

## **Laboratorium 1.**

### **HTML5 i CSS3. Znaczniki HTML i struktura dokumentu**

#### **1. CEL ĆWICZENIA**

Celem ćwiczenia jest zapoznanie się ze składnią języka HTML5 oraz CSS3 i w oparciu o te wiadomości przygotowanie prostej aplikacji internetowej składającej się z zestawu stron w języku HTML.

#### **2. MATERIAŁ NAUCZANIA**

##### **2.1. Wstęp**

Język **HTML** (ang. *HyperText Markup Language*) jest językiem wykorzystywanym do tworzenia dokumentów strony internetowej. Nie jest on językiem programowania. Należy do rodziny języków nazywanych **językami znaczników** (ang. *markup language*). Oznacza to, że jest systemem identyfikacji oraz opisu różnych komponentów dokumentu, takich jak np. nagłówki, akapity czy listy. Podczas gdy HTML wykorzystywany jest do tworzenia struktury i zawartości strony internetowej, **kaskadowe arkusze stylów CSS** (ang. *Cascading Style Sheets*) opisują to, w jaki sposób zawartość ta ma wyglądać. Zatem tworząc witrynę WWW musimy opisać:

- strukturę i zawartość strony (język HTML),
- format elementów (język CSS).

W przypadku braku stylów przeglądarka zastosuje style domyślne, które zazwyczaj są dość ubogie. Oddzielając definicję formatu elementów od samej zawartości strony, otrzymamy dokument, którego format jest niezależny od zawartości. Główną korzyścią takiego rozwiązania jest to, że możemy modyfikować wygląd dokumentu, nie zmieniając jego treści (czyli kodu HTML).

##### **2.2. Testowanie stron WWW**

Każda wykonana strona WWW powinna zostać sprawdzona pod kilkoma względami:

- poprawności składniowej HTML i CSS – służą do tego:  
serwis: <http://validator.w3.org>  
oraz walidator <http://jigsaw.w3.org/css-validator/>
- wyglądu w kilku najpopularniejszych przeglądarkach,
- wpływu rozdzielczości monitora na wygląd strony – jeśli pracujemy na monitorze o dużej rozdzielczości (np. 1680x1050), to test witryn w różnych rozdzielczościach możemy sprawdzić, zmniejszając okno przeglądarki,
- wyglądu przy wyłączonych CSS – możemy tego dokonać, wykorzystując wbudowane możliwości przeglądarek. W przeglądarce Firefox style wyłącza opcja *Widok/Styl strony/Ignoruj style*, a w przeglądarce Internet Explorer – *Widok/Styl/Brak stylu*.

##### **2.3. Tworzenie prostej strony WWW**

Na potrzeby zajęć stworzymy prostą stronę WWW, w oparciu o pięć kroków ilustrujących podstawy tworzenia stron WWW:

**Krok 1.** Należy rozpocząć od treści strony – jako punkt wyjścia dodamy do strony zwykły tekst i zobaczymy, co zrobi z nim przeglądarka.

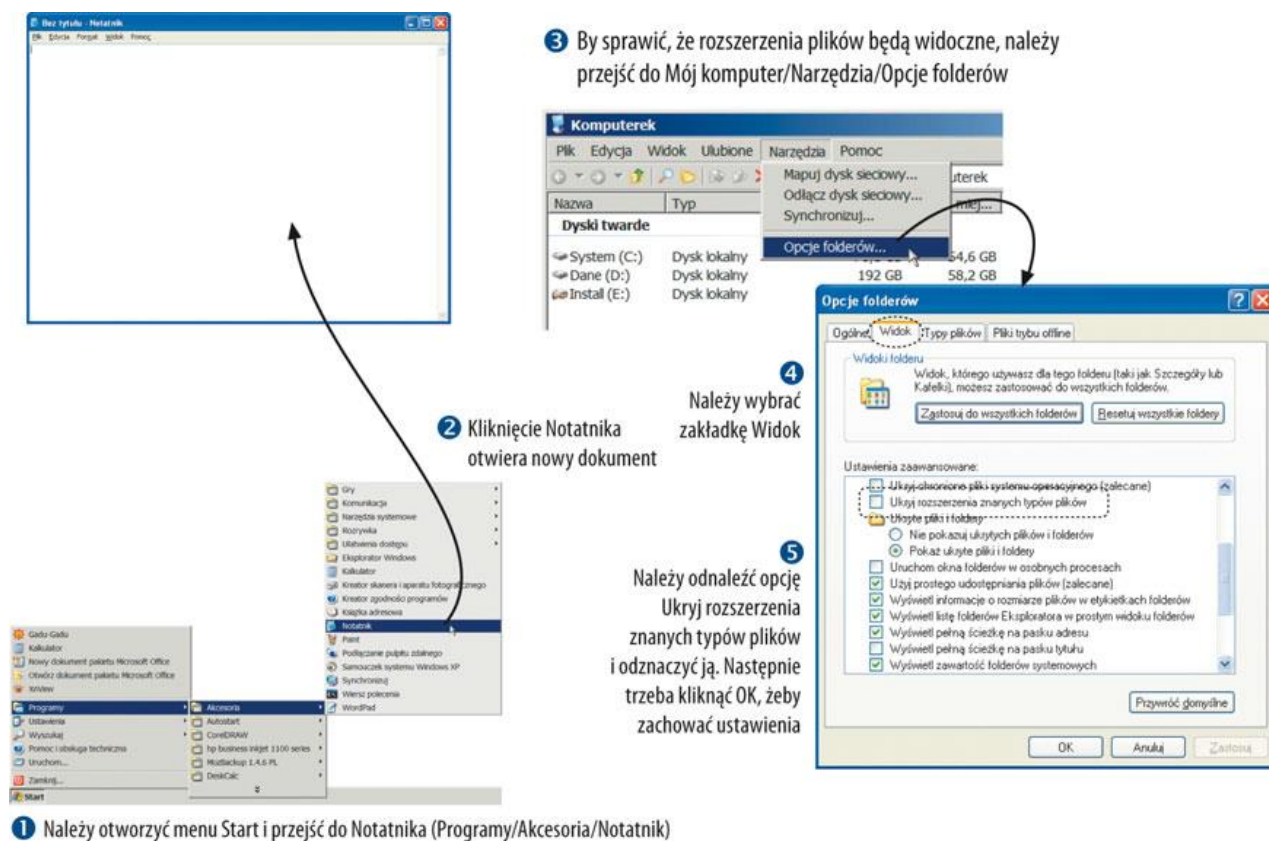
**Krok 2.** Nadanie dokumentowi struktury – zapoznamy się z elementami HTML oraz elementami nadającymi dokumentowi strukturę.

**Krok 3.** Zidentyfikowanie elementów tekstowych – treść strony zostanie opisana za pomocą odpowiednich elementów tekstowych, a także nauczymy się, w jaki sposób należy używać HTML poprawnie.

**Krok 4.** Dodanie obrazka – dodając obrazek do strony, dowiemy się czegoś o atrybutach oraz elementach pustych.

**Krok 5.** Zmiana wyglądu strony za pomocą arkusza stylów – w tym ćwiczeniu będzie można spróbować, jak formatuje się treść strony z wykorzystaniem kaskadowych arkuszy stylów (CSS).

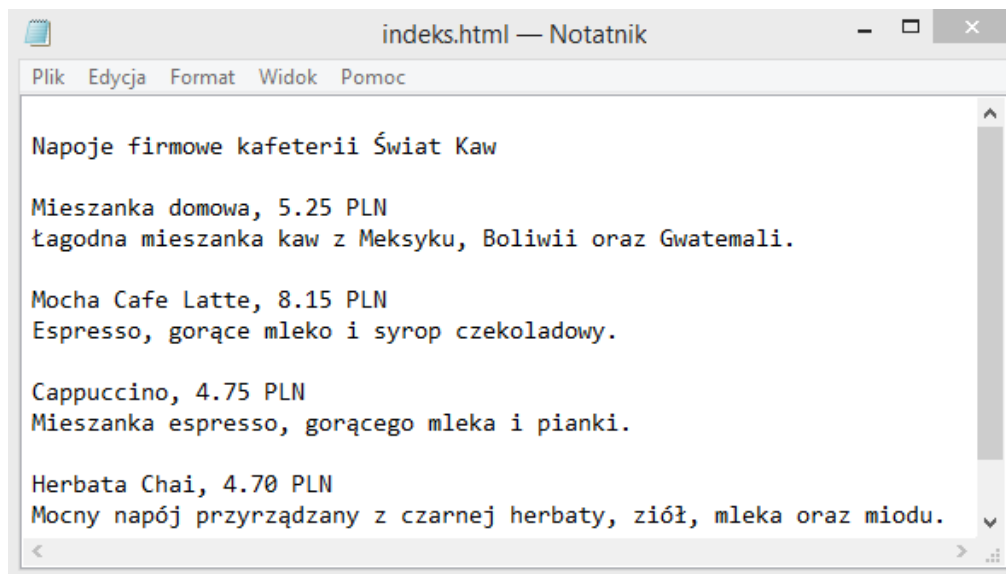
Ćwiczenia można wykonać na komputerze, na którym zainstalowano dowolną przeglądarkę stron WWW (Firefox, Internet Explorer, Google Chrome itp.) oraz prosty edytor tekstowy (np. Notatnik, Notepad++). Pliki tworzone podczas ćwiczeń należy zapisywać katalogu /Nazwisko\_Imie, a później korzystając z menu **Otwórz (Open)** przeglądarki lub jej paska adresu – uruchamiać je w przeglądarce.



Rys.2.1. Rozpoczynanie pracy z Notatnikiem [3]

### Wprowadzanie treści strony

Wprowadź tekst widoczny na rys. 2.2 i zapisz go jako *index.html*. Obejrzyj plik w przeglądarce (np. Firefox – *Plik/Otwórz plik/....*)



Rys.2.2. Treść strony

Treść strony w przeglądarce nie prezentuje się zbyt ciekawie. Tekst napisany jest jednym ciągiem – nie tak jak wyglądało to w oryginalnym dokumencie. Przeglądarka ignoruje złamanie wierszy w dokumencie źródłowym. Po drugie, widać, że samo wpisanie treści strony i nazwanie dokumentu *.html* nie wystarczy. Choć przeglądarka może wyświetlić tekst z takiego pliku, nie oznaczono w żaden sposób struktury jego treści. Z tego powodu niezbędny jest HTML. Znaczniki zostaną dodane w celu uzyskania struktury dokumentu – najpierw w samym dokumencie HTML (krok 2.), a później do zawartości strony (krok 3.). Kiedy przeglądarka będzie знаła strukturę treści, będzie w stanie wyświetlić stronę w bardziej znaczący sposób.

### Nadanie dokumentowi struktury.

W języku HTML **znacznik** (*ang. tag*) to napis otoczony znakami `<` oraz `>`. Bezpośrednio po znaku `<` występuje nazwa znacznika (rys. 2.3). Znacznik otwierający i zamykający otaczają pewien fragment, nadając mu odpowiednie znaczenie. Lista znaczników w języku HTML5 jest ściśle określona. Język nie umożliwia dodawania nowych znaczników. Szczegółowy opis języka HTML5 znajduje się w specyfikacji, która dostępna jest pod adresem: <http://dev.w3.org/html5/spec/>. Natomiast na stronie <http://shebang.pl/artykuly/elementy-html/> dostępne jest zestawienie wszystkich elementów HTML 4.01, XHTML 1.0 oraz HTML 5.



Rys.2.3. Części elementu HTML [3]

Poprawna, pusta strona WWW napisana w języku HTML5 składa się z trzech części (rys.2.4): definicji typu dokumentu (deklaracja DOCTYPE), nagłówka dokumentu (element head), treści dokumentu (element body).

```
<!DOCTYPE html>
  <head>
    <title> Kafeteria Świat Kaw </title>
    <meta charset="utf-8" />
  </head>
<body>
Napoje firmowe kafeterii Świat Kaw
Mieszanka domowa, 5.25 PLN
Łagodna mieszanka kaw z Meksyku, Boliwii oraz Gwatemali.
Mocha Cafe Latte, 8.15 PLN
Espresso, gorące mleko i syrop czekoladowy.
Cappuccino, 4.75 PLN
Mieszanka espresso, gorącego mleka i pianki.
Herbata Chai, 4.70 PLN
Mocny napój przyrządzany z czarnej herbaty, ziół, mleka oraz miodu.
</body>
</html>
```

Rys.2.4. Podstawowa struktura dokumentu HTML z kodowaniem UTF-8

Przeglądarka internetowa rozpoznaje kodowanie znaków strony WWW na podstawie elementu meta zawartego w pliku HTML5. Polskie znaki na stronach internetowych są kodowane na jeden z czterech sposobów: **iso-8859-2**, **windows-1250**, **utf-8**, **znaki specjalne (encje)**.

### Zidentyfikowanie elementów tekstowych

Następnym krokiem jest dodanie znaczników identyfikujących nagłówki oraz podnagłówki (h1 i h2), akapity (p), a także tekst zaakcentowany (em) do treści strony (rys. 2.5).

<!DOCTYPE html>	Informuje przeglądarkę o początku dokumentu HTML
<head>	Rozpoczyna sekcję „nagłówka” strony
<title> Kafeteria Świat Kaw </title>	Określa tytuł strony
<meta charset="utf-8" />	Określa sposób kodowania strony
</head>	Koniec sekcji nagłówka
<body>	Początek treści strony
<h1>Napoje firmowe kafeterii Świat Kaw</h1>	Informuje, że „Napoje...” to nagłówek
<h2>Mieszanka domowa, <em>5.25 PLN</em></h2>	Informuje, że „Mieszanka...” to nagłówek poziomu drugiego oraz, że należy wyróżnić „5.25 PLN”
<p>Łagodna mieszanka kaw z Meksyku, Boliwii oraz Gwatemali.</p>	Początek i koniec akapitu
<h2>Mocha Cafe Latte, <em>8.15 PLN</em></h2>	
<p>Espresso, gorące mleko i syrop czekoladowy.</p>	
<h2>Cappuccino, <em>4.75 PLN</em></h2>	
<p>Mieszanka espresso, gorącego mleka i pianki.</p>	
<h2>Herbata Chai, <em>4.70 PLN</em></h2>	
<p>Mocny napój przyrządzany z czarnej herbaty, ziół, mleka oraz miodu.</p>	
</body>	Koniec treści strony

&lt;/html&gt;

Informuje przeglądarkę, że w tym miejscu kończy się plik HTML

Rys. 2.5. Oznaczenie tekstu za pomocą elementów HTML.

Nagłówki oraz akapity rozpoczynają się od nowych wierszy i nie są zapisane ciągiem jeden po drugim, jak wcześniej. Jest tak, ponieważ są one przykładami elementów blokowych (ang. *blocklevel element*). Każdy element blokowy rozpoczyna się od nowego wiersza i zazwyczaj nad całym elementem oraz pod nim domyślnie dodawany jest jakiś odstęp. Inaczej wygląda to w przypadku tekstu oznaczonego jako zaakcentowany (*em*). Nie rozpoczyna się on od nowego wiersza, ale pozostaje częścią akapitu. Jest tak, ponieważ element *em* jest elementem wewnętrznym (ang. *inline element*). Elementy wewnętrzne nie rozpoczynają nowych wierszy, pozostają na swoim miejscu. Rozróżnienie pomiędzy elementami blokowymi a wewnętrznymi jest istotne. W języku HTML to, czy element jest blokowy, czy wewnętrzny, wpływa na to, jakie elementy może zawierać. Nie można na przykład umieścić elementu blokowego wewnątrz elementu wewnętrznego (na przykład akapitu w środku odnośnika). Elementy blokowe oraz wewnętrzne zachowują się również inaczej, jeśli chodzi o stosowanie arkuszy stylów. Można również zauważyć, że przeglądarka próbuje nadać stronie jakąś hierarchię wizualną poprzez uczynienie nagłówka pierwszego stopnia największym i najbardziej pogrubionym tekstem na stronie, nagłówka drugiego stopnia nieco mniejszym i tak dalej.

Przeglądarka wykorzystuje arkusz stylów, bowiem wszystkie przeglądarki mają wbudowane własne arkusze stylów, które określają domyślny wygląd elementów HTML.

### Dodanie obrazka – elementy puste

Strona internetowa bez obrazka wygląda średnio. Dwoma najważniejszymi metodami umieszczania obrazów na stronach WWW są: element **img**, tła ustalone właściwością **background** w stylach CSS. W ćwiczeniu wykorzystamy element **img**.

Dotychczas wszystkie elementy wykorzystane na stronie internetowej składały się z tekstu otoczonego znacznikami otwierającymi oraz zamykającymi. Spora liczba elementów nie ma zawartości tekstowej, ponieważ są one wykorzystywane jako proste instrukcje. O takich elementach mówi się, że są puste (ang. *empty*). Element obrazka **img** (ang. *image*) jest przykładem tego typu elementu. Przekazuje on przeglądarce, że ma ona pobrać plik graficzny z serwera i wstawić go do tekstu w miejscu występowania znacznika. Inne elementy puste to złamanie wiersza (**br**), linia pozioma (**hr**), a także elementy udostępniające informacje o dokumencie, jednak niewpływające na wyświetlaną treść, jak element **meta**.

Elementy puste nie mają znaczników zamykających. Elementy te zapisuje się w postaci jednego znacznika np. **<img>**, **<br>**, **<hr>**. W celu odróżnienia znaczników pustych od zwykłych znaczników otwierających możemy stosować znak **/**, np. **<img />**, **<br />**, **<hr />**.

Formatami graficznymi, które można stosować na stronach WWW, są: JPEG, PNG, GIF, SVG. Element **img** umieszcza na stronie jeden plik graficzny. Jest to element, który ma dwa podstawowe atrybuty: **src** (ang. *source* – źródło) oraz **alt** (ang. *alternate text* – tekst alternatywny) – rys. 2.6.

Rys. 2.6. Element **img** wraz z atrybutami. [3]

Atrybut **src** ustala adres wstawianego pliku, zaś **alt** zawiera tekst przeznaczony dla tych wszystkich, którzy nie ujrzą obrazu. Jeśli elementu **img** używamy do wstawienia obrazu, którego jedyną rolą jest ozdabianie witryny, wówczas atrybut **alt** należy pozostawić pusty. Oto, co należy wiedzieć na temat atrybutów:

- Atrybuty umieszczane są po nazwie elementu tylko w znaczniku otwierającym, nigdy w zamykającym.
- Do elementu można stosować większą liczbę atrybutów rozdzielanych spacjami w znaczniku otwierającym. Ich kolejność nie jest istotna.
- Atrybuty przyjmują wartości, które następują po znaku równości (=).
- Wartość może być liczbą, słowem, łańcuchem znaków, adresem URL lub miarą, w zależności od celu atrybutu.
- Wartości zawsze należy umieszczać w cudzysłowie. Akceptowane są zarówno cudzysłowy, jak i apostrofy, o ile są one używane w spójny sposób. Jednak to cudzysłowy są przyjętą konwencją zapisu.
- Niektóre atrybuty są wymagane, jak na przykład **src** oraz **alt** w elemencie **img**.
- Nazwy atrybutów dostępnych dla każdego elementu zdefiniowane są w specyfikacjach HTML. Innymi słowy, nie można wymyślić atrybutu dla elementu.

W celu umieszczenia obrazka na stronie przejdź pod adres <https://kawowy.guru/najlepsza-kawa-swiecie/> i zapisz w folderze, w którym znajduje się plik *index.html*, widoczny tam obrazek o nazwie *najlepszakawa\_1.jpg*, a następnie wstawić do dokumentu fragment pokazany poniżej:

```
<h1> Napoje firmowe kafeterii Świat Kaw</h1>
```

### Zmiana wyglądu za pomocą arkusza stylów

CSS zapewnia nam możliwość określenia sposobu, w jaki ma być prezentowana zawartość stron. Dzięki temu strona naszej kafeterii będzie wyglądała nieco atrakcyjnej. Kaskadowe arkusze stylów zostaną omówione szczegółowo na kolejnych zajęciach. Teraz tylko zobaczymy, co znaczy dodanie „warstwy” prezentacji do struktury utworzonej za pomocą znaczników HTML.

Zastosujemy element **style** do dodania osadzonego arkusza stylów do strony internetowej (to tylko jeden z możliwych sposobów dodawania arkuszy stylów, pozostałe omówione zostaną na kolejnych zajęciach). Element **style** umieszczany jest



wewnątrz elementu **head** dokumentu. Należy rozpocząć od dodania elementu style do dokumentu – jak w poniższym kodzie:

```
<style>
...
tutaj definicja typów
...
</style>
```

W chwili obecnej style dokumentów HTML są opisywane wyłącznie w języku CSS. Jest to język domyślny stosowany przez wszystkie przeglądarki. W poprzednich wersjach języka HTML stosowany był atrybut `type="text/css"`: `<style type="text/css">`. W języku HTML5 atrybut ten jest zbędny.

Po umieszczeniu w nagłówku strony elementu style należy wewnątrz niego zapisać style CSS, które upiększą stronę. W przypadku naszej strony będzie to:

```
<head>
    <title> Kafeteria Świat Kaw </title>
    <meta charset="utf-8" />
    <style>
    body
    {
        background: rgb(250, 194, 133);
        margin-top: 5%;
        margin-right: 5%;
        margin-bottom: 5%;
        margin-left: 5%;
        border-color: black;
        border-style: solid;
        border-width: 5px;
        text-align: center;
        font-family: cursive;
    }
    </style>
</head>
```

A strona ostatecznie będzie wyglądała następująco:



Rys. 2.7. Ostateczny wygląd strony „Kafeteria Świat Kaw”

### 3. ĆWICZENIA DO SAMODZIELNEGO WYKONANIA

Twój kolega stara się o pracę webmastera i musi przygotować stronę internetową ze swoim CV. Poprosił Cię o wykonanie takiej strony:

- Musisz stworzyć stronę internetową zawierającą podstawowe informacje o nim: imię, nazwisko, przebieg dotychczasowej edukacji, osiągnięcia zawodowe, zainteresowania itp. Na stronie ma się znaleźć również Jego zdjęcie.
- Za formatowanie dokumentów (krój i wielkość czcionki, kolory, marginesy) powinien być odpowiedzialny odpowiedni arkusz stylu (CSS).
- Nie wolno zapomnieć o odpowiednich nagłówkach dokumentu HTML: autor, opis zawartości, słowa kluczowe itd.
- Zdefiniuj odpowiednie obramowania i marginesy wokół elementów funkcjonalnych strony (np. wokół menu) – rozmiar i kolor wg uznania.
- Aplikację należy tworzyć z wykorzystaniem edytora tekstu (np. Notepad++, Notatnik) – środowiska wspomagające graficzne tworzenie dokumentów HTML nie są w tym ćwiczeniu dozwolone.
- Powstały dokument należy sprawdzić przy pomocy narzędzi (walidatorów) udostępnionych na stronach World Wide Web Consortium - <http://www.w3.org/>
- Należy przeanalizować sposób wyświetlania dokumentu w różnych przeglądarkach.

**Pliki stworzonej strony należy wysłać prowadzącemu w celu uzyskania zaliczenia laboratorium nr 1.**

### 4. LITERATURA

1. Gajda W., *HTML5 i CSS3. Praktyczne projekty*, Helion, Gliwice, 2013
2. Freeman E., Freeman E., *Head First HTML with CSS & XHTML*, Helion, Gliwice, 2007
3. Robbins J., *Projektowanie stron internetowych. Przewodnik dla początkujących webmasterów po HTML5, CSS3 i grafice*. Wydanie IV, Helion, Gliwice, 2013
4. [http://wazniak.mimuw.edu.pl/index.php?title=Aplikacje\\_WWW](http://wazniak.mimuw.edu.pl/index.php?title=Aplikacje_WWW)