



ILUSTROWANY
PRZEWODNIK

TWÓRZ WITRYNY, KTÓRE BĘDĄ
RZĄDZIĆ W SIECI
ZAPRZYJAŻNIJ SIĘ Z JĘZYKIEM CSS
POZNAJ ZASADY PRZYGOTOWYWANIA
I OBRÓBKĘ GRAFIKI
ODKRYJ SKUTECZNE SPOSOBY
NA PROMOCJĘ W INTERECIE

TWORZENIE STRON WWW

WYDANIE II

ALEKSANDRA
TOMASZEWSKA



Hellen



» Idź do

- Spis treści
- Przykładowy rozdział

» Katalog książek

- Katalog online
- Zamów drukowany katalog

» Twój koszyk

- Dodaj do koszyka

» Cennik i informacje

- Zamów informacje o nowościach
- Zamów cennik

» Czytelnia

- Fragmenty książek online

» Kontakt

Helion SA
ul. Kościuszki 1c
44-100 Gliwice
tel. 32 230 98 63
e-mail: helion@helion.pl
© Helion 1991–2011

Tworzenie stron WWW. Ilustrowany przewodnik. Wydanie II

Autor: [Aleksandra Tomaszewska](#)

ISBN: 978-83-246-2621-2

Format: 140×208, stron: 224



Twórz witryny, które będą rządzić w sieci

- Zaprzyjaj się z językiem CSS
- Poznaj zasady przygotowywania i obróbki grafiki
- Odkryj skuteczne sposoby na promocję w Internecie

Projektować strony WWW może niemal każdy. Jednak tworzenia projektów wysokiej jakości to już spore wyzwanie. Internet jest pełen nieczytelnych, przeładowanych treścią albo grafiką witryn, które przynoszą swoim właścicielom więcej szkody niż pożytku. Technologia pędzi do przodu, podczas gdy niektórzy projektanci wyraźnie za nią nie nadążają. Jeśli chcesz mieć pewność, że efekty Twojej pracy to zawsze produkt najwyższej jakości, tak pod względem funkcjonalności, jak i estetyki, a także jeśli chcesz nauczyć się projektować witryny wyróżniające się spośród milionów przeciętnych półproduktów, nadszedł czas, by zacząć wreszcie działać z rozmachem.

W drugim wydaniu książki „Tworzenie stron WWW. Ilustrowany przewodnik” znajdziesz przydatne narzędzia, ułatwiające pracę projektantom stron WWW. Dowiesz się, dlaczego warto używać aplikacji Macromedia Dreamweaver oraz Adobe GoLive i czemu lepiej mieć kilka aplikacji, które robią to samo, niż jedną aplikację, która robi wszystko. Przeczytasz także o tym, czym różnią się formaty GIF, JPEG i PNG oraz którego z nich warto używać w jakiej sytuacji. Nauczysz się podstaw języków HTML i XHTML i odkryjesz, jak rozwinąć skrzydła w dziedzinie pisania skryptów, ponieważ poznasz bliżej język skryptowy JavaScript. W zakończeniu znajdziesz praktyczne porady dotyczące tego, co i w jakich proporcjach powinno znajdować się w doskonale zaprojektowanej witrynie.

- Zasady pisania i formatowania tekstu oraz edytory stron WWW
- Praca z elementami graficznymi przy użyciu takich narzędzi, jak Photoshop i Fireworks
- Kaskadowe arkusze stylów – formatowanie i programy do tworzenia arkuszy
- Tworzenie dynamicznych stron WWW, czyli JavaScript w akcji
- Przygotowywanie i optymalizowanie animacji Flash
- Język HTML – podstawy i struktura dokumentu

Projektuj pierwszorzędne witryny i twórz nową jakość w Internecie

SPIS TREŚCI

Rozdział 1.	Trzy, dwa, jeden... START!	5
	Blog	5
	Serwisy stron WWW	14
Rozdział 2.	Edytory WYSIWYG, czyli chłopcy od brudnej roboty . .	26
	Edytory HTML	27
	Edytory WYSIWYG	29
	Korzystanie z edytora WYSIWYG	36
Rozdział 3.	Grafika	65
	Dodawanie obrazków do strony	69
	Tworzenie grafiki	76
	Darmowa grafika	83
Rozdział 4.	HTML — język internetu	88
	Składnia języka	88
Rozdział 5.	Wykorzystanie formularzy i ramek	109
	Formularze	109
	Ramki	116
	Dodatkowe znaczniki	119
Rozdział 6.	Co to znaczy mieć styl, czyli słów kilka o CSS	120
	Formatowanie tekstu	128
	Dekoracja tekstu	131
	Kolor i tło	136
	Marginesy	137
	Programy do tworzenia arkuszy stylów	138



Rozdział 7. Tworzenie dynamicznych stron WWW, czyli JavaScript w akcji	141
Typy danych	144
Wyrażenia i warunki	146
Wprowadzanie poleceń	149
Okna	159
Rozdział 8. Animacje Flash	169
Przygotowywanie grafiki	170
Animacje	182
Eksportowanie i publikowanie filmów	189
Rozdział 9. Promocja	191
Wyszukiwarki internetowe	193
Reklama	200
Katalogi internetowe	201
Ogłoszenia prasowe	202
Poczta elektroniczna	203
Banery reklamowe	205
Rozdział 10. Poradnik, czyli od czego zacząć	207
Struktura witryny	212
Szablon witryny	213
Projektowanie zawartości stron	215

ROZDZIAŁ 6

Co to znaczy mieć styl, czyli słów kilka o CSS

Przyciski, paski menu, zdjęcia, obrazy — nieodłączne elementy stron WWW. Dotychczas dowiedziałeś się, jak je tworzyć i budować z nich całość za pomocą edytorów tekstowych lub aplikacji WYSIWYG. Poznałeś również podstawy języka HTML. Czas zaprzyjaźnić się z CSS, czyli kaskadowymi arkuszami stylów. Styl w wirtualnym świecie nie wymaga dużych nakładów finansowych, a jedynie zainwestowania odrobiny czasu w opracowanie stylistyki strony i konsekwentne realizowanie przyjętych założeń.

Kaskadowe arkusze stylów (ang. *cascading style sheets* — CSS) są stosowane do formatowania stron i pozwalają na uzyskanie doskonałej kontroli nad jednolitym wyglądem poszczególnych stron i całej witryny. Wyobraź sobie witrynę składającą się z dużej liczby stron WWW, nad którą pracuje cała grupa programistów. Mając wyobrażenie o stopniu skomplikowania zadania polegającego na stworzeniu kilku stron, na których używana będzie ta sama stylistyka — krój, kolor, wielkość czcionki, kolorystyka elementów czy wygląd tabel, wyobraź sobie, jak trudno zapanować nad tymi wszystkimi elementami, gdy masz do ogarnięcia kilkaset stron projektowanych przez kilka różnych osób. Do tego celu niezawodnie przydają się kaskadowe arkusze stylów.

Korzystanie z CSS oznacza również oszczędność miejsca, ponieważ styl każdego elementu jest definiowany jednokrotnie w pliku definicji stylów, a nie przy każdym wystąpieniu elementu na stronie, czyli w przypadku takich elementów jak na przykład akapit czy nagłówek kilkadziesiąt razy na każdej stronie.

WSKAZÓWKA

Szczegółowe informacje na temat kaskadowych arkuszy stylów znajdziesz pod adresem <http://www.w3.org/Style/CSS/>.

Definicje stylów można wprowadzać do opisu strony na kilka sposobów. Pamiętaj, że każdy z tych sposobów ma swoje wady i zalety, a wybór najlepszego sposobu zależy od preferencji programisty oraz sytuacji, w której styl ma być zastosowany.

Sposób I

```
<p style="color:blue; font-family:Arial;
font-size:12pt;"> Zawartość akapitu </p>
```

Definicja stylu jest umieszczona bezpośrednio w znaczniku. Jest to analogiczna składnia do umieszczania w znaczniku atrybutów tego znacznika.

Zaleta: prostota użycia.

Wada: konieczność powtarzania definicji dla każdego kolejnego znacznika umieszczanego na stronie.

Sposób II

```
<html><head>
<style type="text/css">
p
{ color:blue;
font-family:Arial;
font-size:12pt;
}
</style></head>
<body>
<p> Pierwszy akapit </p>
<p>Drugi akapit</p>
</body></html>
```

Definicja stylu jest umieszczona w nagłówku dokumentu. Tak zdefiniowany styl odnosi się do wszystkich znaczników tego typu w dokumencie.

Zaleta: jednokrotna definicja stylu dla elementu w obrębie strony.

Wada: dla tego samego znacznika na innych stronach musisz ponownie zdefiniować style znacznika wewnątrz nagłówków tych stron.

Sposób III

```
<html><head>
<link href="style.css" rel="stylesheet"
type="text/css"></head>
<body>
<p> Pierwszy akapit </p>
<p>Drugi akapit</p>
</body></html>
```

Definicja stylu jest umieszczona w osobnym pliku o rozszerzeniu *.css, który jest powiązany z dokumentem strony poprzez użycie znacznika <link>. Atrybut href wskazuje ścieżkę dostępu do pliku arkusza stylów. Wywołanie zewnętrznego arkusza stylów znajduje się w nagłówku i dotyczy całej strony WWW, czyli wszystkich elementów danego typu.

Zaleta: do tak zdefiniowanego arkusza można podłączyć wiele stron witryny i wszystkie będą miały zdefiniowane te same style. Zmiana stylu danego elementu na wszystkich stronach witryny wymaga jedynie zmiany odpowiednich wpisów w arkuszu stylów.

**WSKAZÓWKA**

Pamiętaj, że zarówno element `<style>`, jak i element `<link>` muszą być umieszczone wewnątrz nagłówka strony.

Najbardziej zalecanym, najefektywniejszym i dającym najlepsze wyniki sposobem używania arkuszy stylów jest sposób III. W pliku zewnętrznego arkusza stylów znajdują się wyłącznie wpisy definiujące style określonych elementów. Do tak zdefiniowanego arkusza można podłączyć wiele stron witryny i wszystkie będą miały zdefiniowane te same style. Dzięki temu w jednym miejscu możesz definiować wygląd całej witryny i nim zarządzać.

Plik arkusza stylów może mieć postać:

```
p
{ color:blue;
  font-family:Arial;
  font-size:12pt;
}
```

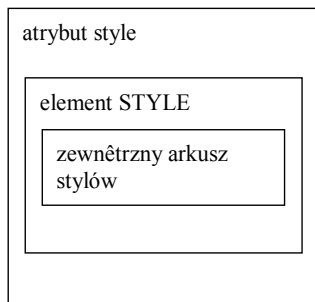
Kaskadowe arkusze stylów nieprzypadkowo noszą taką nazwę, bo jak już wiesz, style można wprowadzać na trzech różnych poziomach:

- ▶ za pomocą atrybutu `style` (sposób I) — styl definiowany kolejno dla poszczególnych elementów na stronie;
- ▶ za pomocą elementu `style` (sposób II) — styl elementu jest określany dla całej strony;
- ▶ za pomocą zewnętrznego arkusza stylów — styl jest definiowany dla danego typu elementu dla wszystkich stron powiązanych z arkuszem stylów.

Filozofia CSS łączy w sobie elastyczny sposób kontroli stylów stron WWW i poszczególnych elementów na stronach z odpowiednią hierarchią stylów, która pozwala unikać konfliktów, jeśli style elementów są definiowane na kilku poziomach. Sposób I (atrybut `style`) ma najwyższą rangę w hierarchii, sposób II (element `style`) ma wyższą rangę niż styl definiowany przez zewnętrzny arkusz stylów, ale niższą niż atrybut `style`. Najniższą rangę ma styl przypisywany przez zewnętrzny arkusz stylów. Takie rozwiązanie pozwala uniknąć konfliktów, które mogłyby wyniknąć, jeśli na stronie użylibyśmy wszystkich trzech metod definiowania stylów. Hierarchia — struktura stylów — przypomina stopnie wodospadu, stąd bierze się w nazwie określenie *kaskadowy* (rysunek 6.1).

Rysunek 6.1.

Style poszczególnych elementów są uporządkowane w określonej kolejności



Hierarchizacja stylów obejmuje nie tylko różne sposoby definiowania stylów, ale również sytuację, gdy do jednej strony podłączonych jest kilka różnych zewnętrznych arkuszy stylów. W takim wypadku przeglądarka uwzględnia kolejność wprowadzania stylów. Zewnętrzny arkusz wprowadzony jako pierwszy dominuje nad arkuszem, który jest wprowadzony jako następny w kolejności. Problem polega na tym, że jeśli w zewnętrznych arkuszach stylów występują wzajemne konflikty, trudno przewidzieć efekt graficzny, dopóki strona nie zostanie wyświetlona w przeglądarce.

Do określenia wartości atrybutów arkusze stylów wykorzystują względne i bezwzględne jednostki miary oraz określone definicje kolorów.

Jednostki względne:

- ▶ px — piksel — jednostka definiowana w oparciu o pojedyncze punkty świetlne monitora;
- ▶ em — proporcje wysokości do czcionki danego elementu — określa zależność pomiędzy poszczególnymi wielkościami;
- ▶ ex — proporcje do wysokości litery;
- ▶ % — procent — określenie wielkości względem wartości domyślnej.

WSKAZÓWKAW

Istnieje także możliwość importowania arkusza stylów wewnątrz innego arkusza stylów. W tym celu należy użyć następującej konstrukcji:

```
@import url(style.css) ;
body {background-image: url(obraz.gif) ;}
```

W takiej sytuacji importowany arkusz ma niższy priorytet niż arkusz, do którego import następuje.

Zasada działania jednostki em jest następująca — przy zdefiniowaniu określonej wartości (na przykład czcionki o wielkości 12 pt) jest ona przyjmowana jako 1 em. Jeśli określe inny atrybut jako 2 em, to będzie on dwukrotnością wcześniej zdefiniowanej wartości (czyli w tym wypadku będzie to 24 pt).

Jednostki bezwzględne:

- ▶ in — cal — jednostka najczęściej używana w USA, rzadko używana w Polsce (1 in = 2,54 cm);
- ▶ pt — punkt — wywodzi się z typografii, gdzie jest standardową jednostką miary (72 pt = 1 in);
- ▶ cm — centymetr — jednostka pochodna systemu metrycznego, często stosowana w Europie;



- ▶ mm — milimetr — jednostka pochodna systemu metrycznego, często stosowana w Europie;
- ▶ pc — pica — jednostka wywodząca się z typografii (1 pica = 12 pt).

W kaskadowych arkuszach stylów można definiować kolory poprzez użycie ich nazw angielskich, jak *blue* czy *yellow*. Wszystkie kolory, które nie zostały zdefiniowane przez użycie ich nazw, muszą być określone przez wykorzystanie modelu RGB. Polega on na przypisaniu liczb całkowitych z przedziału od 0 do 255 każdej z wartości składowych palety RGB. Na przykład `color: rgb(0,12,255)`. Kolor można również zapisać, definiując procent nasycenia danej barwy, na przykład `color: rgb(10%, 50%, 0%)`, a także stosując metodę szesnastkową, na przykład `color: #ff2c00` (tabela 6.1).

Tabela 6.1. Tablica nazw kolorów oraz ich odpowiedników w systemie szesnastkowym

black	#000000	green	#008000
silver	#C0C0C0	lime	#00FF00
gray	#808080	olive	#808000
white	#FFFFFF	yellow	#FFFF00
maroon	#800000	navy	#000080
red	#FF0000	blue	#0000FF
purple	#800080	teal	#008080
fuchsia	#FF00FF	aqua	#00FFFF

Schemat konstrukcji stylu można zapisać w następujący sposób:

selektor {cecha: wartość; cecha: wartość; itd.}

- ▶ *selektor* — znacznik języka HTML, np. `p` to paragraf, `h1` — nagłówek pierwszy, a `table` — tabela.
- ▶ *cecha* — wpis określony specyfikacją kaskadowych arkuszy stylów.

Przykładowe właściwości określone przez specyfikację kaskadowych arkuszy stylów to `background-color`, `border-style`, `font-family` itd. Wartość jest zazwyczaj przypisana do danej cechy, ale może się zdarzyć, że będzie podobna dla kilku zupełnie różnych cech. Ogólnie przyjmujemy, że selektor to dowolny znacznik języka HTML. Możemy jednak wyróżnić kilka rodzajów selektorów:

- ▶ Selektory proste — selektor będący pojedynczym znacznikiem języka HTML, odnoszący się jedynie do tego elementu. Jest to najczęściej spotykany rodzaj selektora.

- ▶ Selektor uniwersalny — używany, gdy jakieś cechy i wartości są uniwersalne i powinny dotyczyć wszystkich elementów na stronie.
- ▶ Potomek — selektor zbudowany w oparciu o zależności panujące pomiędzy poszczególnymi znacznikami języka HTML.

Przykłady selektorów uniwersalnych:

```
*
{color: blue;}
BODY
{color: blue}
```

Tak zapisane selektory działają dla wszystkich elementów strony — nagłówki, akapity, listy itd. będą miały kolor niebieski. Przykłady potomków:

```
p
{color: blue}
p > span
{font-style: italic}
```

Kolor tekstu akapitu będzie niebieski, ale dodatkowo, jeśli wewnątrz akapitu pojawi się znacznik `<span>`, otoczony nim tekst będzie pochylony.

Specyfikacja kaskadowych arkuszy stylów dopuszcza również możliwość grupowania selektorów. Jeśli na przykład chciałbyś, aby wszystkie listy, które wystąpią w dokumencie, miały kolor czerwony, to zamiast definiować styl dla każdego typu listy, możesz zastosować następujący zapis:

```
ul, ol, dl
{
color: red
}
```

Selektory ID pozwalają określić styl określonego elementu na stronie, co pozwala różnicować formatowanie elementów tego samego typu na stronie. Wyobraź sobie, że wszystkie akapity na stronie mają mieć kolor granatowy, ale w jednym tekst powinien być czerwony, ponieważ zawiera on szczególnie ważne informacje. W tym celu możesz zdefiniować style w następujący sposób:

```
p {color: navy}
p#wazne {color: red}
```

Aby utworzyć akapit zawierający tekst w kolorze czerwonym, musisz go wywołać przez użycie zdefiniowanej nazwy:

```
<p id="wazny"> Bardzo ważne informacje wyróżnione kolorem
czerwonym</p>
```

Pozostałe akapity będziesz tworzyć, używając znacznika `<p>`.



Definicja identyfikatora może również mieć następującą postać:

```
#duzy_czerwony_tekst  
{font-size: 20pt; color: red}
```

Identyfikator ten możesz wywołać w dowolnym znaczniku, korzystając z atrybutu ID.

Kolejnym podstawowym elementem specyfikacji kaskadowych arkuszy stylów jest klasa. Klasę konstruuje się w następujący sposób:

```
selektor.nazwa_klasy {cecha: wartość}
```

Tak zdefiniowana klasa może być wywołana tylko w znaczniku, dla którego została utworzona. Aby wywołać klasę w dokumencie, należy użyć atrybutu CLASS.

Na przykład tworząc klasę akapitu zawierającego tekst w kolorze czerwonym, należy wpisać definicję:

```
p.czerwony {color: red}
```

Aby wywołać tę klasę w dokumencie, wpisz:

```
<p class="czerwony"> Czerwony tekst </p>
```

UWAGA

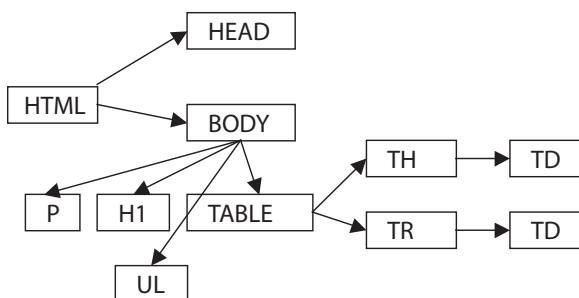
Deklaracja selektora ID musi być poprzedzona znakiem #, ale znaku tego nie należy używać do odwoływania się do wartości atrybutu ID.

UWAGA

Pamiętaj, aby definiując klasę, umieścić znak kropki między selektorem a nazwą klasy.

Dziedziczenie i kaskadowość to cechy, którym kaskadowe arkusze stylów zadzięczają swój dynamiczny rozwój i pozycję. O kaskadowości była mowa już wcześniej w tym rozdziale, a teraz warto napisać kilka słów o dziedziczeniu.

Jeśli chcesz określić kolor tekstu w komórce tabeli, zrobisz to przez użycie elementu <td>. Możesz to również zrobić przez użycie elementu <table> i wtedy wszystkie komórki tabeli, zarówno w wierszach, jak i w nagłówkach, będą miały określony dla tabeli kolor. Ale możesz to również zrobić, definiując kolor tekstu dla elementu <body>. Wtedy nie tylko tekst w komórkach tabeli będzie miał określony kolor, ale również określony kolor będzie mieć tekst w akapitach, nagłówkach i listach. Na tym właśnie polega dziedziczenie w kaskadowych arkuszach stylów — elementy nadrzędne domyślnie przekazują zdefiniowane właściwości elementom, które znajdują się niżej w hierarchii (rysunek 6.2).

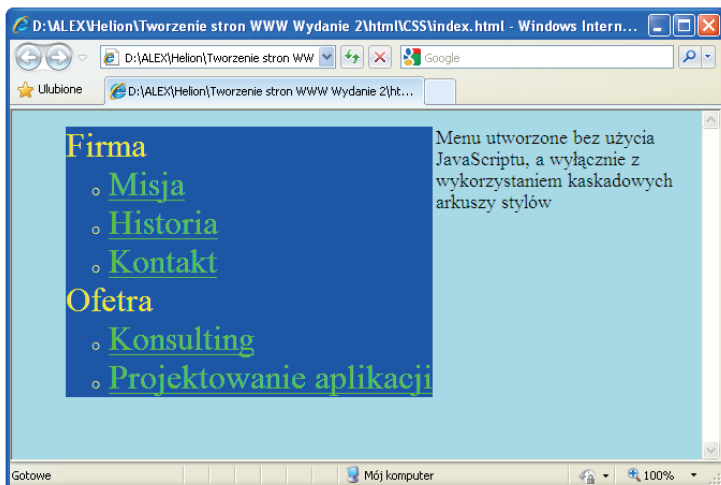


Rysunek 6.2. Schemat dziedziczenia w kaskadowych arkuszach stylów

W naszym przykładzie kolor niebieski dla tekstu, zdefiniowany w elemencie `<body>`, będzie oznaczał również niebieski kolor tekstu we wszystkich komórkach tabeli. Jeśli jednak określimy atrybut elementu podrzędnego, to nie przekaze on tej właściwości w górę hierarchii. W naszym przykładzie niebieski tekst zdefiniowany dla komórki tabeli nie będzie oznaczał niebieskiego tekstu dla wszystkich elementów wewnątrz elementu `<body>`.

Dziedziczenie działa domyślnie, czyli dla tych elementów, dla których nie określono innej wartości właściwości niż ta, którą wskazano dla elementu nadrzędnego. Możemy zdefiniować czerwony tekst w komórkach tabeli, mimo że dla elementu `<body>` zdefiniowano, że tekst ma być niebieski.

Większość ludzi ogranicza wykorzystanie kaskadowych arkuszy stylów do określania koloru tekstu w dokumencie lub kroju czcionki w nagłówkach. Jednak istnieją również ciekawsze zastosowania kaskadowych arkuszy stylów — mogą one służyć do tworzenia menu, do czego większość projektantów zaangażowałaby JavaScript (rysunek 6.3).



Rysunek 6.3. Menu utworzone z wykorzystaniem kaskadowych arkuszy stylów



```
<html>
<head><link rel="stylesheet" href="menu.css" type="text/css">
</head>
<body>
<ul id="menu">
<li>Firma
<ul>
<li><a href="#" "> Misja</a></li>
<li><a href="#" "> Historia</a></li>
<li><a href="#" "> Kontakt</a></li>
</ul>
</li>
<li>Ofetra
<ul>
<li><a href="#" "> Konsulting</a></li>
<li><a href="#" "> Projektowanie aplikacji</a></li>
</ul>
</li>
</ul>
</body>
</html>
```

Menu utworzone bez użycia JavaScriptu, a wyłącznie z wykorzystaniem kaskadowych arkuszy stylów wykorzystuje osobny plik o nazwie *menu.css* o następującej zawartości:

```
body{background-color: lightblue; font-size: larger}
#menu { background-color: blue; float: left }
#menu li {font-size: x-large; color: yellow}
#menu li a {color: lime}
#menu li a hover {color: navy; background-color: white;
font-size: larger }
```

Formatowanie tekstu

Specyfikacja kaskadowych arkuszy stylów pozwala na kontrolowanie następujących właściwości tekstu:

- ▶ wyrównanie poziome i pionowe,
- ▶ wcięcie,
- ▶ odstępy między wierszami,

- ▶ odstępy między wyrazami,
- ▶ odstępy między literami,
- ▶ dekoracja tekstu,
- ▶ przekształcanie tekstu,
- ▶ kontrola pustej przestrzeni.

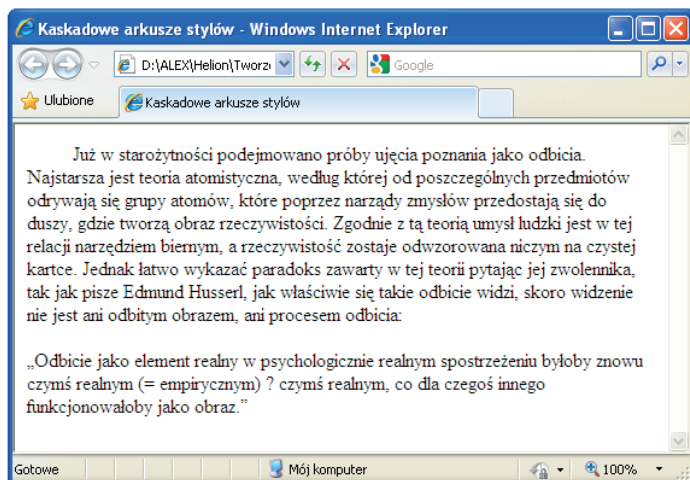
Pionowe i poziome wyrównanie tekstu:

- ▶ do prawej — `text-align: left`,
- ▶ do lewej — `text-align: right`,
- ▶ wyśrodkowanie — `text-align: center`,
- ▶ wyjustowanie bloku tekstu — `text-align: justify`,
- ▶ do góry — `vertical-align: top`,
- ▶ do środka — `vertical-align: middle`,
- ▶ do dołu — `vertical-align: bottom`,
- ▶ indeks dolny — `vertical-align: sub`,
- ▶ indeks górny — `vertical-align: super`.

Wcięcie pierwszego wiersza akapitu

Aby uzyskać efekt wcięcia pierwszego wiersza w akapicie, należy użyć parametru `text-indent` i określić głębokość tego wcięcia w jednostkach względnych lub bezwzględnych (rysunek 6.4). Na przykład:

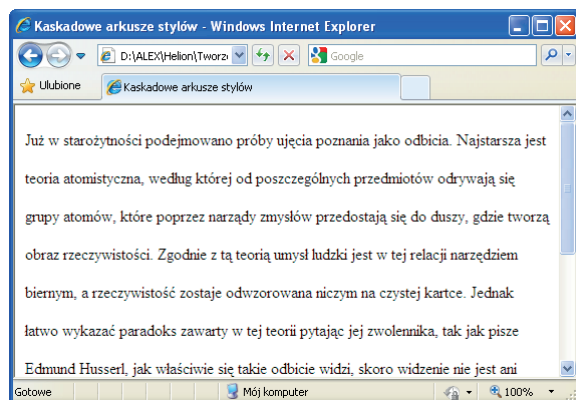
```
p {text-indent: 1 cm}
```



Rysunek 6.4. Wcięcie akapitu uzyskano przez użycie parametru `text-indent`

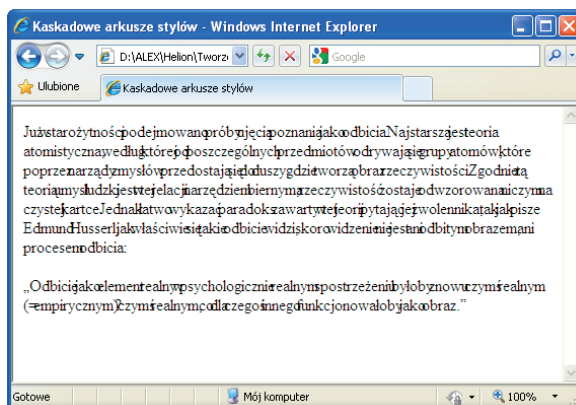
Odstępy

Za sterowanie odstępem pomiędzy wierszami tekstu odpowiada parametr `line-height`, który może przyjmować wartości dodatnie i ujemne. Przy użyciu wartości dodatnich odstępy między wierszami się zwiększają, a gdy użyjesz wartości ujemnych, tekst będzie się zagęszczał (rysunek 6.5).



Rysunek 6.5. Parametr `line-height` umożliwia dokładne określenie odstępu między wierszami

Kaskadowe arkusze stylów pozwalają również definiować odstępy pomiędzy poszczególnymi wyrazami tekstu przy wykorzystaniu parametru `word-spacing`. Również i w tym wypadku możliwe jest podanie wartości dodatnich (zwiększenie odstępu między wyrazami) i ujemnych (zmniejszenie odstępu między wyrazami) (rysunek 6.6).



Rysunek 6.6. Słowa zostały ściśnięte, ponieważ parametr `word-spacing` ma wartość ujemną

Twórcy kaskadowych arkuszy stylów poszli jeszcze o krok dalej i pozwolili projektantom stron wpływać nawet na odstępy pomiędzy literami w tekście. Efekt ten można uzyskać, korzystając z parametru `letter-spacing`.

Na przykład:

```
p {letter-spacing: 5pt}
```

Dekoracja tekstu

W kaskadowych arkuszach stylów twórcy stron mogą kontrolować wygląd tekstu za pomocą właściwości `text-decoration`. Korzystając z tego parametru, mogą sprawić, że tekst będzie:

- ▶ podkreślony — `text-decoration: underline`,
- ▶ przekreślony — `text-decoration: line-through`,
- ▶ nadkreślony — `text-decoration: overline`,
- ▶ migający — `text-decoration: blink`,
- ▶ bez dekoracji tekstu — `text-decoration: none`.

Korzystając z tego parametru, możesz na przykład wyłączyć podkreślenia wszystkich łączów na stronie:

```
A {text-decoration: none;}
```

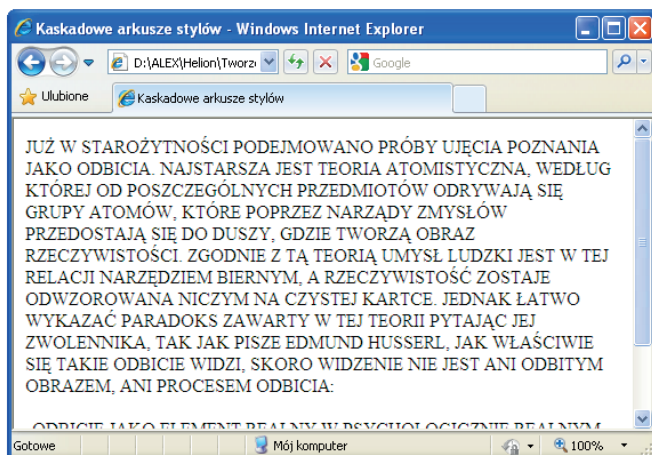
Małe i wielkie litery

Nie trudząc się zbytnio, możesz zamienić cały tekst na stronie na wielkie litery, chociaż oryginalnie w pliku jest on wpisany małymi literami. Na taką wygodę pozwala parametr `text-transform`. Może on przyjmować następujące wartości:

- ▶ tylko wielkie litery — `text-transform: uppercase` (rysunek 6.7),
- ▶ tylko małe litery — `text-transform: lowercase`,
- ▶ każda pierwsza litera w słowie jest zamieniana na wielką — `text-transform: capitalize`,
- ▶ bez zmian — `text-transform: none`.

Zamiast wstawiać do dokumentu twarde spacje, możesz użyć parametru `white-space`, który przyjmuje następujące wartości:

- ▶ `normal` — spacje są redukowane do jednej,
- ▶ `pre` — zachowanie dokładnej liczby spacji znajdujących się w tekście,
- ▶ `nowrap` — blokuje automatyczne łamanie wierszy.



Rysunek 6.7. Wszystkie litery zostały automatycznie zamienione na wielkie przez użycie parametru `text-transform: uppercase`

Formatowanie wyglądu czcionki

Zgodnie ze specyfikacją kaskadowych arkuszy stylów możliwe jest definiowanie następujących własności czcionki:

- ▶ rodzaju — atrybut `font-family`,
- ▶ rozmiaru — atrybut `font-size`,
- ▶ wagi — atrybut `font-weight`,
- ▶ stylu — atrybut `font-style`.

Pod pojęciem rodzaju czcionki kryje się rozróżnienie pomiędzy krojami czcionek lub rodzinami czcionek. Za pomocą kaskadowych arkuszy stylów możesz określić rodzaj czcionki, z którego chcesz korzystać przy wyświetlaniu wskazanego fragmentu tekstu. Konstrukcja stylu jest budowana według następującej definicji:

selektor {font-family: nazwa własna czcionki lub nazwa rodzajowa czcionki}

nazwa własna czcionki — np. Arial, Times New Roman, Verdana;

nazwa rodzajowa czcionki — jedna z pięciu wartości: serif, sans-serif, monospace, cursive lub fantasy.

Czcionki szeryfowe (serif) charakteryzują się tym, że każda litera posiada ozdóbki zwane szeryfami. Dodatkowo czcionki tego typu są proporcjonalne, gdyż każda litera ma inną szerokość. Typowym przedstawicielem tej grupy czcionek jest czcionka Times New Roman. Przykład czcionki szeryfowej (serif) przedstawiono na rysunku 6.8.


Rysunek 6.8.

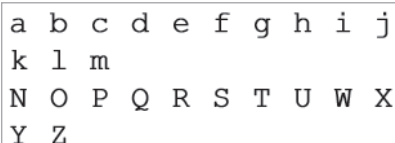
Czcionka szeryfowa

Czcionki bezszeryfowe nie mają żadnych ozdobników na czcionce, ale są one również proporcjonalne. Ten rodzaj czcionek jest najczęściej używany do tworzenia tekstu przeznaczonego do czytania (treść strony). Typowym przedstawicielem tej grupy czcionek jest czcionka Arial. Przykład czcionki bezszeryfowej (sans-serif) przedstawiono na rysunku 6.9.


Rysunek 6.9.

Czcionka bezszeryfowa

Czcionki tego typu są nieproporcjonalne, co oznacza, że każda litera ma taką samą szerokość. Ten rodzaj czcionek automatycznie kojarzy się z tekstem napisanym na maszynie do pisania. Typowym przedstawicielem tej grupy czcionek jest czcionka Courier New (rysunek 6.10).


Rysunek 6.10.

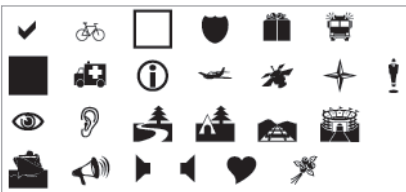
Przykład czcionki typu monospace

Czcionki typu cursive charakteryzują się dużym podobieństwem do pisma odręcznego, mogą też zawierać różnego rodzaju ozdobniki w postaci zawijasów. Przykładem czcionki typu cursive może być czcionka Comic Sans MS (rysunek 6.11).


Rysunek 6.11.

Przykład czcionki typu cursive

Czcionki typu fantasy to czcionki, których nie da się zakwalifikować do żadnego z opisanych powyżej rodzajów czcionek (rysunek 6.12).

**Rysunek 6.12.**

Przykład czcionki typu fantasy

**UWAGA**

Należy pamiętać, że czcionki zdefiniowane w arkuszu stylów są pobierane z komputera użytkownika przeglądającego stronę. Jeśli w systemie nie ma zainstalowanej czcionki określonej przez arkusz stylów, przeglądarka zastąpi ją domyślną czcionką dla danego systemu. Dlatego przy projektowaniu stron nie należy używać zbyt wymyślnych i rzadko spotykanych krojów czcionek. Warto trzymać się najpopularniejszych, jak Times, Arial czy Helvetica.

Za rozmiar czcionki użytej na stronie odpowiada parametr `font-size`. Rozmiar czcionki może być podany w jednostkach względnych lub bezwzględnych. Możesz w związku z tym używać następujących definicji rozmiarów czcionki:

- ▶ `font-size: 20 pt,`
- ▶ `font-size: 12 px,`
- ▶ `font-size: 200%.`

Ciekawym rozwiązaniem jest również możliwość korzystania ze zdefiniowanych wielkości: `xx-small`, `x-small`, `medium`, `large`, `x-large`, `xx-large`. Dodatkowo istnieje również możliwość określenia względnych rozmiarów za pomocą opcji `larger` oraz `smaller`. Na przykład:

```
body
{font-size: large;}
.wiekszy
{ font-size: larger;}
.mniejszy
{font-size: smaller;}
```

Pojęcie wagi czcionki odnosi się do jej grubości, a do definicji grubości czcionki używany jest atrybut `font-weight`. Wartość może być określona za pomocą liczb z przedziału od 100 do 900 oraz nazw `bold`, `bolder`, `lighter` i `normal`. Styl czcionki jest definiowany przez parametr `font-style`, który może przyjmować jedną z dwóch wartości — `italic` (kursywa) lub `oblique`.

Kontrola nad wyglądem list za pomocą kaskadowych arkuszy stylów polega na sterowaniu właściwościami użytej czcionki oraz określaniu typu listy, użytego punktora graficznego oraz definiowaniu pozycji elementów listy względem punktora.

Typy list dzielimy na punktowane, numerowane i definicje. Od typu listy uzależnione są rodzaje punktorów, których możemy użyć w liście. Dla list numerowanych punktarami mogą być litery (małe lub wielkie) oraz cyfry (arabskie lub rzymskie). Dla list punktowanych możemy użyć takich punktorów jak kwadrat, koło lub okrąg.

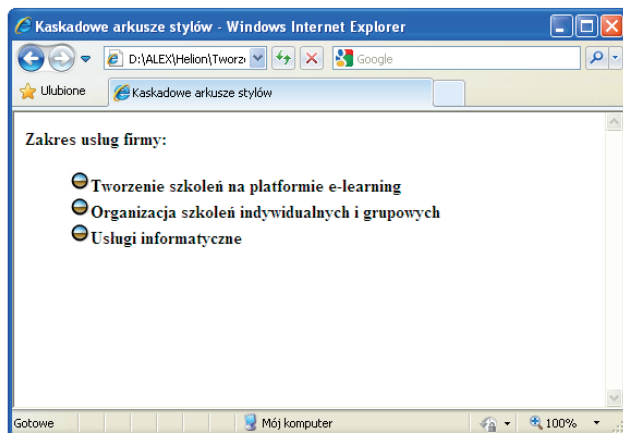
Typ listy określa właściwość `list-style-type`. Dla listy punktowanej może ona przyjmować wartości:

- ▶ `disc` — punktator ma postać koła,
- ▶ `circle` — punktator ma postać okręgu,
- ▶ `square` — punktator ma postać kwadratu.

Dla listy numerowanej właściwość `list-style-type` może przyjmować wartości:

- ▶ `decimal` — odpowiada liczbom arabskim,
- ▶ `lower-roman` — odpowiada małym liczbom rzymskim,
- ▶ `upper-roman` — odpowiada dużym liczbom rzymskim,
- ▶ `lower-alpha` — odpowiada małym literom,
- ▶ `upper-alpha` — odpowiada dużym literom,
- ▶ `none` — brak wypunktowania.

Czasem projektanci chcą zabłysnąć zupełnie nową i niestandardową koncepcją listy, i właśnie na takie okazje twórcy kaskadowych arkuszy stylów przygotowali parametr `list-style-image`, który pozwala użyć jako punktora listy wskazanego pliku graficznego w formacie GIF, JPG lub PNG. Należy jednak pamiętać, że obrazek nie powinien być zbyt duży, gdyż będzie psuł efekt listy (rysunek 6.13). Czasem może również wystąpić konieczność regulacji położenia tekstu względem punktora. Do tego celu służy parametr `list-style-position`.



Rysunek 6.13. Punktator w liście można zastąpić niewielkim plikiem graficznym

Parametr `list-style-position` przyjmuje następujące wartości:

- ▶ `outside` — punktator znajduje się wyraźnie poza listą,
- ▶ `inside` — punktator jest schowany w tekście listy.



Jeżeli lista zawiera krótkie wpisy w każdym punkcie, to różnica pomiędzy ustawieniami parametru `list-style-position` będzie dla użytkownika trudna do zauważenia. Stosowanie tego parametru jest zasadne, jeśli każda pozycja listy składa się z więcej niż jednego wiersza.

Kolor i tło

Każdy element strony, który możesz sformatować przez użycie kaskadowych arkuszy stylów, może mieć zdefiniowany kolor i tło. Jako tło elementy mogą mieć zdefiniowany kolor lub użycie wskazanego obrazu. Aby przypisać kolor do elementu strony, należy użyć parametru `color` i podać wartość koloru przez użycie jego zwyczajowej nazwy angielskiej, zapisu z palety rgb lub określenie wartości szesnastkowej koloru. Szczegółowe sposoby definiowania kolorów znajdziesz wcześniej w tym rozdziale.

Tło elementu możesz określić poprzez podanie koloru tła:

```
selektor {background-color: wartość;}
```

lub wskazanie obrazu, który ma zostać użyty jako tło obiektu:

```
selektor {background-image: URL(obraz.gif)}
```

Obraz użyty jako tło elementu strony musi być zapisany w formacie JPG, GIF lub PNG.

Domyślnie obraz użyty jako tło elementu jest powielany do rozmiaru elementu, jeśli jego rozmiar jest mniejszy niż rozmiar elementu, którego tło stanowi. Jednak za pomocą kaskadowych arkuszy stylów możliwe jest wprowadzenie kontroli nad powielaniem oraz zablokowanie przewijania tła.

Za powielanie obrazu w obrębie elementu, dla którego zdefiniowane jest tło będące obrazem, odpowiada parametr `background-repeat`, który może przyjmować następujące wartości:

- ▶ `no-repeat` — tło nie będzie powielane i zostanie wyświetlone tylko jeden raz,
- ▶ `repeat-x` — tło będzie powielane wyłącznie w poziomie,
- ▶ `repeat-y` — tło będzie powielane wyłącznie w pionie,
- ▶ `repeat` — tło będzie powielane w pionie i w poziomie.

Dodatkowo istnieje możliwość zatrzymania tła, aby nie przesuwało się wraz z zawartością całej strony. Do tego celu służy parametr `background-attachment` z ustawioną wartością `fixed`. Na przykład:

```
body  
{background-image: url(tlo.gif);
```

```
background-attachment: fixed;
}
```

Specyfikacja kaskadowych arkuszy stylów pozwala nie tylko zdefiniować sposób powtarzania obrazu umieszczonego w tle elementu, ale również pozwala określić pozycję obrazu względem okna przeglądarki lub jego lewego górnego rogu. Właściwość tę określa parametr `background-position`, który może przyjmować jedną z następujących wartości:

- ▶ `center` — tło wyśrodkowane,
- ▶ `left` — tło z lewej strony okna,
- ▶ `right` — tło z prawej strony okna,
- ▶ `top` — tło znajduje się na górze okna,
- ▶ `bottom` — tło znajduje się na dole okna,
- ▶ wartość liczbową — określa odległość tła od lewego górnego rogu okna przeglądarki.

Wartości parametrów można ze sobą łączyć, dzięki czemu można na przykład umieścić tło w prawym, górnym rogu okna przeglądarki, stosując zapis:

```
background-position: right top
```

Jeśli zdecydowałeś się na użycie wartości liczbowych do określenia położenia tła, możesz stosować wartości względne lub bezwzględne.

Szczegółowy opis wartości względnych i bezwzględnych stosowanych w kaskadowych arkuszach stylów znajdziesz wcześniej w tym rozdziale.

Marginesy

Podczas tworzenia stron WWW można definiować dwa rodzaje marginesów: zewnętrzne i wewnętrzne. Marginesy zewnętrzne strony to margines lewy, prawy, górny i dolny. Do określania szerokości marginesu należy użyć parametru `margin` w połączeniu z określeniem położenia marginesu:

- ▶ `margin-left` — lewy margines.
- ▶ `margin-right` — prawy margines,
- ▶ `margin-top` — górny margines,
- ▶ `margin-bottom` — dolny margines.

Parametr przyjmuje wartości liczbowe względne i bezwzględne. Marginesy są definiowane w sposób niezależny od siebie (górny może być inny niż dolny, a prawy może być inny niż lewy) i mogą przyjmować wartości ujemne.

Przykładowa definicja marginesów na stronie:



```
body
{
margin-left: 1 cm;
margin-right: 2 cm;
margin-top: 1 cm;
margin-bottom: 3 cm;
}
```

UWAGA

Pamiętaj, że marginesy różnych elementów się sumują. Jeśli na przykład ustawisz lewy margines dla elementu `<body>` na 2 cm, a następnie umieścisz na stronie tabelę, dla której ustawisz lewy margines na 2 cm, to lewa krawędź tabeli zostanie odsunięta od lewej krawędzi strony o 4 cm.

WSKAZÓWKA

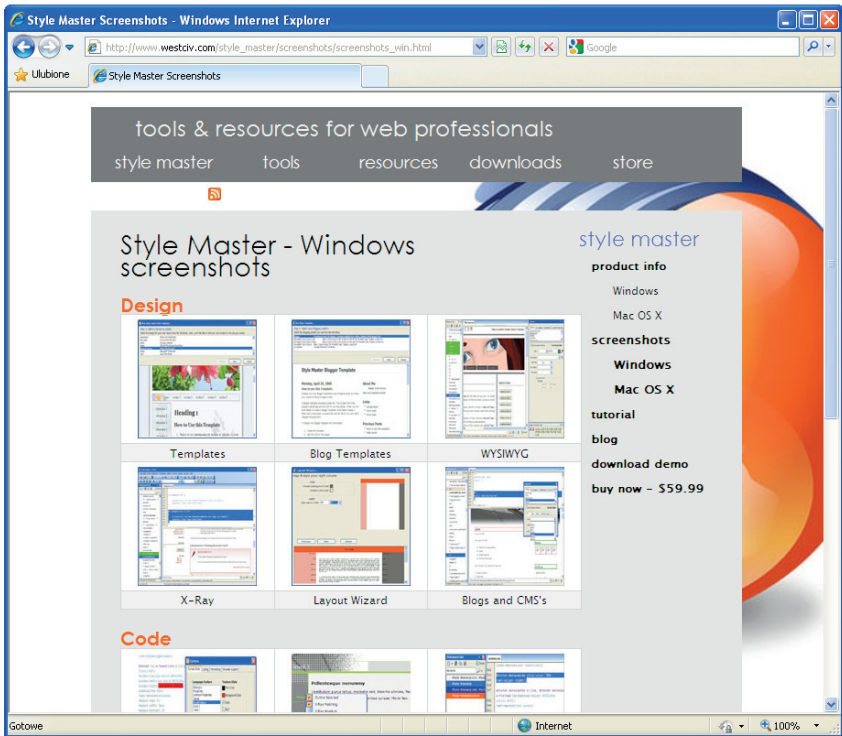
Jeśli chcesz ustawić identyczne wartości dla wszystkich marginesów, możesz to zrobić, używając wyłącznie parametru `margin`, na przykład:

```
body
{
margin: 12 pt
}
```

Margines wewnętrzny jest definiowany za pomocą parametru `padding` w połączeniu z określeniem miejsca na stronie, którego dotyczy margines. Określenie to należy podać w języku angielskim: `right` (prawy), `left` (lewy), `top` (górny) i `bottom` (dolny).

Programy do tworzenia arkuszy stylów

Mam dla Ciebie dobrą wiadomość — nie zawsze będziesz musiał tworzyć kaskadowe arkusze stylów samodzielnie, ponieważ istnieją programy, które zrobią za Ciebie „brudną robotę”. W takie funkcje wyposażone są dobre edytory WY-SIWYG (jak na przykład Dreamweaver, o czym mówiliśmy szczegółowo w rozdziale 2.), ale możesz również pobrać programy do tworzenia arkuszy stylów z sieci. Do takich programów należy Style Master, którego testową wersję możesz pobrać pod adresem http://www.westciv.com/style_master (rysunek 6.14).



Rysunek 6.14. Program Style Master pozwala w prosty sposób tworzyć kaskadowe arkusze stylów

Jeśli zainteresowała Cię tematyka kaskadowych arkuszy stylów, możesz pogłębić swoją wiedzę, odwiedzając witryny poświęcone tej tematyce:

- ▶ <http://www.csszengarden.com> (rysunek 6.15),
- ▶ <http://www.kurshtml.booo.pl/>,
- ▶ <http://www.w3schools.com/css/>.



Rysunek 6.15. W sieci WWW znajdziesz wiele witryn szczegółowo omawiających tematykę kaskadowych arkuszy stylów

Tworzenie stron WWW WYDANIE II

Projektować strony WWW może niemal każdy. Jednak tworzenie projektów wysokiej jakości to już spore wyzwanie. Internet jest pełen nieczytelnych, przeładowanych treścią albo grafiką witryn, które przynoszą swoim właścicielom więcej szkody niż pożytku. Technologia pędzi do przodu, podczas gdy niektórzy projektanci wyraźnie za nią nie nadążają. Jeśli chcesz mieć pewność, że efektem Twojej pracy będzie produkt najwyższej jakości, tak pod względem funkcjonalności, jak i estetyki, a także jeśli chcesz nauczyć się projektować witryny wyróżniające się spośród milionów przeciętnych półproduktów, nadszedł czas, by zacząć wreszcie działać z rozmachem.

W drugim wydaniu książki „Tworzenie stron WWW. Ilustrowany przewodnik” znajdziesz przydatne narzędzia ułatwiające pracę projektantom stron WWW. Dowiesz się, dlaczego warto używać Macromedia Dreamweaver oraz Adobe GoLive i czemu lepiej mieć kilka aplikacji, które robią to samo, niż jedną aplikację, która robi wszystko. Przeczytasz także o tym, czym różnią się formaty GIF, JPEG i PNG oraz którego z nich warto używać w jakiej sytuacji. Nauczysz się podstaw języków HTML i XHTML i odkryjesz, jak rozwinąć skrzydła w dziedzinie pisania skryptów, ponieważ poznasz bliżej język skryptowy JavaScript. W zakończeniu znajdziesz praktyczne porady dotyczące tego, co i w jakich proporcjach powinno znajdować się w doskonale zaprojektowanej witrynie.

ZASADY PISANIA I FORMATOWANIA TEKSTU ORAZ EDYTORZY STRON WWW
PRACA Z ELEMENTAMI GRAFICZNYMI PRZY UŻYCIU TAKICH
NARZĘDZI, JAK PHOTOSHOP I FIREWORKS

KASKADOWE ARKUSZE STYLÓW – FORMATOWANIE I PROGRAMY
DO TWORZENIA ARKUSZY

Tworzenie dynamicznych stron WWW, czyli JavaScript w akcji

Przygotowywanie i optymalizowanie animacji Flash

Język HTML – podstawy i struktura dokumentu

Projektuj pierwszorzędne witryny i twórz
nową jakość w Internecie



Cena: 24,90 zł

Zamów najnowszy katalog:

☎ <http://helion.pl/katalog>

Zamów informacje o nowościach:

☎ <http://helion.pl/newses>

Zamów cennik:

☎ <http://helion.pl/cennik>

Nr katalogowy: 5538



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



Zamówienia telefoniczne:
0 801 339900



0 601 339900



**Wydawnictwo
Helion**

ul. Kołłąszki 1c, 44-100 Gliwice
☎ 44-100 Gliwice, skr. poczt. 462
☎ 32 230 98 63
<http://helion.pl>
e-mail: helion@helion.pl

helion.pl
księgarnia
internetowa

ISBN 978-83-246-2621-2



Informatyka w najlepszym wydaniu