

Wybrane zagadnienia HTML5 i CSS3

Spis treści

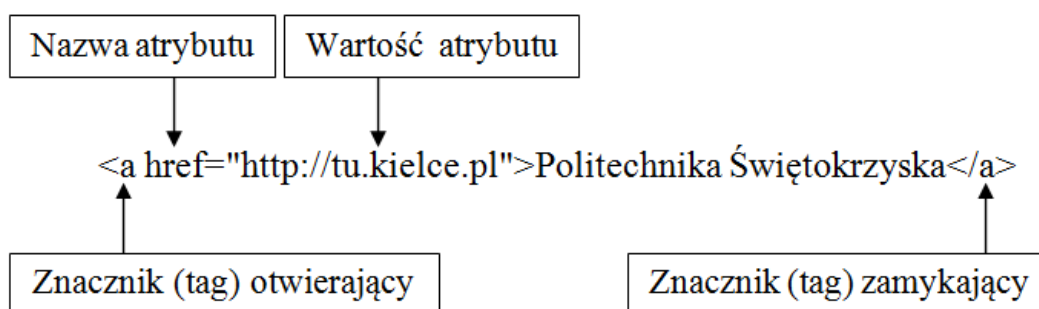
Część I – HTML5	2
1. Wprowadzenie do HTML5	2
2. Struktura dokumentu HTML5	3
3. Elementy blokowe i liniowe. Znaki specjalne.....	4
4. Grafika.....	7
5. Listy wypunktowane i numerowane.....	8
6. Odsyłacze	9
7. Tabele	9
8. Walidacja kodu HTML	10
Część II – CSS3.....	11
9. Wprowadzenie do CSS3.....	11
10. Wybrane własności arkuszy stylów	12
11. Formatowanie elementów za pomocą klas i identyfikatorów	16
12. Jednostki miar.....	17
13. Kolory.....	18

Część I – HTML5

1. Wprowadzenie do HTML5

HTML (ang. *HyperText Markup Language*) to hipertekstowy język znaczników. Służy do określenia struktury dokumentu wyświetlanego przez przeglądarkę internetową i stanowi przez to podstawę działania każdej strony internetowej. Do szczegółowego formatowania wyglądu (np. czcionek, kolorów, obramowania i marginesów) powinno wykorzystywać się kaskadowe arkusze stylów **CSS** – zostaną one omówione nieco później.

Składnię języka HTML stanowią **znaczniki** (ang. *tags*), które umieszcza się w nawiasach ostrych, np. `<html>`, `<div>`. Najczęściej używa się ich w parach – pierwszy nazywa się znacznikiem otwierającym, a drugi znacznikiem zamykającym, np. `<div>...</div>` – rysunek 1. Istnieją również znaczniki samodzielne, np. `
`, `<meta>`, `<hr>`, w przypadku których przyjęto w tej instrukcji zasadę pomijania zamykającego ukośnika (przykładowo `
`), choć ta forma zapisu również jest całkowicie poprawna z punktu widzenia HTML5. Duża część znaczników może posiadać **atrybuty**, które określają szczegóły ich wyglądu lub zachowania. Umieszcza się je przed nawiasem `>` znacznika otwierającego. Składnia atrybutu zbudowana jest z nazwy atrybutu oraz wartości umieszczonej w cudzysłowie: *nazwa*="wartość". Znacznik może posiadać więcej niż jeden atrybut. Należy pamiętać, że każdy tag posiada swoją własną listę atrybutów dozwolonych w języku HTML5, a według nowoczesnego podejścia do ich szczegółowego formatowania wykorzystuje się kaskadowe arkusze stylów CSS. Kod HTML powinien być formatowany z wykorzystaniem odpowiednich wcięć, co zwiększa jego czytelność. Istotne jest także prawidłowe używanie znaczników zamykających – nie wolno o nich zapominać!



Rysunek 1. Budowa znacznika HTML5

Źródło: Opracowanie własne.

Zadaniem przeglądarki internetowej jest odczyt znaczników i ich interpretacja w taki sposób, aby użytkownik widział poprawnie sformatowaną stronę internetową zamiast nieczytelnego kodu. Najpopularniejsze rozszerzenie pliku zawierającego kod HTML to **.html** (choć niekiedy spotyka się jeszcze starsze rozszerzenie **.htm**).

Strony internetowe powinno tworzyć się według ściśle określonych standardów, za których ustanawianie odpowiada organizacja **W3C** (ang. *World Wide Web Consortium*). Nie mają one co prawda mocy prawnej, ale gwarantują właściwe wyświetlanie zaprojektowanych stron przez wszystkie współczesne przeglądarki internetowe (Google Chrome, Mozilla Firefox, Internet Explorer, Opera, Safari itd.). Aktualnie najpopularniejszym standardem jest **HTML5**, który uważa się za rozwinięcie HTML4 i XHTML1. Został on oficjalnie zatwierdzony przez W3C 28 października 2014 roku. Napisano go z myślą o kompatybilności wstecznej – strona internetowa stworzona w HTML5, ale wykorzystująca przestarzałe znaczniki typu `` i `<frame>` będzie wyświetlana poprawnie, choć takie podejście jest niezalecane. W HTML5 postawiono duży nacisk na **semantykę**, czyli nadawanie znaczenia poszczególnym treściom. Standard HTML5 jest wspierany przez następujące przeglądarki:

- Google Chrome 8 i późniejsze wersje,
- Internet Explorer 9 i późniejsze wersje,
- Mozilla Firefox 3.5 i późniejsze wersje,
- Opera 10.5 i późniejsze wersje,
- Safari 4 i późniejsze wersje.

Według StatCounter (źródło: <http://gs.statcounter.com>) w okresie 01.11.2015r. – 19.11.2015r. najpopularniejszą przeglądarką internetową na komputery stacjonarne i laptopy był **Google Chrome**, z którego korzystało aż 57,70% internautów na całym świecie. Na drugim miejscu uplasowała się **Mozilla Firefox 5+** (wersja co najmniej 5) z udziałem w rynku na poziomie 15,84%, na trzecim **Internet Explorer 11** z wynikiem 11,40%. Z programu **Internet Explorer 8**, który nie obsługuje języka HTML5, korzystało 2,07% wszystkich użytkowników (według www ranking.pl w Polsce odsetek ten wynosił w tym samym czasie ok. 0,80%). Powyższe liczby dowodzą, że jedynie niewielka liczba przeglądarek internetowych używanych przez użytkowników desktopów nie wspiera języka HTML5.

Do napisania strony internetowej w języku HTML5 można wykorzystać najzwyklejszy edytor tekstu, np. Notatnik. Istnieją jednak darmowe aplikacje, które pozwalają na kolorowanie składni kodu, edycję kilku plików jednocześnie (dzięki obsłudze zakładek), czy też łatwe ustawienie kodowania znaków. Przykładem jest program **Notepad++**, który będzie wykorzystywany podczas zajęć laboratoryjnych. Można pobrać go w wersji niewymagającej instalacji (portable) ze strony producenta: <https://notepad-plus-plus.org>.

2. Struktura dokumentu HTML5

Składnia dokumentu napisanego w HTML5 rozpoczyna się od deklaracji typu dokumentu `<!DOCTYPE html>` – jej celem jest wskazanie przeglądarce internetowej rodzaju dokumentu, który ma przetworzyć. W kolejnym kroku wyznacza się ramy strony internetowej poprzez wprowadzenie znaczników `<html>...</html>`. Dobrym zwyczajem jest dodanie w tym miejscu informacji o języku, w którym tworzy się stronę – w tym celu wewnątrz znacznika `<html>`

umieszcza się atrybut **lang** wraz z odpowiednią wartością. W przypadku języka polskiego jest to **pl**, a angielskiego **en**.

Nagłówek dokumentu definiuje się wewnątrz znaczników **<head>...</head>**. Mimo, że efektów jego działania nie widać na pierwszy rzut oka, jest on bardzo istotny. Na początku zamieszcza się w nim informację dotyczącą sposobu kodowania znaków. Nieprawidłowe kodowanie może spowodować błędne wyświetlanie niektórych znaków (np. ąćęłńóśźź). Najpopularniejszy standard to obecnie **UTF-8**. W tym celu dodaje się tag **<meta>** z atrybutem **charset="UTF-8"**. Dzięki temu wszystkie polskie znaki będą wyświetlane poprawnie (pod warunkiem prawidłowego ustawienia sposobu kodowania znaków w edytorze tekstowym). Uogólniając, znacznik **<meta>** definiuje informacje profilowe strony, tzw. **metainformacje**. Dzięki niemu można także zdefiniować m.in. opis strony: atrybut **name="opis_strony"** oraz słowa kluczowe oddzielone przecinkiem: atrybut **name="słowa_kluczowe"**. Odpowiedni opis i właściwie dobrane słowa kluczowe pozwalają na prawidłowe odnajdowanie utworzonych stron przez wyszukiwarki internetowe (Google, Bing, Ask itd.). Pomędzy znacznikami **<title>...</title>** wprowadza się tytuł strony internetowej – będzie on widoczny na górnym pasku okna przeglądarki. W nagłówku umieszcza się również definicję stylów CSS.

Pomędzy znacznikami **<body>...</body>** wyznacza się miejsce, w którym umieszcza się właściwą treść strony internetowej (tzw. „ciało dokumentu”), czyli wszystko to, co ma zostać wyświetlone na ekranie.

Szablon strony internetowej w języku HTML5 prezentuje się następująco:

```
<!DOCTYPE html>
<html lang="pl">
<head>
  <meta charset="UTF-8">
  <meta name="description" content="Króki opis strony">
  <meta name="keywords" content="Słowa, kluczowe, oddzielone, przecinkiem">
  <title>Moja pierwsza strona internetowa w HTML5</title>
</head>
<body>
  <!-- Sposób zapisu komentarza w HTML5 -->
  <p>Witaj Internecie!</p>
  <p>Test polskich znaków: ąćęłńóśźź</p>
</body>
</html>
```

3. Elementy blokowe i liniowe. Znaki specjalne

Język HTML zawiera znaczniki, które można podzielić na tzw. **elementy liniowe** – mogą one występować w jednym wierszu obok siebie i zajmują tylko tyle miejsca, ile potrzebują oraz

tw. **elementy blokowe** – są one fragmentami zajmującymi całą, samodzielną i wydzieloną powierzchnię. Nie można ustawić w jednej linii więcej niż jednego elementu blokowego. Umieszcza się je wyłącznie jeden pod drugim. **Wszystkie poniższe znaczniki oprócz
 i <hr> posiadają tag zamykający!**

A. Wybrane znaczniki blokowe:

Znacznik	Opis działania
<p>	Akapit – podstawowy element zawierający tekst. Za jego pomocą wydziela się fragmenty dokumentu tworzące zwartą myśl.
<h1> – <h6>	Nagłówki – dzielą stronę internetową na logiczne części. Można je zobrazować jako tytuły (nagłówki) w gazecie lub książce, gdyż pełnią analogiczną rolę. W języku HTML wyróżnia się sześć poziomów nagłówków, które należy stosować hierarchicznie. Tytułem stopnia pierwszego, <h1>, oznacza się treści najważniejsze, a poza tym domyślnie są one pisane czcionką o największym dopuszczalnym dla nagłówków rozmiarze.
 	Łamanie linii – znacznik ten wykorzystywany jest do przenoszenia treści do kolejnego wiersza bez kończenia bieżącego akapitu. <u>Nie posiada znacznika zamykającego.</u>
<hr>	Pozioma linia – tag rysujący poziomą linię. W HTML5 oznacza tematyczne przejście, np. zmianę tematu artykułu. <u>Nie posiada znacznika zamykającego.</u>
<pre>	Blok preformatowany – tworzy blok tekstu uwzględniający wszystkie tzw. białe znaki w kodzie źródłowym, np. dodatkowe spacje, tabulatory i złamanie wiersza (w przeciwieństwie do akapitu).
<blockquote>	Blok cytowany – tworzy blok tekstu, w którym umieszcza się cytaty. Tekst zawarty pomiędzy tymi znacznikami wyświetlany jest z automatycznie dodanym wcięciem i cudzysłowem.
<address>	Adres – tworzy blok tekstu, w którym umieszcza się informacje kontaktowe dotyczące autora strony internetowej. Domyślnie pisany kursywą.
<div>	Blok dokumentu (kontener) – za jego pomocą grupuje się i pozycjonuje różne elementy dokumentu. Każdy fragment dokumentu objęty blokiem można swobodnie formatować. W HTML5 stracił on jednak na znaczeniu na rzecz elementów semantycznych służących do nakreślenia struktury strony internetowej – są <u>one przedstawione w kolejnych wierszach tabeli.</u>

<code><header></code>	Nagłówek dokumentu – tworzy blok dokumentu, w którym umieszcza się najczęściej logo, motto, hasło reklamowe oraz formularz wyszukiwania. Nowy element semantyczny języka HTML5.
<code><nav></code>	Nawigacja – tworzy blok dokumentu zawierający odnośniki do innych stron lub wyznaczonych fragmentów obecnej strony. Nowy element semantyczny języka HTML5.
<code><article></code>	Artykuł – tworzy blok dokumentu zawierający samodzielny fragment treści, np. wpis na blogu lub post na forum. Nowy element semantyczny języka HTML5.
<code><section></code>	Sekcja dokumentu – tworzy blok dokumentu obejmujący określoną część tematyczną. Powinien on ponadto zawierać nagłówek. Nowy element semantyczny języka HTML5.
<code><aside></code>	Fragment treści – tworzy blok dokumentu, w którym umieszcza się treści odseparowane od głównej zawartości strony i niekoniecznie powiązane z nią tematycznie, np. reklamy, panel boczny z dodatkowymi informacjami. Nowy element semantyczny języka HTML5.
<code><footer></code>	Stopka dokumentu – tworzy blok dokumentu umieszczany na dole strony i zawierający np. informacje o autorze, prawach autorskich, linki. Nowy element semantyczny języka HTML5.

B. Wybrane znaczniki liniowe:

Znacznik	Opis działania
<code></code>	Pogrubienie tekstu ze wzmocnieniem wagi – służy do pogrubienia fragmentu tekstu, który ma się wyróżniać na tle pozostałych fragmentów oraz ma istotną wagę w odniesieniu do całości – znaczenie semantyczne.
<code></code>	Pogrubienie tekstu – służy do pogrubienia tekstu, który nie jest ważniejszy niż reszta dokumentu (w przeciwieństwie do znacznika <code></code> , choć daje on identyczny widoczny dla użytkownika efekt).
<code></code>	Emfaza, kursywa – służy do wyróżnienia kursywą wybranych fragmentów tekstu. Znaczniki te są używane, gdy należy uwypuklić pewien fragment tekstu i podkreślić jego znaczenie.
<code><i></code>	Kursywa – służy do wyróżnienia kursywą wybranych fragmentów tekstu, które pozbawione są emfazy (w przeciwieństwie do <code></code> , choć daje on identyczny widoczny dla użytkownika efekt), np. słów obcojęzycznych i terminów technicznych.
<code><s></code>	Przekreślenie – służy do przekreślenia wybranego fragmentu tekstu i zaakcentowania przez to jego nieaktualności lub niedokładności.
<code><u></code>	Podkreślenie – służy do podkreślenia wybranego fragmentu tekstu.

<code><small></code>	Drobny druk – fragment tekstu zawarty między tymi znacznikami jest pisany mniejszą czcionką. Powinien być używany tylko i wyłącznie do oznaczania tych fragmentów, które mają być „pisane drobnym drukiem”, np. klauzule u dołu umowy.
<code><sup></code>	Indeks górny.
<code><sub></code>	Indeks dolny.
<code></code>	Zakres – wydziela określony fragment dokumentu, który można dowolnie formatować (tak jak znacznik <code><div></code> , ale w przeciwieństwie do niego jest elementem liniowym).

Niektórym elementom liniowym i blokowym, które nie zostały opisane powyżej, zostaną poświęcone osobne podtematy.

Nierzadko zachodzi potrzeba, by umieścić na stronie internetowej znak, którego nie można wprowadzić bezpośrednio z klawiatury lub posiadający specjalne znaczenie. Należy wtedy skorzystać z tzw. **znaków specjalnych**. Każdy z nich posiada unikalny kod zaczynający się od znaku **&** i kończący się **średnikiem**. Przykładowe znaki specjalne (**encje**):

- **<** <
- **>** >
- **&** &
- **≤** ≤
- ** ** spacja niełamiwa
- **Δ** Δ
- **€** €
- **®** ®
- **∑** Σ
- **©** ©

Spośród wymienionych powyżej, znaki **<**, **>**, **&** i **spacja** są znakami o specjalnym znaczeniu i nie można wstawiać ich bezpośrednio z klawiatury. Pełną listę znaków specjalnych i ich kodów można znaleźć na stronie <http://algorytmy.pl/doc/xhtml/?id=109>.

4. Grafika

Grafiki i obrazy są nieodłącznym elementem stron internetowych. Najczęściej używane w Internecie formaty plików graficznych to **JPEG** (ang. *Joint Photographic Experts Group*), **PNG** (ang. *Portable Network Graphics*) i **GIF** (ang. *Graphics Interchange Format*).

W HTML5 pojawił się nowy, semantyczny element blokowy `<figure>...</figure>`. Komponent ten definiuje zbiór elementów multimedialnych (obrazy, diagramy, wideo itd.) we

wspólnym kontenerze, który następnie można formatować jako całość. Towarzyszy mu najczęściej para znaczników `<figcaption>...</figcaption>`, dzięki którym można umieścić podpis pod rysunkiem.

Do umieszczenia grafiki w dokumencie służy znacznik ``. Nie posiada on znacznika zamykającego. Wymaga atrybutów *alt* (alternatywny opis grafiki w przypadku jej niewczytania) oraz *src* (adres URL grafiki). Opcjonalnie można użyć *width* (szerokość) i *height* (wysokość). Przykłady użycia:

- grafika znajduje się w tym samym folderze, co strona internetowa:
``
- grafika znajduje się w folderze podrzędnym:
``
- grafika znajduje się w folderze nadrzędnym:
``
- grafika znajduje się na zewnętrznym serwerze o znanym adresie:
``

Przykładowy kod, za pomocą którego można wstawić grafikę wraz z podpisem:

```
<figure>
  
  <figcaption>Podpis grafiki...</figcaption>
</figure>
```

5. Listy wypunktowane i numerowane

Za stworzenie listy wypunktowanej odpowiadają znaczniki `...`, a numerowanej tagi `...`. Każdy kolejny punkt listy musi zawierać się między znacznikami `...`. Tag `` może posiadać następujące atrybuty:

- *start* – numer początkowy,
- *type* – rodzaj numeracji (*I*, *A*, *a*, *I*, *i*),
- *reversed* – numerowanie w dół (odwrócone: 10, 9, 8, 7...).

Znacznik `` nie posiada żadnych atrybutów dozwolonych przez język HTML5. Przykładowy kod tworzący listę numerowaną:

```
<ol>
  <li>Punkt pierwszy</li>
  <li>Punkt drugi</li>
  <li>Punkt trzeci</li>
  <li>Punkt czwarty</li>
</ol>
```


6. Odsyłacze

Charakterystyczną cechą stron internetowych jest obecność na nich **odsyłaczy hipertekstowych**. Tworzy się je za pomocą znacznika `<a>`. Użytkownik końcowy nie musi wiedzieć, gdzie umieszczone są elementy, do których odwołuje się dany odnośnik. Odsyłacze mogą prowadzić do:

- fragmentów tej samej strony internetowej,
- dowolnej strony w Internecie o znanym adresie:
`Politechnika Świętokrzyska`
- dowolnych plików lub stron znajdujących na tym samym serwerze:
`Podstrona 1`
`Podstrona 2`
- plików innych rodzajów (np. grafiki lub skryptów).

Atrybut **href** definiuje adres URL, do którego ma nastąpić przekierowanie. Inny przydatny atrybut to **target** – definiuje docelowe okno otwieranego dokumentu. Przykładowa wartość to **blank** – dokument otworzy się w nowym oknie lub na nowej karcie.

Funkcję odsyłacza może pełnić również grafika (obrazek):

```
<a href="http://tu.kielce.pl"></a>
```

7. Tabele

Tabele służą do tworzenia zestawień danych tabelarycznych. Nie należy używać ich do tworzenia szkieletów stron internetowych, gdyż do tego celu służą bloki dokumentu takie jak `<header>`, `<nav>`, `<article>`, `<section>`, `<aside>`, `<footer>`, `<div>`. Poniżej omówiony zostanie prostszy model tworzenia tabel, który zawiera pięć znaczników.

Ramy tabeli tworzą znaczniki `<table>...</table>` – pomiędzy nimi należy umieścić całą zawartość. Para znaczników `<caption>...</caption>` służy do nadania tytułu dla tworzonej tabeli. Wiersze umieszcza się między tagami `<tr>...</tr>`. Do tworzenia poszczególnych komórek służą znaczniki `<td>...</td>`. To właśnie komórki zawierają konkretne dane, np. tekst lub grafikę. Aby tabela posiadała obramowanie, w znaczniku `<table>` umieszcza się atrybut **border** z określoną wartością (bez podania wartości tworzy się obramowanie o domyślnej szerokości). Nagłówki wiersza, dla podkreślenia ich większego znaczenia, umieszcza się między znacznikami `<th>...</th>` (zamiast `<td>`) – znaczenie semantyczne, a ponadto tekst zawarty między tymi tagami jest domyślnie pogrubiony. Do szczegółowego formatowania tabel służą kaskadowe arkusze stylów CSS. Przykładowy kod tworzący tabelę posiadającą nagłówek i jeden wiersz:

```
<table border="1">  
<caption>Przykładowa tabela</caption>
```

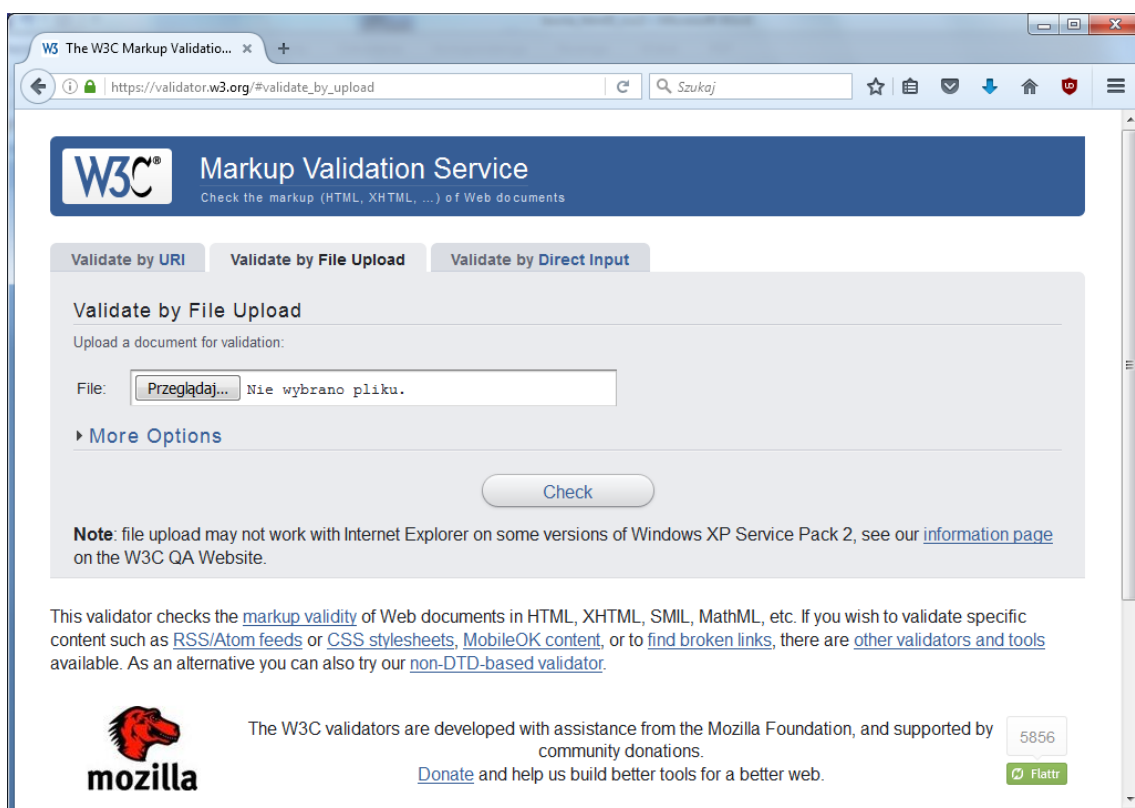
```

<tr>
  <th>Nagłówek 1</th>
  <th>Nagłówek 2</th>
</tr>
<tr>
  <td>Komórka 1</td>
  <td>Komórka 2</td>
</tr>
</table>

```

8. Walidacja kodu HTML

Utworzone strony WWW, nawet pomimo poprawnego wyświetlania przez przeglądarkę internetową, mogą zawierać różne błędy. W ich odnalezieniu bardzo pomocny jest **walidator**, czyli narzędzie pozwalające sprawdzić poprawność składni kodu HTML. Jeden z nich udostępnia na swojej stronie organizacja W3C: <http://validator.w3.org> – rysunek 2. Po wybraniu opcji **Validate by File Upload**, wskazaniu odpowiedniego pliku i kliknięciu przycisku **Check**, następuje proces walidacji. W wyniku otrzymuje się liczbę odnalezionych błędów wraz ze wskazaniem ich umiejscowienia i krótkim opisem.



Rysunek 2. Walidator HTML

Źródło: <http://validator.w3.org>.

Część II – CSS3

9. Wprowadzenie do CSS3

Współczesne strony internetowe powinny być budowane według reguły mówiącej, że **kod HTML** odpowiada za podział struktury dokumentu na sekcje, czyli np. nagłówki, akapity, listy, grafiki, odsyłacze, a **arkusz stylów CSS** jest odpowiedzialny za ich szczegółowe formatowanie, np. zdefiniowanie czcionek, kolorów, marginesów i obramowania.

Kaskadowe arkusze stylów CSS (ang. *Cascading Style Sheets*) umożliwiają przeniesienie formatowania stron internetowych (nawet tych najbardziej złożonych) do jednego dokumentu, czyli tzw. **arkusza stylów**. Innymi słowy CSS służy do odseparowania struktury strony internetowej od formy jej prezentacji. Dzięki temu zabiegowi kod samej strony internetowej pozostaje prosty i czytelny, a zmiana formatowania i układu całej tej strony wraz z podstronami polega tylko i wyłącznie na modyfikowaniu arkusza stylów. Arkusz stylów zawiera listę **reguł (dyrektyw)**, które definiują sposób, w jaki ma być wyświetlona zawartość wybranego dokumentu przez przeglądarkę internetową.

1. Osadzanie stylów w dokumencie. Składnia arkuszy CSS

Kaskadowe arkusze stylów CSS można osadzić w dokumencie na trzy sposoby:

- **style wewnętrzne (lokalne)** – informacje można zagnieźdzać wewnątrz konkretnego znacznika w atrybucie *style*. Takie rozwiązanie może wydawać się wygodne, lecz powoduje nieczytelność kodu dokumentu i utrudnia późniejszą jego modyfikację. Przykład użycia zmieniający kolor tekstu na czerwony w akapicie:

```
<p style="color: red">Tekst pisany na czerwono</p>
```

- **wewnętrzny arkusz stylów** – osadzenie całego kodu arkusza stylów za pomocą pary znaczników `<style>...</style>` umieszczonych w sekcji dokumentu `<head>`. Rozwiązanie to odseparowuje co prawda formatowanie od znaczników na stronie, ale cały kod ciągle znajduje się w jednym pliku. Poniżej podano przykład zmieniający tło strony internetowej na jasnoniebieskie i kolor tekstu w akapicie na zielony:

```
<head>
```

```
  <style>
```

```
    body {background-color: lightblue;}
```

```
    p {color: green;}
```

```
  </style>
```

```
</head>
```

- **zewnętrzny arkusz stylów** – podłączenie zewnętrznego arkusza stylów do dokumentu za pomocą znacznika `<link>` umieszczonego w sekcji `<head>`. Rozwiązanie to wydaje

się być najbardziej użyteczne, gdyż dzięki niemu można używać dokładnie tego samego formatowania na wielu różnych stronach jednocześnie. Zewnętrzny plik arkusza stylów powinien być zwyczajnym plikiem tekstowym, posiadać rozszerzenie **.css** i zawierać co najmniej jedną **regulę**. Przykład sposobu dołączenia zewnętrznego pliku o nazwie style.css do dokumentu:

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

Każda reguła zamieszczona w arkuszu stylów nakłada co najmniej jeden styl na jeden lub więcej elementów. Reguła ta ma ogólną postać:

```
selektor {
  własność: wartość;
  własność: wartość;
}
```

W skład powyższej reguły wchodzi trzy elementy:

- **selektor** – służy do identyfikacji elementu, który ma zostać sformatowany,
- **własność (atrybut)** – określa typ formatowania (np. zmiana barwy, czcionki itd.), który zostanie nałożony na określony element. W ramach jednej reguły można zadeklarować dowolną liczbę własności,
- **wartość** – w jej miejsce wstawiany jest walor własności.

10. Wybrane własności arkuszy stylów

Istnieje bardzo dużo własności, które można nadawać selektorom. W poniższej tabeli przedstawiono jedynie ich niewielką liczbę. Pełną, alfabetyczną listę można znaleźć na stronie: <http://webmaster.helion.pl/index.php/kurs-css/6/151-alfabetyczna-lista-stylow>.

Kategoria	Własność	Opis
Kolor i tło	<i>background-color</i>	definiuje kolor tła elementu, np. <i>background-color: aqua;</i>
	<i>background-image</i>	definiuje element graficzny jako tło, np. <i>background-image: url('/grafika/tlo.jpg');</i>
	<i>color</i>	definiuje kolor elementu, np. <i>color: red;</i>
Tekst	<i>font-family</i>	definiuje rodzaj czcionki, np. <i>font-family: Arial;</i>

	<i>font-size</i>	definiuje rozmiar czcionki, np. <i>font-size: 18px;</i>
	<i>font-weight</i>	definiuje stopień pogrubienia czcionki; przykładowe wartości: <ul style="list-style-type: none"> • <i>normal</i> – domyślna grubość, • <i>bold</i> – czcionka pogrubiona, • <i>bolder</i> – czcionka grubsza niż aktualna, • <i>lighter</i> – czcionka cieńsza niż aktualna, np. <i>font-weight: bold;</i>
	<i>font-style</i>	definiuje styl czcionki; przykładowe wartości: <ul style="list-style-type: none"> • <i>normal</i> – czcionka normalna, • <i>italic</i> – kursywa, np. <i>font-style: italic;</i>
	<i>font</i>	zbioreczy zapis własności czcionki, np. <i>font: 15px arial;</i>
	<i>text-decoration</i>	definiuje wyróżnienie tekstu; przykładowe wartości: <ul style="list-style-type: none"> • <i>none</i> – brak, • <i>underline</i> – podkreślenie, • <i>overline</i> – przekreślenie, np. <i>text-decoration: overline;</i>
	<i>text-align</i>	definiuje poziome wyrównanie tekstu; przykładowe wartości: <ul style="list-style-type: none"> • <i>left</i> – wyrównanie do lewej strony, • <i>right</i> – wyrównanie do prawej strony, • <i>center</i> – wyśrodkowanie, • <i>justify</i> – wyrównanie do lewej i do prawej, np. <i>text-align: center;</i>
	<i>vertical-align</i>	definiuje pionowe wyrównanie tekstu; przykładowe wartości: <ul style="list-style-type: none"> • <i>bottom</i> – wyrównanie do dolnej części elementu, • <i>middle</i> – wyrównanie do środkowej części elementu, • <i>top</i> – wyrównanie do górnej części elementu, • <i>sub</i> – indeks górny, • <i>super</i> – indeks dolny, np. <i>vertical-align: middle;</i>

	<i>text-indent</i>	definiuje wcięcie tekstu, np. <i>text-indent: 3pt;</i>
	<i>line-height</i>	definiuje odstęp między liniami, np. <i>line-height: 100%;</i>
Obramowanie	<i>border-width</i>	definiuje szerokość obramowania, np. <i>border-width: 4px;</i>
	<i>border-color</i>	definiuje kolor obramowania, np. <i>border-color: blue;</i>
	<i>border-style</i>	definiuje styl obramowania; przykładowe wartości: <ul style="list-style-type: none"> • <i>none</i> – brak obramowania, • <i>dashed</i> – linia przerywana, • <i>solid</i> – linia ciągła, • <i>double</i> – linia podwójna, np. <i>border-style: dashed;</i>
	<i>border</i>	zbiorczy zapis własności obramowania, np. <i>border: solid blue;</i>
Przestrzeń i odstępy	<i>margin-left</i>	definiuje lewy margines, np. <i>margin-left: 20px;</i>
	<i>margin-right</i>	definiuje prawy margines
	<i>margin-top</i>	definiuje górny margines
	<i>margin-bottom</i>	definiuje dolny margines
	<i>margin</i>	zbiorczy zapis właściwości marginesów (obszaru występującego po zewnętrznej stronie obramowania) w kolejności: top, right, bottom, left, np. <i>margin: 10px 50px 10px 20px;</i>
	<i>padding</i>	zbiorczy zapis właściwości dopełnienia (obszaru występującego po wewnętrznej stronie obramowania elementu, tzw. marginesu wewnętrznego) w kolejności: top, right, bottom, left, np. <i>padding: 3cm;</i>
Wymiary	<i>width</i>	szerokość, np. <i>width: 100px;</i>
	<i>height</i>	wysokość
Pozycjonowanie elementów	<i>position</i>	definiuje rodzaj pozycjonowania; przykładowe wartości: <ul style="list-style-type: none"> • <i>static</i> – pozycjonowanie tradycyjne: każdy element ma swoje miejsce, • <i>relative</i> – pozycjonowanie względne: element

		<p>zmienia pozycję bez wpływu na pozostałą część dokumentu,</p> <ul style="list-style-type: none"> • absolute – pozycjonowanie absolutne: określa stałą pozycję względem boku okna przeglądarki, np. <p>position: relative;</p>
	top	określa pozycję względem górnej krawędzi, np. top: 15px;
	right	określa pozycję względem prawej krawędzi
	bottom	określa pozycję względem dolnej krawędzi
	left	określa pozycję względem lewej krawędzi
	float	<p>definiuje pływanie elementów: umożliwia przepływać elementom we wskazanym kierunku; przykładowe wartości:</p> <ul style="list-style-type: none"> • left – element znajduje się po lewej stronie, • right – element znajduje się po prawej stronie, np. <p>float: left;</p>
	clear	<p>tamowanie elementów: „zrzuca” element w dół, przykładowe wartości:</p> <ul style="list-style-type: none"> • left – wymusza nową linię pod obiektem umiejscowionym po lewej stronie, • right – wymusza nową linię pod obiektem umiejscowionym po prawej stronie, • none – element zajmuje pozycję domyślną, • both – wymusza nową linię pod obiektem bez względu na jego położenie, np. <p>clear: both;</p>
Listy	list-style-type	<p>definiuje styl listy; przykładowe wartości:</p> <ul style="list-style-type: none"> • disc – koło, • circle – okrąg, • square – kwadrat, np. <p>list-style-type: disc;</p>
Tabele	border-collapse	<p>definiuje, czy obramowania poszczególnych komórek tabeli mają się na siebie nakładać; przykładowe wartości:</p> <ul style="list-style-type: none"> • collapse – obramowania komórek są ze sobą połączone,

		<ul style="list-style-type: none"> <i>separate</i> – obramowania komórek pozostają od siebie oddzielone, np. <i>border-collapse: collapse;</i>
	<i>border-spacing</i>	definiuje szerokość odstępów między komórkami tabeli, np. <i>border-spacing: 10px;</i>

Należy pamiętać, że każda grupa znaczników ma ściśle określony zespół własności (atrybutów), które można jej przypisywać. Każda własność posiada z kolei ściśle określoną listę wartości, które można im nadawać.

11. Formatowanie elementów za pomocą klas i identyfikatorów

Jeśli strona internetowa jest nieco bardziej złożona, zachodzi wtedy potrzeba wybrania konkretnych elementów, na które trzeba nałożyć formatowanie. Styl do wybranego znacznika można przypisać za pomocą tzw. **selektora**. Poniżej omówione zostaną także **klasy** i **identyfikatory**.

Selektor prosty to znacznik języka HTML przypisany do elementu, który w danym momencie podlega formatowaniu. Przykładowo styl zdefiniowany dla selektora `<p>` będzie odnosił się tylko i wyłącznie do akapitów znajdujących się pomiędzy znacznikami `<p>...</p>`:

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

Selektor uniwersalny umożliwia zdefiniowanie formatowania dla wszystkich elementów strony internetowej. Dostępne są dwa takie selektory: `*` oraz **body**, np.:

```
body {
    color: lightgreen;
}
```

Możliwe jest również **grupowanie selektorów**. Jest to przydatne, gdy identyczne wartości mają zostać nałożone na wszystkie z tych selektorów. Poszczególne selektory należy rozdzielać przecinkami, np.:

```
h1, h2, h3 {
    color: red;
}
```

Identyfikatory umożliwiają przypisanie formatowania do konkretnego znacznika, który ma własną, unikalną nazwę – **id**. Identyfikator służy do nadania unikalnego formatowania dla

wyłącznie jednego elementu na stronie internetowej (w przeciwieństwie do klas). W deklaracji CSS przed nazwą identyfikatora umieszcza się znak #, np.:

```
#Kontener {  
    border-width: 4px;  
}
```

Wybrany znacznik HTML musi posiadać atrybut *id*, np.:

```
<div id="Kontener">
```

Klasy pozwalają na przypisywanie formatowania do wybranych elementów, ale można wykorzystywać je wielokrotnie (w dowolnej liczbie elementów). Składnia reguły CSS rozpoczyna się od **kropki**, a następnie zawiera nazwę klasy, np.:

```
.Kontener2 {  
    border-width: 4px;  
}
```

Elementy, które mają zostać sformatowane za pomocą klasy, muszą posiadać atrybut *class*, np.:

```
<div class="Kontener2">
```

W sytuacji, gdy do jednego znacznika odwołuje się więcej niż jeden selektor, to jako pierwszy zostanie nałożony ten bardziej ogólny. Wiąże się z tym zasada kaskadowości. Mówi ona, że jako pierwsze uwzględniane są zewnętrzne arkusze stylów, następnie style wpisane do nagłówka *<head>*, a dopiero na samym końcu style przypisane lokalnie poszczególnym znacznikom. Gdy na dany element jest nałożonych kilka różnych źródeł stylów, jako ostatni zostanie nałożony ten, który znajduje się najbliżej formatowanego elementu.

12. Jednostki miar

Wyróżnia się dwa rodzaje jednostek miar:

- **jednostki względne (relatywne):**
 - **px** – piksele – jednostka ta opiera się na pojedynczych punktach widocznych na ekranie monitora, tzw. pikselach,
 - **em** – proporcje wysokości w odniesieniu do czcionki danego elementu,
 - **ex** – proporcje w stosunku do wysokości litery x (małej litery w aktualnej czcionce),
 - **%** – procenty – określają wielkości względem wartości domyślnej.
- **jednostki bezwzględne (absolutne):**
 - **in** – cale – jednostka ta wywodzi się z amerykańskiego systemu miarowego; 1 in = 2,54 cm,

- **pt** – punkty; 1 pt = 1/72 in,
- **cm** – centymetry; 1 cm = 0,39 in,
- **mm** – milimetry; 10 mm = 1 cm,
- **pc** – pika; 1 pc = 12 pt.

13. Kolory

Wartości kolorów można określać za pomocą:

- **nazwy własnej koloru** – kilkanaście kolorów podstawowych może być zapisywanych za pomocą ich nazw w języku angielskim, np. **Black, Red, Green, Blue**,
- **wartości RGB** (ang. *Red, Green, Blue*) – zapis kolorów przyjmuje tutaj liczby całkowite z przedziału od 0 do 255 dla każdej z wartości składowych, np. **rgb(255,128,0)**. Oznaczenie koloru jest więc złożeniem wartości kodowych trzech składowych barw podstawowych: czerwonej (*red*), zielonej (*green*) i niebieskiej (*blue*),
- **wartość szesnastkowej (heksadecymalnej)** – zapis ten zawiera sześć cyfr szesnastkowych (trzy bajty), które zgodnie z modelem RGB odpowiadają kolejno za barwę czerwoną, zieloną oraz niebieską. Każdy bajt przyjmuje wartości z przedziału od 0 (00) do 255 (FF), np. **#0000FF**.

Wybrane kolory podstawowe:

Kolor	Nazwa po angielsku	Zapis szesnastkowy
cyjan	<i>aqua</i>	#00FFFF
czarny	<i>black</i>	#000000
niebieski	<i>blue</i>	#0000FF
fukcja	<i>fuchsia</i>	#FF00FF
zielony	<i>green</i>	#008000
szary	<i>grey</i>	#808080
limonka	<i>lime</i>	#00FF00
wiśniowy	<i>maroon</i>	#800000
granatowy	<i>navy</i>	#000080
oliwkowy	<i>olive</i>	#808000
fioletowy	<i>purple</i>	#800080
czerwony	<i>red</i>	#FF0000
srebrny	<i>silver</i>	#C0C0C0
morski	<i>teal</i>	#008080
biały	<i>white</i>	#FFFFFF
żółty	<i>yellow</i>	#FFFF00

Więcej o kolorach można przeczytać na stronie <http://algorytmy.pl/doc/xhtml/?id=0108>.

Opracowano na podstawie:

1. Danowski B., *Tablice informatyczne. CSS3*, Helion, Gliwice 2012.
2. MacDonald M., *HTML5: Nieoficjalny podręcznik*, Helion, Gliwice 2012.
3. Mazur D., *HTML5 i CSS3. Definicja nowoczesności*, Wydawnictwo Naukowe PWN SA, Warszawa 2015.
4. *HTML(5) Tutorial*, <http://www.w3schools.com/html>.
5. *Kurs języka HTML i CSS*, <http://webmaster.helion.pl>.
6. *Kurs HTML5*, <http://how2html.pl>.
7. <https://pl.wikipedia.org>.