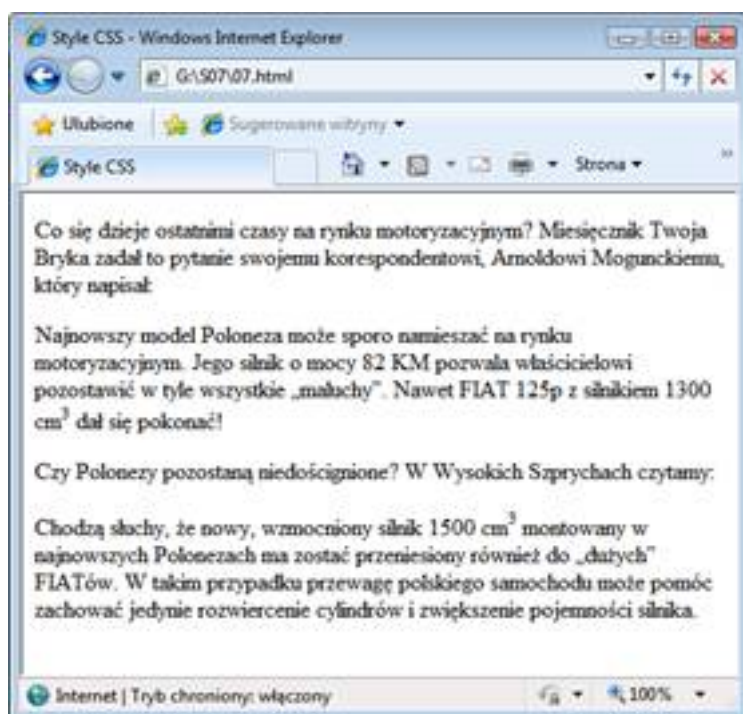
Dodał Administrator  
piątek, 23 lipiec 2010 09:20

---

```
color: brown;  
  
}  
  
span.marka {  
  
color: blue;  
  
font-weight: bold;  
  
}
```

Zbiór ten zapisz pod dowolną nazwą, najlepiej nadając mu rozszerzenie nazwy .css. Przykładowo może być to styl.css.

Ponieważ zewnętrzna deklaracja stylu jest już gotowa, w pliku źródłowym usuń cały element <style>. Jeżeli teraz spróbujesz wyświetlić zawartość pliku XHTML, zobaczysz, że został on pozbawiony formatowania (rysunek 7.13) — nic dziwnego, skoro usunęliśmy wszystkie deklaracje stylu elementów, a nie dodaliśmy jeszcze informacji o tym, że znajdują się one w pliku styl.css.



Rysunek 7.13. Tekst pozbawiony warstwy stylu wciąż jest czytelny — to powinien być jeden z podstawowych celów każdego

Trzeba naprawić ten błąd. Aby połączyć plik XHTML z plikiem zawierającym deklaracje stylu elementów, wystarczy dopisać wewnątrz elementu <head> następujący wiersz:

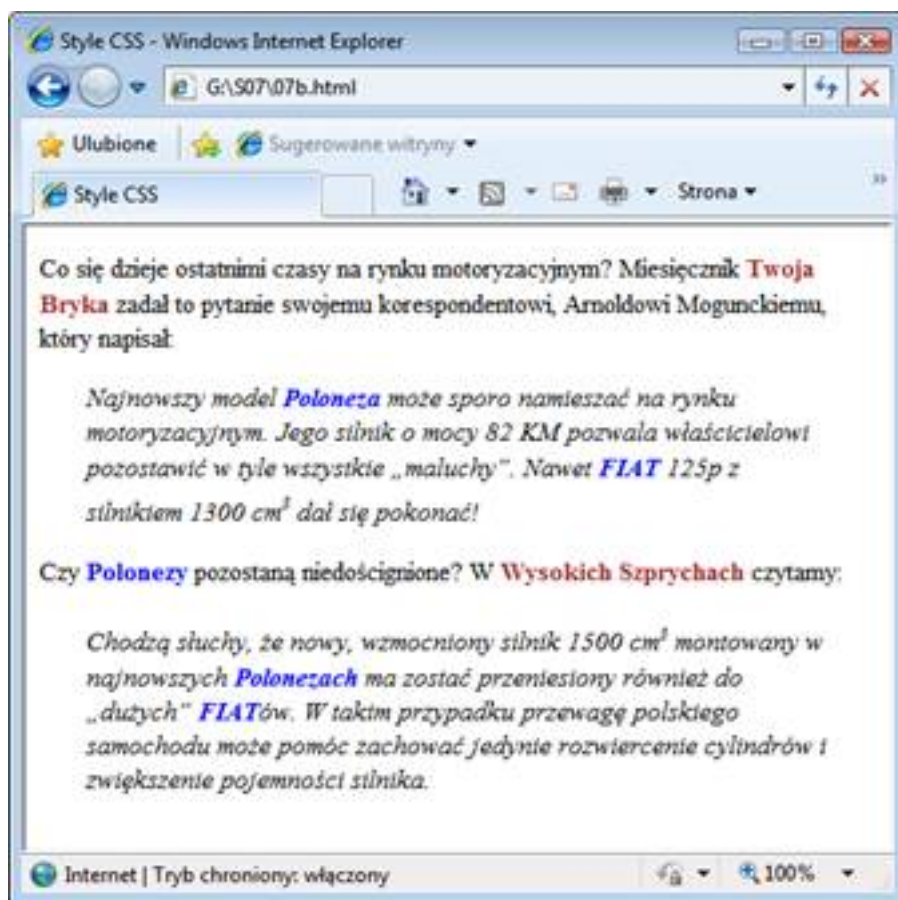
```
<link rel="stylesheet" href="styl.css" type="text/css" />
```

Atrybut href zawiera nazwę pliku (lub ścieżkę dostępu do niego), z którego odczytane zostaną deklaracje stylu. Oczywiście do tego samego pliku deklaracji stylu może odwoływać się wiele plików XHTML i o to właśnie nam chodziło, gdyż dzięki takiemu prostemu zabiegowi wygląd wszystkich stron będzie jednolity, a ewentualne zmiany

## Kaskadowe arkusze stylu

Dodał Administrator  
piątek, 23 lipiec 2010 09:20

deklaracji będą natychmiast obowiązywały na wszystkich stronach WWW (rysunek 7.14).



Rysunek 7.14. Po dopisaniu znacznika tworzącego odniesienie do zapisanego w odrębnym pliku arkusza stylu strona WWW

Oto pełny kod strony:

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pl" lang="pl">
```

```
<head>
```

```
<title>Style CSS</title>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

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

```
</head>
```

```
<body>
```



## Kaskadowe arkusze stylu

Dodał Administrator  
piątek, 23 lipiec 2010 09:20

---

<p>Co&nbsp;się dzieje ostatnimi czasy na&nbsp;ryнку motoryzacyjnym?

Miesięcznik <span class="tytuł">Twoja Bryka</span> zadał

to&nbsp;pytanie swojemu korespondentowi, Arnoldowi Mogunckiemu,

który napisał:</p>

<p class="cytat">Najnowszy model <span class="marka">Poloneza</span>

może sporo namieszać na&nbsp;ryнку motoryzacyjnym. Jego&nbsp;silnik

o&nbsp;mocy 82&nbsp;KM pozwala właścicielowi pozostawić w&nbsp;tyle

wszystkie &bdquo;maluchy&rdquo;. Nawet&nbsp;<span class="marka">FIAT</span>&nbsp;125p

z&nbsp;silnikiem 1300&nbsp;cm<sup>3</sup> dał&nbsp;się pokonać!</p>

<p>Czy&nbsp;<span class="marka">Polonezy</span> pozostaną

niedoścignione? W&nbsp;<span class="tytuł">Wysokich Szprychach</span>

czytamy:</p>

<p class="cytat">Chodzą słuchy, że&nbsp;nowy, wzmocniony silnik

1500&nbsp;cm<sup>3</sup> montowany w&nbsp;najnowszych

<span class="marka">Polonezach</span> ma&nbsp;zostać przeniesiony

również do&nbsp;&bdquo;dużych&rdquo; <span class="marka">FIAT</span>ów.

W&nbsp;takim przypadku przewagę polskiego samochodu może pomóc zachować

jedynie rozwiercenie cylindrów i&nbsp;zwiększenie pojemności silnika.</p>

</body>

</html> **Łączenie arkuszy stylu**

Tworząc zewnętrzny arkusz stylu, nie jesteś do niego uwiązany. Każdy plik XHTML może zawierać kilka elementów <link> odwołujących się do wielu osobnych plików zawierających deklaracje stylu elementów. Dzięki temu możesz przygotować jeden krótki plik zawierający deklaracje używane na wszystkich Twoich stronach, a osobno zestaw jeszcze krótszych plików używanych tylko na co niektórych z nich. W efekcie użytkownika będzie czekało pobieranie z Sieci tylko tych naprawdę niezbędnych deklaracji.

Nikt Ci też nie broni, by w pliku XHTML, w którym używasz (za pomocą elementu <link>) zewnętrznego arkusza stylu, umieścić element <style> i wpisać dowolne deklaracje osadzone. Możesz nawet zmodyfikować niektóre deklaracje poczynione w zewnętrznym arkuszu stylu. Zawsze ostatnia deklaracja liczy się najbardziej.

## Podsumowanie

Kaskadowe arkusze stylu nie tylko unifikują sposób definiowania wyglądu elementów języka XHTML, ale też

## Kaskadowe arkusze stylu

Dodał Administrator  
piątek, 23 lipiec 2010 09:20

---

uwalniają twórcę strony WWW od konieczności modyfikowania znacznych partii kodu przy okazji każdej zmiany pierwotnego planu dotyczącego wyglądu strony.

Projektując strony WWW, staraj się trzymać następujących zasad:

- jeśli projektujesz więcej niż jedną stronę WWW (a tak jest najczęściej — pojedyncza strona WWW zazwyczaj jest mało przydatna), umieść wszystkie definicje stylu w zewnętrznym arkuszu stylu,
- definiując styl wykorzystywany tylko na jednej stronie, skorzystaj ze stylu osadzonego, by jak najbardziej uprościć sobie ewentualne późniejsze poprawki,
- z wpisanej definicji stylu korzystaj tylko w ostateczności i tylko w przypadku określania wyglądu pojedynczych elementów strony — na przykład jednego akapitu tekstu lub jednej komórki tabeli,
- pamiętaj, że maksymalne scentralizowanie definicji stylu (najlepiej w zewnętrznym arkuszu stylu) pozwala przyspieszyć proces pobierania strony z Sieci, a więc i zwiększyć zadowolenie jej czytelników,
- 

korzystaj, gdzie się da, z selektorów umożliwiających określenie wyglądu elementów w zależności od tego, wewnątrz którego elementu strony się znajdują; w ten sposób kod XHTML strony będzie maksymalnie zwięzły, a przeglądarka sama będzie formatowała stronę wyłącznie na podstawie informacji o tym, gdzie znajduje się dany akapit tekstu.