



Elementarz HTML i CSS

Znaczniki przydatne w pracy redaktora stron internetowych

Elementarz HTML i CSS

Znaczniki przydatne w pracy redaktora stron internetowych

Treść, korekta, skład i oprawa graficzna

dr inż. Karol Król

HomeProject.pl

Materiały zaczerpnięte

z HTML Tutorial — W3Schools

Kontakt

k.krol@onet.com.pl

Data publikacji

Kraków, 2018

Spis treści

Przedmowa: Dlaczego HTML i CSS?	4
1. Słownik skrótów	5
2. Trójstopniowa architektura web	6
3. Struktura dokumentu hipertekstowego	7
4. Element blokowy DIV	8
5. Wstawianie linków i plików graficznych	9
6. Kaskadowe arkusze stylów (CSS)	10
6a. Sposoby wstawiania stylów	11
6b. Selektory CSS: identyfikator „id”	12
6c. Selektory CSS: klasa „class”	13
6d. Grupowanie selektorów	14
7. Tabele	15
8. Listy HTML i właściwości listy	16
9. Metainformacje	17
10. Walidacja: poprawność składniowa kodu	18

Przedmowa: Dlaczego HTML i CSS?

Większość nowoczesnych witryn internetowych tworzonych jest w oparciu o system zarządzania treścią (CMS). Pozwala on zarządzać treścią witryny bez znajomości HTML i CSS. Znajomość języka znaczników, choćby w podstawowym zakresie, może być jednak przydatna w codziennej pracy redaktora. Wybrane formatowanie tekstu, elementów witryny lub modyfikacje CMS mogą bowiem wymagać ingerencji w kod źródłowy.

Zachęcam do zapoznania się z podstawowym zestawem znaczników HTML i CSS. W ręce redaktorów oddaję niniejszy elementarz z nadzieją, że będzie on pomocny w codziennej pracy.

Autor



1. Słownik skrótów

CSS — Cascading Style Sheets (kaskadowe arkusze stylów)

CMS — Content Management System (system zarządzania treścią)

DTD — Document Type Definition (deklaracja typu dokumentu)

HTML — HyperText Markup Language (hipertekstowy język znaczników)

W3C — World Wide Web Consortium

XHTML — Extensible HyperText Markup Language (rozszerzalny język znaczników hipertekstowych)

Więcej przykładów zastosowania znaczników HTML i CSS znaleźć można na stronie internetowej

www.w3schools.com

2. Trójstopniowa architektura web

Witryny internetowe tworzone są w oparciu o trójstopniową architekturę złożoną z komplementarnych warstw:

- ❖ **struktury**, tzw. „rusztowania”, której podstawą jest kod HTML. Analiza tej warstwy pozwala odpowiedzieć na pytanie „Co jest czym?”;
- ❖ **wyglądu**. Za oprawę wizualną odpowiada kod CSS, który nadaje format obiektom utworzonym przy pomocy HTML. Analiza tej warstwy pozwala odpowiedzieć na pytanie „Jak to wygląda?”;
- ❖ **zachowania**. Za interaktywność odpowiadają języki skryptowe. Analiza tej warstwy pozwala odpowiedzieć na pytanie „Jak to działa?”.

Strukturę dokumentu hipertekstowego tworzą znaczniki HTML, m.in.: akapity <p>, nagłówki <h1>, tabele <table> i pojemniki <div>. Natomiast formatowanie tych elementów np. rozmiar, kolor lub położenie definiowane jest przy pomocy kaskadowych arkuszy stylów CSS. Interaktywność i funkcjonalność uzyskiwana jest przy pomocy języków skryptowych.



3. Struktura dokumentu hipertekstowego

Prosty dokument hipertekstowy (HTML5) ma następującą strukturę:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Tytuł</title>
```

```
</head>
```

```
<body>
```

```
<h1>Nagłówek</h1>
```

```
<p>Paragraf</p>
```

```
</body>
```

```
</html>
```

Deklaracja DTD `<!DOCTYPE html>` definiuje dokument jako HTML5. Element `<html>` jest głównym elementem strony HTML.

Element `<head>` zawiera metadane.

Element `<title>` zawiera tytuł dokumentu.

Element `<body>` zawiera treści, które są prezentowane w oknie przeglądarki internetowej.

Element `<h1>` wstawia duży nagłówek.

Element `<p>` definiuje akapit.

Znaczniki HTML dzielimy na takie, które mają niezależne domknięcie np. otwarcie `<div>` i zamknięcie `</div>`, otwarcie `<p>` i zamknięcie `</p>` oraz na znaczniki, które „zamykają się same w sobie” np. znacznik `
` lub ``.

W dokumencie HTML można wstawiać komentarze. Komentarze nie są interpretowane jako kod HTML.



```
<!-- To jest komentarz -->
```

4. Element blokowy DIV

Element blokowy (*block-level element*) `<div> ...</div>` to jeden z podstawowych znaczników HTML.

Element blokowy zawsze zaczyna się od nowej linii i przyjmuje całą dostępną szerokość (rozciąga się na lewo i na prawo tak daleko, jak to możliwe).*

Element `<div>` jest często używany jako pojemnik (kontener) dla innych elementów HTML.

Wskazówka: Element `<div>` jest często używany razem z arkuszami stylów CSS, do tworzenia struktury strony internetowej.

Przykład tekstu w pojemniku `<div>` o określonym stylu (obramowanie koloru pomarańczowego o szerokości 1px).

*Przeciwieństwem elementów blokowych są elementy wbudowane (*inline element*). Przykładem znacznika, który otwiera element wbudowany jest znacznik ``. Element wbudowany nie jest rozpoczynany od nowej linii i zajmuje jedynie tyle miejsca ile potrzebuje.

To jest **przykład** zastosowania znacznika `` o określonych atrybutach.

Domyślnie znacznik `<div>` jest wyświetlany w oknie przeglądarki z atrybutem: `{display: block}`.

W specyfikacji HTML5 atrybut „align” nie jest już wspierany (nie powinien być wykorzystywany jako atrybut znacznika `<div>`).

5. Wstawianie linków i plików graficznych

W dokumencie hipertekstowym linki, zwane też odnośnikami lub hiperłączami, są wstawiane przy pomocy znacznika <a>:

```
<a href="URL">tekst linku</a>
```

Linkiem może być również plik graficzny:

```
<a href="index.html">  
    
</a>
```

Znacznik wstawia w dokumencie HTML plik graficzny:

```

```

Wartość atrybutu alt powinna opisywać obraz.

Atrybut alt jest wymagany!

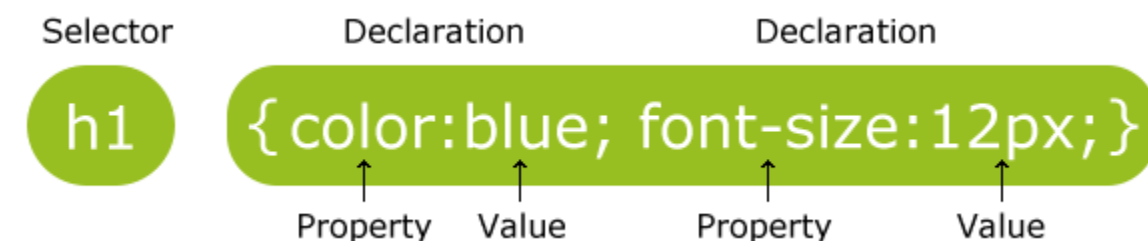


6. Kaskadowe arkusze stylów (CSS)

Kaskadowe arkusze stylów CSS to język opisujący styl dokumentu HTML. CSS opisuje sposób wyświetlania elementów HTML.

Zewnętrzne arkusze stylów są zapisywane w plikach CSS.

Selektor wskazuje na element HTML, który ma przyjąć określony styl.



Blok deklaracji zawiera jedną lub więcej deklaracji rozdzielonych średnikami. Deklaracje opisują atrybuty poszczególnych elementów HTML. Każda deklaracja zawiera nazwę właściwości CSS (*property name*) i wartość (*value*) występują po dwukropku.

Deklaracja CSS zawsze kończy się średnikiem, a bloki deklaracji zawierają się w nawiasie klamrowym.

W poniższym przykładzie zadeklarowano, że tekst umieszczony wewnątrz paragrafu <p> będzie wyrównany do środka i będzie koloru czerwonego:

```
p {  
  color: red;  
  text-align: center;  
}
```

6a. Sposoby wstawiania stylów

Są trzy sposoby wstawiania stylów CSS:

❖ zewnętrzny arkusz stylów (*external style sheet*)

Style przechowywane są w zewnętrznym pliku CSS i z niego pobierane:

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

Zewnętrzny arkusz stylów można zapisać w dowolnym edytorze tekstu. Plik nie powinien zawierać żadnych znaczników HTML. Plik arkusza stylów musi zostać zapisany z rozszerzeniem .css.

❖ inline style

Styl inline może być użyty do zastosowania unikalnego stylu dla pojedynczego elementu.



```
<h1 style="color: brown; margin-left: 30px;">Kursant</h1>
```

❖ style wewnętrzne (*internal style sheet*)

Style wewnętrzne są zdefiniowane w elemencie <style>, w sekcji <head> strony HTML:

```
<head>
<style>
h1 {
    color: maroon;
    margin-left: 40px;
}
</style>
</head>
```

6b. Selektory CSS: identyfikator „id”

Selektor id (*the id selector*) wykorzystuje identyfikator elementu HTML do wybrania konkretnego elementu.

Do wyboru elementu HTML o określonym id wykorzystywany jest znak „kratka” (hash #). Zaraz po nim zapisywana jest nazwa (wartość atrybutu) id.

Identyfikator elementu powinien być unikalny. Selektor id służy do wyboru jednego unikalnego elementu.



Nazwa id nie może zaczynać się od cyfry!



Poniższa reguła stylu zostanie zastosowana do elementu <div> o identyfikatorze id="kursant":

Kod HTML:

```
<div id="kursant">
```

Kod CSS:

```
#kursant {  
    border: 1px solid red;  
    width: 300px;  
    text-align: center;  
    color: red;  
}
```

6c. Selektory CSS: klasa „class”

Selektor klasy (class) wybiera elementy o określonym atrybucie klasy. Aby wybrać elementy z określoną klasą, nazwę klasy należy poprzedzić znakiem kropki (.). W poniższym przykładzie wszystkie elementy HTML opisane klasą `class="kursant"` i prezentujące tekst, będą go wyświetlały w kolorze czerwonym:

```
.kursant {  
  color: red;  
}
```

Można również zastrzec, że klasa będzie mieć wpływ tylko na określone elementy HTML. W poniższym przykładzie formatowanie obejmie jedynie element `<p>` opisany klasą `class="kursant"`:

```
p.kursant {  
  text-align: center;  
  color: red;  
}
```

Atrybut class może być użyty wielokrotnie i na dowolnym elemencie HTML.



Nazwa class nie może zaczynać się od cyfry!



6d. Grupowanie selektorów

W przypadku gdy dane selektory mają jednakowe formatowanie (jednakowe atrybuty i ich wartości) np.

```
h1 {  
  text-align: center;  
  color: red;  
}  
h2 {  
  text-align: center;  
  color: red;  
}  
p {  
  text-align: center;  
  color: red;  
}
```



wskazane jest grupowanie selektorów, co pozwala zmniejszyć ilość kodu:

```
h1, h2, p {  
  text-align: center;  
  color: red;  
}
```

Selektory oddzielane są przecinkami.

```
h1, h2, p { ... }
```

W dokumencie CSS można wstawiać komentarze. Komentarze nie są interpretowane jako kod CSS.

```
/* To jest komentarz */
```



7. Tabele

Do wstawiania tabel w dokumencie hipertekstowym służy znacznik <table>.

Wiersze tabeli tworzone są przy pomocy znacznika <tr> a komórki przy pomocy znacznika <td>.

Prosta tabela HTML:

```
<table border="1">  
<tr>  
<td>komórka 1</td><td>komórka 2</td>  
</tr>  
</table>
```

komórka 1	komórka 2
-----------	-----------

Tabele można formatować przy pomocy CSS:

```
table, th, td {  
  border: 1px solid black;  
}
```

komórka 1	komórka 2
-----------	-----------

8. Listy HTML i właściwości listy

W HTML istnieją dwa główne typy list:

- ❖ nieuporządkowane listy () — elementy listy są oznaczone punktorem,
- ❖ uporządkowane listy () — elementy listy są oznaczone cyframi lub literami.

Właściwości listy CSS umożliwiają:

- ❖ ustawienie różnych znaczników pozycji listy dla uporządkowanych i nieuporządkowanych list,
- ❖ ustawienie grafiki jako znacznika pozycji listy,
- ❖ dodanie koloru tła do list i jej elementów.

Przykład listy (kod HTML):

```
<ul>  
<li>kursant 1</li>  
<li>kursant 2</li>  
<li>kursant 3</li>  
</ul>
```

i prezentacja listy w oknie przeglądarki:

- kursant 1
- kursant 2
- kursant 3

9. Metainformacje

Element `<head>` jest kontenerem na metadane (dane dotyczące danych) i jest umieszczany pomiędzy znacznikiem `<html>` i znacznikiem `<body>`.

Metadane HTML to dane dotyczące dokumentu HTML. Nie są one wyświetlane w oknie przeglądarki internetowej.

Metadane zazwyczaj definiują tytuł dokumentu, zestaw znaków (tzw. stronę kodową dokumentu), style, linki, skrypty i inne metadane.

Następujące znaczniki opisują metadane:

`<title>`, `<style>`, `<meta>`, `<link>`, `<script>`, `<base>`.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Tytuł strony</title>
```

```
</head>
```

```
<body>
```

```
Treści...
```

```
</body>
```

```
</html>
```

10. Walidacja: poprawność składniowa kodu

Walidacja to proces, podczas którego dokumenty hipertekstowe są sprawdzane pod kątem zgodności struktury, użytych elementów i atrybutów z definicjami zawartymi w DTD.

Dokument (hipertekstowy) jest poprawny składniowo, gdy posiada strukturę zgodną z zasadami zdefiniowanymi w specyfikacji.

Jednym z podstawowych (rekomendowanych) narzędzi walidujących jest aplikacja (usługa) W3C Markup Validation Service.

Walidator HTML

<https://validator.w3.org/>

Walidator CSS

<https://jigsaw.w3.org/css-validator/>

Walidacja w naukach technicznych i informatyce oznacza czynności podejmowane w celu potwierdzenia, że użyte procedury, techniki i narzędzia prowadzą do zaplanowanych wyników.



