Zostań webmasterem!

Autor: Paweł Wimmer
ISBN: 83-87211-85-0
Format: B5, stron: 280
Zawiera CD-ROM



SPIS TREŚCI PRZYKŁADOWY ROZDZIAŁ

Nie musisz pozostać na zawsze jedynie biernym uczestnikiem społeczności internetowej.

Możesz zaistnieć jako jej aktywny współtwórca projektując własne witryny WWW w języku HTML!

Wystawienie w Sieci stron WWW wymaga kilku czynności: ich napisania, sprawdzenia poprawności i wysłania na serwer. W podręczniku zaprezentowane więc zostaną nie tylko język HTML, ale i rozmaite czynności i narzędzia webmasterskie.
Celem podręcznika jest zademonstrowanie całej procedury budowania serwisu WWW i prezentowania go we właściwy sposób w Internecie. Język HTML jest tylko składnikiem tej procedury – bardzo ważnym, ale nie jedynym. Takie praktyczne podejście wyrazi się zatem pokazaniem całej gamy działań i odpowiednich narzędzi.
Tak więc:

- opiszemy język HTML jako najważniejszy budulec stron WWW
- określimy podstawowy kanon czynności i narzędzi webmasterskich
- przedstawimy szczególnie wartościowe oprogramowanie, z uwzględnieniem potrzeb polskiego użytkownika
- pokażemy praktyczne sposoby umieszczania witryny w Sieci
- zamieścimy odpowiednie oprogramowanie na płycie, dzięki czemu czytelnik będzie mógł szybko przystąpić do pracy

Dzięki temu kompleksowemu podejściu czytelnik będzie mógł łatwo poznać wszystkie istotne elementy webmasterskiego puzzla. Nie będzie też musiał studiować najdrobniejszych szczegółów języka, gdyż nie są mu one na co dzień potrzebne.
Cały podręcznik jest podporządkowany naczelnej zasadzie prostoty i praktyczności.

Wstęp **8**

HTML w pigułce **10**

Czym jest dokument HTML? 10

Ramy dokumentu 11

Ciało dokumentu 12

Tytuły 12

Akapity 13

Wiersze 14

Pozioma linia 15

Grafika 15

Wykazy 16

Atrybuty czcionki 17

Wielkość i kolor czcionki 18

Odsyłacze 20

Część nagłówkowa dokumentu 21

Tytuł strony 22

Strona kodowa dokumentu 22

Struktura witryny 23

Problem schyłkowych znaczników 23

Koniec wieńczy dzieło 25

Struktura dokumentu **26**

HTML a sprawa polska 26

Podstawowe elementy dokumentu 27

Więcej szczegółów o Head 28

Więcej szczegółów o Body 32

Kilka uwag o wlewaniu tekstu 34

Czcionki **35**

Znaczniki fizyczne – atrybuty 36

Znaczniki logiczne – style 37

Kolory czcionki 37

Wielkość czcionki 38

Zmiana kroju czcionki 39

Elementy blokowe **44**

Specyfika bloku 44

Wyrównywanie bloków 45

Przełamanie wiersza i parametr clear 47

Pozioma linia 48

Blok preformatowany 50

Blok cytowany 51

Cytat w wierszu 51

Adres 52

Środkowanie 53

Komentarz 53

Wycinek dokumentu Div 53

Wycinek dokumentu Span 54

Obramowanie Fieldset 55

Wykazy **56**

Wykaz nieuporządkowany 56

Wykaz uporządkowany ... 58
Lista definicji ... 61
Odsyłacze ... 62
Istota odsyłaczy ... 62
Odsyłacze do etykiet ... 63
Odsyłacze do stron ... 64
Odsyłacze do innych witryn ... 65
Odsyłacze absolutne i względne ... 66
Ścieżki dostępu ... 66
Odwołania do innych dokumentów ... 68
Publikacje elektroniczne w formacie Adobe PDF ... 70
Odsyłacze do poczty elektronicznej ... 71
Odsyłacze do innych usług sieciowych ... 75
Odsyłacze do ramek ... 78
Mapy odsyłaczy na obrazkach ... 78
Grafika ... 81
Uwagi ogólne ... 81
Wstawianie grafiki do dokumentu ... 81
Ikona przy adresie ... 91
Multimedia ... 93
Stare sposoby zagnieżdżania multimediów ... 93
Plik RealAudio lub RealMedia ... 94
Plik Macromedia Flash ... 95
Plik Apple QuickTime ... 96
Nowe sposoby zagnieżdżania multimediów ... 96
Flash ... 98
QuickTime ... 98
RealMedia ... 99
Tabele ... 100
Szkielet tabeli ... 100
Obramowanie ... 101
Pozycjonowanie tabeli i jej zawartości ... 103
Kolory tabeli ... 106
Podpisy i nagłówki kolumn ... 108
Łączenie komórek i zagnieżdżanie tabel ... 110
Nagłówek, ciało i stopka tabeli ... 112
Sterowanie obramowaniem ... 113
Grupy kolumn ... 118
Ramki ... 120
Ogólna struktura strony ... 120
Informacje o Frameset ... 123
Informacje o Frame ... 124
Odsyłacze ... 125
Urozmaicenia ... 126
Ramki w kolumnach i wierszach ... 129
Informacje o Noframes ... 131

Pływające ramki 133
Podstawowa definicja 133
Urozmaicenia 134
Polecenie Object 136
Formularze 140
Ogólne informacje 140
Ramy formularza 141
Rodzaje pól formularza 142
Pola Input 143
Typ Text 143
Typ Checkbox 144
Typ Radio 145
Typ Submit i Reset 146
Pola Select 147
Pole Textarea 149
Podsumowanie 151
Przykład zbiorczy: 151
Znaki specjalne 155
Kolory 157
Style 158
Czym jest styl? 158
Budowa stylu 159
Wstawianie stylów 160
Kaskadowość stylów 163
Klasy selektorów 164
Własności czcionek 165
Własności tekstu 169
Kolor i tło 172
Marginesy 175
Obramowania 177
Wykazy 180
Suwaki 184
Kursory 187
Drukowanie 188
Jednostki miary 191
Selektory pseudoklas 192
Selektory pseudoelementow 193
Znacznik Div 195
Znacznik Span 196
Znacznik @import 196
Narzędzia webmasterskie 198
Kanon narzędzi 198
Polskie litery 200
Edytory HTML 202
WebPager Express 203
ezHTML 204
Website Pro 205

Porównanie edytorów ... 206
Style ... 210
TopStyle Lite ... 210
Edytory nagłówków ... 211
MetaWizard ... 211
Metty ... 212
Edytory kolorów dokumentu ... 213
HTMLcolor ... 213
ColorDetector ... 214
Cięcie i mapowanie grafiki ... 215
Map This! ... 215
MapMakerPro LITE ... 216
GeoHTML ... 217
CoffeeCup Free Viewer Plus ... 218
Sprawdzanie poprawności kodu ... 219
CSE HTML Validator Lite ... 220
TagCheck ... 221
Sprawdzanie poprawności odsyłaczy ... 222
CheckURL ... 222
Xenu Link Sleuth ... 223
LinkProver ... 224
Tworzenie grafiki ... 224
IrfanView ... 226
PhotoPlus ... 227
Flash ... 228
FlashToolSet ... 229
Pobieranie grafik z Internetu ... 230
Aaron's WebVacuum ... 231
Katalogowanie grafik ... 232
FotoAlbum ... 232
Tworzenie dźwięku ... 233
Audacity ... 234
dbPoweramp ... 234
Zakładanie witryny i wysyłanie stron ... 236
Onet.pl ... 236
Wirtualna Polska ... 241
Inne techniki wysyłania stron ... 244
SendToFTP ... 244
Web Publishing Wizard ... 246
FTPEdit ... 247
Własny serwer WWW ... 248
Promocja stron ... 248
MrKrax URL Submitter ... 248
Etykieta webmastera ... 250
Podstawowe przykazanie webmastera ... 250
Uwagi „ideologiczne” ... 251
Uwagi techniczne ... 252
I co dalej? ... 255
Spis ilustracji ... 261
Skorowidz ... 269

HTML w pigułce

Dobrym zwyczajem, spotykanym często w podręcznikach dla początkujących, jest przedstawienie opisywanych zagadnień w postaci krótkiego szkicu, który da pogląd na całość problematyki. W bieżącym rozdziale zaprezentujemy więc podstawowe informacje o budowaniu dokumentu HTML. Po uważnym przestudiowaniu treści tego rozdziału, co zajmie niewiele czasu, każdy będzie potrafił sporządzić prosty dokument, a nawet nazwać się webmasterem.

Pozostałe rozdziały będą stanowiły rozwinięcie tematyki. Warto je przeczytać po opanowaniu i przećwiczeniu informacji zawartych w tym rozdziale. Okaże się, że ich zrozumienie nie nastręczy większych trudności, gdyż będą logicznie wynikały z tego zasobu wiedzy, który czytelnik pozna w pierwszym rozdziale.

Bierzmy się zatem do pracy.

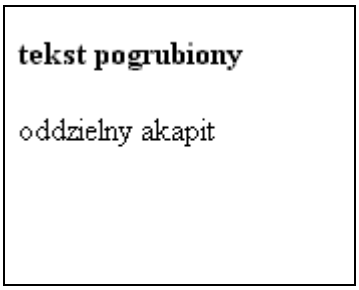
Czym jest dokument HTML?

Dokument HTML jest zwykłym plikiem tekstowym. Można go redagować w najprostszym edytorze tekstów – nawet tak prostym, jak znany wszystkim Notatnik w Windows. Oczywiście, programiści wymyślili specjalne edytory, które znakomicie ułatwiają redagowanie takich dokumentów, ale warto w tym miejscu podkreślić, że dokument HTML nie jest czymś tajemniczym i niezrozumiałym.

Zauważmy od razu, że plik z dokumentem ma charakterystyczne rozszerzenie .htm lub .html. To wiedzą chyba wszyscy. Gdy klikniemy podwójnie na takim pliku, od razu zgłosi się domyślna przeglądarka internetowa (np. Internet Explorer) i wczyta plik, prezentując jego zawartość na ekranie.

I w tym miejscu należy wyraźnie powiedzieć, że dokument HTML tworzony w edytorze oraz to, co zobaczymy w przeglądarce, są ściśle ze sobą powiązane, jak awers i rewers tej samej monety. Wynika to stąd, że dokument zawiera specjalne znaczniki, które są interpretowane przez przeglądarkę. Inaczej mówiąc, znaczniki są poleceniami dla przeglądarki, aby poszczególne fragmenty dokumentu wyświetliła w nakazany sposób. Redagowany przez nas dokument jest źródłem, które ulega przetworzeniu przez przeglądarkę.

Przykładowo, znacznik <b> powie przeglądarce, aby tekst następujący po nim wyświetliła w pogrubieniu (b=bold). Znacznik <p> nakaże przeglądarce wyświetlić następujący po nim tekst, jako kolejny akapit (p=paragraph).



Rysunek 1 - Przykład tekstu pogrubionego i akapitu

Wszystkie znaczniki tworzą język HTML. Zostały one skodyfikowane przez jedną z najważniejszych instytucji współczesnego świata – World Wide Web Consortium (http://www.w3.org). To grupa pracujących społecznie specjalistów, którzy w drodze dyskusji wypracowali obowiązujący standard języka HTML.

Jak więc widać, trzeba się nauczyć kilkunastu czy kilkudziesięciu znaczników, aby można było budować strony internetowe. Powiedzmy więcej – nie trzeba się ich wcale uczyć na pamięć. Wystarczy je rozumieć, a wyspecjalizowany edytor HTML ułatwi nam ich wstawianie. Nie trzeba będzie robić tego ręcznie.

Na początku polecamy jednak używanie Notatnika (czy jakiekolwiek innego edytora, który potrafi zapisywać pliki jako zwykły tekst) i wstawianie znaczników ręcznie. To znakomite ćwiczenie, które pozwoli lepiej „wyczuć" istotę znaczników.

WAŻNE: *w osobnym rozdziale powiemy więcej o narzędziach webmastera. Okazuje się, że w Sieci istnieje mnóstwo darmowych programów, które pozwolą nam tworzyć wszystkie niezbędne elementy strony WWW, nie wydając na to ani grosza. Wiele narzędzi zamieściliśmy na płycie CD dołączonej do podręcznika.*

Ramy dokumentu

Jak powiedzieliśmy, polecenie (znacznik, tag) HTML jest specjalnym ciągiem znaków objętym nawiasami ostrymi. Gdybyśmy, na przykład, chcieli zaznaczyć jakiś fragment tekstu jako tytuł, moglibyśmy objąć go poleceniem `<h1>treść tytułu</h1>`. Wielkość liter jest obojętna - znaczniki wpisane małymi i wielkimi literami są interpretowane w ten sam sposób. Staranniej wykonane edytory HTML pozwalają nawet zdefiniować, czy polecenia HTML będą wpisywane małymi, czy wielkimi literami. Polecamy stosowanie małych liter.

Polecenia powinny się znaleźć w pewnych standardowych ramach. Zostało przyjęte, że dokument HTML zawiera szkielet, który stanowi osnowę dla wszystkich innych poleceń. Spotyka się wprawdzie w Internecie strony, które szkieletu nie zawierają (czasem tylko jego część), ale odradzamy taką praktykę, gdyż w pewnych okolicznościach może ona zaowocować niespodziewanymi efektami.

Cały dokument powinien być objęty parą znaczników <html> </html>. Między nimi powinna zaś się znaleźć para znaczników <head> </head>, która stanowi ramy dla informacji nagłówkowych. Pozostałe informacje powinny być objęte z kolei znacznikami <body> </body>. Wygląda to następująco:

```
<html>
<head>
informacje nagłówkowe
</head>
<body>
właściwa treść (ciało) dokumentu
</body>
</html>
```

Tak właśnie wygląda dokument „z lotu ptaka". Wszystkie te zadziwiające rzeczy, które widzimy na wielu stronach w Internecie, są umieszczane właśnie w tych ramach. Przyjrzyjmy im się nieco bliżej.

Ciało dokumentu

Dla czytelnika naszej strony najważniejsze jest oczywiście ciało dokumentu, gdyż zawiera ono to wszystko, co zobaczymy w przeglądarce. Umieścimy w nim tekst podzielony na akapity, śródtytuły, tabele, obrazki, muzykę – wszystko, co nam tylko przyjdzie do głowy i na co pozwoli język HTML. Poszczególne elementy są umieszczane między znacznikami <body> </body>.

Tytuły

Mianem tytułów określamy specjalnie wyróżnione akapity, które nazywają w jakiś sposób pewne fragmenty tekstu. Niektórzy używają określenia „nagłówki", inni mówią „śródtytuły", a sens jest taki sam. Na potrzeby naszego podręcznika przyjmijmy określenie „tytuły". Mają one różny stopień, od 1 do 6.

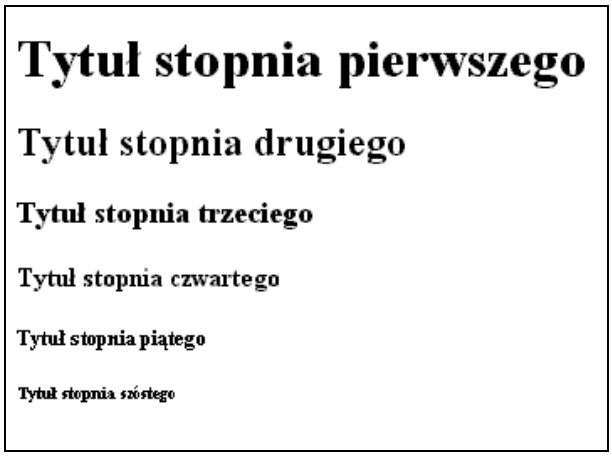
Aby wstawić tytuł stopnia pierwszego (najwyższy w hierarchii), należy podać polecenie:

```
<h1>Treść tytułu</h1>
```

Widzimy od razu, że <h1> rozpoczyna oznaczanie tekstu jako tytułu pierwszego stopnia, natomiast </h1> kończy. Większość znaczników ma taką właśnie konstrukcję. Zwróćmy uwagę na ukośnik wskazujący zamknięcie znacznika.

Analogicznie, tytuły niższych stopni tworzymy za pomocą h2, h3 itd.

I przykład:



Rysunek 2 - Tytuły sześciu stopni

Jak łatwo zauważyć, tytuły są doskonałym narzędziem do rozdzielania logicznych fragmentów tekstu i hierarchizowania ich. Zachodzi tutaj pełna analogia z książkami, gdzie spotykamy rozmaite stopnie śródtytułów, pozwalające wydzielić bloki tekstu i pokazać logikę całej pracy.

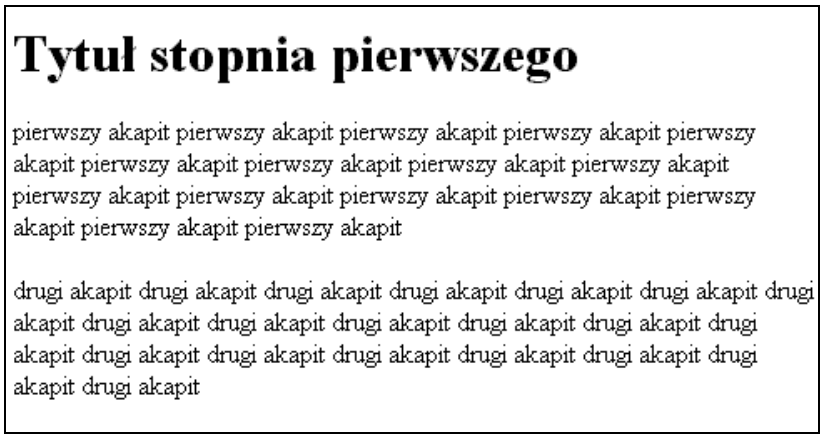
Akapity

Tytuły są nazwami fragmentów tekstu, ale najważniejsze są oczywiście akapity, gdyż to właśnie w nich zawarta jest główna treść naszego dokumentu.

Także i tutaj stosujemy znacznik wskazujący otwarcie akapitu i jego zamknięcie:

```
<p>Treść akapitu</p>
```

Zobaczmy ponownie, jak wyglądają akapity w przeglądarce.



Rysunek 3 - Tytuł i dwa akapity

Jak widać, oddzielne akapity są automatycznie rozdzielane. Co więcej, odległość między nimi jest nieco większa niż między wierszami w ramach jednego akapitu. Przeglądarka zinterpretowała tu w odpowiedni sposób następujący kod:

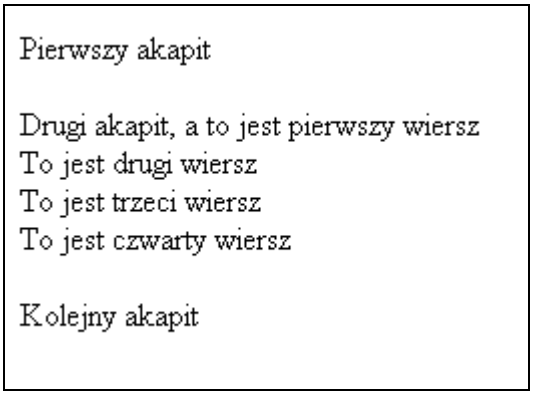
```
<h1>Tytuł stopnia pierwszego</h1>
<p>pierwszy akapit pierwszy akapit pierwszy akapit pierwszy
akapit pierwszy akapit pierwszy akapit pierwszy akapit
pierwszy akapit pierwszy akapit pierwszy akapit pierwszy
akapit pierwszy akapit pierwszy akapit pierwszy akapit
pierwszy akapit pierwszy akapit </p>
<p>drugi akapit drugi akapit drugi akapit drugi akapit drugi
akapit drugi akapit drugi akapit drugi akapit drugi akapit
drugi akapit drugi akapit drugi akapit drugi akapit drugi
akapit drugi akapit drugi akapit drugi akapit drugi akapit
drugi akapit drugi akapit </p>
```

Wiersze

Niekiedy zdarza się, że chcemy przełamać tekst, nie wprowadzając przy tym znaku końca akapitu. Możemy wówczas zastosować pojedyncze polecenie <br>, które przenosi tekst o jeden wiersz w dół, choć nie wprowadza dodatkowej interlinii. Nie stosuje się tutaj znacznika zamykającego.

```
To jest pierwszy wiersz<BR>
To jest drugi wiersz<BR>
To jest trzeci wiersz<BR>
To jest czwarty wiersz<BR>
```

Ilustracja pokaże różnicę między odstępami tworzonymi przez akapity i odstępami tworzonymi przez znaczniki przełamania wiersza.

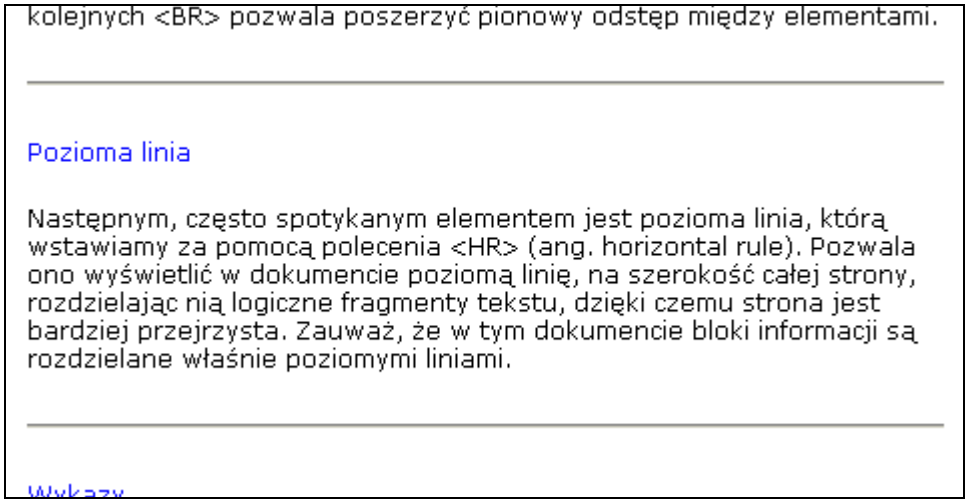


Rysunek 4 - Przełamanie wiersza

Ponadto polecenie <br> ma jeszcze jedną zaletę. Napisanie kilku kolejnych <br> pozwala zwiększyć pionowy odstęp między elementami.

Pozioma linia

Następnym, często spotykanym elementem jest pozioma linia, którą wstawiamy za pomocą polecenia `<hr>` (ang. horizontal rule). Pozwala ono wyświetlić w dokumencie poziomą linię, na szerokość całej strony, rozdzielając nią logiczne fragmenty tekstu, dzięki czemu strona jest bardziej przejrzysta. Ilustracja pokazuje oddzielenie tekstu od innych za pomocą dwóch poziomych linii.



Rysunek 5 - Poziome linie

Grafika

Nie sposób wyobrazić sobie stronę WWW bez grafiki. Może to być zwykły ozdobnik, ale także np. rodzinne zdjęcia, obrazki z wakacji, domowe zwierzaki czy cokolwiek innego. Coraz popularniejsze są cyfrowe aparaty fotograficzne, z których możemy bezpośrednio pobierać obrazki w internetowych formatach. Możemy też skanować zwykłe zdjęcia za pomocą skanera. Ponadto w Internecie znajdziemy ogromne zasoby darmowych grafik, które możemy wykorzystać na swoich stronach (obszerną listę adresów znajdzie Czytelnik na płycie CD).

W dokumentach HTML możemy umieszczać grafiki zapisane w formatach GIF, JPG lub PNG. Szczególnie popularne są te dwa pierwsze formaty. Aby w przeglądarce pojawił się obrazek, należy wpisać polecenie:

```
<img src="nazwa_obrazka">
```

Dla przykładu, wstawmy do dokumentu polecenie, które wyświetli zdjęcie zrobione doraźnie domowemu kotu za pomocą aparatu cyfrowego.

```
<img src="kot.jpg">
```

IMG oznacza image (obrazek), SRC to source (źródło), „kot.jpg” to oczywiście nazwa pliku z obrazkiem. Plik ten powinien być umieszczony w katalogu razem z dokumentem HTML. Dokument wyświetlany w przeglądarce odwołuje się do tego pliku i przywołuje zawartość we wskazanym miejscu.



Rysunek 6 - Ilustracja na stronie WWW

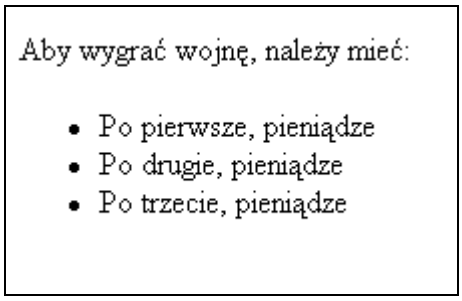
Wykazy

Na stronach WWW powszechnie stosuje się wykazy (zwane też listami). Są to po prostu systematyczne wyliczenia jakichś punktów, które w przeciwieństwie do tekstu o charakterze opisowym w skondensowany sposób prezentują jakąś myśl czy zespół zagadnień. Użytkownicy edytorów tekstów spotykają się z tym elementem - np. w polskiej wersji edytora Word są one znane pod nazwą list numerowanych i list wypunktowanych.

Aby utworzyć szkielet wykazu nieuporządkowanego, należy użyć polecenia `<ul> </ul>` (ang. unordered list). W ramach szkieletu znajdą się poszczególne punkty wykazu, które wprowadzamy za pomocą polecenia `<li>`.

```
<p>aby wygrać wojnę, należy mieć:<p>
<ul>
<li>po pierwsze, pieniądze</li>
<li> po drugie, pieniądze</li>
<li> po trzecie, pieniądze</li>
</ul>
```

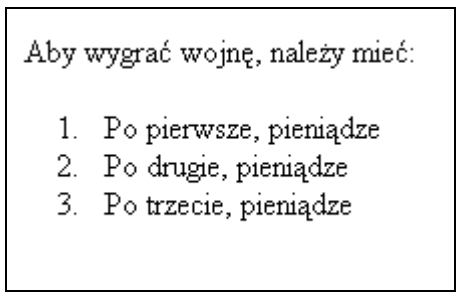
W efekcie uzyskamy:



Rysunek 7 - Wykaz nieuporządkowany

Jest w zasadzie obojętne, czy użyjemy pojedynczego znacznika `<li>`, czy też zastosujemy dodatkowo znacznik zamykający `</li>`. Autorzy edytorów HTML stosują różne konwencje, a wynik w przeglądarce WWW w obu przypadkach jest identyczny. Generalnie jednak zaleca się zamykanie znaczników.

Gdy chcemy utworzyć wykaz uporządkowany, wówczas zamiast `<ul>` i `</ul>` możemy zastosować `<ol>` i `</ol>` (ang. ordered list). Uzyskamy wówczas listę numerowaną za pomocą liczebników arabskich (w dalszej części podręcznika pokażemy inne sposoby numerowania).



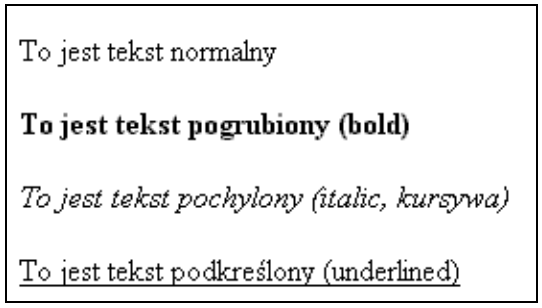
Rysunek 8 - Wykaz uporządkowany

Atrybuty czcionki

Aby tekst w przeglądarce nie wyglądał wszędzie tak samo, stosuje się dodatkowe atrybuty czcionki - pogrubienie (bold), pochylenie (italic) i podkreślenie (underlined). Atrybuty pozwalają uwypuklić jakieś fragmenty tekstu, np. nazwy za pomocą kursywy, a ważniejsze pojęcia za pomocą pogrubienia:

```
To jest tekst normalny
<B>To jest tekst pogrubiony (bold)</B>
<I>To jest tekst pochylony (italic, kursywa)</I>
<U>To jest tekst podkreślony (underlined)</U>
```

Efekt działania tych poleceń:



Rysunek 9 - Podstawowe atrybuty czcionki

Wielkość i kolor czcionki

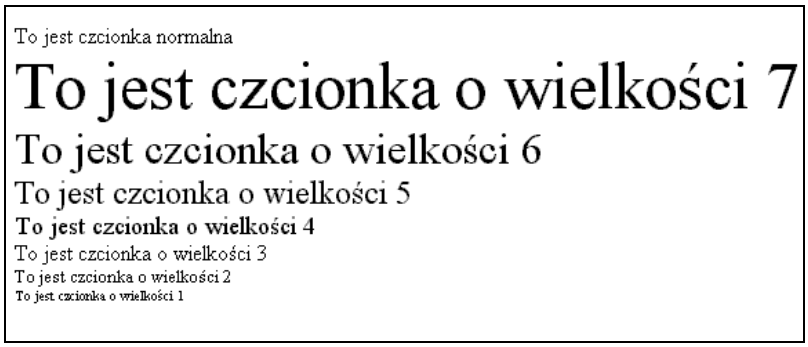
Gdyby autorowi strony znudził się tekst, w którym czcionka ma ten sam kolor i wielkość, zawsze może się posłużyć dodatkowymi poleceniami, zmieniającymi obie cechy. W dokumentach HTML stosuje się nieco inną definicję wielkości czcionki, niż przyjęta w edytorach tekstów. Podstawowa czcionka ma wielkość 3 jednostek (nie mylić z wielkością w punktach typograficznych) i jest zazwyczaj wyświetlana w przeglądarkach WWW jako czcionka 10-12-punktowa. Przeglądarki pozwalają na ogół zdefiniować, jaka wartość punktowa odpowiada czcionce o wielkości 3. W dokumencie możemy stosować czcionkę o wielkości od 1 do 7. Im większa wartość, tym większa czcionka w przeglądarce.

***Uwaga**: Wielkości czcionki w przeglądarce WWW nie należy utożsamiać z wielkością czcionki, za pomocą której redagujemy tekst w edytorze HTML! Czcionka w edytorze nie ma żadnego znaczenia dla postaci strony WWW. Możemy na przykład posługiwać się czcionką System, a i tak w przeglądarce pojawią się czcionki, które definiuje sam dokument lub przeglądarka.*

```
<font size="x">tekst objęty definicją</font>
```

x = 1...7

Widok w przeglądarce:



Rysunek 10 - Czcionki o różnej wielkości

W ten sposób możemy manipulować wysokością czcionki w tekście, uwypuklając jakieś fragmenty. Jest to wygodne, gdyż możemy zmienić wielkość czcionki wewnątrz akapitu, do czego nie nadaje się znacznik tytułu.

W analogiczny sposób możemy zmieniać kolor czcionki. Posługujemy się tutaj poleceniem:

```
<font color="kolor">tekst objęty poleceniem</font>
```

Kolory możemy definiować w postaci tradycyjnej, a więc za pomocą wartości szesnastkowych (np. kolor biały ma wartość #FFFFFF, a czarny #000000), albo za pomocą ich nazw. Wszystkie popularne przeglądarki akceptują wartości heksadecymalne (w formacie RRGGBB, gdzie RR - składowa czerwona, GG - składowa zielona, BB - składowa niebieska), aczkolwiek ich wprowadzanie wymaga wspomagania w edytorze HTML. Niektóre z edytorów pozwalają wybrać barwę za pomocą suwaków i wstawiają odpowiedni kod do edytora.

Internet Explorer i Netscape Navigator interpretują również nazwy kolorów. Początkowo można było się posłużyć jedną z 16 barw, potem paleta rozszerzyła się do 256 kolorów. Najczęściej jednak, przy definiowaniu koloru tekstu wystarczy 16 barw i można przyjąć, że posługiwanie się nazwami kolorów jest najwygodniejszą metodą ich definiowania.

Oto lista szesnastu podstawowych barw – ich nazwy wpisujemy w miejsce „kolor”.

```
black      silver     gray       white
maroon     red        purple     fuchsia
green      lime       olive      yellow
navy       blue       teal       aqua
```

Ich reprezentacja w przeglądarce internetowej (w czarno-białej książce tego nie widać, ale na płycie CD znajduje się przegląd kolorów):



Rysunek 11 - szesnaście podstawowych kolorów

Odsyłacze

Charakterystyczną cechą Internetu jest obecność na stronach hipertekstowych odsyłaczy do innych stron, które znajdują się w światowej sieci. W gruncie rzeczy czytelnik dokumentu nie musi wcale wiedzieć, gdzie znajdują się te strony - ważne jest, aby miał do nich łatwy dostęp. Na tym polega istota hipertekstu - jakiś temat może być opisany za pomocą stron tworzonych przez różnych autorów, mieszkających w różnych krajach, nawet często nie znających się wzajemnie. Gdy na zbudowanych przez nich stronach znajdą się odsyłacze, ich wspólne zainteresowania zostaną zjednoczone za pomocą hipertekstu, który połączy te strony w jedną logiczną całość. Czytelnik jakiegoś dokumentu będzie, po prostu za pomocą kliknięć myszką na odsyłaczach, przenosić się do różnych miejsc w Internecie, skacząc np. z Polski do Japonii, z Japonii do USA, a stamtąd do Francji. Z jego punktu widzenia jest zupełnie obojętne, gdzie strony się znajdują. Dla niego ważny jest łatwy dostęp do informacji.

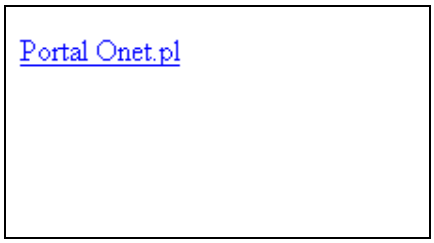
W tym miejscu przedstawimy dwa szczególnie ważne odsyłacze: odsyłacze do innych stron WWW i odsyłacz uruchamiający pocztę elektroniczną. Są one bowiem najpowszechniej stosowane.

Odsyłacz do innej strony jest tworzony za pomocą polecenia:

```
<a href="adres_strony_internetowej">Nazwa tej strony</a>
```

Przykładowy adres mógłby wyglądać następująco:

```
<a href="http://www.onet.pl">Portal Onet.pl</a>
```



Rysunek 12 - Typowy widok odsyłacza internetowego

Ukośnik (slash) na końcu adresu wymusza wczytanie strony domyślnej, zazwyczaj o nazwie index.html (lub index.htm - zależy to od typu serwera WWW).

***Uwaga**: zalecamy nadawanie stronie głównej serwisu nazwy index.htm lub index.html. Jest to powszechnie przyjęta praktyka w Internecie. W ostatnich latach stosuje się także niekiedy default.htm lub default.html.*

Na ekranie przeglądarki wyświetlany jest tekst, który znajduje się między znacznikami `<a href="... ">` i `</a>`. Jest on domyślnie wyświetlany na niebiesko z podkreśleniem (możemy to jednak zmienić). Gdy przesuniemy kursor myszki nad odsyłacz, w pasku stanu przeglądarki zobaczymy kryjący się pod tekstem adres internetowy. Kliknięcie na nim spowoduje skok do strony.

Zwróćmy uwagę na to, że adres internetowy jest podawany w cudzysłowie. Dla działania odsyłacza nie jest to wprawdzie absolutnie konieczne, ale wymaga tego formalna poprawność składni.

Drugim ważnym odsyłaczem jest adres poczty elektronicznej. Bardzo często autorzy stron, którzy przecież dysponują zazwyczaj swoim własnym adresem, wstawiają go na stronie, prosząc o uwagi czytelnika. Jest to dobry obyczaj, który daje czytelnikowi łatwy kontakt z autorem.

Stosujemy tutaj na przykład następujące polecenie:

```
<A HREF="mailto:autor@jego.adres">Imię i nazwisko</A>
```

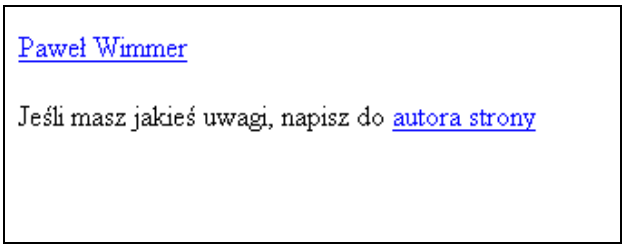
Na przykład:

```
<A HREF="mailto:prwimmer@poczta.onet.pl">Paweł Wimmer</A>
```

Napis `mailto` daje sygnał przeglądarce WWW, że ma uruchomić pocztę elektroniczną. Navigator wywoła swój własny moduł pocztowy Mail, Internet Explorer zaś wywoła Outlook Express (albo Outlook z pakietu MS Office, jeśli jest on domyślnym programem pocztowym). Także i tutaj konsekwentnie stosujmy cudzysłów w adresie. Fragment w postaci:

```
Jeśli masz jakieś uwagi, napisz do <A HREF="mailto:
prwimmer@poczta.onet.pl">autora strony</A>
```

zostanie przedstawiony w przeglądarce następująco:



Rysunek 13 - Przykład adresu poczty elektronicznej

Przesunięcie kursora myszki nad tekst "autora strony" wyświetli w pasku stanu adres poczty elektronicznej, natomiast kliknięcie myszką spowoduje uruchomienie modułu pocztowego, w którym czytelnik strony będzie mógł zredagować list do autora.

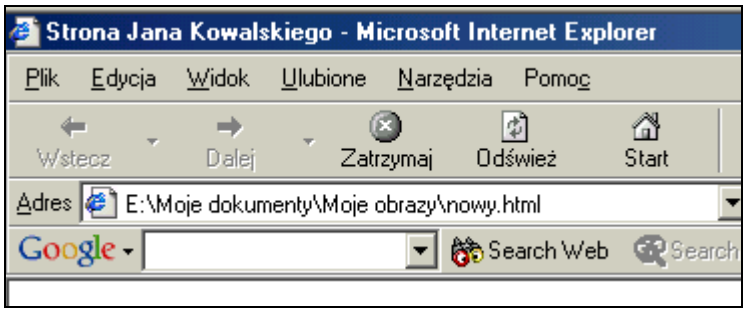
Część nagłówkowa dokumentu

W części nagłówkowej dokumentu można umieścić rozmaite informacje opisowe decydujące, jak nasza strona będzie wyświetlana, a także jak będą ją postrzegać sieciowe wyszukiwarki. Powiemy na ten temat więcej w innym rozdziale, a już teraz zwrócimy uwagę na te elementy, które przydadzą się początkującemu użytkownikowi.

Tytuł strony

Gdy otwieramy jakąś stronę w przeglądarce, w belce tytułowej widzimy zazwyczaj jej tytuł, np. **Strona Jana Kowalskiego**. Nie wolno zapominać o tym elemencie, gdyż pełni on istotną rolę informacyjną. Gdybyśmy o nim zapomnieli, w tym miejscu ukaże się jedynie nazwa fizycznego pliku i jego adres (na dysku lub w Internecie). W części nagłówkowej Head wstawiaj więc zawsze tytuł, na przykład:

```
<head>
<title>Strona Jana Kowalskiego</title>
</head>
```



Rysunek 14 - Tytuł strony

Tytułu strony nie należy mylić z tytułami (śródtytułami, nagłówkami) wewnątrz dokumentu, definiowanymi za pomocą H1, H2 itd.

Strona kodowa dokumentu

Właściwą stroną kodową dokumentu jest ISO-8859-2. Jest to międzynarodowy standard przyjęty przez stosowne ciała kodyfikujące. W Windows standardem jest strona kodowa Windows-1250. Powstaje zatem pytanie, co jest słuszne.

Oczywiście ISO, ale w początkowej fazie nauki sugerujemy stosowanie Notatnika jako edytora i wstawianie do nagłówka dokumentu deklaracji Windows-1250. Tak będzie poręczniej w pierwszych próbach budowania dokumentów HTML, gdyż w Notatniku wykorzystywany jest właśnie ten standard, podobnie jak w wielu innych programach dla Windows, które używamy na co dzień.

```
<head>
<meta http-equiv="content-type" content="text/html;
charset=windows-1250">
</head>
```

Deklarację strony kodowej wstawiaj do każdego dokumentu HTML!

Dopiero po wykonaniu szeregu ćwiczeń i przejściu do systematycznej nauki warto będzie sięgnąć po wyspecjalizowany edytor i stosować już stronę kodową ISO-8859-2. Ma ona postać:

```
<meta http-equiv="content-type" content="text/html;
charset=iso-8859-2">
```

W niezbyt odległej perspektywie wszyscy przejdziemy zapewne na standard Unicode, który w jednej tabeli zawiera znaki należące do wszystkich języków (dzięki temu można będzie łatwiej pisać wielojęzyczne dokumenty), ale na razie ograniczamy się do przyjętych powszechnie standardów.

Struktura witryny

Gdy stawiamy pierwsze kroki, tworzymy zazwyczaj pojedyncze dokumenty. Witryna internetowa składa się jednak na ogół z wielu powiązanych ze sobą dokumentów. Łączymy je za pomocą odsyłaczy, zaś główny dokument zapisujemy w pliku index.htm lub index.html. Inne mogą mieć dowolne nazwy, nawet wielowyrazowe. **Ale uwaga**, nie używaj w nazwach plików polskich liter i nie stosuj spacji. Tak więc nie twórz pliku „kącik humoru.htm", lecz „kacik_humoru.htm". W wielowyrazowych nazwach łącz wyrazy podkreśleniem.

Na dalszym etapie nauczysz się tworzenia wielu stron i łączenia ich ze sobą, ale na początku warto przećwiczyć tworzenie pojedynczych dokumentów.

***Bardzo ważna uwaga**: zapisuj zawsze pliki małymi literami. Nie Index.htm czy INDEX.HTM, lecz index.htm. Wynika to stąd, że serwery internetowe pracują zazwy-*

czaj pod kontrolą systemu operacyjnego Unix, który rozróżnia wielkość liter. Zatem pliki takie są dla nich różnymi plikami, choć pozornie nazywają się tak samo. Na lokalnym dysku nie ma to znaczenia, ale kłopot może się pojawić właśnie na serwerze. Konsekwentnie stosuj więc małe litery.

W rozdziale o narzędziach webmasterskich powiemy o sposobach posyłania plików na serwer. Jednak już teraz zapamiętaj, że struktura witryny na lokalnym dysku powinna być identyczna ze strukturą na serwerze. W przeciwnym razie pojawią się kłopoty z odsyłaczami.

Problem schyłkowych znaczników

World Wide Web Consortium uznało rozmaite znaczniki HTML, które pojawiały się we wcześniejszych wersjach języka, za schyłkowe (deprecated). Oznacza to, że organizacja zaleca sukcesywne rezygnowanie z ich stosowania w dokumentach HTML. Typowym przykładem mogą być znaczniki sterujące wyświetlaniem czcionki, jak wielkość, krój czy kolor.

Nie oznacza to oczywiście, że na skutek jakiejś arbitralnej decyzji tracimy nagle możliwość formatowania czcionek czy innych elementów. Starsze polecenia są po prostu zastępowane przez nowe – w tym konkretnym przypadku definicje czcionki powinny być po prostu zastąpione przez polecenia określone w specyfikacji stylów. Dla przykładu, zamiast „starej" definicji:

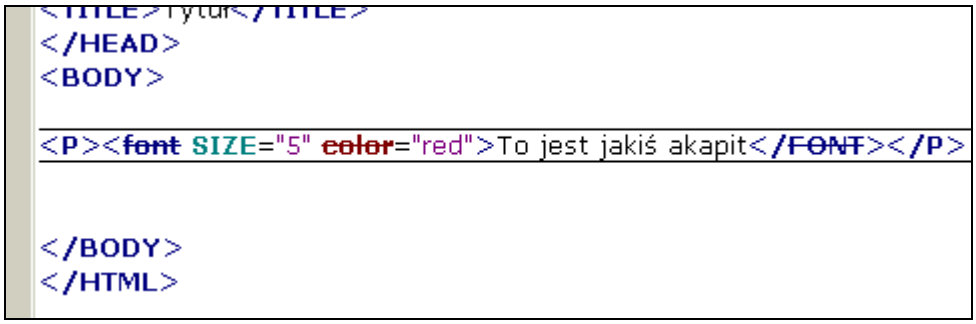
```
<p><font face="verdana" color="red">Treść akapitu</font></p>
```

możemy użyć nowej, stosując style:

```
<p style="font-family: verdana; color: red">To jest jakiś
akapit</p>
```

Efekt będzie oczywiście ten sam.

Warto od razu powiedzieć, że polski edytor Pajączek zawiera mechanizm wyświetlający „przestarzałe" znaczniki , dzięki czemu użytkownik może się od razu zorientować czy warto zastosować jakieś polecenie, czy też może rozważyć użycie nowocześniejszej techniki. Jeśli jakiś znacznik jest wyświetlany w edytorze z przekreśleniem, oznacza to właśnie, że ma status „deprecated".



Rysunek 15 - Pajączek zaznacza schyłkowe znaczniki

Narzuca się zatem pytanie, po co uczyć się starszych znaczników, skoro deprecjonuje je tak ważna instytucja jak W3C. Wyrażając tutaj jedynie swoją własną opinię, pragnę podkreślić następujące okoliczności, które przemawiają za starszymi znacznikami.

- Trudno sobie wyobrazić naukę języka HTML bez poznania tych intuicyjnie zrozumiałych i wygodnych w stosowaniu znaczników. Byłoby to tak, jak gdyby początkujący adept rachunku prawdopodobieństwa zaczynał poznawanie tej pasjonującej dyscypliny od aksjomatycznej definicji prawdopodobieństwa, a nie od prostych ćwiczeń z rzucaniem kostką do gry, które kiedyś natchnęły jakieś genialne umysły do stworzenia całej teorii.
- W wielu sytuacjach użycie starszych poleceń jest po prostu szybsze i wygodniejsze.
- Można być niemal pewnym (tak zresztą zaleca W3C), że przeglądarki internetowe będą zawsze interpretowały klasyczne znaczniki, gdyż odejście od tego oznaczałoby katastrofę w niezliczonych milionach dokumentów opublikowanych dotychczas w Sieci. Nie sposób przecież skłonić miliony osób do podjęcia jakiejś gremialnej akcji poprawiania stron zgodnie z nowymi zasadami. Póki co, nie mamy jeszcze żadnych automatycznie działających konwerterów, które w cudowny sposób zamieniłyby stare na nowe.

Zalecając oczywiście przenoszenie jak największej ilości informacji do stylów, bynajmniej nie warto straszyć webmasterów tym, że użyte tu i ówdzie polecenia `<font>`, `<u>` czy `<center>` nagle przestaną funkcjonować. Można się jedynie spodziewać, że powolutku, w ciągu wielu lat, coraz więcej osób będzie wykorzystywać nowocześniejsze techniki, zaś udział dokumentów sformatowanych w tradycyjny sposób będzie sukcesywnie maleć.

Polecenie	Zamiennik
<applet>	<object>
<basefont>	Arkusze stylów
	Align=center lub arkusze stylów
<dir>	
<font>	Arkusze stylów
<isindex>	<input>
<menu>	
<s>	Arkusze stylów
<strike>	Arkusze stylów
<u>	Arkusze stylów

***Uwaga**: Stosując polecenia schyłkowe, warto zarazem stosować prolog Transitional, dzięki któremu przeglądarka powinna wyświetlać nie tylko aktualne polecenia HTML 4, lecz i znaczniki ze statusem deprecated. Inna sprawa, że jak na razie przeglądarki są tolerancyjne i nie dyskryminują tych znaczników zależnie od użytego prologu.*

Dokładniejszą informację o prologu znajdzie Czytelnik w rozdziale Struktura dokumentu (punkt Podstawowe elementy dokumentu strona **Błąd! Nie zdefiniowano zakładki.**.).

Koniec wieńczy dzieło

Na zakończenie tego rozdziału mam dla wszystkich czytelników dobrą nowinę. Jesteście już webmasterami! Może jeszcze nie w każdym calu, ale pokazane wyżej techniki w pełni wystarczą do sporządzenia swojej internetowej wizytówki. Tak jak kiedyś umiejętność pisania i czytania stała się cywilizacyjnym wymogiem, tak dzisiaj umiejętność zapisywania informacji na wielkiej tablicy zwanej World Wide Web decyduje o przewadze nad tymi, którzy tej sprawności jeszcze nie posiedli.

Polecam zatem przećwiczenie opisanych technik – wypróbowanie rozmaitych układów elementów, zmianę ich treści – aby w pełni panować nad tworzeniem strony. Potem warto będzie przystąpić do studiowania następnych rozdziałów, które poszerzą naszą wiedzę o przydatne narzędzia.