

# ITA-103 Aplikacje Internetowe

Piotr Bubacz

## Moduł 2

Wersja 1

# Kaskadowe Arkusze Stylów – CSS

## Spis treści

Kaskadowe Arkusze Stylów – CSS.....	1
Informacje o module.....	2
Przygotowanie teoretyczne.....	3
Przykładowy problem .....	3
Podstawy teoretyczne.....	3
Przykładowe rozwiązanie.....	9
Porady praktyczne .....	11
Uwagi dla studenta .....	12
Dodatkowe źródła informacji.....	12
Laboratorium podstawowe.....	14
Problem 1 (czas realizacji 20 min) .....	14
Problem 2 (czas realizacji 25 min) .....	17
Laboratorium rozszerzone .....	21
Zadanie 1 (czas realizacji 45 min).....	21
Zadanie 2 (czas realizacji 45 min).....	21
Zadanie 3 (czas realizacji 90 min).....	21
Zadanie 4 (czas realizacji 90 min).....	21

## Informacje o module

### Opis modułu

W tym module znajdziesz informacje dotyczące kaskadowych arkuszy stylów i zalet wynikających z ich stosowania. Zobaczysz, w jaki sposób używać selektorów i jak utworzyć układ strony wykorzystując pozycjonowanie CSS. Dodatkowo nauczysz się, jak zmienić istniejący dokument HTML wykorzystujący formatowanie przy pomocy tabel, aby był zgodny ze standardem XHTML i przyjazny dla CSS.

### Cel modułu

Celem modułu jest przedstawienie możliwości wykorzystania kaskadowych arkuszy stylów do określenia wyglądu dokumentu XHTML, jak również przedstawienie zasad pracy z arkuszami stylów w Visual Studio.

### Uzyskane kompetencje

Po zrealizowaniu modułu będziesz:

- wiedział czym są i gdzie warto stosować kaskadowe arkusze stylów
- potrafił tworzyć nowe strony przyjazne dla kaskadowych arkuszy stylów
- rozumiał potrzebę przystosowania istniejących dokumentów HTML do wymagań standardu XHTML

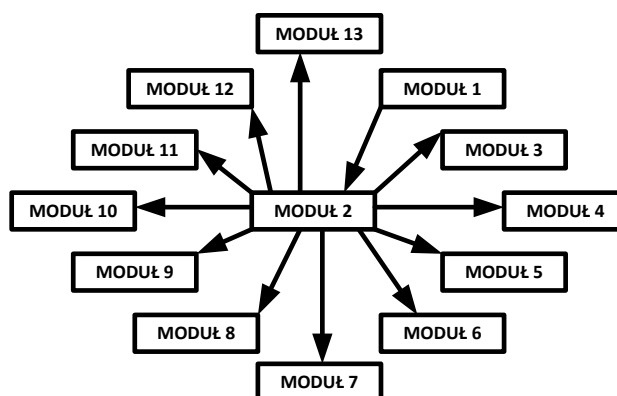
### Wymagania wstępne

Przed przystąpieniem do pracy z tym modulem powinieneś:

- znać podstawy języka XHTML (patrz: moduł „Wprowadzenie do XHTML”)
- rozumieć zasady grupowania elementów przy pomocy znaczników `<div>` i `<span>`
- znać zasady pracy w środowisku Visual Studio, w szczególności tworzenia stron internetowych zgodnych z XHTML

### Mapa zależności modułu

Zgodnie z mapą zależności przedstawioną na Rys. 1, przed przystąpieniem do realizacji tego modułu należy zapoznać się z materiałem zawartym w module „Podstawy HTML”.



Rys. 1 Mapa zależności modułu

## Przygotowanie teoretyczne

### Przykładowy problem

Mając już pewne doświadczenie z językiem HTML, na pewno próbowałeś określić układ elementów na stronie przy pomocy tabel, a jej wygląd przy pomocy znaczników takich jak `<i>`, `<b>` czy `<font>`. Stosując takie podejście prędzej czy później zauważysz jego oczywiste wady. Przy bardziej złożonych stronach kod HTML staje się nieczytelny, a próba zmiany stylu lub układu elementów pociąga za sobą konieczność wprowadzenia poprawek w wielu miejscach w kodzie. Co więcej, jeśli strona, którą tworzysz, jest tylko jedną z wielu stron w serwisie, to zmiany najprawdopodobniej trzeba wprowadzić również w pozostałych dokumentach. Rozwiązaniem tych problemów może być oddzielenie treści dokumentu od jego prezentacji. Niestety język HTML nie wspiera takiej możliwości. Do tego celu powszechnie wykorzystuje się inne narzędzie – *kaskadowe arkusze stylów* lub w skrócie CSS (ang. *Cascading Style Sheets*).

O ile tworzenie reguł CSS nie jest trudne, o tyle nauka efektywnego posługiwania się nim może zająć dużo więcej czasu, niż w przypadku języka HTML. Musisz nie tylko wiedzieć, jakie właściwości mają poszczególne elementy HTML, lecz również jak właściwości te są interpretowane przez najpopularniejsze przeglądarki. Ponadto musisz nauczyć się odpowiednio przygotowywać dokumenty HTML i poprawnie pozycjonować elementy na stronie.

### Podstawy teoretyczne

Kaskadowe arkusze stylów to mechanizm używany do opisywania wyglądu dokumentów HTML. Wśród głównych zalet wynikających z jego stosowania wymienić można:

- **Łatwiejsze zarządzanie wyglądem dokumentów** – CSS pozwala grupować informacje o wyglądzie poszczególnych elementów strony, jak również wykorzystywać je wielokrotnie, nawet przez wiele dokumentów.
- **Mniejszy rozmiar stron** – stosowanie arkuszy stylów pozwala zmniejszyć rozmiar dokumentów HTML i ogólnie ograniczyć ilość danych, jakie musi pobrać użytkownik, by wyświetlić je w przeglądarce – w przypadku, gdy wiele stron wykorzystuje ten sam arkusz CSS, jest on pobierany przez przeglądarkę tylko raz.
- **Oddzielenie treści od wyglądu dokumentów** – dzięki CSS można oddzielić zawartość stron od ich wyglądu, dzięki czemu ich treść staje się łatwiej dostępna np. dla wyszukiwarek internetowych lub czytników ekranowych, wykorzystywanych przez osoby niewidome.

### Reguły CSS

Arkusz CSS stanowi zbiór reguł, które informują przeglądarkę, jak wyświetlać określone elementy na stronie. Każda reguła składa się z dwóch części: tzw. *selektora* i definicji, którą tworzy zbiór par *właściwość-wartość*:

```
selektor { właściwość: wartość[; właściwość: wartość[;...n] ] }
```

Selektor określa, jakich elementów strony dotyczy reguła, definicja zaś mówi, jakie wartości powinny przyjąć wybrane właściwości tych elementów, takie jak kolor czy styl tekstu. Przykładowo następująca reguła:

```
h1 { font-family: Arial; font-style: italic; }
```

określa, że tekst wewnątrz wszystkich elementów h1 na stronie powinien zostać wyświetlony krojem Arial, pismem pochyłym.

### Rodzaje selektorów

Selektory to jedne z najważniejszych aspektów CSS. Dzielą się one na kilka rodzajów, z których te najczęściej używane omówimy w kolejnych punktach.

### *Selektor typu*

Selektor typu odnosi się do wszystkich elementów określonego rodzaju. Przykładowo reguła:

```
h1 { font-weight: bold; }
```

zostanie zastosowana dla wszystkich elementów h1 w dokumencie.

### *Selektor klasy*

Selektor klasy odnosi się do wszystkich elementów o określonej wartości atrybutu class. Przykładowo reguła:

```
.akapit { font-weight: bold; }
```

zostanie zastosowana dla elementów dowolnego rodzaju, którym przypisano klasę akapit, np.:

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

### *Selektor ID*

Selektor ID działa na podobnej zasadzie, co selektor klasy, z tymże wybiera elementy o określonej wartości atrybutu id. Ponieważ wartość tego atrybutu jest unikatowa, selektor tego typu zawsze odnosi się do co najwyżej jednego elementu na stronie. Przykładowo reguła:

```
#stopka { font-weight: bold; }
```

zostanie zastosowana dla elementu dowolnego rodzaju posiadającego atrybut id="stopka", np.:

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

### *Łączenie selektorów*

By precyzyjniej określić elementy, poszczególne rodzaje selektorów można łączyć ze sobą. Przykładowo reguła:

```
p.akapit { font-weight: bold; }
```

zostanie zastosowana wyłącznie dla takich elementów p, którym przypisano klasę akapit.

### **Definiowanie stylów dla dokumentu HTML**

Reguły możemy definiować:

- bezpośrednio w dokumencie HTML, w nagłówku strony:

```
<head>
...
  <style type="text/css">
    h1 { color: olive; }
    p { color: gray; }
  </style>
</head>
```

- w dokumencie HTML jako wartość atrybutu style wybranych elementów:

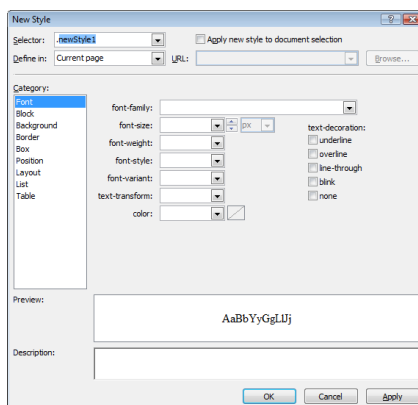
```
<h1 style="color:olive">...</h1>
<p style="color:gray">...</p>
```

- w zewnętrznym pliku CSS – wymagane jest wskazanie go przy pomocy elementu link, np.:

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

## Praca z CSS w środowisku Visual Studio 2008

Środowisko Visual Studio udostępnia wizualny edytor reguł CSS (Rys. 2), który można przywołać wybierając z menu **Format** -> **New Style**.



Rys. 2 Okno New Style, kategoria Font

Najważniejsze pola w edytorze to:

- **Selector** – selektor dla reguły.
- **Define In** – miejsce przechowywania reguły, może być nim nagłówek aktualnej strony lub zewnętrzny arkusz stylów.
- **URL** – lokalizacja zewnętrznego arkusza stylów.
- **Preview** – podgląd zdefiniowanego formatowania.
- **Description** – podgląd definicji reguły.
- **Category** – lista kategorii, na które podzielone zostały właściwości elementów HTML.

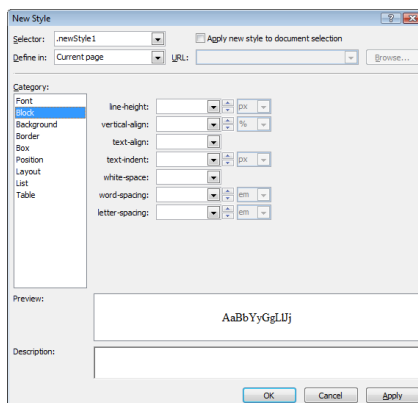
### Kategoria Font

W kategorii **Font** (Rys. 2) znajdują się opcje pozwalające określić sposób wyświetlania tekstu. Przykładowe właściwości:

- **font-family** – krój pisma
- **font-size** – stopień pisma
- **color** – kolor tekstu

### Kategoria Block

W kategorii **Block** (Rys. 3) znajdują się opcje pozwalające określić sposób ułożenia tekstu wewnątrz elementów blokowych.



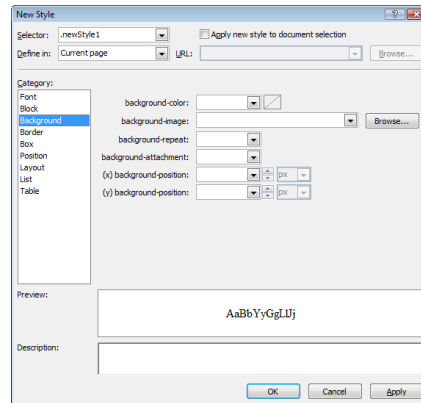
Rys. 3 Okno New Style, kategoria Block

Przykładowe właściwości:

- **line-height** – odległość między sąsiednimi liniami tekst
- **vertical-align** – ułożenie tekstu w pionie
- **text-align** – ułożenie tekstu w poziomie

### Kategoria Background

W kategorii **Background** (Rys. 4) znajdują się opcje pozwalające określić wygląd tła elementu.



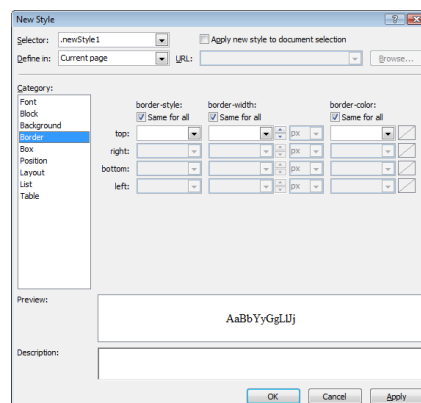
Rys. 4 Okno New Style, kategoria Background

Przykładowe właściwości:

- **background-color** – kolor tła
- **background-image** – obraz wyświetlany w tle
- **background-repeat** – powtarzanie obrazu wyświetlanego w tle w pionie i/lub w poziomie

### Kategoria Border

W kategorii **Border** (Rys. 5) znajdują się opcje pozwalające określić wygląd obramowania wokół elementu.



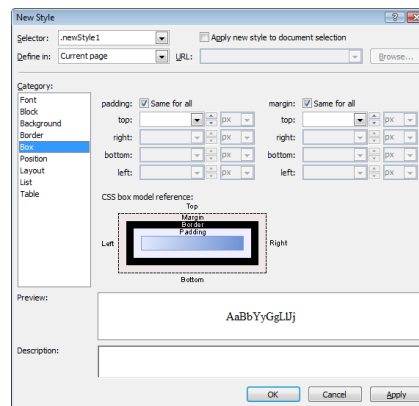
Rys. 5 Okno New Style, kategoria Border

Opcja **Same for all** umożliwia zastosowanie tych samych ustawień dla wszystkich krawędzi obramowania. Przykładowe właściwości:

- **border-style** – styl obramowania
- **border-width** – grubość obramowania
- **border-color** – kolor obramowania

## Kategoria Box

W kategorii **Box** (Rys. 6) znajdują się opcje pozwalające określić marginesy zewnętrzne i wewnętrzne elementów.



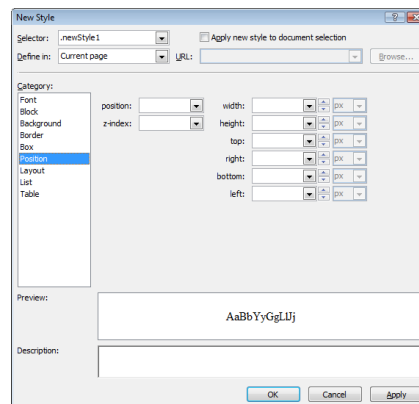
Rys. 6 Okno New Style, kategoria Box

Opcja **Same for all** umożliwia zastosowanie tych samych ustawień dla wszystkich marginesów. Właściwości:

- **padding** – rozmiar marginesu wewnętrznego
- **margin** – rozmiar marginesu zewnętrznego

## Kategoria Position

W kategorii **Position** (Rys. 7) znajdują się opcje pozwalające określić rozmiar i położenie elementów.



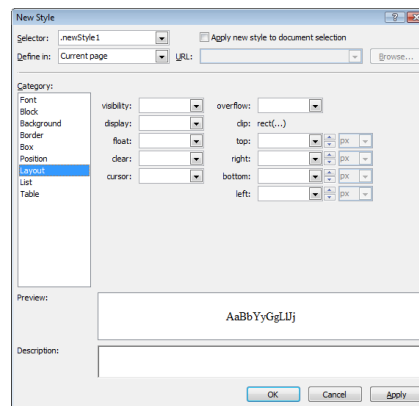
Rys. 7 Okno New Style, kategoria Position

Przykładowe właściwości:

- **position** – rodzaj pozycjonowania (statyczne, względne lub bezwzględne)
- **width** – szerokość elementu
- **height** – wysokość elementu
- **left** – położenie lewej krawędzi elementu (zależy od rodzaju pozycjonowania)
- **top** – położenie górnej krawędzi elementu (zależy od rodzaju pozycjonowania)

## Kategoria Layout

W kategorii **Layout** (Rys. 8) znajdują się opcje pozwalające określić zaawansowane pozycjonowanie elementów na stronie.



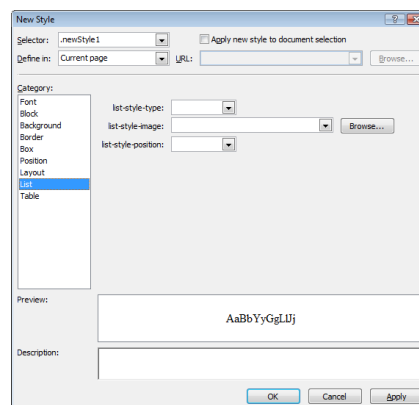
Rys. 8 Okno New Style, kategoria Layout

Przykładowe właściwości:

- **visibility** – widoczność elementu
- **float** – przyciąganie elementu do krawędzi bocznych innych elementów (tzw. *pływanie*)
- **clear** – zakaz możliwości przyciągnięcia innych elementów do krawędzi bocznych elementu

## Kategoria List

W kategorii **List** (Rys. 9) znajdują się opcje pozwalające określić wygląd list.



Rys. 9 Okno New Style, kategoria List

Właściwości:

- **list-style-type** – rodzaj listy (numerowana lub punktowana)
- **list-style-image** – obrazu używany jako punkt w listach punktowanych
- **list-style-position** – położenie punktora

## Podsumowanie

W tym rozdziale przedstawione zostały najważniejsze pojęcia związane z kaskadowymi arkuszami stylów. Dowiedziałeś się, czym są reguły i jak stosować je do wybranych elementów strony wykorzystując selektory.

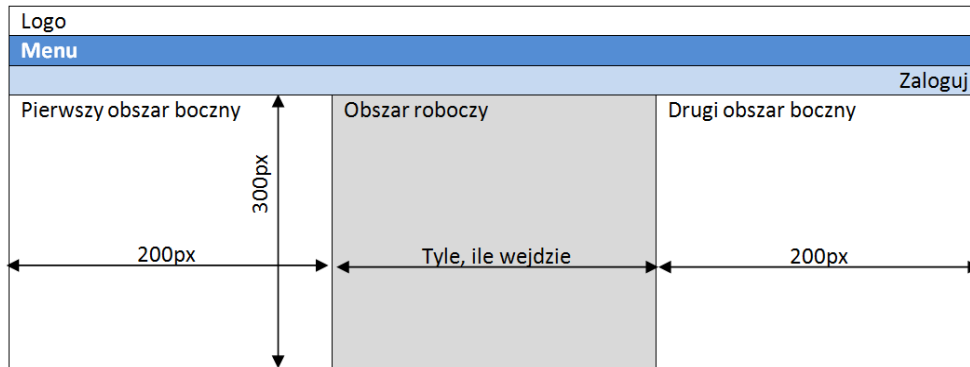
Choć do edycji kaskadowych arkuszy stylów wystarczy najprostszy edytor tekstu, Visual Studio udostępnia dużo wygodniejszy, wizualny edytor stylów, wyposażony m.in. w dynamicznie aktualizowany podgląd formatowania i możliwość zarządzania umiejscowieniem reguł.



## Przykładowe rozwiązanie

### Przygotowanie dokumentu HTML przyjaznego dla CSS

Zanim będziemy mogli zdefiniować wygląd elementów na stronie przy pomocy kaskadowych arkuszy stylów, powinniśmy najpierw odpowiednio przygotować dokument HTML. Wyobraźmy sobie, że chcemy przy pomocy HTML i CSS stworzyć stronę o ułożeniu elementów jak na Rys. 10.



Rys. 10 Szkic wyglądu strony

Pierwszym problemem, z jakim musimy sobie poradzić, jest podzielenie strony na główne obszary. Jeśli przyjrzyś się rysunkowi, zauważysz, że można wyodrębnić dwa główne elementy, oznaczone na rysunku pogrubioną linią: nagłówek strony oraz jej treść. Możemy zdefiniować je za pomocą znaczników `<div>`. Dodatkowo każdemu z nich warto nadać atrybut `id`, aby można było jednoznacznie je identyfikować. By ułatwić pozycjonowanie, możemy dodać jeszcze jeden, dodatkowy znacznik `<div>`, który obejmie całą stronę:

```
<body>
  <div id="Strona">
    <div id="Naglowek"></div>
    <div id="Tresc"></div>
  </div>
</body>
```

Teraz należy przeanalizować zawartość głównych elementów. Rozpocznijemy od nagłówka. Składa się on z trzech głównych elementów: logo, menu oraz obszaru logowania. Postępując według tej samej zasady, co poprzednio, powinniśmy otrzymać:

```
<div id="Naglowek">
  Logo
  <div id="Menu">Menu</div>
  <div id="ObszarLogowania">Zaloguj</div>
</div>
```

W analogiczny sposób możemy wyróżnić trzy obszary, w których prezentowana będzie treść strony:

```
<div id="Tresc">
  <div id="PierwszyObszarBoczny">Pierwszy obszar boczny</div>
  <div id="DrugiObszarBoczny">Drugi obszar boczny</div>
  <div id="ObszarRoboczy">Obszar roboczy</div>
</div>
```

Możemy teraz podejrzeć tak przygotowany dokument w przeglądarce. Jak łatwo zauważyć, na razie wyglądem strona w ogóle nie przypomina tej z rysunku. Mimo to prace nad kodem HTML możemy praktycznie zakończyć.

Podjęcie hierarchiczne ułatwia nam zarządzanie wyglądem strony. Zmiana właściwości elementu nadrzędnego powoduje zmianę wszystkich elementów podrzędnych. Podział hierarchiczny może

wynikać z dwóch czynników, po pierwsze z logicznego podziału strony, po drugie z podziału strony na elementy podobne.

### ***Definiowanie reguł CSS dla dokumentu – definiowanie ułożenia elementów***

W kolejnym kroku powinniśmy utworzyć arkusz stylów (np. plik **Style.css**) i dołączyć go do dokumentu przy pomocy znacznika `<link>`. W pliku zdefiniujemy reguły, które pozwolą wyświetlić obok siebie obszary, w których prezentowana jest treść. Aby odwołać się do tych elementów posłużymy się selektorem ID:

```
#PierwszyObszarBoczny { width:200px; height:300px; float:left; }  
#DrugiObszarBoczny   { width:200px; height:300px; float:right; }
```

Zdefiniowaliśmy rozmiar obu elementów oraz przyciąganie elementu do krawędzi bocznych innych elementów (w tym przypadku elementu nadrzędnego `<div id="Tresc">`). Możemy teraz uruchomić stronę w przeglądarce i zweryfikować poprawność reguł. Obszary boczne zgodnie z oczekiwaniami wyświetlane są przy przeciwległych krawędziach strony.

Teraz zajmiemy się obszarem roboczym (`<div id="ObszarRoboczy">`). W pierwszym kroku spróbujemy określić jego aktualny zasięg. W tym celu zdefiniujemy dla niego kolor tła, dodając następującą regułę CSS:

```
#ObszarRoboczy { background-color: #EEE; }
```

Możemy teraz zapisać plik CSS i odświeżyć stronę w przeglądarce. Jak zauważymy, obszar roboczy jest aktualnie wyświetlany na całą szerokość strony. Powinniśmy odsunąć go od lewej i prawej krawędzi o 200px, aby zrobić miejsce dla obszarów bocznych. W tym celu zastosujemy marginesy. Dodatkowo wprowadzimy marginesy wewnętrzne, aby treść prezentowana w obszarze głównym nie zlewała się z treścią wyświetlaną w pozostałych obszarach. Po zmodyfikowaniu zdefiniowana wcześniej reguła CSS powinna mieć następującą postać:

```
#ObszarRoboczy { background-color: #EEE;  
padding: 50px 20px;  
margin: 0 200px; }
```

Jeśli teraz zapiszemy plik CSS i odświeżymy stronę w przeglądarce, zauważymy, że obszar główny jest wyświetlany tak, jak oczekiwaliśmy.

### ***Definiowanie reguł CSS dla dokumentu – definiowanie wyglądu elementów***

Otrzymaliśmy stronę, której elementy są odpowiednio ułożone. W kolejnym kroku określimy wygląd poszczególnych elementów. Rozpocniemy od zdefiniowania wyglądu elementu na najwyższym poziomie hierarchii. Dla treści strony internetowej takim elementem jest znacznik `<body>`. Do określenia jego właściwości, w pliku CSS, wykorzystamy selektor typu:

```
body {  
font-family: Verdana;  
font-size: 8pt;  
color: #000033; }
```

Określimy teraz nie tylko wygląd tekstu wyświetlanego bezpośrednio w znaczniku `<body>`, ale również wszystkich znaczników zagnieżdżonych, w tym przypadku wszystkich tekstów na stronie. Zajmiemy się teraz elementami nagłówka.

Określimy właściwości elementu `<div id="Menu">`. Dla tego elementu określimy inny kolor tła oraz zmniejszymy wyświetlany tekst. W pliku CSS dodajemy następującą regułę:

```
#Menu {  
background-color: #55BBFF;  
font-size: 7.5pt; }
```

Dla elementu `<div id="ObszarLogowania">` określimy wyrównanie do prawej, inny kolor tła oraz marginesy wewnętrzne. Reguła w pliku CSS będzie miała postać:

```
#ObszarLogowania {  
    text-align: right;  
    background-color: #C3DBEA;  
    padding: 5px 10px; }
```

Jeśli teraz zapiszemy plik CSS i odświeżymy stronę w przeglądarce, zauważymy, że wszystkie elementy są wyświetlane zgodnie ze szkicem strony przedstawionym na Rys. 10. Teraz jedyne co pozostało, to dodać treść do naszej nowej strony.

## Porady praktyczne

### *Uwagi ogólne*

- Pamiętaj, że Twoje strony powinny wyświetlać się poprawnie we wszystkich najpopularniejszych przeglądarkach internetowych, co ze względu na różnice w interpretacji CSS może być czasem trudne.
- Sprawdzaj zgodność kodu HTML oraz CSS ze standardami przy pomocy walidatorów na stronach W3C. Jeśli Twoja strona spełnia wymagania standardów, to umieść na niej informację o zgodności. Możesz w tym celu wykorzystać kod zawarty na stronach walidatorów W3C.
- Staraj się umieszczać w kodzie CSS komentarze. W ten sposób nawet rok po jego napisaniu będziesz mógł szybko przypomnieć sobie, do czego służą poszczególne reguły.
- Organizuj arkusze stylów, np. dzieląc je na sekcje. Dobra organizacja pliku ułatwia jego modyfikację i przeglądanie w przyszłości.
- Jeśli w regule określasz wartość 0, możesz pominąć jednostkę (np. zamiast `0px` możesz napisać `0`).

### *Stosowanie stylów w dokumentach XHTML*

- Możliwe jest dobieranie stylów na podstawie atrybutów `class` i `id`, jak również definiowanie ich bezpośrednio jako wartość atrybutu `style`. Najczęściej powinieneś stosować klasy, ponieważ umożliwiają one wielokrotne wykorzystanie tych samych definicji, natomiast zdecydowanie powinieneś unikać stosowania atrybutu `style`. Jeśli chcesz nadać unikatowy styl jakiemuś elementowi, skorzystaj z atrybutu `id`.
- Jako wartość atrybutu `class` możesz podać wiele klas, np. `<div class="pierwszy drugi">`. Powinieneś jednak uważać, aby wskazane style nie określały tych samych własności, bo różne przeglądarki mogą odmiennie to zinterpretować.
- W nazwach elementów i klas zawsze używaj słów określających ich znaczenie (np. `id="Menu"` lub `class="ObszarBoczny"`), a nie ułożenie na stronie lub wygląd (np. `id="LewyObszarBoczny"` lub `class="NiebieskieTło"`). W końcu jedną z największych zalet CSS jest umożliwienie kompletnej zmiany wyglądu stron bez potrzeby modyfikowania dokumentów HTML.

### *Definiowanie układu elementów na stronie*

- Staraj się nie używać tabel do układania elementów. Stosując je utrudniasz dostęp do treści i możliwość zmiany wyglądu strony w przyszłości.
- Układ elementów na stronie można zdefiniować wykorzystując statyczne pozycjonowanie lub pływanie. Oba podejścia mają wady. Pływanie jest trudne do opanowania, a pozycjonowanie ogranicza elastyczność układu strony. O wyborze jednego z nich powinny zdecydować wymagania względem strony.

### **Określanie wielkości pisma**

- Wielkość pisma możemy określić m.in. w pikselach, punktach lub firetach (em). Wybór właściwej jednostki jest trudny. Wybierając piksele mamy największą kontrolę nad wyglądem strony, ale blokujemy możliwość zmiany wielkości tekstu w przeglądarce. Problemu tego nie mamy, jeśli zastosujemy firety, jednak wówczas musimy upewnić się, że po zmianie rozmiaru tekstu w przeglądarce strona nadal wygląda tak, jak tego chcemy.
- Rozmiar tekstu można określić wykorzystując wartości symboliczne: `xx-small`, `x-small`, `small`, `medium`, `large`, `x-large`, `xx-large` oraz `smaller` i `larger`. Ich użycie jest jednak niepraktyczne, bowiem nie wiemy dokładnie, jaką wielkość przyjmie tekst wyświetlany w przeglądarce.

### **Skrócona notacja**

Tam, gdzie to możliwe, powinieneś używać skróconej notacji CSS oraz grupować style, np. zamiast:

```
.styl1 { margin-top: 11px; margin-right: 11px; margin-bottom: 11px;
        margin-left: 11px; }
.styl2 { margin-top: 11px; margin-right: 11px; margin-bottom: 11px;
        margin-left: 11px; }
```

powinieneś napisać:

```
.styl1, .styl2 { margin: 11px; }
```

### **Uwagi dla studenta**

Jesteś przygotowany do realizacji laboratorium jeśli:

- rozumiesz zasadę i potrzebę budowania dokumentów XHTML przyjaznych dla CSS
- umiesz posługiwać się selektorami
- umiesz zmienić istniejący dokument w języku HTML na zgodny z XHTML i przyjazny dla CSS
- potrafisz podać przykłady przestarzałych znaczników HTML, które można zastąpić przy pomocy CSS
- umiesz zdefiniować w pliku CSS podstawowe własności związane z wyglądem i położeniem elementu na stronie
- umiesz podać przykład zastosowania CSS, jego wady i zalety
- wiesz, w jaki sposób można określić reguły CSS w Visual Studio 2008

Pamiętaj o zapoznaniu się z uwagami i poradami zawartymi w tym module. Upewnij się, że rozumiesz omawiane w nich zagadnienia. Jeśli masz trudności ze zrozumieniem tematu zawartego w uwagach, przeczytaj ponownie informacje z tego rozdziału i zajrzyj do notatek z wykładów.

### **Dodatkowe źródła informacji**

1. Eric A. Meyer, *CSS według Erica Meyera. Kolejna odłona*, Helion, 2005

W książce autor prezentuje możliwości CSS na przykładzie różnych projektów prezentując krok po kroku ich rozwiązanie.

2. Dave Shea, Molly E. Holzschlag, *Zen stosowania CSS*, Helion, 2006

Jest to książka autora witryny CSS Zen Garden. Na przykładzie różnych projektów opisuje on różnorodne aspekty tworzenia stron, m.in. projektowanie, układ, typografia czy efekty specjalne.

3. *Working with CSS Overview*, <http://msdn2.microsoft.com/en-us/library/bb398931.aspx>

Na stronie opisano możliwości pracy z CSS w Visual Studio 2008.

4. *Cascading Style Sheets*, <http://www.w3.org/Style/CSS>

Na stronie znajdziesz wszystkie informacje związane ze standardem CSS.

5. *CSS Zen Garden*, <http://www.csszengarden.com>

Na stronie znajdziesz przykłady zastosowania wielu stylów do jednej strony.

6. *CSS Vault*, <http://cssvault.com>

Na stronie znajdziesz przykłady zastosowania CSS w różnych projektach.

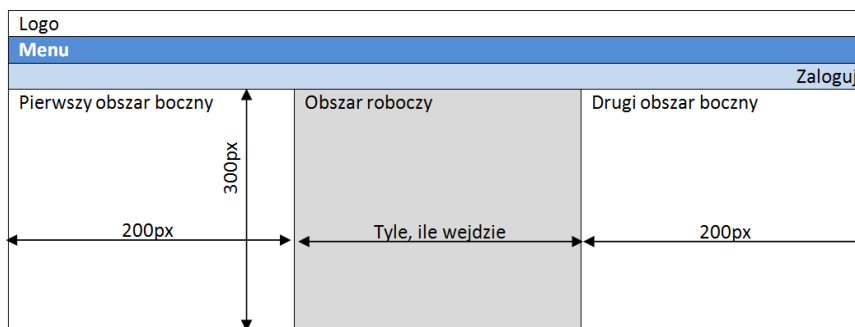
7. *Open Source Web Design*, <http://www.oswd.org>

Na stronie znajdziesz setki gotowych projektów szablonów stron.

## Laboratorium podstawowe

### Problem 1 (czas realizacji 20 min)


Jesteś właścicielem niewielkiej firmy komputerowej wykonującej różnego rodzaju zlecenia. Właśnie dzisiaj przyszedł do Ciebie Twój dobry klient i poprosił o przygotowanie dla niego strony internetowej jego firmy zajmującej się handlem nieruchomościami. Przedstawił Ci swoją koncepcję układu elementów na stronie (Rys. 11) i zaproponował, abyś w oparciu o nią przygotował szablon wyglądu. Ucieszyłeś się i przyjąłeś zlecenie, gdyż dodatkowe pieniądze są zawsze potrzebne, jednak po chwili odkryłeś, że wszyscy Twoi pracownicy są już obłożeni pracą na następnych kilka dni. Postanowiłeś zatem sam wykonać zadanie.

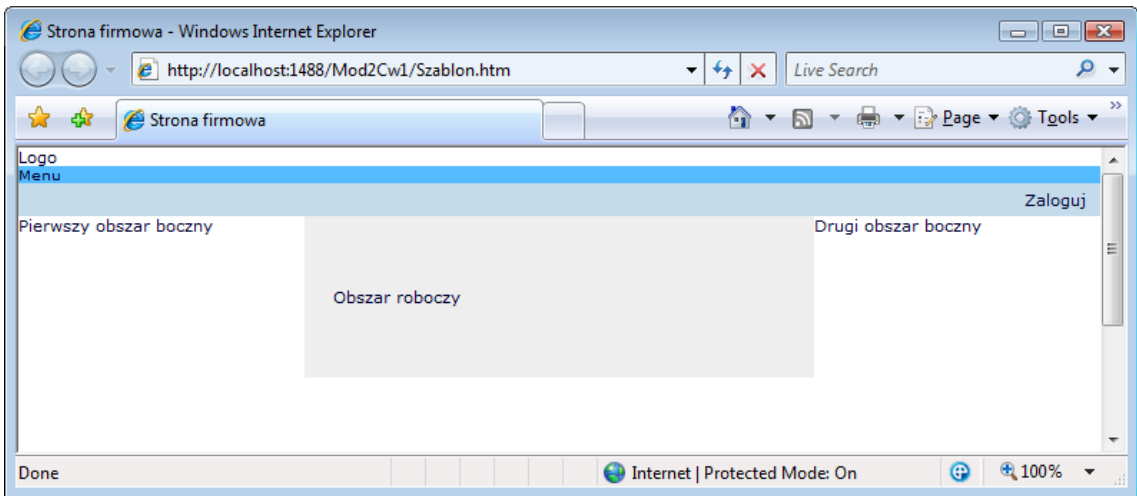


Rys. 11 Szkic wyglądu strony

Pamiętaj, że strona musi być zgodna ze standardem XHTML oraz CSS. Powinieneś zweryfikować to za pomocą walidatorów na stronie W3C.

Zadanie	Tok postępowania
1. Utwórz nową stronę w Visual Studio 2008	<ul style="list-style-type: none"> <li>• Otwórz Visual Studio 2008.</li> <li>• Z menu wybierz <b>File -&gt; New -&gt; Web Site</b>.</li> <li>• Z listy <b>Visual Studio installed templates</b> wybierz <b>Empty Web Site</b>.</li> <li>• Z listy <b>Location</b> wybierz <b>File System</b>, a w polu obok określ lokalizację dla pliku w dowolnym miejscu na dysku.</li> <li>• Z listy <b>Language</b> wybierz <b>Visual C#</b>.</li> <li>• Kliknij <b>OK</b>.</li> </ul>
2. Dodanie strony do projektu. Utworzenie głównych elementów strony	<ul style="list-style-type: none"> <li>• Wybierz <b>Website -&gt; Add New Item</b>.</li> <li>• Z listy <b>Visual Studio installed templates</b> wybierz <b>HTML Page</b>.</li> <li>• W polu <b>Name</b> wpisz <b>Szablon.htm</b>.</li> <li>• Kliknij <b>OK</b>.</li> <li>• Zmień typ dokumentu z <b>XHTML 1.0 Transitional</b> na <b>XHTML 1.1</b>, modyfikując pierwszą linię pliku do postaci: <pre>&lt;!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd"&gt;</pre> </li> <li>• Zmień tytuł strony na „<b>Strona firmowa</b>”.</li> <li>• Wewnątrz znacznika <b>&lt;body&gt;</b> dodaj: <pre>&lt;div id="Strona"&gt;   &lt;div id="Naglowek"&gt;&lt;/div&gt;   &lt;div id="Tresc"&gt;&lt;/div&gt; &lt;/div&gt;</pre> </li> </ul>
3. Podział głównych	<ul style="list-style-type: none"> <li>• Wewnątrz znacznika zawierającego nagłówek (<b>id="Naglowek"</b>) dodaj: <pre>Logo</pre> </li> </ul>



elementów	<pre>&lt;div id="Menu"&gt;Menu&lt;/div&gt; &lt;div id="ObszarLogowania"&gt;Zaloguj&lt;/div&gt;</pre> <ul style="list-style-type: none"> <li>Wewnątrz znacznika zawierającego treść (<b>id="Tresc"</b>) dodaj: <pre>&lt;div id="PierwszyObszarBoczny"&gt;Pierwszy obszar boczny&lt;/div&gt; &lt;div id="DrugiObszarBoczny"&gt;Drugi obszar boczny&lt;/div&gt; &lt;div id="ObszarRoboczy"&gt;Obszar roboczy&lt;/div&gt;</pre> </li> </ul>
4. Sprawdzenie wyniku prac	<ul style="list-style-type: none"> <li>W oknie <b>Solution Explorer</b> kliknij prawym przyciskiem myszy plik <b>Szablon.htm</b>, a następnie wybierz <b>View In Browser</b>.</li> </ul>  Czy w przeglądarce internetowej strona wygląda tak, jak na szkicu? Dlaczego nie?
5. Dodanie nowego arkusza stylów	<ul style="list-style-type: none"> <li>Wybierz <b>Website</b> -&gt; <b>Add New Item</b>.</li> <li>Z listy <b>Visual Studio installed templates</b> wybierz <b>Style Sheet</b>.</li> <li>W polu <b>Name</b> wpisz <b>Style.css</b>.</li> <li>Kliknij <b>OK</b>.</li> </ul>
6. Określenie pozycji w obszarach roboczych	<ul style="list-style-type: none"> <li>Przełącz widok strony na <b>Design</b> lub <b>Split</b>.</li> <li>Wybierz <b>Format</b> -&gt; <b>New Style</b>.</li> <li>W polu <b>Selector</b> wpisz <b>#PierwszyObszarBoczny</b>.</li> <li>W górnej części okna z listy <b>Define in</b> wybierz <b>Existing Style Sheet</b>, a z listy rozwijanej <b>URL</b> wybierz <b>Style.css</b>.</li> <li>Z listy <b>Category</b> wybierz <b>Layout</b>, a następnie określ właściwość <b>float: left</b>.</li> <li>Z listy <b>Category</b> wybierz <b>Position</b>, a następnie określ właściwości <b>width: 200 px</b> oraz <b>height: 300 px</b>.</li> <li>Kliknij <b>OK</b>.</li> <li>Postępuj podobnie z selektorem <b>#DrugiObszarBoczny</b>, tylko zmień właściwość <b>float</b> na <b>right</b>.</li> </ul>
7. Określenie podstawowego wyglądu zawartości strony	<ul style="list-style-type: none"> <li>W pliku <b>Style.css</b> standardowo jest już dodany selektor znacznika <b>&lt;body&gt;</b>. Kliknij go prawym przyciskiem myszy i wybierz <b>Build Style</b>.</li> <li>Określ właściwości <b>font-family: Verdana</b>, <b>font-size: 8 pt</b> i <b>color: #000033</b>.</li> <li>Z listy <b>Category</b> wybierz <b>Background</b>, a następnie określ właściwość <b>background-color: white</b>.</li> <li>Z listy <b>Category</b> wybierz <b>Box</b>, a następnie określ właściwość <b>margin: 0</b>. Upewnij się, że opcja <b>Same for all</b> jest zaznaczona.</li> <li>Kliknij <b>OK</b>.</li> <li>Otwórz plik <b>Szablon.htm</b>. W oknie <b>Solution Explorer</b> chwyć myszką plik <b>Style.css</b> i upuść go w sekcji nagłówek strony za znacznikiem <b>&lt;/title&gt;</b>. Zobacz jak zmienił się podgląd pliku.</li> </ul>
8. Określenie wyglądu pozostałych elementów strony	<ul style="list-style-type: none"> <li>Przy pomocy narzędzi Visual Studio lub w kodzie pliku <b>Style.css</b> określ następujące style: <ul style="list-style-type: none"> <li><b>#Menu</b>: kolor tła: <b>#55BBFF</b>, wielkość tekstu: <b>7.5pt</b>.</li> <li><b>#ObszarLogowania</b>: wyrównanie tekstu (właściwość <b>text-align</b>): <b>do prawej</b>, kolor tła <b>#C3DBEA</b>, margines wewnętrzny (ang. <i>padding</i>) z góry i dołu <b>5px</b>, z lewej i prawej <b>10px</b>.</li> <li><b>#ObszarRoboczy</b>: kolor tła <b>#EEE</b>, margines wewnętrzny (ang. <i>padding</i>) z góry i dołu <b>50px</b>, z lewej i prawej <b>20px</b>, margines z góry i dołu <b>0</b>, z lewej i prawej <b>200px</b>.</li> </ul> </li> </ul>


<p>9. Sprawdzenie wyglądu strony w przeglądarce</p>	<ul style="list-style-type: none"> <li>• W oknie <b>Solution Explorer</b> kliknij prawym przyciskiem myszy plik <b>Szablon.htm</b> i wybierz <b>View In Browser</b>.</li> <li>• Strona w przeglądarce powinna wyglądać podobnie jak na Rys. 12.</li> </ul>
 <p>Rys. 12 Wygląd strony po zastosowaniu stylów</p>	
<p>10. Sprawdzenie poprawności na stronie W3C</p>	<ul style="list-style-type: none"> <li>• Wejdź na stronę <a href="http://validator.w3.org">http://validator.w3.org</a> i zweryfikuj poprawność dokumentu XHTML.</li> <li>• Czy Twój dokument jest zgodny ze standardem XHTML? Jeśli nie, to przeanalizuj błędy i spróbuj je poprawić.</li> <li>• Wejdź na stronę <a href="http://jigsaw.w3.org/css-validator">http://jigsaw.w3.org/css-validator</a> i zweryfikuj poprawność arkusza CSS.</li> <li>• Czy Twój arkusz stylów jest zgodny ze standardem CSS? Jeśli nie, to przeanalizuj błędy i spróbuj je poprawić.</li> </ul>
<p>11. Dodanie informacji o zgodności ze standardami</p>	<ul style="list-style-type: none"> <li>• Wewnątrz znacznika zawierającego obszar roboczy (<b>id="ObszarRoboczy"</b>) dodaj następujący kod:             <pre>&lt;p&gt; &lt;a href="http://validator.w3.org/check?uri=referer"&gt;   &lt;img style="border:0;width:88px;height:31px"     src="http://www.w3.org/Icons/valid-xhtml11-blue"     alt="Valid XHTML 1.1" /&gt;&lt;/a&gt; &lt;a href="http://jigsaw.w3.org/css-validator/"&gt;   &lt;img style="border:0;width:88px;height:31px"     src="http://jigsaw.w3.org/css-validator/images/vcss-blue"     alt="Poprawny CSS!" /&gt;&lt;/a&gt; &lt;/p&gt;</pre> </li> </ul>
<p>12. Zamiana położenia obszarów bocznych</p>	<ul style="list-style-type: none"> <li>• W pliku <b>Style.css</b> w regule z selektorem <b>#PierwszyObszarBoczny</b> zmień właściwość <b>float</b> na <b>right</b>.</li> <li>• W regule z selektorem <b>#DrugiObszarBoczny</b> zmień właściwość <b>float</b> na <b>left</b>.</li> <li>• Czy oba obszary zamieniły się miejscami? Jakiego kroku musiałbyś wykonać, gdyby dokument sformatowany był z użyciem tabel?</li> </ul>
<p>13. Przygotowanie propozycji własnej stylistyki</p>	<ul style="list-style-type: none"> <li>• Przygotuj arkusz CSS zawierający Twoją propozycję nowego wyglądu strony. Zmień kolory, krój tekstu i inne właściwości.</li> </ul>





## Problem 2 (czas realizacji 25 min)

Akurat wtedy, kiedy skończyłeś pracę nad szablonem, przyszedł do Ciebie kolejny klient. Przyniósł ze sobą stronę przygotowaną kilka lat wcześniej i poprosił, abyś dostosował ją do obecnych standardów. Bardzo zależało mu na zachowaniu obecnego wyglądu i układu elementów, jednak z możliwością łatwego wprowadzania zmian w przyszłości. Spojrzałeś w kod i zauważyłeś, że do rozmieszczania elementów strony używane są tabele, zaś wygląd definiowany jest przy użyciu przestarzałych znaczników, takich jak <b>, <i> czy <font>. Uświadomiłeś sobie, że praktycznie trzeba będzie przepisać kod od początku, ale ponieważ klientowi w potrzebie nigdy się nie odmawia, zgodziłeś się podjąć zadania.

Zadanie	Tok postępowania
1. Utwórz nową stronę w Visual Web Developer 2008 Express Edition	<ul style="list-style-type: none"> <li>Otwórz Visual Web Developer 2008 Express Edition.</li> <li>Z menu wybierz <b>File</b> -&gt; <b>New Web Site</b>.</li> <li>Z listy <b>Visual Studio installed templates</b> wybierz <b>Empty Web Site</b>.</li> <li>Z listy <b>Location</b> wybierz <b>File System</b>, a w polu obok określ lokalizację pliku w dowolnym miejscu na dysku.</li> <li>Z listy <b>Language</b> wybierz <b>Visual C#</b>.</li> <li>Kliknij <b>OK</b>.</li> </ul>
2. Dodanie oryginalnej strony do projektu	<ul style="list-style-type: none"> <li>Dodaj do projektu istniejący plik <b>StronaGlowna.htm</b>. Wybierz <b>Website</b> -&gt; <b>Add Existing Item</b>, a następnie wskaż jego lokalizację.</li> <li>Otwórz stronę klikając dwukrotnie plik <b>StronaGlowna.htm</b> w oknie <b>Solution Explorer</b>.</li> </ul>
3. Sprawdzenie poprawności dokumentu	<ul style="list-style-type: none"> <li>Źródło dokumentu przekopiuj na stronę <a href="http://validator.w3.org">http://validator.w3.org</a> do pola w zakładce <b>Validate by Direct Input</b>. <ul style="list-style-type: none"> <li>Wciśnij przycisk <b>Check</b>.</li> </ul> </li> <li> Ile błędów jest na stronie? <ul style="list-style-type: none"> <li>Z listy <b>Doctype</b> wybierz <b>XHTML 1.1</b> i wciśnij przycisk <b>Revalidate</b>.</li> </ul> </li> <li> Ile błędów jest wyświetlanych teraz?</li> </ul>
4. Utworzenie szablonu strony	<ul style="list-style-type: none"> <li>Wybierz <b>Website</b> -&gt; <b>Add New Item</b>.</li> <li>Z listy <b>Visual Studio installed templates</b> wybierz <b>HTML Page</b>.</li> <li>W polu <b>Name</b> wpisz <b>Szablon.htm</b>.</li> <li>Kliknij <b>OK</b>.</li> <li>Zamień zawartość pliku na: <pre>&lt;!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd"&gt; &lt;html xmlns="http://www.w3.org/1999/xhtml"&gt; &lt;head&gt;&lt;title&gt;Lorem ipsum dolor sit amet&lt;/title&gt;&lt;/head&gt; &lt;body&gt;   &lt;div id="Strona"&gt;     &lt;div id="Naglowek"&gt;       &lt;h1&gt;&lt;/h1&gt;       &lt;h2&gt;&lt;/h2&gt;     &lt;/div&gt;     &lt;div id="Tresc"&gt;       &lt;div id="Obszar1"&gt;         &lt;div class="Ramka"&gt;&lt;/div&gt;         &lt;div class="Ramka"&gt;&lt;/div&gt;         &lt;div class="Ramka Ostatnia"&gt;&lt;/div&gt;       &lt;/div&gt;       &lt;div id="Obszar2"&gt;&lt;/div&gt;     &lt;/div&gt;   &lt;/div&gt; &lt;/body&gt; &lt;/html&gt;</pre> </li> </ul>

	<pre> &lt;div id="ObszarGlowny"&gt;&lt;/div&gt; &lt;div id="Obszar3"&gt;   &lt;div class="Ramka"&gt;&lt;/div&gt;   &lt;div class="Ramka Ostatnia"&gt;&lt;/div&gt; &lt;/div&gt; &lt;/div&gt; &lt;div id="Stopka"&gt;&lt;/div&gt; &lt;/div&gt; &lt;/body&gt; &lt;/html&gt; </pre> <p> Zauważ, że został zmieniony standardowy typ dokumentu z <b>XHTML 1.0 Transitional</b> na <b>XHTML 1.1</b>.</p> <ul style="list-style-type: none"> <li>• Przekopiuj treść z pliku <b>StronaGlowna.htm</b> do nowo utworzonego szablonu na podstawie wskazówek poniżej.</li> <li>• Do wnętrza znacznika <b>&lt;h1&gt;</b> przekopiuj tekst z pierwszej wiersza tabeli.</li> <li>• Do wnętrza znacznika <b>&lt;h2&gt;</b> przekopiuj tekst z drugiej wiersza tabeli.</li> <li>• Do wnętrza znacznika o <b>id="Stopka"</b> przekopiuj tekst z ostatniego wiersza tabeli.</li> <li>• Tekst zapisany stylem pogrubionym umieść wewnątrz znaczników <b>&lt;h3&gt;</b>.</li> <li>• Resztę tekstu umieść wewnątrz znaczników <b>&lt;p&gt;</b>.</li> <li>• Wewnątrz znacznika o <b>id="ObszarGlowny"</b>, pomiędzy znacznikami <b>&lt;h3&gt;</b> i <b>&lt;p&gt;</b>, dodaj następujący kod: <pre> &lt;img src="Pictures/WebPage.png" alt="Web page" /&gt; </pre> </li> </ul>
<p>5. Dodanie nowego arkusza stylów</p>	<ul style="list-style-type: none"> <li>• Wybierz <b>Website -&gt; Add New Item</b>.</li> <li>• Z listy <b>Visual Studio installed templates</b> wybierz <b>Style Sheet</b>.</li> <li>• W polu <b>Name</b> wpisz <b>Style.css</b>.</li> <li>• Kliknij <b>OK</b>.</li> </ul>
<p>6. Określenie podstawowego ułożenia strony</p>	<ul style="list-style-type: none"> <li>• Do pliku <b>Style.css</b> dodaj następujące reguły: <pre> #Obszar1 { float: left; width: 620px; } #Obszar2 { float: right; width: 200px; } #Strona { width: 620px; } #Tresc { float: left; width: 620px; } .Ramka { float: left; width: 200px; } </pre> </li> </ul>
<p>7. Sprawdzenie wyglądu strony w przeglądarce</p>	<ul style="list-style-type: none"> <li>• Otwórz plik <b>Szablon.htm</b>. W oknie <b>Solution Explorer</b> chwyć myszką plik <b>Style.css</b> i upuść go w sekcji nagłówka strony za znacznikiem <b>&lt;/title&gt;</b>.</li> <li>• W oknie <b>Solution Explorer</b> kliknij prawym przyciskiem myszy plik <b>Szablon.htm</b> i wybierz <b>View In Browser</b>. Zobacz jak wygląda projektowana strona.</li> </ul>
<p>8. Określenie wyglądu strony</p>	<ul style="list-style-type: none"> <li>• Przy pomocy narzędzi Visual Studio lub w kodzie pliku <b>Style.css</b> określ style dla wymienionych niżej elementów.</li> <li>• Dla elementu <b>body</b>: <ul style="list-style-type: none"> <li>– Kolor tła: <b>#708090</b></li> <li>– Kolor czcionki: <b>#F5F5F5</b></li> </ul> </li> <li>• Dla elementów <b>h1</b>: <ul style="list-style-type: none"> <li>– Rodzinę fontów: <b>Arial, Helvetica, sans-serif</b></li> <li>– Margines górny: <b>0</b></li> <li>– Wyrównanie: <b>wyśrodkowane</b></li> </ul> </li> </ul>

	<ul style="list-style-type: none"><li>• Dla elementów <b>h2</b>:<ul style="list-style-type: none"><li>– Wielkość tekstu: <b>1em</b></li><li>– Styl tekstu: <b>kursywa</b></li><li>– Margines: <b>0</b></li></ul></li><li>• Dla elementów <b>h3</b>:<ul style="list-style-type: none"><li>– Wielkość fontów: <b>1em</b></li><li>– Styl fontów: <b>pogubienie</b></li><li>– Margines: <b>0</b></li></ul></li><li>• Dla elementów <b>img</b>:<ul style="list-style-type: none"><li>– Pływanie: <b>do lewej</b></li><li>– Margines: <b>10px 10px 10px 0</b></li><li>– Szerokość: <b>172px</b></li></ul></li><li>• Dla elementów <b>p</b>:<ul style="list-style-type: none"><li>– Margines: <b>0</b></li></ul></li><li>• Dla elementu o <b>id="Obszar2"</b> dodaj:<ul style="list-style-type: none"><li>– Margines górny: <b>10px</b></li></ul></li><li>• Dla elementu o <b>id="Strona"</b> dodaj:<ul style="list-style-type: none"><li>– Margines lewy: <b>auto</b></li><li>– Margines prawy: <b>auto</b></li></ul></li><li>• Dla elementu o <b>id="Tresc"</b> dodaj:<ul style="list-style-type: none"><li>– Margines górny i dolny: <b>10px</b></li><li>– Margines prawy i lewy: <b>0</b></li></ul></li><li>• Dla elementu o <b>id=" Obszar3"</b> oraz <b>id="ObszarGlowny"</b>:<ul style="list-style-type: none"><li>– Pływanie: <b>do lewej</b></li><li>– Margines górny: <b>10px</b></li><li>– Szerokość: <b>410px</b></li></ul></li><li>• Dla elementu o <b>id="Stopka"</b> dodaj:<ul style="list-style-type: none"><li>– Styl fontów: <b>pogubienie</b></li><li>– Wyrównanie: <b>wyśrodkowane</b></li></ul></li><li>• Dla elementów klasy <b>Ramka</b> dodaj:<ul style="list-style-type: none"><li>– Margines prawy: <b>10px</b></li></ul></li><li>• Dla elementów klasy <b>Ostatnia</b>:<ul style="list-style-type: none"><li>– Margines prawy: <b>0</b></li></ul></li><li>• Dla elementów klasy <b>Ikona</b>:<ul style="list-style-type: none"><li>– Obramowanie: <b>0</b></li><li>– Szerokość: <b>88px</b></li><li>– Wysokość: <b>31px</b></li></ul></li></ul>
9. Sprawdzenie wyglądu strony w przeglądarce	<ul style="list-style-type: none"><li>• W oknie <b>Solution Explorer</b> kliknij prawym przyciskiem myszy plik <b>Szablón.htm</b> i wybierz <b>View In Browser</b>. Zobacz jak teraz wygląda projektowana strona.</li></ul>

	<ul style="list-style-type: none"><li>• Strona powinna wyglądać podobnie do strony <b>StronaGlowna.htm</b>.</li></ul>
10. Sprawdzenie poprawności na stronie W3C	<ul style="list-style-type: none"><li>• Wejdź na stronę <a href="http://validator.w3.org">http://validator.w3.org</a> i zweryfikuj poprawność dokumentu XHTML.</li><li>•  Czy Twój dokument jest zgodny ze standardem XHTML? Jeśli nie, to przeanalizuj błędy i spróbuj je poprawić.</li><li>• Wejdź na stronę <a href="http://jigsaw.w3.org/css-validator">http://jigsaw.w3.org/css-validator</a> i zweryfikuj poprawność arkusza CSS.</li><li>•  Czy Twój arkusz stylów jest zgodny ze standardem CSS? Jeśli nie, to przeanalizuj błędy i spróbuj je poprawić.</li></ul>
11. Dodanie informacji o zgodności ze standardem	<ul style="list-style-type: none"><li>• Wewnątrz znacznika o <b>id="Stopka"</b> dodaj następujący kod: <pre>&lt;p&gt; &lt;a href="http://validator.w3.org/check?uri=referer"&gt;   &lt;img class="Ikona"     src="http://www.w3.org/Icons/valid-xhtml11-blue"     alt="Valid XHTML 1.1" /&gt;&lt;/a&gt; &lt;a href="http://jigsaw.w3.org/css-validator/"&gt;   &lt;img class="Ikona"     src="http://jigsaw.w3.org/css-validator/images/vcss-blue"     alt="Poprawny CSS!" /&gt;&lt;/a&gt; &lt;/p&gt;</pre></li></ul>
12. Przygotowanie propozycji własnej stylistyki	<ul style="list-style-type: none"><li>• Przygotuj arkusz CSS zawierający Twoją propozycję nowego wyglądu strony. Zmień kolory, krój tekstu i inne właściwości.</li></ul>

## Laboratorium rozszerzone

### Zadanie 1 (czas realizacji 45 min)

Przygotowałeś dla klienta stronę internetową z wykorzystaniem XHTML i CSS. Ten bardzo się ucieszył, że jest ona zgodna ze standardami i można łatwo wprowadzać w niej modyfikacje, a ponieważ zbliża się rocznica powstania jego firmy, zaproponował, abyś z tej okazji przygotował propozycję nowego układu oraz wyglądu strony. Ponieważ nie masz dużego doświadczenia w dziedzinie projektowania, postanowiłeś poszukać inspiracji na jednej z witryn oferujących gotowe rozwiązania. Z pomocą wyszukiwarki trafiłeś na stronę Open Source Web Design (<http://www.oswd.org>), oferującą darmowe szablony. Spróbuj zaadoptować style z któregoś z projektów do strony przygotowanej w poprzednim zadaniu.

### Zadanie 2 (czas realizacji 45 min)

Spółka usługowo-handlowa Kowalski-Sale posiada witrynę internetową wykonaną 4 lata temu. Firma się rozwija i w związku z tym chciałaby odświeżyć wizerunek oraz rozszerzyć informacje na swojej stronie. Niestety nie jest to łatwe, gdyż do zdefiniowania wyglądu i ułożenia elementów zostały użyte tabele i przestarzałe znaczniki. Przeanalizuj dokumenty zawarte w archiwum Kowalski-Sale.zip i opracuj nową wersję strony zgodną ze standardami XHTML i CSS. Przygotuj co najmniej dwie różne wersje układu i wyglądu strony.

### Zadanie 3 (czas realizacji 90 min)

Starasz się o świetnie płatną pracę w dużej firmie tworzącej aplikacje internetowe. Jesteś jednym z kilku kandydatów, którzy przeszli do drugiej tury rozmów. Twój przyszły pracodawca postanowił sprawdzić Twoje umiejętności posługiwania się kaskadowymi arkuszami stylów, prosząc Cię o przygotowanie pliku CSS dla strony głównej witryny CSS Zen Garden (<http://csszengarden.com>).

Podpowiedź: w serwisie znajdują się przykładowe pliki CSS przygotowane przez innych użytkowników. Zobacz, jak został w nich zdefiniowany wygląd strony głównej. Pamiętaj, że w firmie działa system antyplagiatowy!

### Zadanie 4 (czas realizacji 90 min)

Jesteś niezależnym projektantem stron internetowych. Twoim nowym zleceniem jest przygotowanie profesjonalnie wyglądającej galerii zdjęć. Twój klient określił następujące wymagania:

- Strona powinna być zgodna ze standardem XHTML 1.1 oraz CSS 2.1.
- Wygląd strony musi być zdefiniowany w zewnętrznym pliku CSS.
- Na głównej stronie galerii ma być wyświetlana lista miniatur zdjęć wraz z odnośnikami do zdjęcia. Każda miniatura musi być umieszczona w wystylizowanej w CSS ramce. Ilość miniatur w wierszu ma być zależna od szerokości okna przeglądarki.

Jako zawartość galerii wykorzystaj wybrane przez siebie zdjęcia umieszczone w serwisie Live Spaces (<http://www.spaces.live.com>).