

HTML i CSS

– wykład powtórzeniowy



Szablon dokumentu HTML

```
<!doctype html>
<html>
  <head>
    <meta charset="UTF-8" />
    <title>Tytuł strony...</title>
  </head>
  <body>
  </body>
</html>
```



1. Pierwszy wiersz tworzy tak zwana deklaracja typu dokumentu, reprezentowana przez znacznik **<!DOCTYPE>**. Deklaracja typu dokumentu informuje przeglądarkę WWW, jak ściśle powinna się ona trzymać standardu języka HTML w czasie interpretowania kodu strony. Powyższa deklaracja również podkreśla, że strona WWW zawiera kod zapisany w zgodzie ze standardami HTML
2. Pomiedzy znacznikami **<html>** i **</html>** umieszczmy wszystkie elementy składające się na stronę WWW. Za znacznikiem **</html>** nie może znajdować się już nic.
3. Kolejne dwa wiersze zawierają znaczniki otwierający i zamykający elementu **head**, deklarującego obszar nagłówka strony. Nagłówek strony nie jest nigdzie wyświetlany.
4. Elementy wyświetlane na stronie umieszczamy pomiędzy znacznikami **<body>** **</body>**.

Znaczniki meta umieszczone w sekcji nagłówka dokumentu HTML służą do wskazywania zakodowanych informacji i określania **metadanych** (nie pojawiają się na stronie internetowej).

```
<meta charset="UTF-8" />
```

```
<meta name="description" content="Opis strony"/>
```

```
<meta name="keywords" content="słowo_kluczowe_1, słowo_kluczowe_2"/>
```

```
<meta name="author" content="Autor" />
```

```
<meta name="copyright" content="właściciel praw autorskich" />
```

```
<meta name="robots" content="follow"/>
```

```
<meta name="robots" content="noindex"/>
```



Styl wpisany

Używamy ich najczęściej gdy chcemy zmodyfikować styl pojedynczego elementu.

<znacznik style="własność: wartość">

...

</znacznik>

Przykład:

```
<body style= "background-color: #c4c59c;  
              min-width: 800px;  
              max-width: 800px;  
              margin: 0 auto;" >  
  
</body>
```



Styl osadzony

Ogólna postać stylu osadzonego wygląda następująco:

```
<style type=„text/css”>
  <!--
    selektor{własność: wartość; własność: wartość...}
  -->
</style>
```

Rodzaje arkuszy stylu: styl osadzony



Przykład:

```
<style type="text/css">
  <!--
    body
    {
      background-color: black;
      min-width: 800px;
      max-width: 800px;
      margin: 0 auto
    }
    h1 {color: blue; text-align: center}
    h2 {color: blue; text-align: left}
    p
    {
      color: white;
      font-family: arial, sans-serif;
      font-size: small
    }
  -->
</style>
```

Rodzaje arkuszy stylu: zewnętrzne arkusze stylu

Jak przygotować plik CSS?

Zewnętrzny arkusz stylów możemy stworzyć w dowolnym edytorze. Plik nie powinien zawierać znaczników HTML.

Przykład zawartości pliku:

```
hr {color: red}
h1 {text-align: center}
p {
    text-align: center;
    text-size: large
}
```

Tak przygotowany plik zapisujemy na dysku - nazwa dowolna, rozszerzenie **.css** np. "styl.scc"



Rodzaje arkuszy stylu: zewnętrzne arkusze stylu

Jak dołączyć plik CSS do dokumentu HTML?

Polecenie link umieszczamy w sekcji HEAD dokumentu:

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

gdzie: style.css – nazwa arkusza stylu (nie musi być właśnie taka)

KLASY, czyli jak nadać różne style dla jednego typu elementu.

element.nazwa_klasy {właściwość: wartość...}

Przykład:

```
h1.zielony {color: green; text-align:center}
```

```
h1.niebieski {color: blue; text-align:center}
```

```
p.arial {font-family: arial, sans-serif}
```

```
p.courier_new {font-family: "courier new", serif}
```

Zastosowanie przykładu z poprzedniej strony w dokumencie HTML:

```
<body>
  ....
  <h1 class="zielony">Ten nagłówek będzie zielony</h1>
  <h1 class="niebieski">Ten nagłówek będzie niebieski</h1>
  <p class="arial">Ten akapit będzie pisany czcionką ARIAL</p>
  <p class="courier_new">Ten akapit będzie pisany czcionką COURIER NEW</p>
  ....
</body>
```

KLASY – niezwiązane z konkretnym znacznikiem



Klasy niezwiązane z konkretnym obiektem.

.nazwa_klasy {właściwość: wartość...}

Przykład:

```
.zielony {color: green; text-align: center}
.niebieski {color: blue; text-align: center}
.arial {font-family: arial, sans-serif}
.courier_new {font-family: "courier new", serif}
```

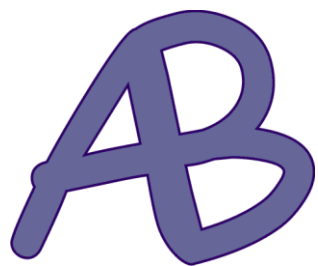


Zastosowanie przykładu z poprzedniej strony w dokumencie HTML:

Zastosowanie:

```
<body>
  ....
  <h1 class="zielony">Ten nagłówek będzie zielony</h1>
  <p class="zielony">Teraz kolorem zielonym pisany jest akapit</h1>
  <p class="arial">Ten akapit będzie pisany czcionką ARIAL</p>
  <h1 class="courier_new">Ten nagłówek będzie pisany czcionką COURIER NEW</h1>
  ....
</body>
```

ciąg dalszy nastąpi...



II. Formatowanie tekstu



Elementy liniowe – mogą być wmontowane w blok tekstu (np. w akapit)

- ✓ Formatujemy kolor, podkreślenie itp.
- ✓ Nie można im nadać marginesów (wewnętrznych i zewnętrznych), interlinii itp.

To jest przykładowy tekst. To jest przykładowy tekst. **To jest przykładowy tekst.** To jest przykładowy tekst. To jest przykładowy tekst.
To jest przykładowy tekst. To jest przykładowy tekst. To jest przykładowy tekst. To jest przykładowy tekst. To jest przykładowy tekst.

- ✓ Do elementów liniowych należą:
<a>, <sup>, <sub>, <pre>...
- ✓ Do elementów liniowych odnoszą się własności z rodziny „font” (np. font-family)



Elementy blokowe – wyznaczają bloki tekstu. Blok jest wpisany w prostokąt.

- ✓ Można formatować marginesy, interlinię, obramowania, kolor tła (dla całego bloku), wcięcia...
- ✓ Blokom można nadawać wszystkie aspekty formatu elementów liniowych.

- ✓ Do elementów liniowych należą: <p>, , , <h1>...
- ✓ Do bloków odnoszą się własności z rodziny „text” (np. text-align)

To jest przykładowy tekst. To jest przykładowy tekst. To jest przykładowy tekst. To jest przykładowy tekst. To jest przykładowy tekst. To jest przykładowy tekst.

To jest przykładowy tekst. To jest przykładowy tekst. To jest przykładowy tekst. To jest przykładowy tekst. To jest przykładowy tekst. To jest przykładowy tekst.

To jest przykładowy tekst. To jest przykładowy tekst. To jest przykładowy tekst. To jest przykładowy tekst. To jest przykładowy tekst. To jest przykładowy tekst.



Kontenery – znaczniki, które same w sobie niczemu nie służą. Grupują element, którym możemy nadać wspólne formatowanie

- ✓ kontener na elementy liniowe (to nie do końca książkowe określenie)

`...`

- ✓ kontener na elementy blokowe

`<div>....</div>`



Jedną z najważniejszych decyzji, które należy podjąć przed rozpoczęciem pisania reguł stylistycznych, jest rodzaj użytych jednostek:

- ✓ bezwzględne (punkty, cale itd.)
- ✓ czy względne (procenty, em).

Stara metoda polegała na stosowaniu pikseli, obecnie nastąpił zwrot ku jednostkom względnym. Oto zalety i wady tej metody:



Zalety:

1. Rozmiar tekstu zmienia się proporcjonalnie w każdym miejscu po użyciu opcji zmiany rozmiaru tekstu w menu Widok przeglądarki. Jest to bardzo przyjazne podejście do użytkownika.
2. Dopracowując szczegóły projektu można **proporcjonalnie zmienić rozmiar całego tekstu na stronie, modyfikując tylko wartość dla elementu body**. Zmieniany jest tylko rozmiar podstawowy, a wszystkie pozostałe rozmiary dostosowują się proporcjonalnie do tej wartości.



Wady:

1. Przez chwilę nieuwagi można spowodować, że tekst w zagnieżdżonych elementach będzie bardzo mały ponieważ rozmiar czcionki jest dziedziczony.
2. Użytkownik może zburzyć układ strony, który nie został przystosowany do zmiany rozmiaru tekstu. Jeśli na przykład użytkownik ustawi rozmiar czcionki na dużą wartość, pływające kolumny mogą poprzesuwać się w różne miejsca, jeśli przestaną mieścić się w tych, które im pierwotnie wyznaczono.

(Na wykładzie przyjrzymy się tworzeniu zaawansowanych układów stron przy użyciu CSS, zajmiemy się także problemem i sposobami jego rozwiązania)

Rozmiar czcionki



Rozmiar czcionki możesz określić za pomocą atrybutu stylu font-size:

font-size: 16px;



Jednostki bezwzględne:

- **px** – punkty ekranu
- **pt** – punkty (1 punkt to 1/72 cala)

Uwaga! Jednostek **pt** używamy do definiowania wydruku. Aby przygotować wydruk ze strony w oddzielnym pliku CSS definiujemy rozmiary czcionki za pomocą pt.

$$1\text{pt} = \frac{96}{72}\text{px} = \frac{4}{3}\text{px} = 1,33\text{px}$$

$$1\text{px} = \frac{72}{96}\text{pt} = \frac{3}{4}\text{pt} = 0,75\text{pt}$$



Imienne wartości absolutne:

xx-small - najmniejsza

x-small - bardzo mała

small - mała

medium - średnia

large - duża

x-large - bardzo duża

xx-large - największa

rozmiar **xx-small**

rozmiar **x-small**

rozmiar **small**

rozmiar **medium**

rozmiar **large**

rozmiar **x-large**

rozmiar **xx-large**



Jednostki względne (odnoszące się do jednostki referencyjnej).

em – jednostka pochodzi od szerokości małej litery „m”

Rozmiar czcionki ustalany w oparciu o jednostki em jest relatywny w stosunku do rodzica elementu. Rodzicem wszystkich elementów pierwszego poziomu (uwaga na zagnieżdżanie) jest element `<html>`. Domyślna wartość rozmiaru czcionki dla tego elementu w większości przeglądarek to **16px** (1em = 16px).

% - procent – działają na podobnej zasadzie jak jednostki em i odnoszą się do rodzica w ustalaniu rozmiaru czcionki.



Imienne wartości względne:

`smaller` - mniejsza od bieżącej

`larger` - większa od bieżącej

rozmiar `smaller`

czcionka bieżąca

rozmiar `larger`



Przy wyborze kroju pisma masz do dyspozycji wszystkie kroje zainstalowane na Twoim komputerze. Pamiętaj jednak, że inni użytkownicy wcale nie muszą nimi dysponować! Z tego powodu powinieneś ograniczać się c standardowych rodzin krojów pisma:

- ✓ **serif** — czcionka szeryfowa, np. Times New Roman,
- ✓ **sans-serif** — czcionka bezszeryfowa, np. Arial,
- ✓ **cursive** oraz **fantasy** — ozdobne czcionki pochylę (rzadko używane, gdyż każda przeglądarka może podstawić w ich miejsce całkowicie inne kroje pisma — często w ogóle nieprzystające do zamysłu autora strony),
- ✓ **mono** — czcionka nieproporcjonalna (o stałej szerokości wszystkich znaków) używana najczęściej do zapisywania tekstu programów komputerowych, tworzenia prymitywnych tabel lub wyróżniania poleceń programów; np. Courier New.



Wyboru kroju pisma dokonuje się za pomocą atrybutu stylu o nazwie font-family:

font-family: „times new roman”, serif;

/ font Times New Roman, jeżeli nie jest dostępny to inna czcionka szeryfowa/

Inne parametry czcionki i akapitu

- ✓ Tekst preformatowany (spacje i końce linii mają znaczenie) –
`<pre>...</pre>`
- ✓ Tekst pisany czcionką o stałej szerokości (tekst maszynowy) –
`<tt>...</tt>`
- ✓ Marginesy akapitów –
`<p style="margin-top: 5pt; margin-bottom: 3pt;">...</p>`
`<p style="margin: 10px 10px 10px 10px;">`
/gdzie wartości oznaczają margin: górny prawy dolny lewy;/
- ✓ Marginesy wewnętrzne –
`<p style="padding: 20px;">`

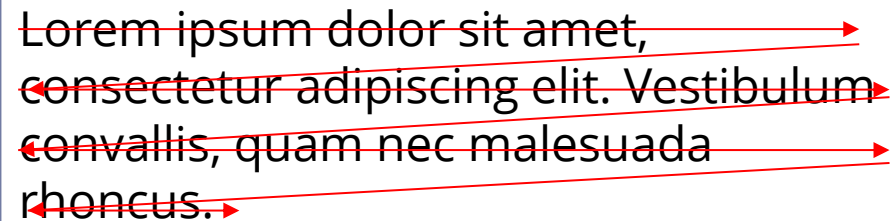


W CSS dostępnych jest osiem własności do formatowania tekstu.

1. `text-indent`
2. `letter-spacing`
3. `word-spacing`
4. `text-decoration`
5. `text-align`
6. `line-height`
7. `text-transform`
8. `vertical-align`

Bardzo ważne jest, aby używając własności tekstu, wiedzieć, że w CSS każda linia tekstu znajduje się w prostokątnej ramce.

Na przykład fragment tekstu w akapicie dla CSS jest długą, cienką linią w prostokątnej ramce, podzieloną na poszczególne wiersze.



Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Vestibulum
convallis, quam nec malesuada
rhoncus.



Własność text-align:

Przykład:

```
p {text-align:right;}  
Wartości: left, right, center, justify
```

Tekst zostaje wyrównany w poziomie wewnątrz zawierającego go elementu. Dlatego własność tę stosuje się do elementu kontenera. Jeśli chcemy aby nagłówek <h1> znajdował się na środku elementu <div>, własność text-align należy zastosować do elementu <div>, a nie <h1>.



Własność text-indent

Przykład:

Przykład: `P {text-indent: 3em;}`

Wartości: wszystkie jednostki miary (wartości mogą być dodatnie i ujemne)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum convallis, quam nec malesuada rhoncus, odio metus vestibulum quam, in laoreet ex odio scelerisque mi. Sed sit amet egestas augue. Proin fermentum sit amet nibh eget facilisis. Suspendisse ultrices et lacus at placerat..

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum convallis, quam nec malesuada rhoncus, odio metus vestibulum quam, in laoreet ex odio scelerisque mi. Sed sit amet egestas augue. Proin fermentum sit amet nibh eget facilisis. Suspendisse ultrices et lacus at placerat..



Własność letter-spacing i word-spacing

Przykład:

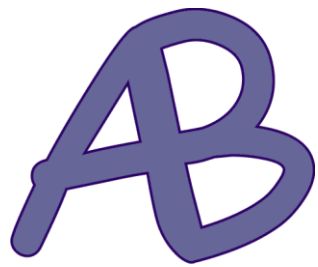
Przykład: `p {letter-spacing: 2em;}`

`p {word-spacing: 2em;}`

Wartości: wszystkie jednostki miary (wartości mogą być dodatnie i ujemne)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum convallis, quam nec malesuada rhoncus, odio metus vestibulum quam, in laoreet ex odio scelerisque mi. Sed sit amet egestas augue.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum convallis, quam nec malesuada rhoncus, odio metus vestibulum quam, in laoreet ex odio scelerisque mi. Sed sit amet egestas augue.



Listy

```
<ul>
  <li>Pozycja 01</li>
  <li>Pozycja 02</li>
  <li>Pozycja 03</li>
  <li>Pozycja 04</li>
</ul>
```

```
<ol>
  <li>Pozycja 01</li>
  <li>Pozycja 02</li>
  <li>Pozycja 03</li>
  <li>Pozycja 04</li>
</ol>
```

```
<ol>
  <li>Punkt 01</li>
  <li>
    <ul>
      <li>Podpunkt 01</li>
      <li>Podpunkt 02</li>
      <li>Podpunkt 03</li>
    </ul>
  </li>
  <li>Punkt 03</li>
  <li>Punkt 04</li>
</ol>
```

Jak zmienić lub usunąć znaki punktora na liście elementów?

Wygląd punktora listy elementów można zmodyfikować za pomocą właściwości

list-style-type

Oto dopuszczalne wartości właściwości **list-style-type**:

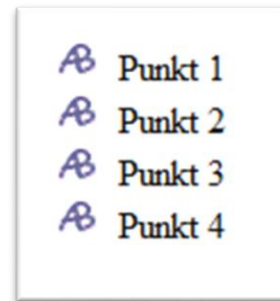
- disc,
- circle,
- decimal-leading-zero,
- decimal,
- lower-roman,
- upper-roman,
- lower-greek,
- lower-alpha,
- lower-latin,
- upper-alpha,
- upper-latin,
- Hebrew,
- Armenian,
- Georgian,
- none.



























Jak zdefiniować punktor użytkownika?

Aby zastosować własny obrazek jako punktor listy numerowanej, należy zamiast właściwości `list-style-type` użyć właściwości **`list-style-image`**, która pozwala podać adres URL zawierający ścieżkę i nazwę pliku graficznego.

```
ul{list-style-image: url(plik.gif);}
```



Właściwość **cursor** może przyjmować szereg wartości reprezentujących ikony

 auto	 move	 no-drop	 col-resize
 all-scroll	 pointer	 not-allowed	 row-resize
 crosshair	 progress	 e-resize	 ne-resize
 default	 text	 n-resize	 nw-resize
 help	 vertical-text	 s-resize	 se-resize
 inherit	 wait	 w-resize	 sw-resize

np.:

```
<div class=„klasa" style = "cursor:all-scroll">ZAWARTOŚĆ</div>
```



```
<dl>  
  <dt>HTML</dt>  
  <dd>Język znaczników, przeznaczony do tworzenia stron WWW.</dd>  
  <dt>CSS</dt>  
  <dd>Język do definiowania stylów stron internetowych.</dd>  
  <dt>JavaScript</dt>  
  <dd>Język służący do programowania dynamicznych stron WWW.</dd>  
</dl>
```

HTML

Język znaczników, przeznaczony do tworzenia stron internetowych.

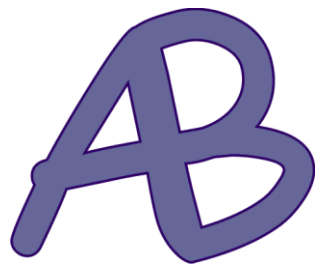
CSS

Język do definiowania stylów stron internetowych.

JavaScript

Język służący do programowania dynamicznych stron internetowych.

Wykład 0 - część IV

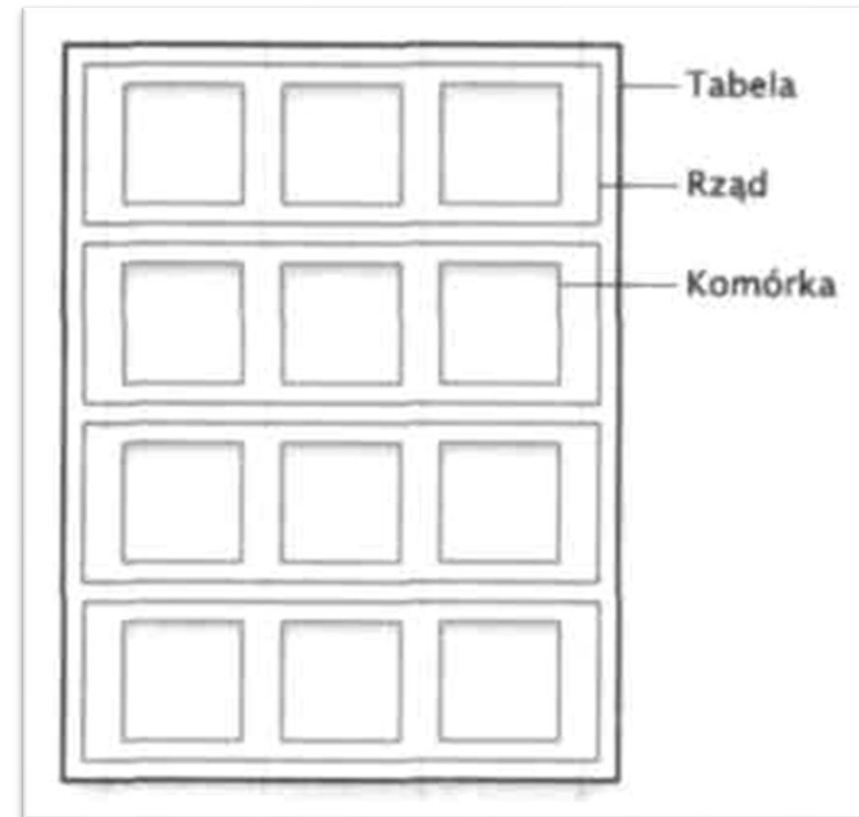


TABELE

Struktura tabeli - HTML

Najprostsza tabela składa się z elementu `<table>` oraz komórek `<td>` pogrupowanych w rzędy `<tr>`. Podstawowa struktura kodu tabeli 3x4 wygląda zatem następująco:

```
<table>           //TABELA//
  <tr>            //RZĄD//
    <td> </td>    //KOMÓRKA//
    <td> </td>
    <td> </td>
  </tr>
  <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>
```





Struktura tabeli - HTML 5

```

<table summary="Opis">           //Opis - nie jest wyświetlany//
  <thead>                        //Nagłówek tabeli//
    <tr>                        //wiersz nagłówka (może ich być więcej niż 1)//
      <th>Nagłówek1</th>        //Komórka nagłówka//
      <th>Nagłówek2</th>
    </tr>
  </thead>                      //koniec nagłówka//
  <tfoot>                       //stopka tabeli//
    <tr>                       //wiersz stopki//
      <td>Stopka1</td>         //komórka stopki//
      <td>Stopka2</td>
    </tr>
  </tfoot>                     //koniec stopki//
  <tbody>                       //sekcja body tabeli//
    <tr>                       //wiersz//
      <td>a11</td>             //komórka//
      <td>a12</td>
    </tr>
    <tr>
      <td>a21</td>
      <td>a22</td>
    </tr>
    <tr>
      <td>a31</td>
      <td>a32</td>
    </tr>
  </tbody>
</table>

```

Naglowek 1	Naglowek 2
a11	a12
a21	a22
a31	a32
Stopka 1	Stopka 2



Jak zmienić szerokość i pozycję tabeli?

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

- szerokość może być podana w procentach (np. 100% szerokości okna); pikselach (px) czy też w centymetrach (cm)



Obramowania tabeli

<table border="3">

- wywodzi się z poprzednich wersji języka HTML i umożliwia sterowanie obramowaniem wszystkich komórek tabeli na raz. Szerokość obramowania podaje się jako wartość atrybutu **border**, podanie zera jako szerokości usuwa obramowanie.

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


- wykorzystuje atrybuty stylu CSS (atrybut stylu **border**, którego pierwszy parametr określa szerokość obramowania, drugi — styl, a trzeci — kolor).

UWAGA: ta konstrukcja powoduje, że obramowaniem otaczana jest cała tabela, lecz nie poszczególne komórki.

Obramowanie tabeli

PRZYKŁAD:

```
<table style="border: 5px double red" summary="Tabela testowa">
....
// tabela z poprzedniego przykładu//
....
</table>
```

Kolumna 1	Kolumna 2	Kolumna 3	Kolumna 4
Żaba	 Jaszczurka	Kot	Kameleon
Morświn	Rybitwa	Szczupak	Okofi



Obramowanie komórek

W CSS można przypisać atrybut stylu **border** każdej komórce tabeli z osobna.

```
<td style="border: 1pt solid green;">...</td>
```

```
<td style="border: 2pt dashed blue;">...</td>
```

```
<td style="border: 3pt dotted red;">...</td>
```

Dotyczy to także komórek nagłówka i stopki tabeli

```
<th style="border: 2pt solid green"> Nagłówek</th>
```

```
<td style="border: 3 double red>Stopka</td>
```

Kolumna 1	Kolumna 2	Kolumna 3	Kolumna 4
1	2	3	4
5	6	7	8



Kolor tła i tekstu tabeli

Aby zmienić kolor tła tabeli i jej komórek za pomocą stylów CSS, wykorzystuje się atrybuty z rodziny **background** (głównie **background-color**). Zmiana koloru tła tabeli automatycznie indukuje zmianę koloru tła wszystkich komórek, których barwy tła nie określiłeś. Upraszcza to kompozycję wielobarwnych tabel.

```
<table style="background-color: red>  
<td style="background-color: blue>...</td>
```

Aby zmienić rozmiar, kolor i krój czcionki używamy poznanych już atrybutów CSS (color: font-family: font-size: itp). Można je stosować do całej tabeli **<table>**, lub do pojedynczych komórek **<td>**.

Kolumna 1	Kolumna 2	Kolumna 3	Kolumna 4
1	2	3	4
5	6	7	8



Jak określić marginesy komórek tabeli?

<table cellpadding="5" cellspacing="5"> – działa dla całej tabeli

cellpadding – określa szerokość marginesu wewnątrz komórki — określa odsunięcie zawartości komórki (tekstu, obrazu) od krawędzi.

cellspacing – określa szerokość marginesu wokół komórki — a więc rozsuniecie komórek. Nadanie mu wartości 0 spowoduje zlikwidowanie „obramowania” powstałego z tła tabeli.

W razie CSS można sterować szerokością marginesów wewnętrznych pojedynczych komórek.

<td style="padding: 10px">...</td>



Wyrównanie tekstu w tabeli

Do wyrównywania zawartości komórki — czyli elementu **td** lub **th** — służy atrybut stylu **text-align**, który może przyjmować następujące wartości:


- ✓ left — wyrównanie do lewego marginesu,
- ✓ right — wyrównanie do prawego marginesu,
- ✓ center — wyśrodkowanie,
- ✓ justify — wyrównanie do obu marginesów (justowanie).

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

Wyrównanie tekstu w tabeli

Przykład:

Lewo, góra	Środek, góra	Prawo, góra
Lewo, środek	 Środek, środek	Prawo, środek
Lewo, dół	Środek, dół	Prawo, dół



Wyrównanie tekstu w tabeli

Przykład - rozwiązanie

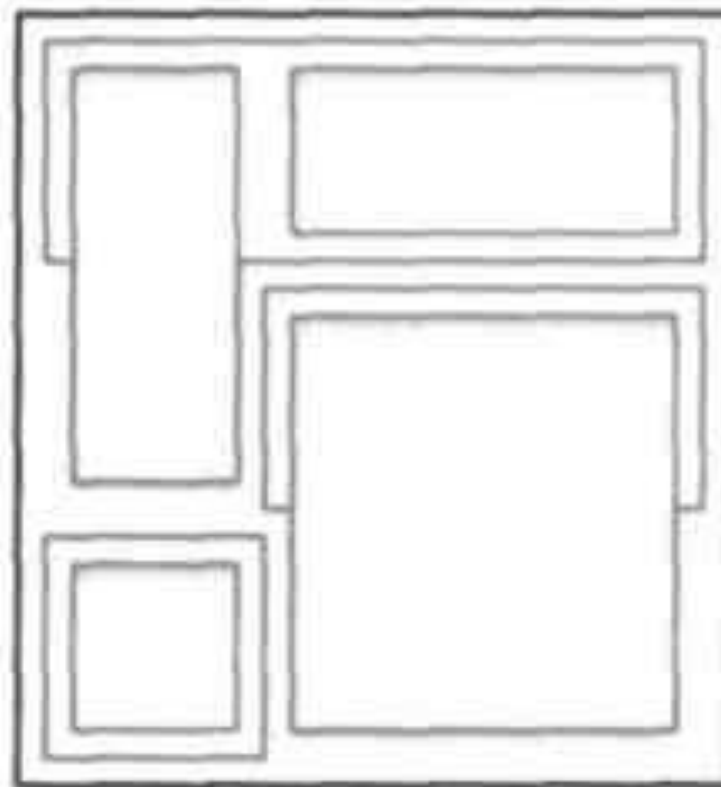
```
<table cellpadding="3" cellspacing="0" border="1"
        style="width:100%;" sunrnary="scalanie">

  <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>
```

Scalanie komórek tabeli





Scalanie komórek tabeli

Jak scalić komórki w poziomie?

Aby scalić sąsiadujące komórki tabeli, należy usunąć zbędne elementy `<th>` lub `<td>`, zaś dużą, scaloną komórkę uzupełnić atrybutem **colspan** informującym o liczbie komórek odpowiadających scalonej komórce.

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>
```



Jak scalić komórki w pionie?

W podobny sposób można uzyskać efekt rozpięcia komórki na kilku wierszach.

W tym przypadku skorzystamy z atrybutu **rowspan** elementu **td** z najwyższego wiersza oraz usuniemy elementy **td** z wierszy poniżej.

Scalanie komórek tabeli

Jak scalić komórki w pionie - przykład

```
<table summary="scalanie">
  <tr>
    <td rowspan="2">Nagłówek</td>
    <td>Komórka 1.2</td>
    <td>Komórka 1.3</td>
    <td>Komórka 1.4</td>
  </tr>
  <tr>
    <td>Komórka 2.2</td> //w drugiej kolumnie mamy już tylko
                           3 komórki (zamiast 4)//
    <td>Komórka 2.3</td>
    <td>Komórka 2.4</td>
  </tr>
</table>
```

Nagłówek	Komórka 1.2	Komórka 1.3	Komórka 1.4
	Komórka 2.2	Komórka 2.3	Komórka 2.4



Pozycjonowanie tabel na stronie

Centrowanie tabeli:

```
<div style="text-align: center">
  <table style="border: 1px solid green; margin: auto;"
summary= "Tabela testowa 2">
  <tr>
    <td>Komórka 1</td>
    <td>Komórka 2</td>
  </tr>
</table>
</div>
```

Użyto tu dwu mechanizmów:

```
<table style="margin: auto;">
<div style="text-align: center">...</div>
```

Ma to na celu zapewnienie poprawnego działania strony w różnych przeglądarkach.



Pozycjonowanie tabel na stronie

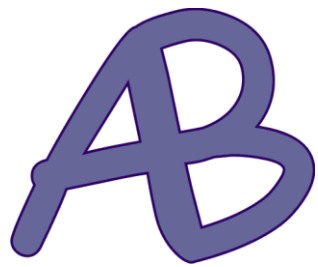
Wyrównie do prawej:

```
<table style="margin-left: auto;">  
<div style="text-align: right">....</div>
```

Ma to na celu zapewnienie poprawnego działania strony w różnych przeglądarkach.



- `<p style="background-color: red; color: blue;"> ..</p>`
- zdefiniowane kolory dla akapitu (niebieski tekst na czerwonym tle);
- `<body style="color: #ffffff; background-color: #000000">`
- zdefiniowano kolory dla całej strony (biały tekst na czarnym tle)
- `...` - zmiana koloru fragmentu akapitu, pojedynczego wyrazu lub nawet litery.



Odnośniki



Ścieżki dostępu

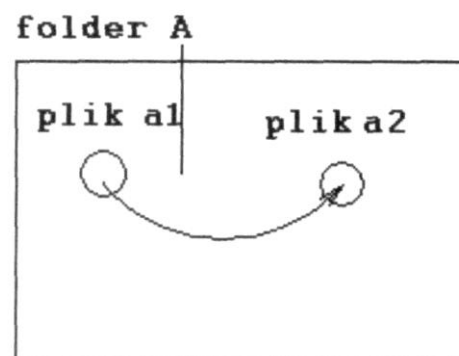
Jakie są rodzaje ścieżek dostępu?

- ✓ **bezwzględne** (np. `http://www.netscape.corn/`) — podany adres od początku do końca (bezwzględnie) definiuje położenie strony WWW i może być stosowany w takiej samej postaci niezależnie od strony WWW, na której go użyjemy;
- ✓ **względne** (np. `../teksty/opis.html` lub też `nowy.html`) — podany adres definiuje położenie strony WWW wyłącznie względem aktualnej strony, użycie go na innej stronie WWW (umieszczonej w innym katalogu tego samego serwera lub na innym serwerze) uniemożliwi odszukanie wskazywanego przez odnośnik dokumentu.

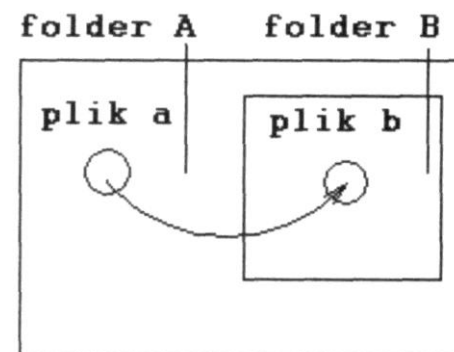
Uwaga:

W żadnym wypadku nie wolno stosować **bezwzględnych adresów dyskowych** np. `c:\Moje Dokumenty\plik.html`

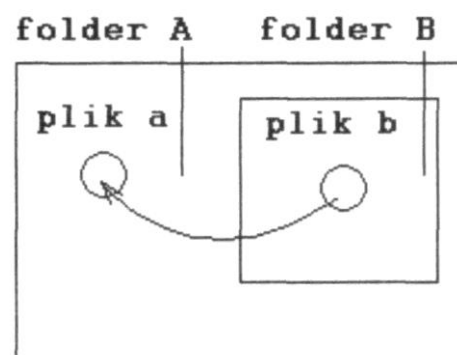
Ścieżki dostępu



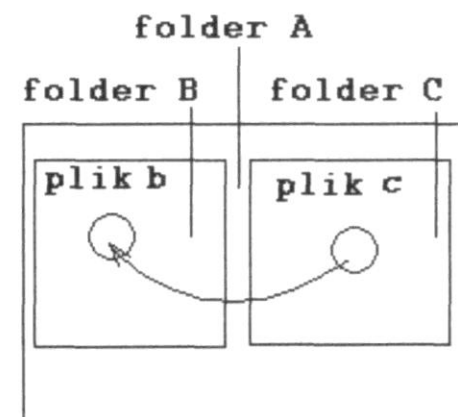
Adres względny: a2.html



Adres względny: B/b.html



Adres względny: ../a.html



Adres względny: ../B/b.html

Wartość atrybutu href	Znaczenie
Ścieżki względne	
href="a.html "	Plik a.html znajduje się w katalogu bieżącym.
href="A/a.html "	Plik a.html znajduje się w katalogu A, który znajduje się w katalogu bieżącym.
href="A/B/a.html "	Plik a.html znajduje się w katalogu B, który jest folderem podrzędnym katalogu A. Folder A jest z kolei umieszczony w katalogu bieżącym.
href="../a.html "	Plik a.html znajduje się w katalogu znajdującym się o poziom wyżej niż bieżący (tzn. w katalogu nadrzędnym).
href="../../a.html "	Plik a.html znajduje się w katalogu znajdującym się o dwa poziomy wyżej niż bieżący.
Ścieżki bezwzględne	
href="/B/A/a.html "	Plik a.html znajduje się w katalogu A, który znajduje się w katalogu B, który z kolei znajduje się w katalogu głównym.



Definiowanie odnośników

- ✓ Odnośnik do innego dokumentu:

```
<a href="url_dokumentu.html">Odnosnik</a>
```

- ✓ Odnośnik do innej strony WWW

```
<a href="http://www.helion.pl/">Wydawnictwa Helion</a>
```

/domyślnie wczytywany jest index.html

- ✓ Odnośnik do podstrony

```
<a href="http://www.bartoszewski.pr.radom.pl/pipu_d/index.html">  
Strona Artura Bartoszewskiego</a>
```

- ✓ odnośnik do pliku

```
<a href="http://www.bartoszewski.pr.radom.pl/nti_d/c0/xhtml1_cw_0  
.pdf">Odnosnik</a>
```

- ✓ Odnośnik pocztowy

```
<a href="mailto:jan.kowalski@moj.adres.pi">Wyślij do  
mnie list</a>
```



Definiowanie odnośników

Uwaga:

Struktura adresu URL zgodna jest z formatami systemu UNIX. Oznacza to, że wszystkie nazwy i ścieżki dostępu są czułe na wielkość znaku (za wyjątkiem nazw komputerów i adresów e-mail). Aby utworzyć odnośnik do pliku w katalogu podrzędnym bieżącego pliku, należy skorzystać z prawego ukośnika, / , który umieszczany jest między nazwą katalogu a nazwą pliku.

Oto przykład:

```
Lista <a href="2005/wszystkie.html">moich publikacji  
wydanych w 2007r</a>.
```




Definiowanie odnośników

Odnośnik „wmontowany” w akapit

<p>

Wyświetl **dodatkowe informacje**
na ten temat.

</p>

UWAGA:

Nie można umieszczać elementów `a` w innych elementach `a` ani otaczać znacznikami `<a>` elementów blokowych. Jedynie elementy liniowe mogą zawierać się w elemencie `<a>`. Z tego powodu nie można zrobić odnośnikiem na przykład całego wiersza tabeli bez powtarzania `a` w każdej komórce.



Literatura:

W prezentacji wykorzystano fragmenty i przykłady z książek:

- Sokół M.; *abc języka HTML i XHTML*. Helion, Gliwice 2006.
- Wyke-Smith Ch.; *CSS Witryny szyte na miarę*. Helion, Gliwice 2008.