

Tabele zazwyczaj kojarzą się z wypełnioną gęsto tekstem lub liczbami, podzieloną liniami na wiersze i kolumny kartką papieru; w dokumentach elektronicznych też najczęściej przyjmują taką postać. Na stronach WWW tabele zyskują jednak nowe znaczenie — choć można je wykorzystywać do zapisywania danych, to najczęściej służą jednak do definiowania układu strony — na przykład dzielenia strony na różnej szerokości kolumny wypełnione tekstem na wzór szpalt w gazecie.

Mechanizm tabel jest mocnym punktem języka XHTML — możliwe jest definiowanie rodzaju obramowania komórek tabeli (bądź zrezygnowanie z niego), ustalanie marginesów komórek tabeli, a nawet zagnieżdżanie tabel (tworzenie tabeli wewnątrz komórki innej tabeli).

Osiągnięcie mistrzostwa w operowaniu tabelami wymaga miesięcy ciężkiej pracy, jednak po przeczytaniu kilkunastu kolejnych stron powinieneś opanować podstawy tworzenia i modyfikowania tabel, które pozwolą Ci samodzielnie poszerzać swoją wiedzę. **Jak w języku XHTML opisuje się tabelę**

Kod opisujący tabelę zawarty jest wewnątrz elementu `<table>`. Niestety, olbrzymia elastyczność języka XHTML spowodowała, że nie wystarczy utworzyć element `<table>`, podać liczbę wierszy i kolumn tabeli i rozpocząć wprowadzanie danych. By stworzyć tabelę, konieczne jest stworzenie odrębnych elementów dla każdego wiersza i każdej komórki tabeli! Powoduje to, że tworzenie tabel jest jednym z najbardziej skomplikowanych i frustrujących zadań, przed którymi stają początkujący twórcy stron WWW.

Projektując tabele w języku XHTML, musisz cały czas pamiętać o podstawowej zasadzie: najważniejszym elementem tabeli jest wiersz, który z kolei podzielony jest na komórki. Wierszowi tabeli odpowiada element `<tr>`, zaś pojedynczej komórce — element `<th>` (w przypadku komórek pełniących funkcję nagłówka) oraz `<td>` (w przypadku zwykłych komórek danych). Elementy `<th>` oraz `<td>` muszą być zawarte w elemencie `<tr>`, a ten z kolei — w elemencie `<table>`. Struktura kodu tabeli wygląda zatem następująco:

```
<table>
```

```
<tr>
```

```
<td>
```

```
</td>
```

```
<td>
```

```
</td>
```

```
</tr>
```

```
<tr>
```

```
<td>
```

```
</td>
```

```
<td>
```

```
</td>
```

```
</tr>
```

Tabele

Dodał Administrator
czwartek, 22 lipiec 2010 08:59

</table>

Projektowanie tabeli najlepiej rozpocząć od... narysowania jej na kartce papieru. Zaufaj mi — pozwoli Ci to zaoszczędzić sporo czasu, gdyż pomoże w przełożeniu podziału na wiersze i komórki na prawidłowy kod XHTML tabeli. **Tworzenie tabeli**

Poniżej przedstawiłem kroki niezbędne do zbudowania najprostszej tabeli, zawierającej dwa wiersze po dwie komórki każdy. Jeden z wierszy będzie pełnić funkcję nagłówka tabeli, drugi zaś funkcję wiersza danych.

Wyróżnianie wiersza pełniącego funkcję nagłówka tabeli nie ma na celu wyłącznie zmiany jego wyglądu: ten sam efekt wizualny mógłbyś osiągnąć również, tworząc nagłówek w sposób identyczny jak zwykły wiersz danych. Jak już jednak podkreślałem w poprzednich rozdziałach, równie istotna jak wygląd dokumentu XHTML jest jego struktura logiczna.

Przygotuj sobie szkielet pustej strony WWW, a następnie wewnątrz elementu <body> wprowadź następujący kod XHTML:

```
<table>
```

```
<tr>
```

```
<th>Nagłówek 1</th>
```

```
<th>Nagłówek 2</th>
```

```
</tr>
```

```
<tr>
```

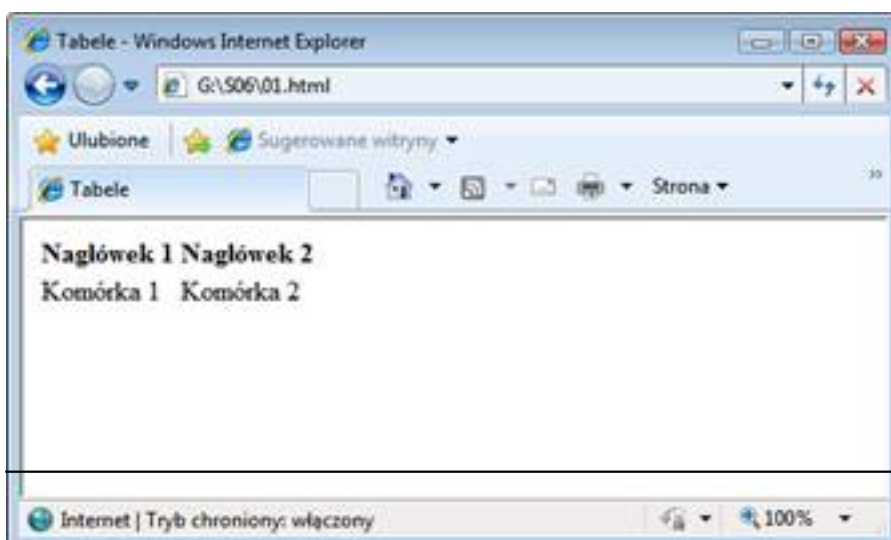
```
<td>Komórka 1</td>
```

```
<td>Komórka 2</td>
```

```
</tr>
```

```
</table>
```

Powyższy kod tworzy tabelę (element <table>) składającą się z dwóch wierszy (elementów <tr>). W pierwszym z nich znajdują się dwie komórki nagłówka (elementy <th>, rysowane po kolei od lewej do prawej), a w drugim dwie zwykłe komórki danych (elementy <td>). Ponieważ kolejne wiersze powielają zawsze wygląd pierwszego (szczególnie chodzi tu o szerokości poszczególnych komórek), obie komórki danych będą umieszczone idealnie poniżej komórek nagłówka, tworząc dwie kolumny (rysunek 6.1).



Rysunek 6.1. Prosta tabela

Tworząc tabele i wypełniając je treścią, pamiętaj, że element `<td>` koniecznie musi zawierać jakąś treść. Jeśli komórka tabeli ma pozostać pusta, wewnątrz elementu `<td>` (a więc pomiędzy znacznikami `<td>` i `</td>`) musi być zawarty znak twardej spacji (w kodzie XHTML odpowiada mu encja ` `). **Zmiana szerokości i**

pozycji tabeli

Przełęczarka WWW zawsze stara się nadać tabeli najmniejszą możliwą szerokość. O ile zatem w tabeli będzie dostatecznie mało danych — jak w powyższym przykładzie — cała tabela zajmie tylko kilka centymetrów szerokości okna. Jeżeli chcesz użyć tabeli w celu kształtowania wyglądu strony — na przykład podzielenia jej na obszary wypełniane różną treścią — może to zniweczyć Twe wysiłki.

Określenie szerokości całej tabeli i jej ewentualne wyśrodkowanie lub dosunięcie do prawej krawędzi okna nie jest na szczęście skomplikowane. Za szerokość odpowiada znany Ci już pewnie atrybut stylu `width`, a szerokość możesz podawać w pikselach (px), punktach (pt), centymetrach (cm) lub względem szerokości okna (%). Szczególnie ta ostatnia opcja może być przydatna, jeżeli Twoim celem jest utworzenie tabeli o szerokości całego okna przeglądarki:

```
<table style="width: 100%;">
```

```
...
```

```
</table>
```

Jeżeli spróbowałeś na własną rękę stworzyć tabelę węższą od okna przeglądarki i wycentrować ją lub dosunąć do brzegu ekranu za pomocą atrybutu stylu `text-align`, pewnie zauważyłeś, że nie reaguje ona na Twoje wysiłki. Aby zmienić wyrównanie tabeli, należy bowiem użyć atrybutu stylu `margin`:

```
-
```

aby wyśrodkować tabelę, jako wartości lewego i prawego marginesu należy podać `auto`,

```
-
```

aby dosunąć tabelę do prawej krawędzi, jako wartość lewego marginesu należy podać `auto`, a prawego — `0`.

Przy okazji możesz też określić szerokość górnego i dolnego marginesu tabeli. Oto przykład (rysunek 6.2):

```
<table style="border: 1px solid blue;">
```

```
<tr>
```

```
<td>Komórka 1</td>
```

```
<td>Komórka 2</td>
```

```
</tr>
```

```
</table>
```

Tabele

Dodał Administrator
czwartek, 22 lipiec 2010 08:59

```
<table style="border: 1px solid green; margin: 10pt auto 10pt auto;">
```

```
<tr>
```

```
<td>Komórka 1</td>
```

```
<td>Komórka 2</td>
```

```
</tr>
```

```
</table>
```

```
<table style="border: 1px solid red; margin: 10pt 0 10pt auto;">
```

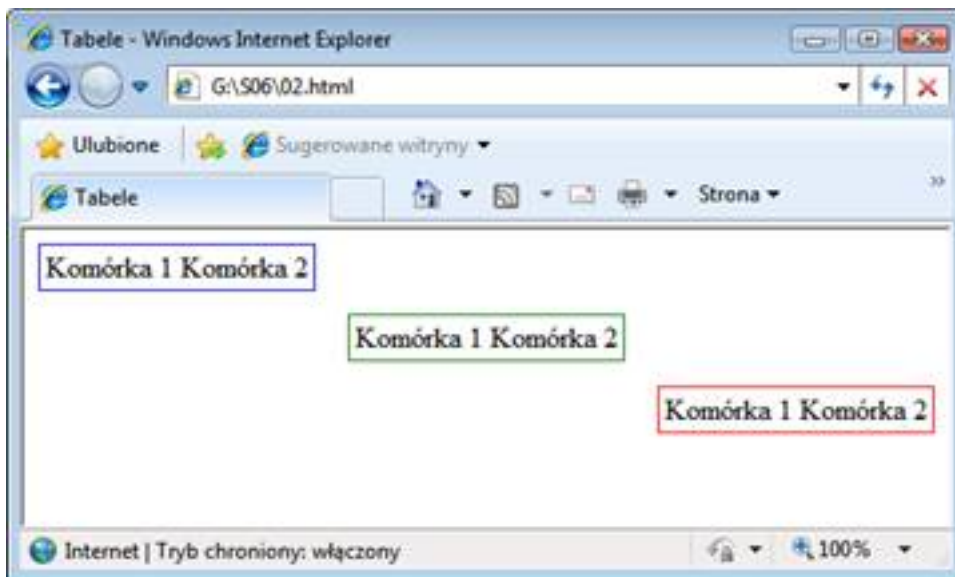
```
<tr>
```

```
<td>Komórka 1</td>
```

```
<td>Komórka 2</td>
```

```
</tr>
```

```
</table>
```



Rysunek 6.2. Trzy sposoby wyrównywania tabeli względem marginesów strony

Wskazówka

Podana powyżej metoda wyrównywania całej tabeli względem marginesów nie funkcjonuje poprawnie w wersjach przeglądarki Microsoft Internet Explorer starszych niż 7.0. W ich przypadku konieczne jest zawarcie tabeli w bloku `<div>` i narzucenie sposobu wyrównywania tekstu w ramach tego bloku za pomocą atrybutu stylu `text-align`. Jeżeli zależy Ci na zgodności ze starszą wersją przeglądarki Internet Explorer, najlepiej jest połączyć obydwie metody wyrównywania. **Zmiana szerokości kolumny tabeli**

Tabele

Dodał Administrator
czwartek, 22 lipiec 2010 08:59

Rzadko kiedy przeglądarka uda się we właściwy sposób dopasować szerokość kolumn tabeli do zawartej w jej komórkach treści. Czasem też nie może być mowy o automatycznym modyfikowaniu szerokości wybranych kolumn. Na szczęście język XHTML pozwala bardzo dokładnie kontrolować szerokość kolumn tabeli i określać ją w sposób dokładny — co do jednego piksela — lub względny.

Oczywiście służy do tego ten sam atrybut stylu `width`. Tym razem jednak musisz go zastosować nie do elementu `<table>` (w którym to przypadku modyfikował szerokość całej tabeli), lecz do elementu `<td>` lub `<th>`. Ponieważ pamiętasz pewnie, że wszystkie komórki w jednej kolumnie tabeli muszą mieć tę samą szerokość, wystarczy przypisać ten atrybut stylu do jednej komórki w całej kolumnie — najlepiej do komórki znajdującej się w pierwszym wierszu. Oto przykład (rysunek 6.3):

```
<table border="2">

<tr>

<th style="width: 3cm;">Kolumna 1</th>

<th style="width: 40%;">Kolumna 2</th>

<th>Kolumna 3</th>

<th style="width: 50pt;">Kolumna 4</th>

</tr>

<tr>

<td>Żaba</td>

<td>Jaszczurka</td>

<td>Kot</td>

<td>Kameleon</td>

</tr>

<tr>

<td>Morświn</td>

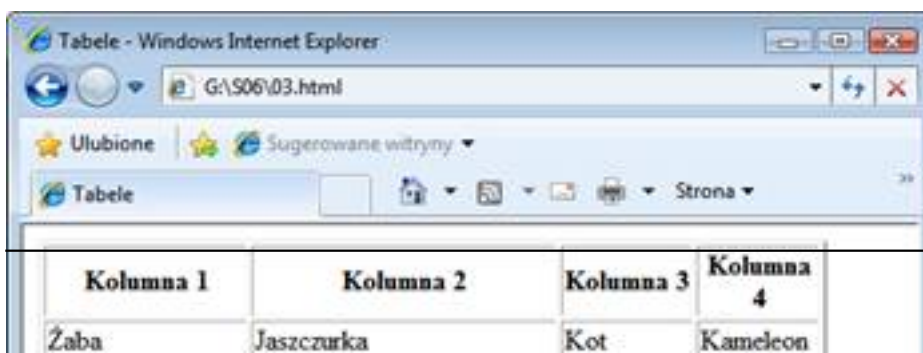
<td>Rybitwa</td>

<td>Szczupak</td>

<td>Okoń</td>

</tr>

</table>
```



Tabele

Dodał Administrator
czwartek, 22 lipiec 2010 08:59

Rysunek 6.3. Tabela z narzuconą szerokością poszczególnych kolumn

Kolumny, których szerokość nie zostanie ręcznie określona, nadal będą automatycznie rozszerzane lub zwężane przez przeglądarkę w zależności od ich zawartości oraz od szerokości pozostałych kolumn tabeli. Nie uda Ci się też sztucznie zwęzić kolumn zawierających długie słowa — minimalną szerokością kolumny będzie zawsze długość najdłuższego słowa zapisanego w jej komórkach, gdyż przeglądarki WWW nie potrafią automatycznie dzielić wyrazów.

Wskazówka

Aby oznakować miejsca, w których bardzo długie wyrazy **mogą** zostać podzielone w razie potrzeby, zastosuj encję ­. **Zmiana obramowania tabeli**

Na pewno zauważyłeś, że w poprzednich przykładach utworzyłem ramkę wokół tabeli. Nie jest to zadanie trudne i może być wykonane na dwa sposoby.

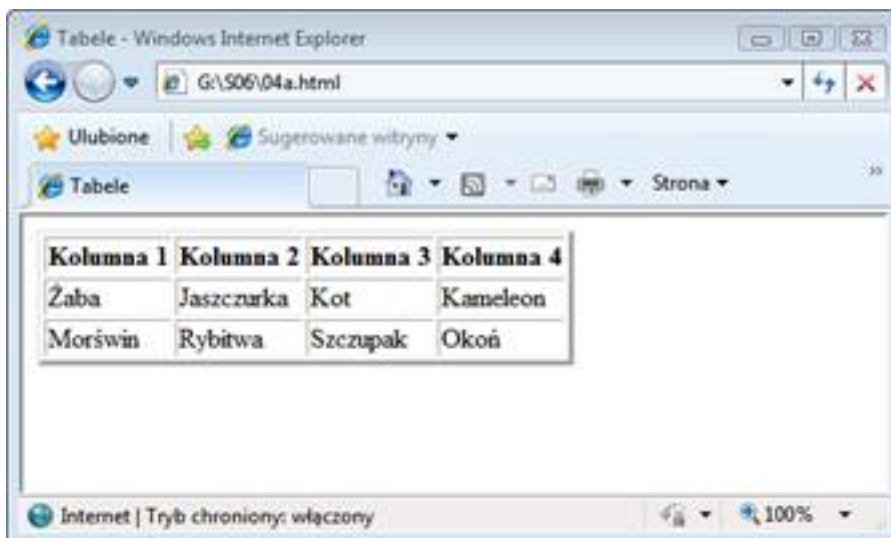
Pierwszy z nich wywodzi się z poprzednich wersji języka XHTML i umożliwia sterowanie obramowaniem wszystkich komórek tabeli naraz. Szerokość obramowania podaje się jako wartość atrybutu border, przy czym podanie zera jako szerokości usuwa obramowanie:

```
<table border="3">
```

...

```
</table>
```

Sposobu tego powinieneś używać wyłącznie przy projektowaniu tabel i diagnozowaniu ich układu. Stworzona w ten sposób ramka jest bardzo czytelna, lecz mało elegancka, a kod wykorzystujący atrybut border nie jest zgodny z najnowszymi normami (rysunek 6.4).



Rysunek 6.4. Obramowanie stworzone za pomocą atrybutu border jest funkcjonalne, lecz mało estetyczne

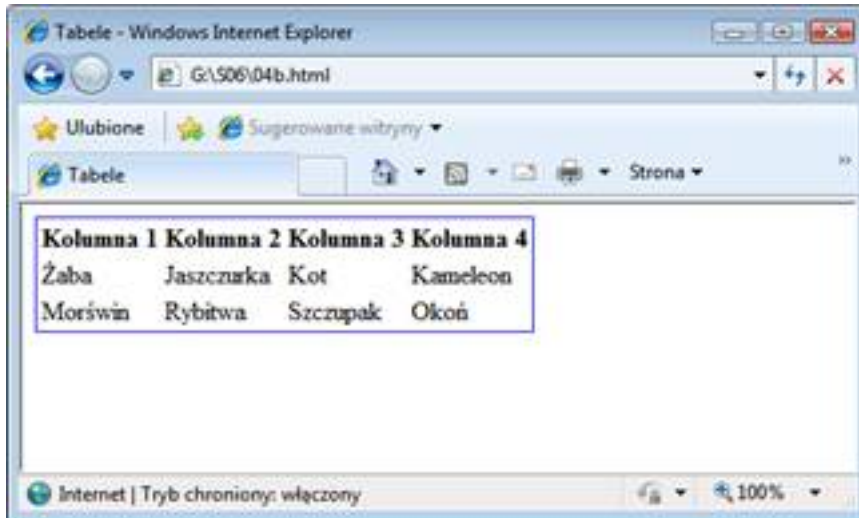
Tabele

Dodał Administrator
czwartek, 22 lipiec 2010 08:59

Drugi, ładniejszy sposób wykorzystuje atrybuty stylu CSS, a konkretniej znany Ci atrybut stylu border, którego pierwszy parametr określa szerokość obramowania, drugi — styl, a trzeci — kolor. Niestety, przypisanie go samemu elementowi <table>:

```
<table style="border: 1px solid blue;">
```

powoduje, że obramowaniem otaczana jest cała tabela, lecz nie poszczególne komórki (rysunek 6.5).



Rysunek 6.5. Atrybut stylu border zastosowany do elementu <table> wpływa wyłącznie na obramowanie otaczające całą tab

Aby uzyskać pożądany efekt, należy przypisać atrybut stylu border każdej komórce tabeli z osobna. Nagrodą za trud włożony w modyfikowanie kodu będzie możliwość dowolnego sterowania obramowaniem każdej komórki niezależnie (rysunek 6.6):

```
<table style="border: 1px solid blue;">  
<tr>  
<td style="border: 2pt solid green;">Żaba</td>  
<td style="border: 3pt dotted red;">Jaszczurka</td>  
<td style="border: 2pt dashed blue;">Kot perski</td>  
<td>Kameleon</td>  
</tr>  
</table>
```



Tabele

Dodał Administrator
czwartek, 22 lipiec 2010 08:59

Rysunek 6.6. Określenie obramowania poszczególnych komórek wymaga sporego nakładu pracy, jednak pozwala uzyskać ś

Najczęściej jednak autorowi stron WWW chodzi o to, by nowy rodzaj obramowania narzucić wszystkim komórkom tabeli lub przynajmniej grupom podobnych do siebie komórek. W przypadku dużych tabel — składających się z setek lub tysięcy komórek — osobne ustalanie wyglądu obramowania każdej z nich byłoby niezmiernie pracochłonne.

Na szczęście można wykorzystać mechanizm kaskadowych arkuszy stylu CSS (a dokładniej selektory dziedziczenia), by zautomatyzować pracę i przenieść powielanie definicji stylu na przeglądarkę. W pierwszym kroku należy usunąć wszelkie definicje stylu z kodu XHTML oraz nadać tabeli unikatowy identyfikator, korzystając z atrybutu `id` elementu `<table>`:

```
<table id="tabela1">
<tr>
<td>Żaba</td>
<td>Jaszczurka</td>
<td>Kot perski</td>
<td>Kameleon</td>
</tr>
</table>
```

W kolejnym kroku przygotuj obszar definicji stylu dokumentu (lub otwórz do edycji istniejącą definicję stylu, jeżeli dany plik XHTML już ją zawiera) i wprowadź opis obramowania tabeli o nadanym przed chwilą identyfikatorze:

```
table#tabela1 {
border: 1px solid blue;
}
```

Obramowanie tabeli powinno już być widoczne. Zmiana obramowania poszczególnych komórek — pozornie bardzo skomplikowana — będzie jednak równie prosta dzięki wykorzystaniu selektora dziedziczenia. Konieczne jest stworzenie definicji CSS opisującej styl wszystkich elementów `<td>` stworzonych w ramach elementu `<table>` o unikatowym identyfikatorze `tabela1` (rysunek 6.7):

```
table#tabela1 td {
border: 3px dotted red;
}
```



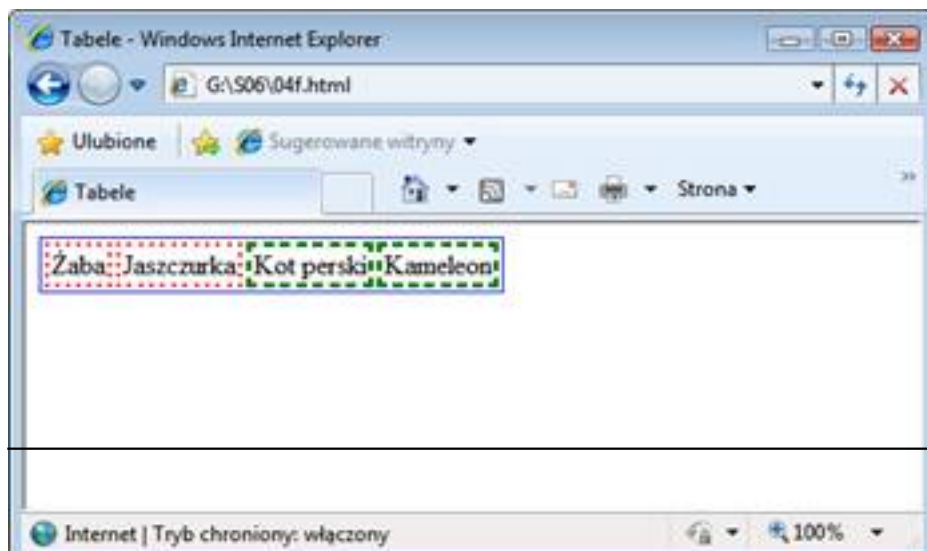
Tabele

Dodał Administrator
czwartek, 22 lipiec 2010 08:59

Rysunek 6.7. Jeżeli wszystkie komórki tabeli mają dysponować identycznym obramowaniem, warto użyć selektorów dziedziczenia

Oczywiście jeżeli różne grupy komórek tabeli mają dysponować różnym obramowaniem, możesz stworzyć klasy stylu elementu `<td>` i przypisać im różne definicje stylu (rysunek 6.8):

```
table#tabela1 {  
    border: 1px solid blue;  
}  
  
table#tabela1 td.czerwona_ramka {  
    border: 3px dotted red;  
}  
  
table#tabela1 td.zielona_ramka {  
    border: 3px dashed green;  
}  
...  
<table id="tabela1">  
    <tr>  
        <td class="czerwona_ramka">Żaba</td>  
        <td class="czerwona_ramka">Jaszczurka</td>  
        <td class="zielona_ramka">Kot perski</td>  
        <td class="zielona_ramka">Kameleon</td>  
    </tr>  
</table>
```



Tabele

Dodał Administrator
czwartek, 22 lipiec 2010 08:59

Rysunek 6.8. Korzystając z klas stylu, można zróżnicować wygląd poszczególnych komórek tabeli, nie rezygnując jednak ze

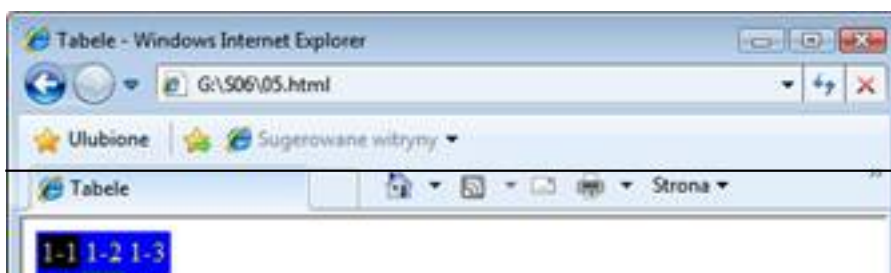
Wskazówka

Aby scalić ze sobą obramowania poszczególnych komórek i zlikwidować odstępy między nimi, nadaj elementowi `<table>` atrybut `border-collapse` o wartości `collapse`. **Zmiana koloru tła i tekstu tabeli**

Zmiana koloru tła tabeli i jej komórek — realizowana tylko i wyłącznie za pomocą stylów CSS — przebiega w sposób analogiczny do zmiany obramowania tabeli, tyle że wykorzystuje się atrybuty z rodziny `background` (głównie `background-color`), a zmiana koloru tła tabeli automatycznie zmienia kolor tła wszystkich komórek, których barwy tła nie określiłeś. Nie muszę chyba podkreślać, że bardzo upraszcza to kompozycję wielobarwnych tabel.

Oto przykład tabeli o trzech kolumnach i trzech wierszach, w której trzem komórkom leżącym na przekątnej nadane zostały odmienne barwy tła niż całej reszcie tabeli (rysunek 6.9):

```
<table style="border: none; color: yellow; background-color: blue;">
<tr>
<td style="background-color: black; color: silver;">1-1</td>
<td>1-2</td>
<td>1-3</td>
</tr>
<tr>
<td>2-1</td>
<td style="background-color: green; color: cyan;">2-2</td>
<td>2-3</td>
</tr>
<tr>
<td>3-1</td>
<td>3-2</td>
<td style="background-color: red; color: white;">3-3</td>
</tr>
</table>
```



Rysunek 6.9. Efekt zmiany koloru tła całej tabeli i poszczególnych jej komórek

Zauważ, że użyłem też atrybutu stylu `color`, aby określić kolor tekstu zawartego wewnątrz komórek tabeli. Działa on w sposób opisany już w tym kursie w części poświęconej formatowaniu akapitów tekstu oraz stylom CSS. Zmieniając kolor tła tabeli, zawsze warto dobrać odpowiednio kontrastowy kolor tekstu — czarny tekst na niebieskim tle będzie ledwie czytelny dla większości użytkowników Twojej strony WWW, a dla daltonistów najprawdopodobniej absolutnie niewidoczny.

W przypadku gdyby ten sam zestaw barw miał zostać użyty w wielu komórkach, powtarzanie definicji stylu wewnątrz każdego wyróżnionego elementu `<td>` byłoby pracochłonne i niepraktyczne. Najprostszym rozwiązaniem jest stworzenie klas stylu elementów `<td>` i przypisanie im zestawu atrybutów gwarantującego pożądany wygląd komórek. Jeszcze lepszym sposobem na poradzenie sobie z tym problemem — likwidującym ryzyko powstania konfliktów nazw klas między kilkoma tabelami — jest połączenie mechanizmu klas stylu z selektorami dziedziczenia opisanymi w poprzednim punkcie. **Zmiana marginesów komórek tabeli**

Jeżeli przyjrzyj się dokładnie rysunkowi zamieszczonemu w poprzednim ćwiczeniu, zauważysz pewnie, że wokół komórek tabeli rysowana jest ramka koloru tła całej tabeli. Spowodowane jest to standardowymi marginesami komórek: komórki są odsunięte od siebie, a pomiędzy nich wyziera tło tabeli. W języku XHTML można jednak sterować prawie każdym aspektem wyglądu strony, zatem i te marginesy możesz zmienić.

Za domyślne marginesy wszystkich komórek tabeli odpowiadają atrybuty `cellpadding` i `cellspacing` elementu `<table>`. Pierwszy określa szerokość marginesu wewnątrz komórki — czyli o ile ma być odsunięta zawartość komórki (tekst, obraz) od jej krawędzi. Drugi z kolei określa szerokość marginesu wokół komórki — czyli o ile komórki mają być odsunięte od siebie. Właśnie nadanie wartości 0 temu drugiemu — `cellspacing` — spowoduje zlikwidowanie „obramowania” powstałego z tła tabeli.

Oto kod czterech przykładowych tabel. Każda z nich składa się tylko z jednej komórki, ale wyraźnie możesz dostrzec sposób, w jaki działają dwa opisane powyżej atrybuty (rysunek 6.10):

```
<table style="background-color: blue;" cellpadding="5" cellspacing="5">
<tr>
  <td style="background-color: yellow;">CellPadding=5, CellSpacing=5</td>
</tr>
</table>
```

```
<br />
```

```
<table style="background-color: blue;" cellpadding="0" cellspacing="5">
<tr>
  <td style="background-color: yellow;"><b>CellPadding=0</b>, CellSpacing=5</td>
</tr>
</table>
```

Tabele

Dodał Administrator
czwartek, 22 lipiec 2010 08:59

```
<br />
```

```
<table style="background-color: blue;" cellpadding="5" cellspacing="0">
```

```
<tr>
```

```
<td style="background-color: yellow;">CellPadding=5, <b>CellSpacing=0</b></td>
```

```
</tr>
```

```
</table>
```

```
<br />
```

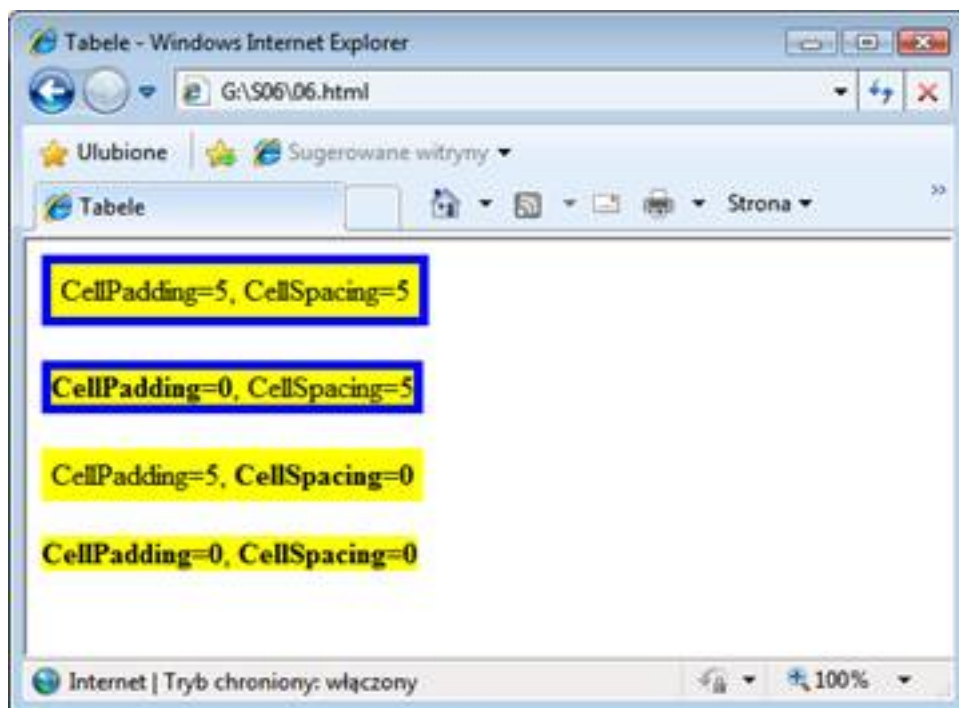
```
<table style="background-color: blue;" cellpadding="0" cellspacing="0">
```

```
<tr>
```

```
<td style="background-color: yellow;"><b>CellPadding=0</b>, <b>CellSpacing=0</b></td>
```

```
</tr>
```

```
</table>
```



Rysunek 6.10. Efekty uzyskane dla różnych kombinacji wartości atrybutów cellspacing i cellpadding

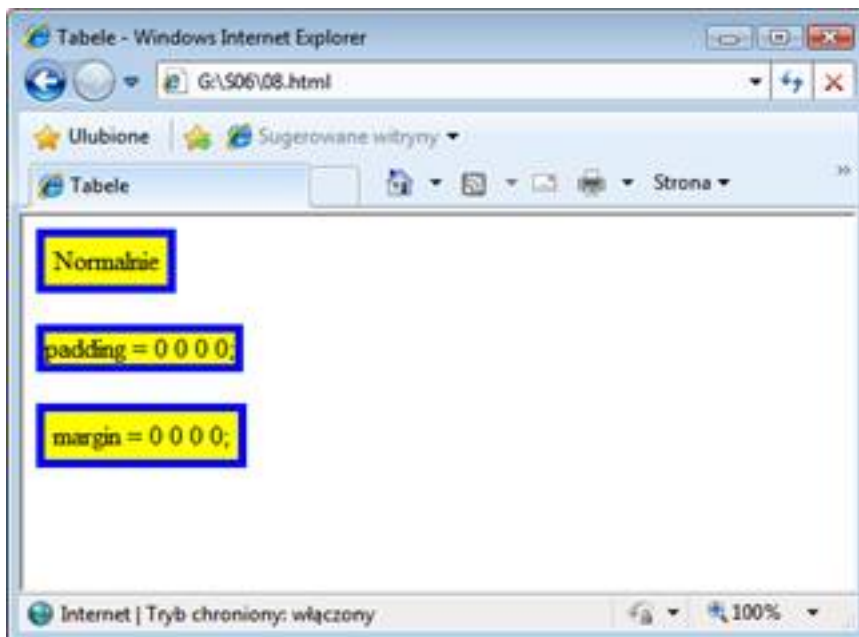
Atrybuty cellspacing i cellpadding określają szerokości marginesów wspólnie dla wszystkich komórek tabeli, ale w razie potrzeby możesz też sterować szerokością marginesów wewnętrznych pojedynczych komórek. Służy do tego znany Ci już atrybut stylu padding. Niestety, szerokości marginesów zewnętrznych nie zmienisz — pozostaje Ci usuwanie ich na poziomie globalnym za pomocą atrybutu cellspacing="0" lub atrybutu stylu border-collapse:

Tabele

Dodał Administrator
czwartek, 22 lipiec 2010 08:59

collapse; (rysunek 6.11):

```
<table style="background-color: blue;" cellpadding="5" cellspacing="5">  
  <tr><td style="background-color: yellow;">Normalnie</td></tr>  
</table>  
  
<br />  
  
<table style="background-color: blue;" cellpadding="5" cellspacing="5">  
  <tr><td style="background-color: yellow; padding: 0 0 0 0;">padding = 0 0 0 0;</td></tr>  
</table>  
  
<br />  
  
<table style="background-color: blue;" cellpadding="5" cellspacing="5">  
  <tr><td style="background-color: yellow; margin: 0 0 0 0;">margin = 0 0 0 0;</td></tr>  
</table>
```



Rysunek 6.11. Wpływ atrybutów stylu margin i padding na marginesy komórek tabeli

Rozbudowywanie tabeli

Wstawienie nowego wiersza tabeli jest operacją banalną — wystarczy w odpowiednim miejscu kodu wstawić nowy element `<tr>` i uzupełnić go odpowiednią liczbą elementów `<td>` lub `<th>`. Musisz jedynie pamiętać, by w nowo wstawianym wierszu tabeli liczba komórek była identyczna z liczbą komórek znajdujących się w pozostałych wierszach.

Tabele

Dodał Administrator
czwartek, 22 lipiec 2010 08:59

Niestety, struktura kodu tabeli w języku XHTML — tabela opisana jest wiersz po wierszu, zaś każdy wiersz zawiera identyczną liczbę komórek — powoduje, że rozszerzenie tabeli o dodatkową kolumnę jest znacznie bardziej skomplikowane i pracochłonne. Wstawienie nowej kolumny polega na dołożeniu nowych elementów `<td>` lub `<th>` do każdego istniejącego wiersza tabeli.

Jeśli po rozbudowaniu kodu tabeli o nowy wiersz lub kolumnę tabela dosłownie „rozpada się”, mieszając zawartość komórek lub kończąc się przedwcześnie, na pewno popełniłeś jakiś błąd w kodzie. W takim przypadku jeszcze raz sprawdź, czy zamknąłeś we właściwej kolejności wszystkie elementy języka XHTML składające się na kod tabeli oraz czy wszystkie wiersze tabeli zawierają tyle samo komórek (czyli elementów `<td>` oraz `<th>`). **Scalanie**

komórek tabeli

Jeśli tworzyłeś kiedyś za pomocą edytora tekstów lub arkusza kalkulacyjnego tabelę o skomplikowanym układzie, zetknąłeś się pewnie z funkcją scalania komórek tabeli, umożliwiającą utworzenie wspólnego nagłówka obejmującego swoim zasięgiem kilka wierszy lub kolumn tabeli. Scalanie komórek tabeli jest również dostępne w języku XHTML; można wręcz powiedzieć, że jego rola jest znacznie większa, gdyż tabele umieszczane na stronie WWW służą nie tylko tabelaryzowaniu danych, ale również tworzeniu układu strony oraz rozmieszczaniu jej elementów — takich jak fragmenty tekstu czy osadzone na stronie obrazy. Niestety, tworzenie kodu XHTML odpowiedzialnego za układ tabeli zawierającej scalone komórki jest dość trudne.

Podstawowa trudność przy tworzeniu kodu scalonych komórek polega na tym, że scalanie wymaga złamania zasady obecności takiej samej liczby komórek w każdym wierszu tabeli. Dwie scalone ze sobą komórki opisywane będą w kodzie przez tylko jeden element `<td>`, co znacznie utrudnia weryfikację poprawności kodu tabeli.

Scalanie komórek tabeli w poziomie

Scalanie kilku leżących obok siebie komórek tabeli w poziomie jest względnie łatwe — należy tylko usunąć zbędne elementy `<th>` i `<td>`, zaś kod elementu tworzącego dużą, scaloną komórkę uzupełnić atrybutem `colspan` informującym, jako ile komórek ma być traktowana scalona komórka. Na przykład jeżeli chcesz scalić ze sobą trzy komórki:

```
<td>...</td> <td>...</td> <td>...</td>
```

wynikowy kod będzie miał następującą postać:

```
<td colspan="3">...</td>
```

ponieważ ten jeden element `<td>` ma być traktowany jako trzy komórki. Oto przykład, w którym w drugiej z dwóch identycznych tabel trzy komórki w środkowym wierszu zastąpiłem jedną scaloną, rozciągniętą na powierzchni trzech (rysunek 6.12):

```
<table border="1">
<tr>
<td>1-1</td> <td>1-2</td> <td>1-3</td> <td>1-4</td>
</tr>
<tr>
<td>2-1</td> <td>2-2</td> <td>2-3</td> <td>2-4</td>
</tr>
```

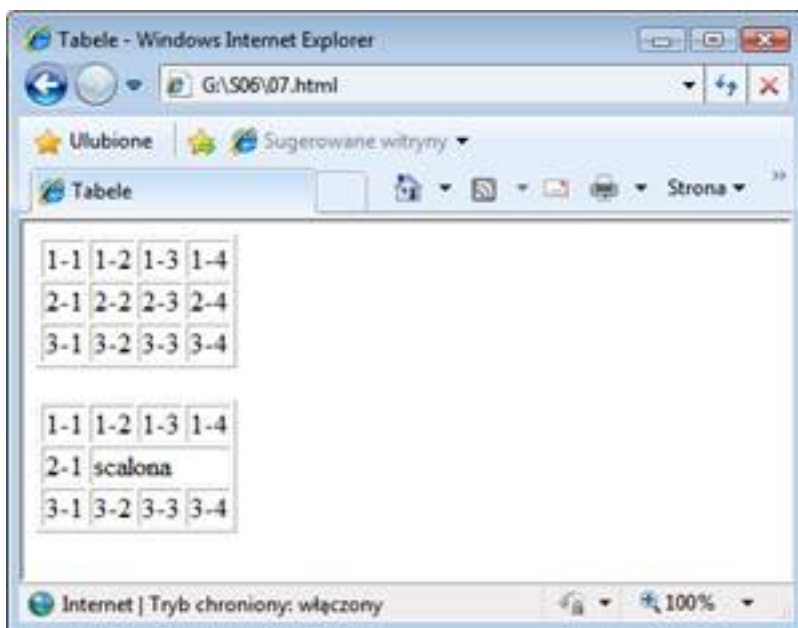
Tabele

Dodał Administrator
czwartek, 22 lipiec 2010 08:59

```
<tr>
  <td>3-1</td> <td>3-2</td> <td>3-3</td> <td>3-4</td>
</tr>
</table>
```

```
<br />
```

```
<table border="1">
  <tr>
    <td>1-1</td> <td>1-2</td> <td>1-3</td> <td>1-4</td>
  </tr>
  <tr>
    <td>2-1</td> <td colspan="3">scalona</td>
  </tr>
  <tr>
    <td>3-1</td> <td>3-2</td> <td>3-3</td> <td>3-4</td>
  </tr>
</table>
```



Rysunek 6.12. W dolnej tabeli trzy elementy `<td>` (odpowiadające trzem kolumnom tabeli) zostały zastąpione jednym, tworzącym

Scalanie komórek w pionie

Scalanie dwóch leżących nad sobą, czyli w pionie, komórek tabeli wymaga rozszerzenia o atrybut `rowspan` elementu `<td>` leżącego w najwyższym wierszu oraz usunięcia elementów `<td>` leżących w wierszach poniżej. Metoda jest zatem identyczna jak w przypadku scalania komórek w poziomie, lecz wykonanie trudniejsze, gdyż modyfikacja kodu będzie obejmowała wiele wierszy tabeli, a więc łatwiej o omyłkę.

Oto przykład, w którym w drugiej z dwóch identycznych tabel dwie komórki w środkowej kolumnie zastąpiłem jedną scaloną, rozciągniętą na powierzchni dwóch (rysunek 6.13):

```
<table border="1">
```

```
<tr>
```

```
<td>1-1</td>
```

```
<td>1-2</td>
```

```
<td>1-3</td>
```

```
</tr>
```

```
<tr>
```

```
<td>2-1</td>
```

```
<td>2-2</td>
```

```
<td>2-3</td>
```

```
</tr>
```

```
<tr>
```

```
<td>3-1</td>
```

```
<td>3-2</td>
```

```
<td>3-3</td>
```

```
</tr>
```

```
</table>
```

```
<br />
```

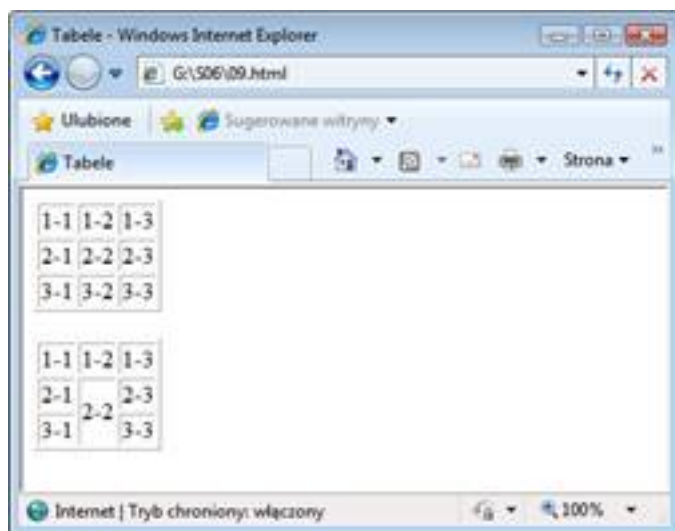
```
<table border="1">
```

```
<tr>
```


Tabele

Dodał Administrator
czwartek, 22 lipiec 2010 08:59

```
<td>1-1</td>
<td>1-2</td>
<td>1-3</td>
</tr>
<tr>
<td>2-1</td>
<td rowspan="2">2-2</td>
<td>2-3</td>
</tr>
<tr>
<td>3-1</td>
<td>3-3</td>
</tr>
</table>
```



Rysunek 6.13. W dolnej tabeli dwa elementy <td> zostały zastąpione jednym, tworząc komórkę rozciągniętą na dwa wiersze

Dzielenie scalonej komórki tabeli

Aby anulować operację scalenia komórek tabeli, należy wykonać dwie operacje:

usunąć atrybut rowspan lub colspan z elementu <td> odpowiadającego za scaloną komórkę (zamienić ją z

Tabele

Dodał Administrator
czwartek, 22 lipiec 2010 08:59

powrotem w najwykleszą komórkę tabeli),

odtworzyć usunięte w czasie scalania elementy `<td>`, tak aby przywrócić odpowiednią liczbę elementów w każdym wierszu (elemente `<tr>`) tabeli.

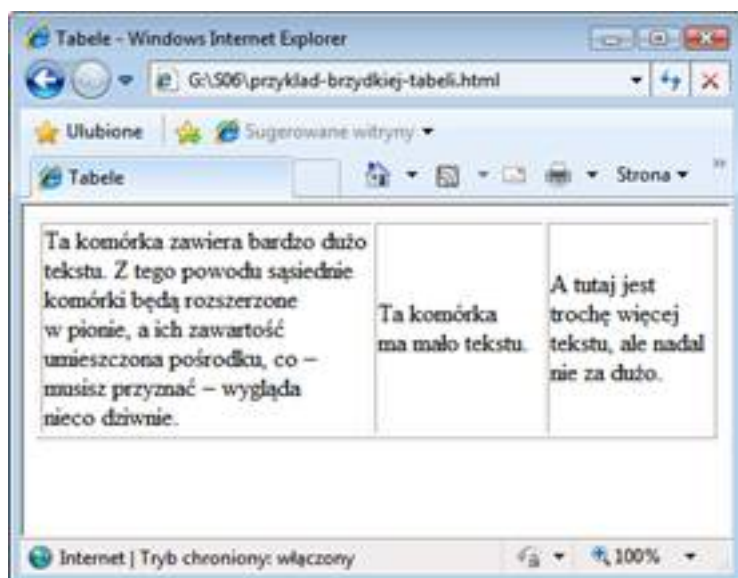
Określenie położenia zawartości komórki tabeli

Tekst zawarty w komórce tabeli może być wyrównywany w dwojaki sposób:

w poziomie — do lewego lub prawego marginesu, do obu marginesów lub wycentrowany,

w pionie — do górnego lub dolnego marginesu albo wycentrowany.

Szczególnie istotna jest możliwość definiowania wyrównywania tekstu w pionie, gdyż przeglądarki WWW najczęściej domyślnie centrują zawartość komórki, co powoduje nieelegancki wygląd tabel zawierających w komórkach na zmianę długie i krótkie fragmenty tekstu (rysunek 6.14).



Rysunek 6.14. Skutek ignorowania kwestii wyrównywania zawartości komórek tabeli w pionie

Do wyrównywania zawartości komórki — czyli elementu `<td>` lub `<th>` — służy znany Ci już dobrze atrybut stylu `text-align`, który może przyjmować następujące wartości:

`left` — wyrównanie do lewego marginesu,

`right` — wyrównanie do prawego marginesu,

Tabele

Dodał Administrator
czwartek, 22 lipiec 2010 08:59

center — wyśrodkowanie,

-

justify — wyrównanie do obu marginesów (justowanie).

Z kolei za wyrównanie zawartości komórki w pionie odpowiada atrybut stylu vertical-align, któremu można przypisać jedną z następujących wartości:

-

top — wyrównanie do górnego marginesu,

-

bottom — wyrównanie do dolnego marginesu,

-

middle — wyśrodkowanie.

Oto przykład tabeli, w której użyłem prawie wszystkich (z wyjątkiem justowania) możliwości wyrównania zawartości (rysunek 6.15):

```
<table cellpadding="3" cellspacing="0" border="1" style="width: 100%;">
<tr style="height: 1.5cm;">
  <td style="width: 33%; text-align: left; vertical-align: top;">Lewo, góra</td>
  <td style="width: 33%; text-align: center; vertical-align: top;">Środek, góra</td>
  <td style="width: 33%; text-align: right; vertical-align: top;">Prawo, góra</td>
</tr>
<tr style="height: 1.5cm;">
  <td style="text-align: left; vertical-align: middle;">Lewo, środek</td>
  <td style="text-align: center; vertical-align: middle;">Środek, środek</td>
  <td style="text-align: right; vertical-align: middle;">Prawo, środek</td>
</tr>
<tr style="height: 1.5cm;">
  <td style="text-align: left; vertical-align: bottom;">Lewo, dół</td>
  <td style="text-align: center; vertical-align: bottom;">Środek, dół</td>
  <td style="text-align: right; vertical-align: bottom;">Prawo, dół</td>
</tr>
</table>
```



Rysunek 6.15. Dziewięć najważniejszych kombinacji atrybutów stylu odpowiedzialnych za wyrównywanie zawartości komórek

W powyższym przykładzie nadałem elementom `<tr>` atrybut stylu `height`, aby ręcznie określić wysokość wierszy tekstu, co podkreśliło możliwość wyrównywania zawartości komórek tabeli w pionie. W praktyce staraj się nigdy nie wykorzystywać możliwości sterowania wysokością wierszy tabeli: tylko automatyczny dobór umożliwi wyświetlanie zawsze pełnej zawartości wszystkich komórek tabeli.

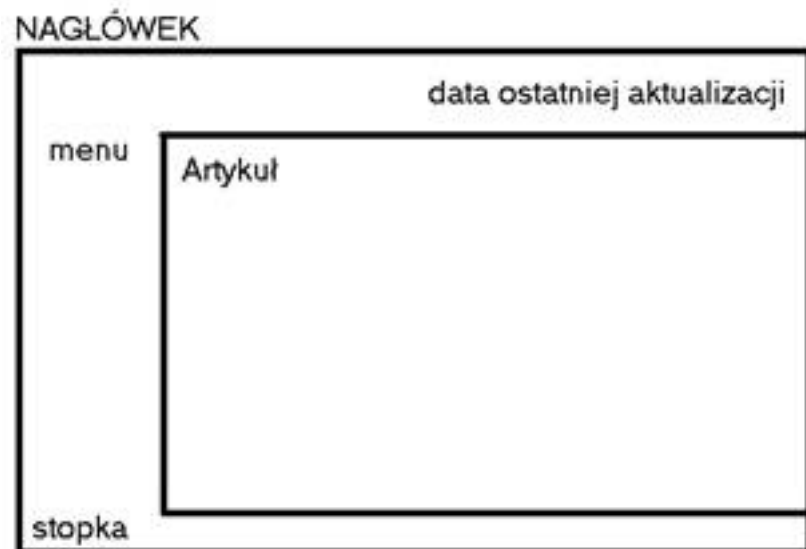
Aby uczynić domyślnym wyrównywanie zawartości do górnej krawędzi komórek, w obszarze definicji stylu strony WWW umieść następującą definicję:

```
td {  
  
vertical-align: top;
```

} Tabela tworząca układ strony

Podstawowym celem tabeli jest przedstawienie zbioru danych w eleganckiej, tabelarycznej postaci. Dostępny w języku XHTML mechanizm tabel jest jednak tak elastyczny, iż umożliwia budowanie — za pomocą tabel — układu całej strony WWW (dzielenie jej na prostokątne fragmenty odpowiadające szpaltom gazety).

W poniższym przykładzie pokażę, w jaki sposób wykorzystać mechanizm tabel do stworzenia prostego układu strony podzielonej na fragmenty (rysunek 6.16).



Rysunek 6.16. Typowy układ graficzny serwisu WWW

Tworząc bardziej skomplikowane tabele, możesz wspomagać się takimi właśnie szkicami. Ułatwiają one określenie liczby wymaganych wierszy i kolumn tabeli oraz oznaczenie komórek, które powinny zostać scalone.

Przyjrzyj się powyższemu schematowi i postaraj się odgadnąć liczbę potrzebnych wierszy i kolumn tabeli. Jeden

Tabele

Dodał Administrator
czwartek, 22 lipiec 2010 08:59

wiersz przypadnie na nagłówek, drugi — na datę ostatniej aktualizacji, trzeci — na menu i treść artykułu, a czwarty — na stopkę strony; razem cztery wiersze. Liczba kolumn powinna z kolei wynosić dwa, gdyż środkowa część strony będzie podzielona właśnie na dwie kolumny; pozostałe wiersze będą wykorzystywały komórki scalone, by zamienić dwie kolumny w jedną.

Oto gotowy kod strony WWW o podanym schemacie, uzupełniony nawet pewnym szcążkowym formatowaniem. Na rysunku 6.17 przedstawiona została strona WWW powstała na bazie tego kodu:

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

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

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

<head>

<title>Serwis internetowy</title>

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

</head>

<body style="margin: 0; padding: 0; font-family: 'trebuchet ms';">

<table cellpadding="3" cellspacing="0" style="width: 100%;">

<tr>

<td colspan="2" style="font-size: 16pt; font-weight: bold;">

Serwis internetowy

</td>

</tr>

<tr>

<td colspan="2" style="background-color: blue; color: yellow; text-align: right;">

Ostatnia aktualizacja: 2006-06-01

</td>

</tr>

<tr>

<td style="width: 3.5cm; background-color: blue; color: yellow; text-align: center; vertical-align: top;">

<a href="strona.html" style="color: cyan;">Strona główna</a><br />

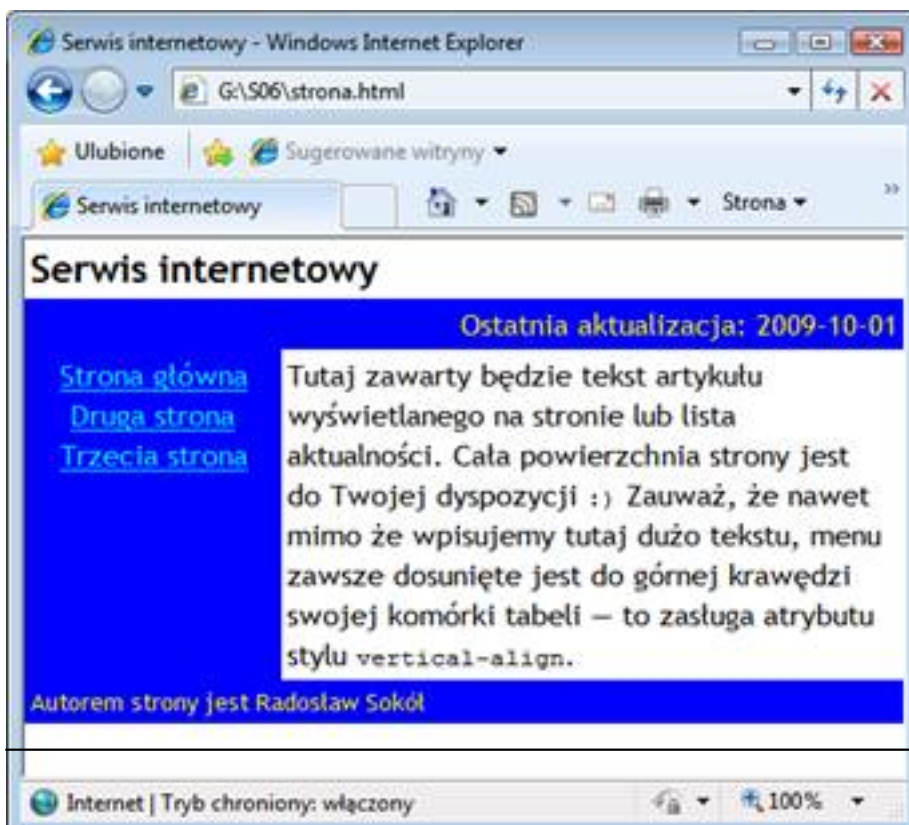
<a href="druga.html" style="color: cyan;">Druga strona</a><br />


```

Tabele

Dodał Administrator
czwartek, 22 lipiec 2010 08:59

```
<a href="trzecia.html" style="color: cyan;">Trzecia strona</a>
</td>
<td style="vertical-align: top;">
Tutaj zawarty będzie tekst artykułu wyświetlanego na&nbsp;stronie
lub&nbsp;lista aktualności. Cała powierzchnia strony jest do&nbsp;Twojej
dyspozycji&nbsp;<tt>:</tt> Zauważ, że nawet mimo że&nbsp;wpisujemy tutaj
dużo tekstu, menu zawsze dosunięte jest do&nbsp;górnjej krawędzi swojej
komórki tabeli&nbsp;&mdash; to&nbsp;zasluga atrybutu stylu <tt>vertical-align</tt>.
</td>
</tr>
<tr>
<td colspan="2" style="background-color: blue; color: yellow; font-size: 10pt;">
Autorem strony jest Radosław&nbsp;Sokół
</td>
</tr>
</table>
</body>
</html>
```



Rysunek 6.17. Strona WWW podzielona na części funkcjonalne z wykorzystaniem tabel XHTML

Podsumowanie

Tabele stanowią bardzo istotny element języka XHTML. Wbrew pozorom rzadko służą one do przedstawiania w postaci tabelarycznej zestawów danych. Najczęstszym zastosowaniem tabel jest dzielenie strony na obszary (takie jak obszar nagłówka, obszar menu czy obszar treści) wypełnione różnorodnym tłem.

Tworząc tabele, pamiętaj o kilku zasadach:

-

kod tabeli wprowadza się zawsze wiersz po wierszu,

-

każdy wiersz tabeli (element `<tr>`) zawiera kilka komórek (elementy `<td>` lub `<th>`),

-

w tabeli niezawierającej scalonych komórek liczba komórek w każdym wierszu musi być identyczna,

-

komórka scalona obejmująca N wierszy lub N kolumn odpowiada N odrębnym elementom `<td>` lub `<th>`, zatem tworząc scaloną komórkę, należy usunąć nadmiarowe komórki tabeli wraz z ich zawartością,

-

błędny kod tabeli najczęściej skutkuje efektem „rozsypywania się” tabeli — pojawiają się puste wiersze lub kolumny, komórki nachodzą na siebie lub też tabela urywa się nagle i przekształca w zwykły tekst.