MODUŁ 2

CSS WPROWADZENIE

Kaskadowe arkusze stylów (CSS - Cascading Style Sheets)

Kaskadowe arkusze stylów (CSS) są używane do oddzielenia zawartości strony internetowej od znaczników i parametrów służących do formatowania jej wyglądu. Dzięki temu zachowany jest pewien porządek, co ułatwia modyfikowanie zarówno treści, jak i sposobu jej wyświetlania. Ponadto, za pomocą CSS można zdefiniować różny sposób formatowania tej samej strony, a nawet zmienić standardowe formatowanie znaczników HTML, czy utworzyć własne znaczniki, nie istniejące w HTML, i przy pomocy CSS określić sposób wyświetlania umieszczonej wewnątrz nich treści.

Dołączanie CSS do strony internetowej

Definicja elementów formatujących (CSS) może zostać dołączona do tworzonej strony na dwa sposoby:

- może być umieszczona wewnątrz znacznika `<style>`, zarówno w sekcji zawierającej treść, jak i w sekcji nagłówkowej, przy czym częściej wykorzystywany jest ten drugi sposób:
  ```
  <head>
  <style>
  definicja stylów CSS
  </style>
  </head>
  ```
- może zostać zapisana w osobnym pliku i dołączona do strony w sekcji `<head>` przy pomocy znacznika `<link>`:

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

W większości przykładów zamieszczonych w e-lekcjach wykorzystywany jest pierwszy sposób. Ponieważ kody CSS i HTML są interpretowane przez przeglądarkę, pliki zawierające zamieszczone przykłady można tworzyć zarówno na dysku lokalnym (wtedy będą one dostępne jedynie na tym dysku) lub w katalogu *public_html* na własnym koncie na serwerze ie.uek.krakow.pl (wtedy będą one widoczne w Internecie).

Selektory, właściwości, wartości

Budowa selektora

Podstawową jednostką definiowaną w CSS jest tzw. selektor. Jeśli jego nazwa jest taka sama, jak nazwa znacznika HTML, definiuje on parametry tego właśnie znacznika (czyli sposobu wyświetlania znajdującej się wewnątrz niego treści). Definicja selektora jest umieszczona **w nawiasach klamrowych** i składa się z par **właściwość** (zwana także cechą), po niej występuje dwukropek a następnie **wartość** nadawana tej właściwości. Na końcu wartości występuje średnik, ostatnia wartość w definicji selektora (przed nawiasem klamrowym zamykającym) nie musi być zakończona średnikiem. Właściwość może mieć wiele wartości, wtedy są one oddzielone spacjami:

```
selektor
```

```
{
właściwość1: wartość1a wartość1b wartość1c;
...
właściwość2: wartość2a
}
```

Wskaż wszystkie zdania prawdziwe (trzy):

- [x] Definicja selektora składa się z par właściwość:wartość
- [ ] Selektor musi mieć taką samą nazwę jak znacznik HTML, którego parametry definiuje
- [x] Podstawową jednostką definiowaną w CSS jest selektor
- [ ] Na końcu definicji właściwości występuje przecinek
- [ ] Na końcu definicji selektora występuje średnik
- [x] Właściwość może mieć wiele wartości
- [ ] Definicja selektora jest umieszczana w nawiasach kwadratowych
- [ ] Właściwość jest oddzielona od wartości średnikiem

ODP: 136

Zmiana ustawień standardowych

Standardowo tekst na stronie internetowej ma kolor czarny. Utwórz plik tekstowy (w *Notatniku* lub na serwerze) i umieść w nim następujący kod:

```
<b><i>
Tekst jest pogrubiony i pochylony, ale czarny<br>
</i></b>
Tekst nie jest wyróżniony, ale też ma kolor czarny
```

Następnie zapisz plik, nadając mu dowolną nazwę i rozszerzenie *htm* (pamiętaj, że jeśli tworzysz plik w *Notatniku*, jego nazwę umieść w cudzysłowie, wtedy *Notatnik* nie dołączy standardowego rozszerzenia txt i plik będzie skojarzony z domyślną przeglądarką).

Jeśli teraz wyświetlisz zawartość pliku w przeglądarce, zobaczysz, że tekst jest wyświetlony w całości kolorem czarnym, gdyż takie jest ustawienie standardowe w przeglądarkach.

Zmiana ustawień standardowych, c.d.

Zmodyfikuj zawartość utworzonego pliku dodając **na początku** kod:

```
<style>
* {color: red;}
</style>
```

Następnie zapisz plik i wyświetl jego zawartość w przeglądarce. Tym razem cały tekst jest wyświetlony kolorem czerwonym.

W przykładzie zmieniono standardowe ustawienie koloru tekstu wskazując, że dla całej treści na stronie internetowej kolor tekstu będzie standardowo czerwony. Wykorzystano w tym celu gwiazdkę - selektor ten oznacza zmianę ustawień standardowych dla całej treści znajdującej się na stronie internetowej. Oczywiście dla poszczególnych fragmentów może to zostać zmienione poprzez wykorzystanie klas, pseudoklas, identyfikatorów lub znaczników HTML.

Przykład wykorzystania selektora

Znaczniki nagłówkowe w HTML (<h1>, <h2>, ..., <h6>) służą do wyróżnienia tekstu, przy czym przeglądarka ma zdefiniowany sposób tego wyróżnienia i to ona "decyduje", jak tekst będzie wyglądał. Korzystając z CSS możemy sami wskazać, jak ma być sformatowany tekst umieszczony wewnątrz znaczników nagłówkowych. Utwórz plik html (w *Notatniku* lub w katalogu *public_html* na serwerze) i umieść w nim kod:

```
<h1>
Tekst wyróżniony przy pomocy znacznika H1
</h1>
```

następnie zapisz plik, nadając mu nazwę *style.htm* (pamiętaj, że jeśli tworzysz plik w *Notatniku*, jego nazwę umieść w cudzysłowie, wtedy *Notatnik* nie dołączy standardowego rozszerzenia txt i plik będzie skojarzony z domyślną przeglądarką).

Jeśli teraz wyświetlisz zawartość pliku w przeglądarce, zobaczysz, że wyświetlony tekst jest pogrubiony i pisany dużą czcionką, gdyż tak przeglądarki interpretują formatowanie za pomocą znacznika `<h1>`. Korzystając z CSS możemy zmienić to standardowe ustawienie. Zobacz, jak...

Przykład wykorzystania selektora, c.d.

Zmodyfikuj utworzony plik do postaci kodu (pomiń umieszczone na początku numery wierszy):

```
1 <head>
2 <style>
3 h1
4 {
5 color:red; font-style:italic; font-size:16pt; letter-spacing:2pt;
6 }
7 </style>
8 </head>
9 <h1>
10 Ten tekst jest wyświetlany przy pomocy znacznika H1, lecz ma
zmieniony format
11 </h1>
```

W sekcji `<head>` wewnątrz znacznika `<style>` ustalono inne niż standardowe formatowanie dla znacznika `<h1>` (wiersze 3−6). W znajdującej się tam definicji selektorem jest `h1`, co oznacza, że wszystkie teksty na stronie internetowej umieszczone wewnątrz znacznika `<h1>` będą miały zdefiniowany w tym miejscu wygląd. Dla selektora określono wartości dla czterech właściwości, dzięki czemu tekst ograniczony znacznikiem `<h1>` będzie miał następujące parametry (wiersz 5):

- kolor czcionki czerwony (`color:red;`)
- czcionka pochylona (`font-style:italic;`)
- rozmiar czcionki 16 punktów (`font-size:16pt;`)
- odległość między znakami 2 punkty (`letter-spacing:2pt;`)

Sprawdź, wyświetlając zawartość pliku w przeglądarce.

W ramach ćwiczenia definicję selektora umieść w osobnym pliku CSS, dołącz go do pliku HTML i zaobserwuj efekt w przeglądarce.
Dla przypomnienia. Aby dołączyć plik CSS do pliku HTML należy w sekcji `<HEAD>` pliku HTML wpisać:
`<link rel="stylesheet" type="text/css" href="`*`katalog/plik.css`*`"/>`

Przykład wykorzystania selektora - formatowanie tabeli

Utwórz plik HTML i umieść w nim kod:

```
1 <style>
```

```
2 table {
3 border: thick dotted blue; }
4 </style>
5 <table>
6 <tr>
7 <td>pierwsza komórka
8 <td>druga komórka
9 <tr>
10 <td>trzecia komórka
11 <td>czwarta komórka
12 </table>
```

W przykładzie zdefiniowano tabelę (wiersze 5-12) oraz ustalono, przy pomocy CSS, sposób formatowania jej krawędzi zewnętrznej (wiersze 2-3). W tym celu w sekcji `<style>` zdefiniowano jeden selektor (*table*), który dotyczy znacznika HTML o takiej samej nazwie (wiersze 2−3). Ustalono w nim następujące parametry ramki (*border*) rysowanej naokoło tabeli: linia gruba (*thick*), kropkowana (*dotted*), koloru niebieskiego (*blue*). Zauważ, że średnik występuje dopiero po nazwie koloru, a poszczególne wartości (*thick dotted blue*) są oddzielone spacjami. Średnik ten można w tym miejscu pominąć, gdyż bezpośrednio po nim występuje nawias klamrowy zamykający definicję selektora.

Selektory potomków

W CSS wykorzystywane są pojęcia selektorów-dzieci i selektorów-potomków. Ich istota jest identyczna, jak w przypadku więzi rodzinnych. W HTML dany element jest dzieckiem innego elementu, jeśli jest jego bezpośrednim podelementem, natomiast jest potomkiem, gdy jest dowolnym podelementem, niekoniecznie bezpośrednim (dzieckiem). W poniższym przykładzie element `<dziecko>` jest zarówno dzieckiem jak i potomkiem elementu `<rodzic>`, ale `<wnuk>` i `<prawnuk>` nie są dziećmi elementu `<rodzic>`, są natomiast jego potomkami. Podobnie, element `<wnuk>` jest dzieckiem i potomkiem elementu `<dziecko>`, ale `<prawnuk>` jest jedynie potomkiem elementu `<dziecko>`, nie jest natomiast jego dzieckiem.

```
<rodzic>
<dziecko>
<wnuk>
<prawnuk>
</prawnuk>
</wnuk>
</dziecko>
</rodzic>
```

W CSS istnieje możliwość definiowania właściwości jedynie dla potomków.

Selektory potomków - przykład

```
1 <style>
2ol ul ul {color:red;}
3 </style>
4 <ol>
5<li>Zima
6 <ul>
7 <li>Grudzień
8 <ul>
9 <li>Mikołaj
10 <li>Boże Narodzenie
11 </ul>
12 <li>Styczeń
13</ul>
```

```
14<li>Wiosna
15</ol>
```

W przykładzie zdefiniowano listę numerowaną <ol> (wiersze 4−15), wewnątrz niej listę wypunktowaną <ul> (wiersze 6−13), a wewnątrz niej kolejną listę wypunktowaną <ul> (wiersze 8−11), w każdym przypadku tworząc elementy list oczywiście przy pomocy znaczników <li>.
W sekcji <style> (wiersze 1−3) zdefiniowano selektor *ol ul ul*, co oznacza, że ustalone w ten sposób formatowanie dotyczy jedynie ostatniej listy <ul>, będącej potomkiem zarówno listy <ol> jak i pierwszej listy <ul>. Kolorem czerwonym wyróżnione więc zostaną elementy <li> znajdujące się w wierszach 9−10 (sprawdź). Jeśli w wierszu 2 usunięte zostanie jedno ul (czyli będzie on miał postać: *ol ul {color: red;}*), będzie to oznaczało, że definiowany selektor dotyczy **wszystkich potomków** (nie tylko dzieci) elementu <ol>, a więc, w przykładzie, obydwu list wypunktowanych, gdyż obie są potomkami <ol> (pierwsza z nich jest także dzieckiem, ale to nie ma znaczenia). Ostatecznie kolorem czerwonym wyróżnione zostaną elementy <li> obydwu list <ul>, znajdujące się w wierszach 7 i 12 (są to elementy pierwszej listy wypunktowanej) oraz 9 i 10 (elementy drugiej listy wypunktowanej) - sprawdź. Jeśli w wierszu drugim usunięte zostaną obydwa *ul*, pozostanie tam jedynie selektor *ol*, co sprawi, że cały tekst będzie wyróżniony kolorem czerwonym, gdyż obydwie listy <ul> są potomkami <ol>, a definicja selektora dotyczy zarówno jego, jak i wszystkich jego potomków (sprawdź).

Selektory potomków - przykład 2

```
1 <style>
2ol ul ul {color:red;}
3 </style>
4 <ol>
5<li>Zima
6 <ul>
7 <li>Grudzień
8 <ul>
9 <li>Mikołaj
10 <li>Boże Narodzenie
11 </ul>
12 <li>Styczeń
13</ul>
14<li>Wiosna
15</ol>
```

Jeśli nazwy selektorów są oddzielone przecinkami a nie spacjami, selektory te są traktowane niezależnie. Definicja zaczynająca się od *ol, ul* oznacza więc, że definiowane są parametry dla wszystkich znaczników <ol> i <ul> występujących na stronie, niezależnie od tego, czy są zagnieżdżone, czy nie, podczas gdy *ol ul* (bez przecinka między nimi) oznacza, iż definiowane są jedynie parametry dla znaczników <ul> będących potomkami <ol>.
Jeśli więc w powyższym przykładzie w wierszu 2 pomiędzy *ol* a *ul* wstawiony zostanie przecinek zamiast spacji (i jednocześnie drugie *ul* będzie usunięte), zapis taki będzie oznaczał, że definiowane są właściwości dwóch niezależnych od siebie selektorów: *ol* i *ul*. W tym przypadku będzie to miało taki sam efekt, jak użycie jedynie *ol*, gdyż wszystkie znaczniki <ul> w kodzie HTML (wiersze 4−15) są potomkami <ol>. Gdyby jednak na stronie występowały elementy <ul> nie będące potomkami <ol>, taka definicja miałaby sens (sprawdź, umieszczając na końcu przykładu dowolną listę <ul>).

Definicja ol, ul {color : red } dotyczy:

- jedynie znaczników ol, dla których istnieje dziecko ul
- jedynie znaczników ul będących potomkami ol
- jedynie znaczników ul, dla których istnieje potomek ol
- **wszystkich znaczników ol i wszystkich znaczników ul**
- jedynie znaczników ol, dla których istnieje potomek ul

Klasy, pseudoklasy, identyfikatory

Klasy

Jeśli przy pomocy CSS zdefiniujemy atrybuty dla danego znacznika HTML, tekst formatowany względem niego będzie miał zawsze taki sam wygląd. Aby zróżnicować ten wygląd można skorzystać z klas. Klasy mogą być związane z konkretnym znacznikiem (wtedy po nazwie selektora występuje kropka i nazwa klasy) lub nie (wtedy definicja klasy zaczyna się od gwiazdki i kropki lub tylko od kropki). Można na przykład utworzyć dwie lub większą liczbę klas dla tego samego znacznika i dzięki temu tekst umieszczony wewnątrz niego w poszczególnych miejscach na stronie może mieć różny wygląd, zależnie od tego, z której klasy skorzystamy. Można także zdefiniować klasę nie związaną z konkretnym znacznikiem i korzystać z niej w różnych miejscach w kodzie HTML.

Klasy - przykład

W pliku HTML umieść następujący tekst:

```
1 <style>
2 p.gruby {font-weight : bold;}
3 p.pochylony {font-style : italic;}
4 </style>
5 <body>
6 <p class="gruby">Ten tekst jest napisany czcionką pogrubioną</p>
7 <p class="pochylony">Ten tekst jest napisany czcionką pochyloną</p>
8 </body>
```

W sekcji <style> (wiersze 1−4) zdefiniowano dwie klasy dla znacznika <p>. Dla pierwszej z nich, o nazwie *gruby* (*p.gruby* − wiersz 2), ustalono jedną właściwość − czcionka ma być pogrubiona. Dla klasy o nazwie *pochylony* (*p.pochylony* − wiersz 3) także ustalono jedną właściwość − czcionka będzie napisana kursywą (czcionką pochyłą). Następnie, w sekcji <body> (wiersze 5−8), umieszczono dwa teksty wewnątrz znaczników <p>, przy czym w pierwszym przypadku, korzystając z parametru *class* użyto klasy *gruby* (wiersz 6), w drugim klasy *pochylony* (wiersz 7).

Przykład klasy nie związanej z konkretnym znacznikiem

```
1 <style>
2 .pochylony {font-style : italic;}
3 </style>
4 <p class="pochylony">Ten tekst jest napisany czcionką pochyloną</p>
5 <h1 class="pochylony">Ten tekst też jest napisany kursywą</p>
```

W przykładzie zdefiniowano jedną klasę o nazwie *pochylony* (wiersz 2). Jej nazwa jest poprzedzona jedynie kropką, co oznacza, że klasa ta może być użyta wewnątrz dowolnego znacznika HTML. W przykładzie użyto jej do pochylenia tekstu wewnątrz znaczników <p> i <h1> (wiersze 4 i 5). Pozostałe atrybuty tekstu w obydwu przypadkach pozostają niezmienione.

Pseudoklasy

Pseudoklasy to takie klasy, których nie trzeba tworzyć, gdyż dla niektórych znaczników HTML zostały predefiniowane. Są one wykorzystywane m.in. do wykonania określonej akcji w odpowiedzi na zachowania użytkowników podczas przeglądania strony. Typowym przykładem mogą tu być linki do stron internetowych, których wygląd jest uzależniony od tego, czy użytkownik „odwiedził" już stronę do której odnosi się dany link, czy umieścił kursor myszy na linku, czy nie wykonał dotychczas żadnej akcji związanej z danym linkiem. Aby zmienić atrybuty pseudoklasy, należy postępować jak w przypadku klas, różnica jest taka, że w wyrażeniu ścieżkowym, po nazwie selektora występuje dwukropek a nie kropka. Ponieważ pseudoklasy są predefiniowane, więc istnieją niezależnie od tego, czy ich atrybuty są definiowane przy pomocy CSS, czy nie.

Pseudoklasy - przykład

Dla znacznika <a> w HTML predefiniowane są następujące pseudoklasy:
active - użytkownik kliknął link, strona jest aktualnie wczytywana,
link - użytkownik nie kliknął linku, strona nie była jeszcze odwiedzona,
visited - link był kliknięty, strona była odwiedzona,
hover - kursor myszy znajduje się nad linkiem.
Jeśli nie zostanie to zmienione, przeglądarki odpowiednio formatują odnośniki na stronie internetowej (np. zawsze są one podkreślone).

Utwórz plik HTML. Umieść w nim poniższy kod, w miejsce *adres_stronyX* wpisując adresy trzech różnych istniejących stron internetowych:

```
<a href="http://adres_strony1">strona pierwsza</a><br>
<a href="http://adres_strony2">strona druga</a><br>
<a href="http://adres_strony3">strona trzecia</a>
```

Następnie wyświetl zawartość pliku w przeglądarce internetowej, kliknij jeden z linków i zaobserwuj, jak przeglądarka wyświetla linki do stron odwiedzonych a jak nieodwiedzonych.

Pseudoklasy - przykład, c.d.

Na początku utworzonego pliku umieść kod:

```
1 <style>
2 a:visited {color : red;}
3 a:link {color : black;}
4 a:hover {background-color: yellow; color : green;}
5 </style>
```

Zmień także adresy stron na inne, w przeciwnym przypadku, gdy odwiedziłeś(-aś) wszystkie strony, nie zaobserwujesz efektu związanego z likami nieodwiedzonymi.

Dodając powyższy kod zdefiniowano format wyświetlania tekstu ograniczonego znacznikiem <a>, gdy na stronie internetowej będą miały miejsce zdarzenia związane z trzema pseudoklasami. Zgodnie z definicją pseudoklas, linki odwiedzone (*visited*) będą miały kolor czerwony (wiersz 2), linki nieaktywne i nieodwiedzone (*link*) będą miały kolor czarny (wiersz 3), a te, nad którymi umieszczony zostanie kursor myszy (*hover*) będą miały kolor zielony na żółtym tle (wiersz 4). Zauważ na stronie internetowej, że wszystkie linki są podkreślone – jest to ustawienie standardowe dla omawianych pseudoklas, a ponieważ nie zostało zmienione w sekcji <style>, więc cały czas jest obowiązujące.

Pseudoklasy - przykład 2

Dla znacznika <p> (oraz innych, np. znaczników nagłówkowych h) istnieje pseudoklasa o nazwie *first-letter*, dzięki której można wyróżnić pierwszą literę akapitu. Standardowo nie są dla niej ustawione żadne szczególne atrybuty, korzystając z CSS można to zmienić. Utwórz plik HTML zawierający poniższy kod:

```
<style>
p:first-letter {
font-size: 24pt; color: red; background-color: yellow}
```

```
</style>
<p>Tekst w tym akapicie ma wyróżnioną pierwszą literę</p>
```

W powyższym przykładzie w sekcji <style> ustalono, że pierwsza litera akapitu (pseudoklasa *first-letter* dla selektora *p*) będzie miała wielkość 24 punktów i będzie napisana kolorem czerwonym na żółtym tle (sprawdź).

Podobnie, używając pseudoklasy *first-line*, można zdefiniować parametry dla pierwszego wiersza (czyli fragmentu tekstu znajdującego się przed pierwszym
).

Identyfikatory

Identyfikatory pełnią tę samą rolę, co klasy i podobnie jak klasy mogą być związane z konkretnym znacznikiem lub nie. W taki sam sposób się je także definiuje, jedyna różnica polega na tym, że w miejsce kropki należy użyć znaku # (hash):

```
znacznik#nazwa_identyfikatora
{
właściwość : wartość1 wartość2 ... ;
...
}
```

lub, gdy identyfikator nie jest związany z konkretnym znacznikiem:

```
#nazwa_identyfikatora
{
właściwość : wartość1 wartość2 ...;
...
}
```

Druga różnica między identyfikatorami a klasami polega na tym, że aby sformatować tekst umieszczony wewnątrz znacznika HTML z wykorzystaniem identyfikatora należy, w miejsce *class*, użyć parametru *id*:

```
id="nazwa_identyfikatora"
```

Identyfikatory - przykład

W pliku HTML umieść następujący kod:

```
1 <style>
2 #kolor { color : blue; }
3 </style>
4 <h1 id="kolor">Tekst jest niebieski, gdyż użyto identyfikatora</h1>
5 <font color="red">Tekst jest czerwony</font>
6 <p id="kolor">Tekst jest niebieski dzięki identyfikatorowi</p>
7 <font color="green">Tekst jest zielony</font>
8 <h1 id="kolor">Tekst znów jest niebieski</h1>
```

W sekcji <style> utworzono identyfikator *#kolor* definiując w nim niebieski kolor tekstu (wiersz 2). Następnie, wewnątrz dwóch różnych znaczników: dwukrotnie w <h1> oraz raz w <p> umieszczono parametr *id="kolor"* (wiersze 4, 6 i 8). Wyświetlony w ten sposób tekst we wszystkich tych przypadkach będzie miał taki kolor, jak zdefiniowano w identyfikatorze, a więc niebieski. Pozostałe atrybuty formatowania (wielkość czcionki, kolor tła, itd.), ponieważ nie zostały ustalone w identyfikatorze, będą miały standardowe wartości. Pomiędzy omówionymi znacznikami, w których użyto identyfikatora kolor, wstawiono tekst wyświetlany w kolorze czerwonym (wiersz 5) oraz zielonym (wiersz 7) – w tych przypadkach użyto HTML-owego znacznika <font>.

Wskaż wszystkie zdania prawdziwe:

- a:hoover to początek definicji pseudoklasy hoover dla znacznika a
- W HTML występuje następujący kod:
  <abc class="a" ...
  <xyz class="a" ...
  Oznacza to, że a jest klasą, której początek definicji zaczyna się od:
  abc,xyz.a
- pseudoklasy są predefiniowane
- #kolor to początek definicji znacznika kolor
- .x to początek definicji identyfikatora x, który może być użyty wewnątrz dowolnego znacznika HTML
- a.x to początek definicji klasy x dla znacznika a
- #kolor to początek definicji identyfikatora kolor

CSS - przykłady

CSS - przykłady

W tej lekcji pokazano trzy praktyczne przykłady wykorzystania CSS:

- do wyświetlenia dodatkowej informacji na stronie w postaci tzw. "dymka",
- do utworzenia menu,
- do zdefiniowania tzw. warstw (pojemników).

Wyświetlenie dodatkowej informacji na ekranie w postaci tzw. "dymka"

Niekiedy po umieszczeniu kursora myszy na danym obiekcie (fragmencie tekstu, rysunku) wyświetlana jest krótka informacja (tzw. „dymek”). W poniższym przykładzie pokazano, jak można to zrobić korzystając z CSS:

```
1 <style>
2 .dymek {position: relative; text-decoration: none;}
3 .dymek span {display: none;}
4 .dymek:hover span {
5 border: solid 1px #FF0000; width: 200px; display: block;
6 position: absolute; top: 50px; background: yellow;
7 color: black; }
8 </style>
9 <body>
10 <br/>
11 <a href="" class="dymek">Link numer 1
12 <span>Informacja w dymku pierwszym</span></a>
13 <br>
14 <a href="" class="dymek">Link numer 2
15 <span>Informacja w dymku drugim</span></a>
16 <body>
```

W sekcji `<style>` (wiersze 1−8) zdefiniowano trzy klasy.
Pierwsza z nich, *dymek* (wiersz 2), jest wykorzystywana w sekcji `<body>` do wyświetlenia tekstu odnośnika (wiersze 11−12 oraz 14−15).

Klasa *.dymek span* (wiersz 3) jest wykorzystywana do ukrycia (*display:none;*) tekstu umieszczonego wewnątrz znacznika . Dzięki temu obydwa teksty znajdujące się wewnątrz znaczników (wiersze 12 i 15) są początkowo niewidoczne. Zauważ, że klasa ta dotyczy tylko tych elementów , które są umieszczone wewnątrz znaczników mających parametr *class="dymek"* (tak jest w tym przypadku: obydwa znaczniki znajdują się wewnątrz znaczników <a>, które są klasy *dymek*). Gdyby znacznik <a> nie miał parametru *class="dymek"*, zdefiniowany wewnątrz niego tekst umieszczony w byłby od razu widoczny, gdyż klasa *.dymek span* nie miałaby tutaj zastosowania.
Trzecia klasa *.dymek:hover span* (wiersze 4−7) zawiera definicję sposobu wyświetlania tekstu umieszczonego wewnątrz znacznika , gdy nad tekstem znajdującym się wewnątrz znacznika należącego do klasy *dymek* umieszczony zostanie kursor myszy (wykorzystano tu pseudoklasę *hover*). W obydwu przypadkach parametrem *href* znacznika <a> jest łańcuch pusty (wiersze 11 i 14), co oznacza tzw. „pusty link", czyli po kliknięciu tekstu umieszczonego wewnątrz znacznika <a> nie zostanie wykonana żadna akcja. Oczywiście, w ramach ćwiczenia, możesz w to miejsce wpisać adresy dowolnych stron internetowych, wtedy będą to pełnoprawne linki. Spróbuj też modyfikować poszczególne atrybuty klas w sekcji <style> (wiersze 1−8) i obserwuj efekt w przeglądarce.

Menu

W przykładzie zostanie pokazane, jak wykorzystać CSS do utworzenia menu. Ponadto, aby zobrazować możliwość umieszczenia kodu CSS w osobnym pliku, utworzone zostaną dwa pliki: jeden zawierający kod HTML, drugi CSS. Utwórz dwa pliki (koniecznie w tym samym katalogu) i wpisz w nich poniższe kody:
plik *menu.htm*:
```
1 <link rel="stylesheet" type="text/css" href="menu.css" />
2 <opcje id="menu">
3 <a href="">Menu pozycja I</a>
4 <a href="">Menu pozycja II</a>
5 <a href="">Menu pozycja III</a>
6 <a href="">Menu pozycja IV</a>
7 <a href="">Menu pozycja V</a>
8 </opcje>
```
plik *menu.css*:
```
1 <style type="text/css">
2 #menu {float: left;}
3 #menu a {width: 150px; display: block; text-decoration: none;
4 background-color: cyan; color: black; padding: 1px;
5 border-left: 3px solid cyan; margin-top: 3px; }
6 #menu a:hover {background-color: #ccc; color: #FF0000;
7 border-style: inset; border-left: 3px solid #FF0000;}
8 </style>
```

W pliku *menu.htm* utworzono stronę internetową, do której dołączony został plik *menu.css* (wiersz 1), czyli kaskadowy arkusz stylów zawierający informacje formatujące dla znaczników zdefiniowanych w pliku *menu.htm*.
W pliku CSS zdefiniowano trzy identyfikatory. Pierwszy *#menu* (wiersz 2) zawiera informację dotyczącą sposobu wyświetlania elementów mających parametr *id="menu"* (a więc wszystkich elementów zdefiniowanych wewnątrz znacznika <opcje> w pliku *menu.htm*) w postaci tzw. obiektów pływających umieszczonych jeden pod drugim przy lewej krawędzi strony (właściwość *float:left;*). Identyfikator *#menu a* (wiersze 3−5) opisuje sposób wyświetlenia elementów <a>, ale tylko tych znajdujących się wewnątrz znacznika, któremu przypisano identyfikator *#menu*. Natomiast *#menu a:hover* (wiersze 6−7) definiuje sposób wyświetlenia tych elementów <a>, gdy umieszczony zostanie na nich kursor myszy (wykorzystana pseudoklasa *hover*). W efekcie na stronie pojawi się menu, a gdy kursor myszy zostanie umieszczony nad jedną z opcji (bez klikania), będzie ona podświetlona zgodnie z parametrami zapisanymi w identyfikatorze *#menu a:hoover*. Po utworzeniu obydwu plików próbuj modyfikować parametry w pliku *menu.css* (np. usuwaj poszczególne właściwości lub zmieniaj ich wartości) i obserwuj efekt w przeglądarce.

W przykładzie użyto nieistniejącego w HTML znacznika <opcje> (plik *menu.htm*, wiersz 2). Wewnątrz

niego zdefiniowano pięć elementów <a> (wiersze 3−7). Ponieważ definicji znacznika <opcje> nie ma w pliku *menu.css*, więc przeglądarka nie użyje żadnego formatowania, zignoruje ten znacznik. Jednak parametrem tego znacznika jest *id="menu"* (plik *menu.htm,* wiersz 2), a identyfikator *#menu* został zdefiniowany w pliku *menu.css* (wiersz 2), więc ustalone w ten sposób parametry formatowania będą obowiązujące w tym miejscu dla znacznika <opcje>.

Warstwy (pojemniki)

Warstwy (lub inaczej pojemniki) są wykorzystywane do podziału strony internetowej na fragmenty, które mogą być umieszczone w dowolnych miejscach w przeglądarce. W HTML do tworzenia pojemników wykorzystywany jest znacznik . Można zdefiniować dowolną liczbę warstw, umieszczając w nich różną treść, która także może być dowolna (w szczególności może być innym pojemnikiem) i umieszczać je w wybranych miejscach strony internetowej, a nawet ukrywać. Oprócz treści pojemniki mają atrybuty przy pomocy których definiuje się ich rozmiar, położenie, czy wygląd znajdującego się w nich tekstu. Atrybuty te mogą być formatowane przy pomocy CSS.

Warstwy (pojemniki) - przykład

Utwórz dwa pliki (w tym samym katalogu):
warstwy.htm:
```
1 <link rel="stylesheet" type="text/css" href="warstwy.css" />
2 <div id="gora">Nagłówek strony</div>
3 <div id="srodek">Treść strony</div>
4 <div id="dol">Stopka strony</div>
```
warstwy.css:
```
1 #gora {
2 background-color: red;
3 width: 800px;
4 height: 100px;
5 }
6 #srodek {
7 background-color: grey;
8 width: 800px;
9 height: 600px;
10 margin-left: 100px;
11 }
12 #dol {
13 background-color: #ABA;
14 width: 800px;
15 height: 100px;
16 }
```

W pliku *warstwy.htm* umieszczono instrukcje tworzące trzy pojemniki, którym nadano różne identyfikatory id: *gora*, *srodek* i *dol* (wiersze 2−4).
W pliku *warstwy.css* ustalono atrybuty dla każdego pojemnika zdefiniowanego w pliku *warstwy.htm*. Dla wszystkich selektorów zdefiniowano trzy atrybuty:

- kolor tła *background-color* − w pierwszych dwóch przypadkach wykorzystano angielskie nazwy kolorów (wiersze 2 i 7), w trzecim wartość parametru podano w postaci kodu szesnastkowego (wiersz 13),
- szerokość (*width*, wiersze 3, 8 i 14) w pikselach,
- wysokość (*height*, wiersze 4, 9 i 15), także w pikselach.

Identyfikator *#srodek* ma ponadto zdefiniowaną właściwość *margin-left* (wiersz 10), co w daje efekt odsunięcia od lewej krawędzi okna przeglądarki w tym przypadku o 100 pikseli.

Warstwy w HTML są wykorzystywane do:

- organizacji selektorów definiowanych w CSS
- nakładania stron internetowych na siebie
- **podziału strony internetowej na fragmenty**
- wyświetlania zawartości różnych plików zawierających opisy stron internetowych na stronie