



Podstawy HTML i CSS

Bartosz Wrzałka

Czym jest HTML?

HTML (ang. HyperText Markup Language) – hipertekstowy język znaczników, obecnie szeroko wykorzystywany do tworzenia stron internetowych.

Język znaczników – format dokumentu, który obok tekstu zawiera także dodatkowe informacje, które go opisują. Te dodatkowe informacje, na przykład na temat wyglądu czy znaczenia fragmentu dokumentu, są wyrażane poprzez wplecione w tekst znaczniki.



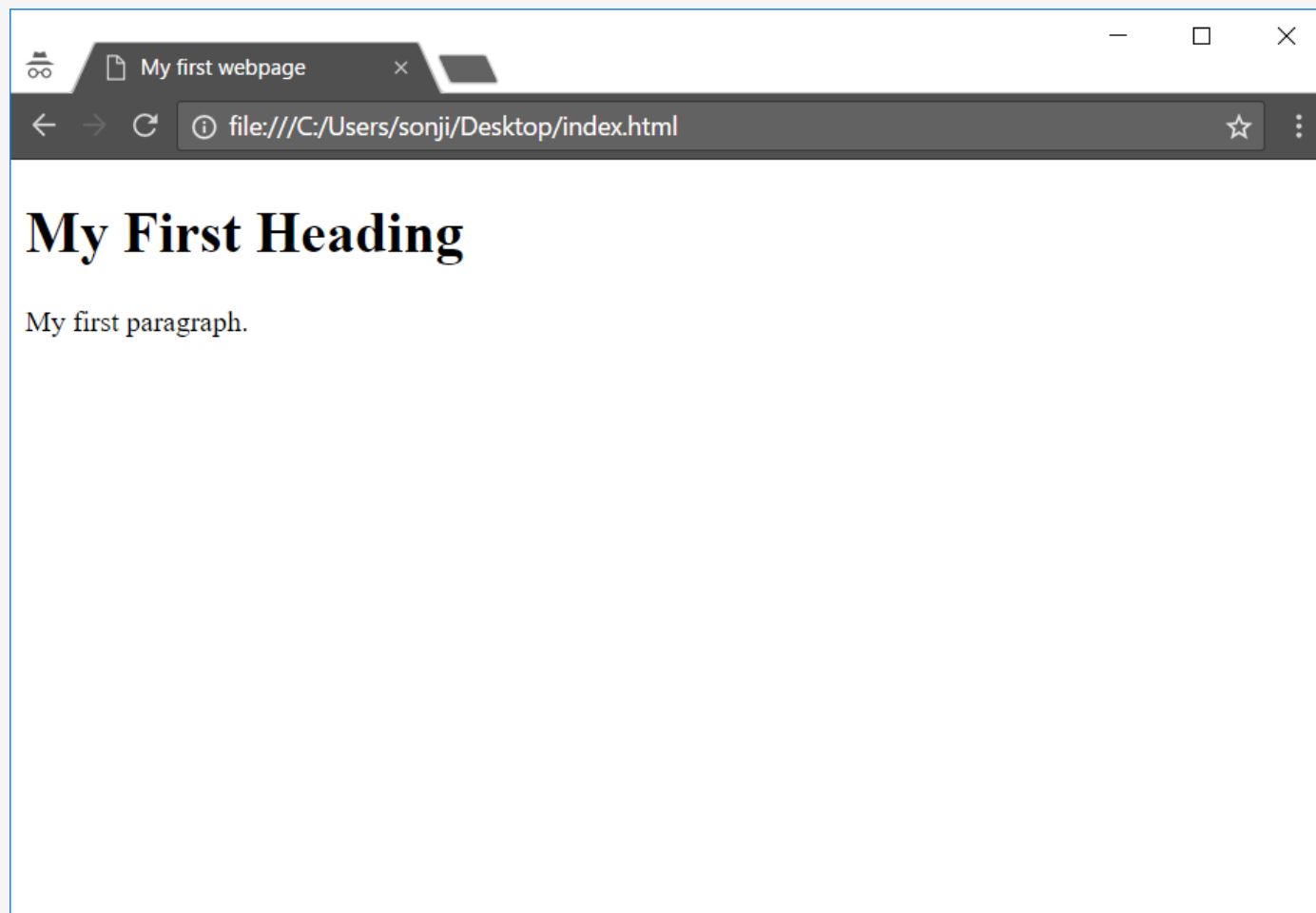


Jak wygląda kod HTML?

```
<!DOCTYPE html>
<html>
<head>
  <title>My first webpage</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <h1>My First Heading</h1>
  <p>My first paragraph.</p>
</body>
</html>
```



Efekt?





Znaczniki HTML

Zapoznajmy się po kolei z tymi najczęściej stosowanymi.

Znaczniki główne

<html> - znacznik określający początek i koniec dokumentu HTML

<head> - znacznik określający początek i koniec nagłówka dokumentu

<body> - znacznik określający początek i koniec zawartości dokumentu



Znaczniki strukturalne

`<div>` - podstawowy kontener treściowy

`<p>` - akapit

`<h1>` - `<h6>` - nagłówki dokumentu



Znaczniki prezentacyjne

**** - wzmocnienie treści

** - wyróżnienie treści

<sub> - indeks dolny

<sup> - indeks górny

`<code>` - tekst czcionką o stałej szerokości znaków

~~~~ - tekst przekreślony



Inne znaczniki

`` - wstawianie obrazków

`` - odróżnienie inline

`<a>` - wstawianie łącza do innego miejsca

`<canvas>` - element, po którym można „malować” skryptami

`<script>` - wstawianie skryptów wewnątrz kodu HTML

`<style>` - wstawianie stylów wewnątrz kodu HTML



Inne znaczniki – cd.

`
` - złamanie linii

`<audio>` - wstawianie dźwięku

`<video>` - wstawianie filmu

`<title>` - nagłówek dokumentu

`<meta>` - dodatkowe informacje o dokumencie

`<iframe>` - załadowanie zawartości innej strony



Znaczniki HTML5

`<section>` - sekcja dokumentu

`<article>` - artykuł na stronie

`<aside>` - treść spoza głównego „flow” dokumentu

`<header>` - nagłówek strony (nie mylić z h1-h6)

`<footer>` - stopka dokumentu

`<nav>` -nawigacja



Znaczniki HTML5 – cd.

`<main>` - główna część dokumentu

`<details>` - dodatkowe informacje

`<summary>` - podsumowanie treści ukrytych w `<details>`

`<progress>` - dodatkowe informacje

`<dialog>` - wyskakujące okienko

`<menuitem>` - element nawigacji



Dlaczego wybór znaczników jest ważny?

1. Przestrzeganie ustalonych standardów.
2. Czytelność kodu – samo opisujący się kod.
3. Domyślne typy wyświetlania.
4. Im lepszy dobór znaczników tym mniej stylowania.
5. Pozycjonowanie.





Zagnieżdżanie znaczników

Jak powinniśmy to robić?



XHTML i HTML - różnice

HTML

CSS i DOM JavaScript nie rozróżnia wielkości liter w nazwach elementów

Niektóre elementy są zawsze puste i nie mają tagu zamykającego `` `
`

dopuszcza skrócone atrybuty, np. `<input checked>`

Ma niezgodną z XML składnię, przez co nie można go bezpośrednio mieszać z XML-owymi językami

XHTML

CSS i DOM działa tylko z nazwami elementów pisanymi dokładnie tak, jak w dokumencie XHTML

Puste są te elementy, które autor zapisze w postaci `<element/>` lub `<element></element>`

XHTML wymaga zawsze atrybutów z wartością, np. `<input checked="checked"/>`

XHTML jest zgodny z XML i ma swoją przestrzeń nazw. Może być łączony z Atom, SVG, MathML, RDF i innymi.

Zadanie 1

Stwórz dokument HTML zgodny z HTML5 zawierający:

- 3 różne znaczniki h1...h6
- 2 paragrafy, które zawierają łamanie tekstów
- teksty wyróżnione na 2 różne sposoby
- tytuł strony





Listy

 Unordered list

Link

Link

Link

· Link

· Link

· Link

 Ordered list

Link

Link

Link

1 Link

2 Link

3 Link

Wyświetlenie dokumentu - block



```
<div style="background:#6ADDFE; height:100px; display:block;"></div>  
<div style="background:#431C6F; height:100px;"></div>  
<div style="background:#F09B00; height:100px;"></div>  
<div style="background:#5A5A55; height:100px;"></div>
```

Wyświetlenie dokumentu – inline-block



```
<div style="background:#6ADDFE; height:100px; display:inline-block;"></div>  
<div style="background:#431C6F; height:100px; display:inline-block;"></div>  
<div style="background:#F09B00; height:100px; display:inline-block;"></div>  
<div style="background:#5A5A55; height:100px; display:inline-block;"></div>
```

Wyświetlenie dokumentu – inline



Tekst Tekst Tekst Tekst

```
<div style="background:#6ADDFD; height:100px; display:inline;">Tekst</div>  
<div style="background:#431C6F; height:100px; display:inline;">Tekst</div>  
<div style="background:#F09B00; height:100px; display:inline;">Tekst</div>  
<div style="background:#5A5A55; height:100px; display:inline;">Tekst</div>
```

Zadanie

Ułóż 3 divy (nadaj im wysokość, tło i tekst w środku) używając display typu:

- block;
- inline;
- inline-block



Tabele



```
<table>
  <thead>
    <tr>
      <th>Column 1</th>
      <th>Column 2</th>
      <th>Column 3</th>
      <th>Column 4</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Cell 1-1</td>
      <td>Cell 1-2</td>
      <td>Cell 1-3</td>
      <td>Cell 1-4</td>
    </tr>
    <tr>
      <td>Cell 2-1</td>
      <td>Cell 2-2</td>
      <td>Cell 2-3</td>
      <td>Cell 2-4</td>
    </tr>
  </tbody>
</table>
```

Column 1	Column 2	Column 3	Column 4
Cell 1-1	Cell 1-2	Cell 1-3	Cell 1-4
Cell 2-1	Cell 2-2	Cell 2-3	Cell 2-4

Tabele



```
<table>
  <thead>
    <tr>
      <th>Column 1</th>
      <th>Column 2</th>
      <th>Column 3</th>
      <th>Column 4</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Cell 1-1</td>
      <td colspan="2">Cell 1-2</td>
      <td rowspan="2">Cell 1-3</td>
    </tr>
    <tr>
      <td>Cell 2-1</td>
      <td>Cell 2-2</td>
      <td>Cell 2-3</td>
    </tr>
  </tbody>
</table>
```

Column 1	Column 2	Column 3	Column 4
Cell 1-1	Cell 1-2		Cell 1-4
Cell 2-1	Cell 2-2	Cell 2-3	



Column 1	Column 2	Column 3	Column 4
Cell 1-1	Cell 1-2		Cell 1-4
Cell 2-1	Cell 2-2	Cell 2-3	

td 203px × 198px



Atrybuty znaczników - przykłady

```
<div id="main-element"></div>
```

```
<div class="main-class"></div>
```

```
<div style="font-size:20px"></div>
```

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

```
<a href="#" title="link"></a>
```

```

```

```
<input role="search" />
```

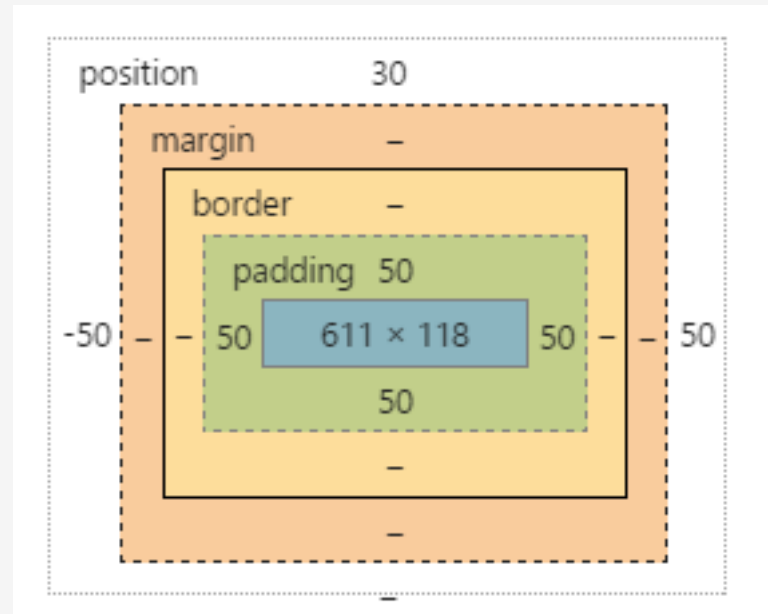
```
<input data-visibility="hidden" />
```



Atrybuty znaczników – do czego służą?

```
<div atrybut="wartość"></div>
```

1. Nadawanie cech znacznikom
2. Odróżnianie znaczników od innych tego typu(class, id)
3. Grupowanie elementów (class)
4. Przechowywanie dodatkowej informacji (data)
5. Podawanie źródeł zawartości (img, iframe)
6. Wyszukiwanie elementów w CSS oraz Javascript





Formularze – znacznik form

1. Jest opakowaniem dla wysyłanych danych
2. Jego główne atrybuty:
 - action
 - pusta wartość
 - link względny/bezwzględny
 - method:
 - get
 - Post



Formularze – rodzaje znacznika input

Rodzaj inputa ustawiamy poprzez atrybut type

- text

```
<input type="text"/>
```
- checkbox

```
<input type="checkbox" checked/>
```
- radio

```
<input type="radio" checked/>
```
- hidden

```
<input type="hidden" value="wartosc1"/>
```
- submit

```
<input type="submit" value="Zapisz"/>
```
- file

```
<input type="file"/>
```
- password

Fragment formularza HTML, który wizualizuje różne typy elementów input:

- Przycisk tekstowy (text input).
- Przycisk wyboru (checkbox), który jest zaznaczony.
- Przycisk wyboru (radio button), który jest zaznaczony.
- Przycisk Zapisz (submit button).
- Przycisk Wybierz plik (file input) i komunikat Nie wybrano pliku.



Formularze – znacznik textarea i select

- textarea

```
<textarea>Tekst</textarea>
```

- select

```
<select>  
  <option value="1">wartość 1</option>  
  <option value="2">wartość 2</option>  
  <option value="3">wartość 3</option>  
</select>
```

Zadanie

Stwórz dokument HTML zawierający:

- Formularz wysyłający dane do URL strony
- Select z kilkoma wartościami
- Inputy typu: text, radio (3x), checkbox, hidden, submit





CSS – Nazewnictwo klas

1. Wielkość liter ma znaczenie

```
<div class="wrapper">Tekst</div>
<style>
  .wrapper {
    background:red;
  }
</style>
```

*Basically*¹, a name must begin with an underscore (`_`), a hyphen (`-`), or a letter (`a` – `z`), followed by any number of hyphens, underscores, letters, or numbers. There is a catch: if the first character is a hyphen, the second character must² be a letter or underscore, and the name must be at least 2 characters long.

```
-?[_a-zA-Z]+[_a-zA-Z0-9-]*
```




Narzędzia developerskie – F12

🔍 📱

Elements Sources Console Network Timeline Profiles Resources Security Audits

<!DOCTYPE html>
<html>
 <head>...</head>
 <body>
 <div class="wrapper">Tekst</div>
 <style>
 .wrapper {
 background:red;
 }
 </style>
 <form action>...</form>
 </body>
</html>

Styles Computed Event Listeners DOM Breakpoints Properties

Filter + 📌 🔍

element.style {
}
div { zzzz.html:55
 font-size: 24px;
}
div { user agent stylesheet
 display: block;
}

margin -
border -
padding -
1016 × 27

html body div.wrapper



Na czym polega kaskadowość?

System (!NICE) = (!important, iNline, Id, Class, Element)

$(1,0,0,0,0) > (0,1,0,0,0) > (0,0,1,1,0) > (0,0,1,0,1) > (0,0,0,0,1)$



CSS – Jednostki wielkości

Px – piksele

Em – obliczana przez przeglądarkę na podstawie font-size rodzica

Rem – obliczana przez przeglądarkę na podstawie font-size najwyżej w hierarchii, czyli html

Vw – 1% szerokości ekranu

Vh – 1% wysokości ekranu

Vmin – 1% mniejszego wymiaru ekranu

Vmax – 1% większego wymiaru ekranu

cm, mm, in, pt, pc, ex, ch

CSS – sposoby dołączania stylów do dokumentu



Inline

```
<div style="font-size:20px"></div>
```

Znacznik style

```
<style>
  div {
    font-size:24px;
  }
</style>
```

Dołączanie plików stylów

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



Domyślnie – static



CSS – pozycjonowanie – relative



Relative - pozycjonowanie elementu oraz dzieci względem niego



CSS – pozycjonowanie – absolute



STATIC
ABSOLUTE



CSS – pozycjonowanie - fixed





```
color: #000000;  
text-decoration: underline;  
font-size: 24px;  
line-height: 100%;  
font-style: italic;  
font-weight: bold;  
font-family: 'Arial';  
text-transform: uppercase;  
letter-spacing: 3px;
```



Obrazek czy tło?

``

- Zalety:
- Grafikę można 'chwycić', zapisać obrazek
- Wstawiamy do HTML'a źródło obrazka, co sprzyja w zmianach treści przez klienta.
- Łatwiej zastosować lazy loading
- Wady:
- Problem z skalowalnością, szczególnie przy RWD, trzeba wspomagać się skryptami które podmieniają obrazki w zależności od rozdzielczości
- Problem z powiększeniem i ucięciem powiększonego fragmentu który wychodzi po za 'ramkę'

`background:url('image.jpg');`

- Zalety:
- Łatwo zastosować media query (RWD)
- Łatwo skalować obrazek, background-size;
- Wady:
- Jest definiowany w CSS w arkuszu, lub jako inline – ciężko go podmieniać.
- Trudno zastosować lazy loading



Dynamiczne

```
a:link {color: blue;}  
a:visited {color: purple;}  
a:hover {color: red;}  
a:active {color: yellow;}
```

Strukturalne

```
li:nth-child(2) {color: purple;}  
li:first-child {color: purple;}  
li:last-child {color: purple;}  
li:only-child {color: purple;}  
li:only-of-type {color: purple;}  
li:empty {color: purple;}
```

Inne

```
input:disabled {color: purple;}  
input:checked {color: purple;}  
input:valid {color: purple;}  
input:not(.valid) {color: purple;}
```



:after :before

1. Dlaczego warto stosować?

- Zmniejszenie HTML
- Bez dostępu do HTML możemy wprowadzać dodatkowe elementy
- Element może mieć tekst (problem gdy są tłumaczenia)

2. Jak użyć?

```
.button:after {  
  content: "";  
  display: inline-block;  
  margin-left: 10px;  
  background: url('icon-arrow.png');  
  width: 20px;  
  height: 10px;  
}
```

Zadanie

Stwórz listę `` z 6 elementami `` i zmień dowolne atrybuty (kolor, czcionka) dla:

- Pierwsze elementu
- $2n+1$ elementu
- Ostatniego elementu



Zadanie

Ostyluj elementowi <a> wszystkie z 4 akcji:

- Link
- Visited
- Hover
- active





CSS – krótsze zapisy - porady

Margin/padding

```
margin-left:25px;  
margin-right:25px;  
margin-top:50px;  
margin-bottom:50px;  
  
margin: 50px 25px;  
margin: 50px;
```

Przy 0 nie stosujemy jednostki

```
margin:0;
```

Border

```
border-width: 1px;  
border-style: solid;  
border-color:#000000;  
border:1px solid #000000;
```

Dla elementów blokowych (np. <div>)
nie musimy stosować ponownie reguł

```
div {  
    width:100%;  
    display: block;
```

Tło

```
background: url('image.png') center top no-repeat ;  
  
background:url('image.png');  
background-position: center top;  
background-repeat: no-repeat;
```



Media queries

Sposób na tworzenie responsywnych stron



Użycie media queries

```
1  <!-- CSS media query on a link element -->
2  <link rel="stylesheet" media="(max-width: 800px)" href="example.css" />
3
4  <!-- CSS media query within a stylesheet -->
5  <style>
6  @media (max-width: 600px) {
7      .facet_sidebar {
8          display: none;
9      }
10 }
11 </style>
```



Flex

Nareszcie metoda dostosowana do dzisiejszych stron



Flexbox froggy

Nauka przez zabawę :D