

Ogólne ramy tabeli

Ramy tabeli tworzone są za pomocą polecenia:

```
<table> </table>
```

Cała zawartość tabeli musi być umieszczona między tymi dwoma znacznikami, które stanowią jej delimitery. W ich ramach są umieszczane definicje rzędów, definicje komórek w rzędach, konkretne dane w komórkach, tytuł tabeli i nagłówki wierszy i kolumn.

Wiersz tabeli

Wiersze tabeli wprowadzamy za pomocą polecenia:

```
<tr> </tr>
```

Wiersz tabeli jest jej konkretyzacją, a sam tworzy z kolei ramy dla komórek z danymi. W ramach `<table>` `</table>` można umieścić wiele kolejnych definicji wierszy `<tr>` `</tr>`, dla przykładu:

```
<table>
<tr> </tr>
<tr> </tr>
<tr> </tr>
</table>
```

Zwróć uwagę, że wprowadzenie definicji tabeli `table`, a w jej ramach pewnej liczby wierszy `tr` nie powoduje jeszcze wyświetlenia tabeli. Dopiero zdefiniowanie komórek, o których piszemy niżej, pozwoli wyświetlić tabelaryczny układ danych.

Komórka w wierszu

Komórki tabeli wprowadzamy za pomocą polecenia:

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

Komórka zawiera już konkretne dane. Między jej znacznikami można umieszczać tekst i grafikę. Konkretne komórki są umieszczane między znacznikami wybranego wiersza, na przykład:

```
<table>
<tr> <td> </td><td> </td><td> </td></tr>
<tr> <td> </td><td> </td><td> </td></tr>
<tr> <td> </td><td> </td><td> </td></tr>
</table>
```

Uwaga: dla przejrzystości obrazu dokumentu w edytorze HTML warto umieszczać definicje wierszy tabeli jedną pod drugą, natomiast definicje kolejnych komórek obok siebie, co symuluje układ wierszy i kolumn w całej tabeli. Oczywiście komórki można opisywać w edytorze także w kolejnych rzędach, co naturalnie nie wpływa na ich faktyczne położenie w przeglądarce, wyznaczone przez definicję wiersza. Sposób pisania to kwestia wygody użytkownika.

Przykład tabeli:

a1 a2 a3 a4 a5

b1 b2 b3 b4 b5

c1 c2 c3 c4 c5

Jak widać, tabela zawiera 3 rzędy, w każdym po 5 komórek, zaś w konkretnych komórkach (w ramach znaczników `<td>` `</td>`) zostały umieszczone dane: od a1 do c5. Jest to najprostszy przykład tabeli, która nie zawiera żadnych obramowań, barw, nagłówków i podpisu. Za chwilę przystąpimy do urozmaicania tabeli, dzięki czemu znacznie wzrośnie jej przejrzystość i wartość informacyjna.

Obramowanie tabeli

Aby tabela zawierała naokoło obramowanie, wystarczy rozszerzyć definicję o atrybut **border**. Jeśli nie podamy szerokości obramowania, przyjmowana jest jej domyślna wartość.

```
<table border> </table>
```

a1 a2 a3 a4 a5

b1 b2 b3 b4 b5

c1 c2 c3 c4 c5

Gdybyśmy podali szerokość obramowania w pikselach, zostanie ona odpowiednio zinterpretowana przez przeglądarkę, na przykład:

```
<table border="10"> </table>
```

a 1	a2	a3	a4	a 5
b 1	b2	b3	b4	b 5
c 1	c2	c3	c4	c 5

Poziome wyrównanie danych w komórkach

Atrybut **align** wykorzystuje się także do poziomego wyrównania zawartości komórki (a nawet całego wiersza **tr**) - środkowania, justowania do lewej i justowania do prawej, z użyciem wartości center, left i right.

```
<td align="left"> </td>
<td align="center"> </td>
<td align="right"> </td>
```

Oto wynik:

a1 - do lewej	a2 - środkowanie	a3 - do prawej
b1 - do lewej	b2 - środkowanie	b3 - do prawej
c1 - do lewej	c2 - środkowanie	c3 - do prawej

Teoretycznie możliwe jest też wyrównanie według znaku (można wstawiać ten atrybut w komórkach, wierszach i >grupach kolumn), co znajduje praktyczne zastosowanie w wyrównywaniu liczb ze znakami po przecinku, pozwalając poprawnie wyrównywać je właśnie według podanego znaku - niestety, żadna z przeglądarek nie interpretuje tego atrybutu.

```
<td align="char" char=",">10,59</td>
<td align="char" char=",">100,48</td>
<td align="char" char=",">12,592</td>
```

Alternatywnie stosuje się też style CSS, dające ten sam efekt.

```
<td style="text-align: left;"> </td>
<td style="text-align: center;"> </td>
<td style="text-align: right;"> </td>
```

a1 - do lewej	a2 - środkowanie	a3 - do prawej
b1 - do lewej	b2 - środkowanie	b3 - do prawej
c1 - do lewej	c2 - środkowanie	c3 - do prawej

Wysokość tabeli

Wysokość tabeli była niekiedy dawniej regulowana za pomocą nieoficjalnego, niewchodzącego w skład HTML atrybutu **height**, o wartościach w pikselach lub procencie widocznego okna, ignorowanego zresztą przez niektóre przeglądarki. Poprawne może być jedynie podanie wysokości za pomocą stylów CSS, na przykład:

```
<table border style="height: 200px; width: 60%; ">
```

Wynik podania tej wartości:

Komórka a1	Komórka a2
Komórka a3	Komórka a4

Wartość podana w procentach będzie interpretowana jedynie wtedy, gdy tabela będzie umieszczona w pojemniku wyższego rzędu o podanej wysokości.

```
<div style="height: 200px; ">
<table border style="height: 50%; width: 60%; ">
```

Komórka a1	Komórka a2
Komórka a3	Komórka a4

Pionowe wyrównanie danych w komórkach

Atrybut **valign** (vertical align) służy do pionowego wyrównania zawartości komórki - do górnego brzegu, do środka i do dolnego brzegu. Używamy wówczas odpowiednio konstrukcji:

```
<td valign="top"> </td>
<td valign="middle"> </td>
<td valign="bottom"> </td>
```

Wynik:

a1 - do górnego brzegu	a2 - do górnego brzegu	a3 - do górnego brzegu
b1 - do górnego brzegu	b2 - do dolnego brzegu	b3 - do środka