1. Cel ćwiczenia

Celem pierwszych zajęć jest utworzenie szkieletu strony HTML. Szkielet ten będzie stanowił podstawę dla ćwiczenia drugiego.

2. Teoria

Obecnie niemal wszystkie strony internetowe tworzone są w standardzie HTML5. Nieliczne tworzone są jeszcze w standardzie XHTML (w tym najmniejszy udział mają te tworzone wedle wersji 5). Ponadto stare serwisy, które nie są od dłuższego czasu aktualizowane, mogą nadal być wykonane w starym standardzie HTML4.

2.1 Informacje wstępne o HTML5

Sam HTML5 stanowi milowy krok w rozwoju witryn internetowych. Powszechnym stało się używanie języka skryptowego JavaScript, który jeszcze w wersji 4 był jedynie dodatkiem. Mało tego, niektóre z elementów HTML, jak płótno (Canvas), film (video) czy muzyka (audio) nie mogą istnieć bez włączonego parsera JavaScript w przeglądarce. Ponadto pojawiły się nowe rozwiązania przechowywania danych lokalnych. Teraz ze stron WWW można tworzyć aplikacje HTML działające nawet w przypadku, gdy nie mamy dostępu do sieci – zapewnia to dodatkowy mechanizm przechowywania przez przeglądarkę plików witryny po stronie klienta. Dodatkowo, prócz tradycyjnych ciasteczek, projektant może korzystać z pamięci lokalnej przeglądarki, która umożliwia przechowywanie znacznie większej ilości danych.

Nowe rozwiązania są także istotne z punktu widzenia wyszukiwarek internetowych. Przykładowo stosowanie semantycznych elementów blokowych pozytywnie wpływa na klasyfikację naszej strony, chociaż standard nie zabrania też stosowania jakże popularnego wydziału (warstwa – div). Niemniej dla robotów Google czy Bing znacznie lepiej wyglądać będzie kod:

```
<figure>
	<img src=’obrazek.jpg’ alt=’Nowy obrazek’/>
	<figcaption>Podpis obrazka</figcaption>
</figure>
```

niż:

```
<div>
	<img src=’obrazek.jpg’ alt=’Nowy obrazek’/>
	<p>Podpis obrazka</p>
</div>
```

W tym miejscu należy także zauważyć, że atrybut alt z opcjonalnego stał się usilnie zalecany (chociaż nadal nie obowiązkowy). Odpowiada on bowiem za łatwiejsze kwalifikowanie zdjęć w wyszukiwarce graficznej, a strony z odpowiednio podpisanymi zdjęciami mają wyższy priorytet niż te z brakującym atrybutem.

Najważniejszym czynnikiem jest przestrzeganie konstrukcji zagnieżdżania elementów. Elementy blokowe mogą zawierać zagnieżdżenia natomiast liniowe – nie. Stąd niepoprawną konstrukcją będzie zapis:

```
<a href=’odnosnik.html’><p>Przycisk</p></a>
```

który był stosowany przez wielu projektantów stron w HTML4 (też nie był zgodny ze standardem). Poprawna wersja wyglądałaby następująco:

Przycisk
lub, jeżeli chcemy nadać naszemu napisowi szczególne cechy:

Przycisk

Druga pozycja jest również poprawna gdyż element span należy do znaczników liniowych. Oczywiście obecnie odnośniki nie muszą być tworzone przy użyciu znacznika a. Coraz częściej tworzy się je poprzez użycie zdarzeń JavaScript, które to zdarzenie można przypisać do dowolnego elementu HTML.

Oczywiście omówione tutaj zmiany w HTML to nie wszystko. Wiele z elementów zostało wycofane z użycia (jak chociażby „kochane” przez wszystkich znaczniki CENTER, FONT, czy MARQUE), inne otrzymały nowe atrybuty (jak np. elementy INPUT, pozwalające na dołączenie ich do wskazanych formularzy bez konieczności zagnieżdżania ich w jakikolwiek formularz), jeszcze inne pozostały, lecz ze zmniejszoną ilością atrybutów (przykładem mogą być znaczniki tabeli, które nie posiadają już atrybutów ramki, koloru czy odstępów pomiędzy polami – wszystko przeniesione zostało do CSS).

2.2 Informacje dotyczące CSS

Obecnie żadna projektowana strona nie może obyć się bez kaskadowych arkuszy stylów (CSS – Cascade Style Sheet). Wynika to z faktu usunięcia wcześniej wspomnianych elementów oraz atrybutów odpowiadających za pozycję i wygląd innych elementów na stronie WWW.

Style można dołączać na różne sposoby – poprzez element <style></style>, w osobnym pliku (podłączanym do czystych plików HTML poprzez znacznik <link>) bądź tworzonych dynamicznie poprzez JavaScript (najmniej efektywne rozwiązanie, aczkolwiek niekiedy konieczne).W celu lepszego wykorzystania kodu CSS powstały także technologie pozwalające na dodawania do arkuszy zmiennych i instrukcji warunkowych (zostaną one przedstawione nieco później).

Głównym celem stworzenia i stosowania CSS jest tzw. rozwijalność projektów witryn WWW. W przypadku zmian wyglądu poszczególnych elementów bezpośrednio w kodzie HTML projektant, chcąc zastosować podobny efekt na kolejnych podstronach (bądź projektach) musiał kopiować cały kod HTML, po czym zmieniać treści danej strony. Gdy dochodziło do zmiany projektowej (np. koloru liter poszczególnych paragrafów) projektant musiał modyfikować WSZYSTKIE pliki HTML danego projektu, co w przypadku większej strony WWW (np. serwis informacyjny) mogło kończy się nawet kilkugodzinnym poszukiwaniem i zamianą wartości. Dzięki CSS wystarczy zmodyfikować wartość w jednym miejscu by wszystkie powiązane ze sobą elementy otrzymały nowe wartości. Do tego nie jest większym problemem zamiany danej palety barw np. na skalę szarości (w przypadku żałoby bądź szczególnego wydarzenia, np. sportowego czy branżowego) – wystarczy podmienić nazwę używanego pliku CSS.

Do poszczególnych elementów strony można odwoływać się na kilka sposobów:
a) poprzez identyfikator – jeżeli w danym projekcie chcemy dokonywać zmian wyglądu tylko jednego, konkretnego elementu w danym dokumencie to możemy skorzystać z atrybutu id. Należy jedynie pamiętać by w danym dokumencie HTML istniał TYLKO JEDEN element z danym identyfikatorem! Nazwę identyfikatora w CSS poprzedzamy krzyżykiem (#).
Przykład zastosowania:
<script>#nowy_test {color: red;}</script>
Witaj świecie

b) poprzez klasę – działa podobnie do identyfikatora lecz pozwala na ustawianie atrybutów wielu elementów w obrębie danego dokumentu (i całego projektu). Nazwę klasy poprzedzamy kropką (.).
Przykład zastosowania:
<script>.nowy_test {color: red;}</script>
Witaj świecie
Moja pierwsza witryna

c) poprzez nazwę elementów – styl można zastosować do poszczególnych elementów HTML, z pominięciem atrybutów id czy class. Tego typu ustawienia będą tyczyć się WSZYSTKICH znaczników o podanej nazwie.
Przykład zastosowania:
<script>p {color: red;}</script>
<p Witaj świecie
Moja pierwsza witryna

d) poprzez dziedziczenie – CSS pozwala na ustawianie stylu elementów, które mają następnie mają się aktywować w przypadku występowania z innymi elementami (bądź wewnątrz innych elementów). Odpowiadają za to kolejno selektory opisane poniżej:
- element1 element2 – nadaj styl elemenent2 za każdym razem gdy występuje wewnątrz element1
- element1 > element2 – nadaj styl element2 za każdym razem gdy jego bezpośrednim rodzicem jest element1
- element1 ~ element2 – nadaj styl element2 za każdym razem, gdy występuje on następnej kolejności po element1 (każdorazowo!). Co istotne, elementy te muszą występować w ramach tego samego elementu nadrzędnego.
- element1 + element2 – nadaj styl element2 pierwszemu elementowi występującemu po element1
- element[atrybut] – nadaj styl dla element jeżeli posiada on zadeklarowany wskazany atrybut

INFORMACJA: Ostatnia pozycja może zawierać także warunki logiczne, które mogą testować wartość wskazanego atrybutu.

Istnieje także jeszcze jeden, specjalny selektor CSS – gwiazdka (*). Pozwala on na zdefiniowanie stylu, który będzie narzucany wszystkim elementom, niezależnie od ich nazwy, identyfikatora czy nazwy elementu.

Przy tworzeniu arkuszy trzeba pamiętać o zachowywaniu priorytetów selektorów. Najważniejszym, z punktu widzenia przeglądarki, zawsze jest styl deklarowany poprzez atrybut style, pozwalający na nadpisanie stylu zadeklarowanego w innym miejscu. Natomiast w plikach CSS priorytet rozkłada się wedle zasady „im szczegółowiej został określony styl, tym pewniejsze jego przypisanie”. Stąd,w przypadku pojedynczych selektorów, wygrywa selektor identyfikatora. Następny po nim jest selektor klasy. Na końcu jest CSS atrybutu. W przypadku łączonych wskazań wszystko zależne jest od ich układu. Najlepiej ilustruje to poniższy przykład (zaczerpnięty ze strony HTML Dog – odnośnik w materiałach):

p posiada szczegółowość 1 (1 HTML selector)
div p posiada szczegółowość 2 (2 HTML selectors, 1+1)
.tree posiada szczegółowość 10 (1 class selector)
div p.tree posiada szczegółowość 12 (2 HTML selectors + a class selector, 1+1+10)
#baobab posiada szczegółowość 100 (1 id selector)
body #content .alternative p posiada szczegółowość 112 (HTML selector + id selector + class selector + HTML selector, 1+100+10+1)

Oczywiście im wyższa szczegółowość, tym większe prawdopodobieństwo przypisania stylu do danego elementu (jak widać ostatni selektor wygra z pozostałymi).

W przypadku gdy istnieją dwie deklaracje o identycznym priorytecie ZAWSZE wygrywa deklaracja ostatnia (nadpisuje poprzednią).

Przed przystąpieniem do projektowania warto jest zawsze zapoznać się ze wszystkimi zmianami w standardzie oraz, co najważniejsze, ze wsparciem poszczególnych rozwiązań w przeglądarkach. Może się bowiem okazać, że chociaż pewne rozwiązania zostały włączone do standardu (np. flexbox) to poszczególne przeglądarki inaczej interpretują ten element (wymaga odpowiednich deklaracji dla minimum 2 głównych przeglądarek WWW).

Chociaż w internecie można natknąć się na projektantów stron, którzy próbują przekonywać do tworzenia stron zgodnych ze starszymi przeglądarkami to nie powinno się do tego stosować chyba, że klient sobie tego zażyczy (za odpowiednią dopłatą). Tworzenie stron z dostosowaniem do starszych, często nie dostosowanych do standardu przeglądarek internetowych pośrednio wpływa na jakość i szybkość działania stron w przeglądarkach nowoczesnych i dobrze napisanych. Wpływ ten wynika chociażby z konieczności pobrania i wykonania dodatkowego kodu, który często jest na serwerach zewnętrznych (pobranie i załączenie takowej biblioteki na swoim serwerze powoduje dodatkowy kłopot w postaci ręcznej aktualizacji biblioteki).

3. Zadania do wykonania:

Należy wykonać szkielet strony WWW wykorzystując HTML oraz CSS. Strona ma zostać zbudowana z wykorzystaniem elementów semantycznych. Arkusze mają być dołączane z osobnych plików CSS. Od początku projektu należy dbać o jakość i schludność kodu. Strona może mieć dowolny wygląd, należy jednak pamiętać o kilku obowiązkowym dołączeniu następujących fragmentów:

- tytuł strony – obecnym trendem jest tworzenie tytułu z odpowiednio dobranych fontów oraz grafiki CSS (gradient, zaokrąglenia, cienie itp.). Należy wykorzystać minimum 3 efekty CSS (do tekstu bądź tła).
- menu główne strony, w którym muszą znaleźć się minimalnie 4 odnośniki. Jeżeli strona ma mieć charakter prywatny podstronami mogą być np. O mnie, Moje osiągnięcia, Moje projekty, Kontakt. Jeżeli strona ma być firmowa – O nas, Oferta, Galeria, Kontakt. W przypadku sklepu – Promocje, Wybór działu, Napisali o nas, Kontakt. Jeżeli zostanie wybrana inna tematyka strony to nazwy odsyłaczy powinny zostać adekwatnie zmienione.
- strona właściwa powinna zostać podzielona minimum na dwie części – treść właściwą oraz przypis boczny (służący np. za podręczny pasek nawigacji, pasek reklam bądź opisów do treści właściwej na stronie WWW).
- stopka strony, która powinna zawierać informacje prawne oraz skrócone informacje o autorze strony.

Zadaniem dodatkowym może być utworzenie ramki z informacją o wykorzystywaniu ciasteczek celem personalizacji strony. Na chwilę obecną można (lecz nie trzeba) dodać zdarzenie ukrycia tego okna po kliknięciu przycisku Anuluj.

4. Materiały do zadania:
http://htmldog.com/guides/css/intermediate/specificity/
https://www.w3schools.com
https://msdn.microsoft.com/en-us/library/hh772960(v=vs.85).aspx
https://developer.mozilla.org/pl/
https://www.html5rocks.com/en/