

Tworzenie Stron Internetowych

odcinek 4

The background of the slide is black, featuring a complex pattern of glowing, multi-colored lines. These lines, in shades of orange, red, and white, radiate from a central point on the right side of the frame, creating a sense of dynamic movement and depth. The lines vary in thickness and brightness, some appearing as sharp, bright streaks while others are more diffuse and ethereal.

tabele

Tabela służy do uporządkowanego przedstawienia zbioru danych. Oprócz tego klasycznego zastosowania, może być wykorzystana do układania elementów na stronie (nie chodzi tu o układ strony).

Podstawową **strukturę tabeli** definiują następujące znaczniki:

<table>...</table> - znacznik tabeli

<tr>...</tr> - znacznik wiersza (*table row*)

<td>...</td> - znacznik pojedynczej komórki (*table data*)

np. tabelę o dwóch kolumnach i dwóch rzędach definiujemy następująco:

```
<table>
  <tr>
    <td>komórka1</td>  <td>komórka2</td>
  </tr>
  <tr>
    <td>komórka3</td>  <td>komórka4</td>
  </tr>
</table>
```

Liczba komórek w każdym wierszu musi być taka sama.

Elementy stanowiące treść tabeli mogą znajdować się **tylko wewnątrz** znaczników **<td>** i w tytule tabeli.

Zaleca się używania wcięć w kodzie tabeli. Ułatwia to jej edycję (przejrzystość).

W komórkach tabel można umieszczać inne elementy HTML, np. akapity, obrazy, listy

Tabele można zagnieżdżać, „podtabela” musi być umieszczona jako zawartość elementu **<td>**.

tabele

Uwaga. Domyślnie tabela zrobiona w html wyświetla się w przeglądarce bez żadnych linii oddzielających komórki. Linie te traktowane są jako warstwa prezentacyjna i wykonuje się je w CSS.

Wybiegając nieco do przodu, jeśli chcemy mieć proste obramowania w tabeli, należy **w kodzie html w części <body>** dodać następujący kod:

```
<style>
  table, th, td {
    border: 1px solid black;
    border-collapse: collapse;
  }
</style>
```

Inne cechy wyglądu tabeli zostaną omówione później, w trakcie zapoznawania z CSS.

tabele

Komórki nagłówkowe <th>

Służą do definiowania nagłówków kolumn lub / i wierszy. Praktycznie nie różnią od zwykłych komórek, ale semantycznie mają inne znaczenie. Tekst w tych komórkach jest domyślnie pogrubiony.

```
<table>
  <tr>
    <th>...</th>  <th>...</th>  <th>...</th>
  </tr>
  <tr>
    <th>...</th>  <td>...</td>  <td>...</td>
  </tr>
  <tr>
    <th>...</th>  <td>...</td>  <td>...</td>
  </tr>
</table>
```

Tytuł tabeli

Musi znajdować się bezpośrednio po znaczniku <table>.

```
<table>
  <caption>tytuł tabeli</caption>
  (...)
</table>
```

tabele

Łączenie komórek (**colspan** i **rowspan**)

Atrybuty te umożliwiają łączyć komórki w pionie i poziomie. Ich użycie zmniejsza ilość elementów **<td>**. Wartość nadawana tym atrybutom określa liczbę łączonych komórek (n).

```
colspan="n"   rowspan="n"
```

Przykład: połączenie dwóch komórek w pierwszym wierszu:

```
<table>
  <tr>
    <td colspan="2">komórki1,2</td>  <td>komórka3</td>
  </tr>
  <tr>
    <td>komórka4</td>  <td>komórka5</td>  <td>komórka6</td>
  </tr>
</table>
```

Przykład: połączenie dwóch komórek w pierwszej kolumnie:

```
<table>
  <tr>
    <td rowspan="2">komórki1,4</td>  <td>komórka2</td>  <td>komórka3</td>
  </tr>
  <tr>
    <td>komórka5</td>  <td>komórka6</td>
  </tr>
</table>
```

tabele

Podział tabeli na sekcje **<thead>**, **<tbody>**, **<tfoot>**

Elementy te umożliwiają grupowanie wierszy tabeli w część nagłówkową, część główną i stopkę. W jednej tabeli część nagłówkowa i stopka może wystąpić tylko raz. Kolejność tych elementów musi być jak w poniższym przykładzie. Jeśli mamy `<caption>`, to element ten musi być przed `<thead>`.

Przykład:

```
<table>
  <thead>
    <tr>
      <th>A</th>  <th>B</th>  <th>C</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>komórka1</td>  <td>komórka2</td>  <td>komórka3</td>
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <td>a</td>  <td>b</td>  <td>c</td>
    </tr>
  </tfoot>
</table>
```

W grupę można połączyć dowolną liczbę wierszy. Jeśli zdecydujemy się grupować wiersze, to każdy wiersz musi się znaleźć w jednej z grup. Grupowanie wierszy ma na celu ich wyróżnienie i łatwiejsze przeglądanie tabeli.

tabele

Łączenie kolumn w grupy `<colgroup>`

Element `<colgroup>` i `<col>` łączą wybrane kolumny w grupę/grupy. Łączenie to pozwala w prosty sposób nadać całej grupie wybrany wygląd. Element ten musi być umieszczony bezpośrednio po opcjonalnym elemencie `<caption>`, przed innymi elementami tabeli.

```
<table>
  <colgroup>
    <col span="2" style="background-color:green">
    <col style="background-color:yellow">
  </colgroup>
  <tr>
    <th>numer</th>
    <th>nazwa</th>
    <th>data</th>
  </tr>
  <tr>
    <td>34768</td>
    <td>XML</td>
    <td>21.03.2017</td>
  </tr>
</table>
```

Fragmety kodu wpisane na szaro stanowią o wyglądzie tabeli (tu kolor tła) i są częścią języka CSS.

Atrybut `span` określa liczbę kolumn do połączenia w grupę (domyślnie 1).

W grupę można połączyć dowolną liczbę kolumn, a w jednej tabeli może być wiele takich grup.

tabele

Zagnieżdżanie tabel

Tabele można zagnieżdżać w sobie. Tabela (tabela) **podrzędna** musi znajdować się w komórkach tabeli **nadrzędnej**. Zagnieżdżanie może być wielopoziomowe.

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

<table border="1"><tr><td>komórka1A</td><td>komórka1B</td></tr><tr><td>komórka1C</td><td>komórka1D</td></tr></table>	komórka1A	komórka1B	komórka1C	komórka1D	komórka2
komórka1A	komórka1B				
komórka1C	komórka1D				
komórka3	komórka4				

formularze

Formularze

Formularze umożliwiają zebranie danych / informacji od użytkownika i przesłanie ich na serwer. Są sposobem na przepływ informacji od użytkownika do autora. Prosty formularz nie zapewnia bezpieczeństwa przysyłanych danych (brak szyfrowania).

Elementami budującymi formularz są element **<form>** oraz **polo formularza**. Dostępne są różne typy pól (tekstowe, wyboru, ...).

```
<form>  
  pola formularza  
</form>
```

Sposób odesłania formularza wskazujemy atrybutem **action**. Może to być np. odesłanie na serwer w celu zapisania do pliku/bazy danych lub odesłanie na adres e-mailowy:

```
<form action="mailto: adres emailowy" method="post" enctype="text/plain">
```

Atrybut **method** określa sposób przesłania na serwer danych wpisanych w formularz:

- **post** – wysłanie w body komunikatu HTTP, służy do przesyłania dużej ilości danych i danych wrażliwych.
- **get** – wysłanie jako zmienne w URL. Domyślny. Nadaje się do przesyłania małej ilości danych niewrażliwych.

Parametr **enctype** (występuje tylko z **post**) pozwala zapisać dane uzyskane w formularzu w treść e-maila w sposób czytelny dla człowieka.

formularze

Pola formularza tworzone są elementami:

- **<input>** – umożliwia wpisanie informacji lub wybór opcji.

Opis pola `<input type="typ" name="nazwa" value="wartość" />`

gdzie **name** to nazwa pola, **value** zawiera: treść odsyłanej odpowiedzi (w przypadku pól wyboru odp. z listy), napis na przycisku (dla submit i reset) lub tekst domyślny (dla text i password). Atrybut **type** może mieć następujące wartości (wybrane przykłady):

- **text** – pole tekstowe (1 linia)

`<input type="text" name="imię" value="wpisz imię">`

- **password** – pole hasła (hasło nie jest szyfrowane)

Podaj hasło `<input type="password" name="hasło">`

- **radio** – pole opcji (można wybrać tylko jedną odpowiedź)

Wybierz rozmiar `<input type="radio" name="rozmiar" value="M"> M`

- **checkbox** – pole wyboru (można wybrać więcej niż jedną odpowiedź)

Wybierz kolor `<input type="checkbox" name="kolor1" value="niebieski"> niebieski`

- **submit** – wysyłanie formularza w sposób zadany w `<form>` atrybutem action (nie posiada atrybutu name, posiada atrybut value)

`<input type="submit" value="wyślij">`

- **reset** – wyczyszczenie całego formularza z wprowadzonych danych

`<input type="reset" value="wyczyść">`

formularze

Pola formularza tworzone są elementami (cd):

- **<label>** – definiuje opis do pola **<input>**. Opcjonalny atrybut **for** pozwala związać opis z polem.

```
<label for="op1">Opcja 1</label>
```

```
<input type="radio" name="wybor" id="op1" value="opcja1">
```

Lub krócej

```
<label>Opcja 1
```

```
  <input type="radio" name="wybor" value="opcja1">
```

```
</label>
```

- **<textarea>** – tworzy pole do wpisania tekstu wielolinijkowego.

```
<textarea name="nazwa" rows="x" cols="y" maxlength="n">tekst informujący</textarea>
```

gdzie **name** jest nazwą pola, **rows** i **cols** jego rozmiarem, **maxlength** maksymalną długością możliwą do wpisania (n znaków).

Opcjonalny *tekst informujący* zostanie wpisany w pole i może to być jakaś uwaga skierowana od użytkownika.

formularze

Pola formularza tworzone są elementami (cd):

- **<select>** – tworzy listę złożoną z kilku opcji do wyboru

```
<select name="nazwa_listy" size="liczba wyświetlanych pozycji" multiple="multiple">
  <option value="opcja1">pierwsza opcja</option>
  <option value="opcja2">druga opcja</option>
  ...
</select>
```

gdzie **name** jest nazwą listy, **size** liczbą wyświetlanych jednocześnie opcji. Jeśli obecny jest atrybut **multiple**, użytkownik ma możliwość wybrania wielu opcji jednocześnie. Atrybut **value** podaje wartość odpowiedzi, która będzie odesłana.

- **<button>** – tworzy klikalny przycisk

```
<button type="button" onclick="alert('naciskasz i masz!')">Naciśnij tu</button>
```

Powyższa linia kodu zawiera JavaScript.

Dodatkowe uwagi:

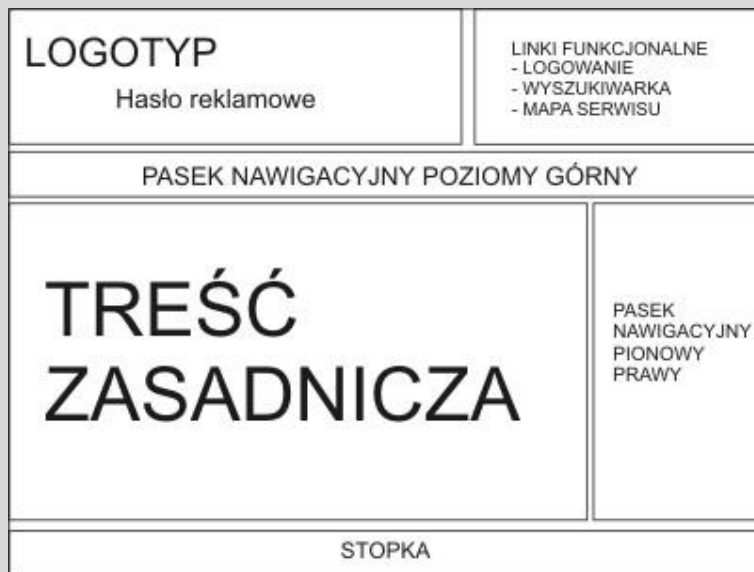
- Pola formularza można grupować i opisywać za pomocą **<fieldset>** i **<legend>**.
- Zdania w formularzu, pola, zestawy opcji itd. należy obejmować jakimiś znacznikami, np. akapitu `<p>` czy bloku `<div>`.

tworzenie układu strony

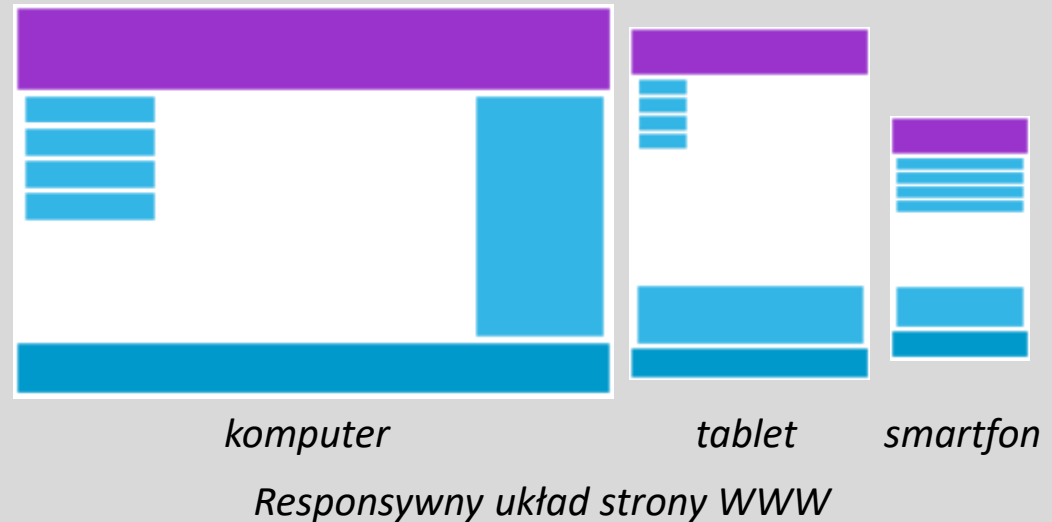
Układ strony WWW to jej podział na kilka obszarów pełniących różne funkcje, np. nagłówek z obrazem-banerem, menu nawigacyjne, treść główna, stopka, itp. Układ tych obszarów może zależeć od wielkości urządzenia na którym wyświetlana jest strona WWW (responsywność).

Kilka uwag do budowania układu strony:

<http://webownia.pl/blog/uklad-strony-schematy-nie-zawsze-zle.html>



Przykładowy układ strony



tworzenie układu strony

Do tej pory układ strony budowany był na **<div>** z atrybutem **id** (nadal można tak robić), np.:

```
<div id="header"> nagłówek </div>
```

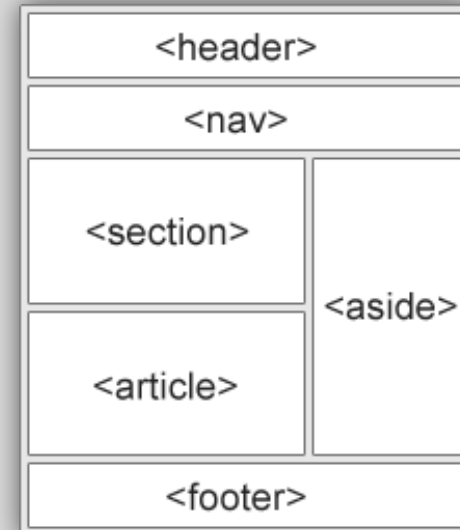
```
<div id="menu"> menu nawigacyjne </div>
```

```
<div id="content"> treść główna strony </div>
```

```
<div id="footer"> stopka </div>
```

W HTML5 dostępne są nowe (blokowe) elementy semantyczne służące budowaniu układu stron internetowych. Te elementy to:

- **<header>** – zawiera nagłówek strony, wprowadzenie do niej (może być tekst lub obraz)
- **<nav>** – definiuje zestaw odnośników będących menu nawigacyjnym
- **<main>** – element z główną, unikalną zawartość strony
- **<section>** – element definiujący części dokumentu (np. rozdział)
- **<article>** – element zawierający treść niezależną, mogącą istnieć bez reszty strony (np. wpis na forum, komentarz, news)
- **<aside>** – definiuje zawartość poboczną, ale związaną tematycznie z resztą strony (np. przypisy)
- **<footer>** – zawiera informację na temat samej strony (np. podaje autora, zasady wykorzystania treści, dane kontaktowe, ...)



Więcej o układzie strony WWW omówione będzie przy CSS.

tworzenie układu strony

Menu nawigacyjne możemy utworzyć wykorzystując listę nieuporządkowaną `` (poprawność semantyczna) lub bez użycia listy.

Przykład 1 (z ``):

```
<ul>
  <li><a href="index.html">strona główna</a></li>
  <li><a href="metody.html">metody pomiarowe</a></li>
  <li><a href="odnosniki.html">odnośniki</a></li>
  <li><a href="kontakt.html">kontakt</a></li>
</ul>
```

Przykład 2 (bez ``):

```
<div>
  <a href="index.html">strona główna</a>
  <a href="metody.html">metody pomiarowe</a>
  <a href="odnosniki.html">odnośniki</a>
  <a href="kontakt.html">kontakt</a>
</div>
```

