

## Instrukcja 3

Starsze implementacje HTML posiadały sporą ilość znaczników i atrybutów, które umożliwiały modyfikację kolorów poszczególnych elementów (tła, obramowania, tekstu), ustawianie właściwości czcionki (krój, wielkość, grubość, pochylenie itp.), wyrównywanie elementów względem siebie (align), centrowanie, itp.

W HTML5 wszystkie te atrybuty, wraz ze znacznikami, zostały bezpowrotnie usunięte. We wcześniejszej wersji, HTML 4.01, były z kolei oznaczone jako przestarzałe. Jak więc ustawiać właściwości tekstu, ułożenia elementów czy też np. szerokości linii?

Rozwiązanie jest dosyć proste – tego typu operacje wykonuje się dzięki kaskadowym arkuszom stylów (Cascade Style Sheet – CSS). W najnowszej, 3 odsłonie, CSS oferuje znacznie więcej niż ustawianie wielkości, koloru czy obramowania danego elementu HTML. Dzięki niemu możemy także dokonywać obrotu elementów w czasie rzeczywistym, chować/pokazywać dany element, tworzyć obramowania z zaokrągleniami, animacje i inne efekty specjalne.

Ze względu na znaczną objętość materiałową w instrukcji zostaną ukazane jedynie najważniejsze, przykładowe możliwości operacji na stylach CSS. Więcej informacji o nich można zdobyć w sieci Internet (np. na stronie <http://www.w3schools.com/cssref/default.asp>).

Składnia CSS jest bardzo prosta i intuicyjna:

```
{font-size:12px; background-color:blue;border-color:red;}
```

font-size, background-color oraz border-color to właściwość konkretnego elementu HTML (np. tagu body, section czy table). To co znajduje się po dwukropku jest wartością – w przykładzie czcionka ma wielkość 12 pikseli, tło kolor niebieski, a obramowanie czerwony. Każdą kolejną właściwość rozdziela się poprzez znak średnika.

Kaskadowe style można włączać do kodu HTML na cztery różne sposoby:

- poprzez atrybut style KAŻDEGO dostępnego znacznika
- poprzez deklarację stylu w nagłówku strony (w znaczniku style)
- poprzez podłączenie pliku ze stylami CSS wykorzystując znacznik <link>
- poprzez atrybut class KAŻDEGO dostępnego znacznika (jednak wcześniej musi zostać dołączony plik CSS/zostać zadeklarowany styl w nagłówku, w którym zdefiniowane zostaną klasy CSS).

### 1. Przykłady podłączenia CSS do istniejącego kodu HTML

a) użycie stylu jednorazowo, dla pojedynczego elementu:

```
<p style="text-align:center;">Tekst w tym paragrafie zostanie wyśrodkowany (o ile to możliwe)</p>
```

b) wstępne zdefiniowanie stylu do konkretnego id elementu na stronie poprzez wpisanie go w tagu style:

```
<!DOCTYPE html>
<html lang="pl">
<head>
  <meta charset="UTF-8"/>
  <title>Pierwsza strona w HTML5</title>
  <style type="text/css">
    #pierwszy {
      color: red;
      background-color: blue;
      text-align: right;
    }
  </style>
</head>
```

```

        font-size: 42px;
    }
</style>
</head>
<body>
<p id="pierwszy">Tekst w tym paragrafie zostanie odpowiednio sformatowany</p>
<p>Ten paragraf nie zostanie zmieniony!</p>
</body>
</html>

```

c) dołączenie arkusza CSS poprzez znacznik link (wraz z wykorzystaniem klas)

```

//to jest plik styl.css
#pierwszy {
    color: red;
    background-color: blue;
    text-align: right;
    font-size: 42px;
}
/*W plikach css możemy wstawiać komentarze – to jest właśnie przykład jednego z nich;
komentarz może być wieloliniowy */
.prawo {
    text-align: right;
}

h1 {
    color: blue;
    text-align: center;
}

p.wiekszeZnaki {
    font-size: 17pt;
}

//plik HTML

<!DOCTYPE html>
<html lang="pl">
<head>
    <link rel="stylesheet" href="styl.css" type="text/css"/>
    <meta charset="UTF-8"/>
    <title>Pierwsza strona w HTML5</title>
</head>
<body>
<h1>Każdy nagłówek h1 będzie zmieniony!</h1>
<h4>Ten już nie</h4>
<p id="pierwszy">Tekst w tym paragrafie zostanie odpowiednio sformatowany</p>
<p>Ten paragraf nie zostanie zmieniony!</p>
<p class="wiekszeZnaki">Paragraf dostanie większe znaki</p>
<span class="wiekszeZnaki">Ponieważ element nie jest paragrafem to klasa nie zostanie dla niego
zastosowana!</span>
<h4 class="prawo">Ta klasa jest ogólna; zostanie zastosowana</h4>

```

```
<p class="prawo">Tutaj również</p>
</body>
</html>
```

## 2. Więcej o właściwościach, identyfikatorach, selektorach oraz klasach.

Selektorem jest w zasadzie konkretny element HTML (znacznik, np. div, span, p, h1), którego właściwości chcemy zmienić. Specjalną wersją selektora są identyfikatory oraz klasy, o których więcej można dowiedzieć się z podpunktów b) oraz c).

a) właściwość – jest to pewien atrybut (parametr) elementu na stronie WWW (czyli znacznika). Dzięki właściwościom mamy możliwość zmiany pewnych cech pisma, obramowania, tła czy koloru elementu. W najnowszej, 3 odsłonie CSS, można ponadto w sposób dynamiczny modyfikować rozmiary poszczególnych elementów (przykładowo powiększyć tekst po najechaniu na niego), ich ułożenie (uciekający element), czy też tworzyć transformacje (obrót na wszystkich 3 płaszczyznach – X, Y oraz Z). W następnym punkcie instrukcji (3) zostaną przedstawione wybrane właściwości wraz z przykładem ich użycia.

b) identyfikator – style CSS można przypisywać do konkretnego identyfikatora (atrybut znacznika „id”). Jako wartość przyjmuje dowolny ciąg znaków. W danym dokumencie HTML identyfikator elementu powinien być UNIKATOWY. Oznaczałoby to, że w ten sposób przypisywane style są mało użyteczne i równie dobrze do wskazanego elementu można bezpośrednio użyć atrybutu style (przykład a) z 1 punktu). Trzeba jednak pamiętać, że plik ze stylami możemy wykorzystywać na wielu stronach HTML – zarówno należących do naszego aktualnego projektu, jak i do przyszłych (bądź równoległych). Wtedy zamiast ustawiać każdemu elementowi o wskazanym id te same wartości wystarczy nadać je odpowiedniemu id (prostsze i wygodniejsze rozwiązanie). W pliku CSS nazwę identyfikatora poprzedza się znakiem tzw. krzaka (hash), `#`.

Przykład:

```
//plik styl.css
```

```
#tytul {
    background-color: yellow;
    color: red;
    border-style: solid;
    border-width: 2px;
    border-color: blue;
}
```

```
//plik index.html
```

```
<!DOCTYPE html>
<html lang="pl">
<head>
    <link rel="stylesheet" href="styl.css" type="text/css"/>
    <meta charset="UTF-8"/>
    <title>Pierwsza strona w HTML5</title>
</head>
<body>
<div id="tytul">
    <p>Paragraf tytułu</p>
    <p>Drugi paragraf tytułu</p>
```

```

        <span>Element blokowy tekstu</span>
</div>
<!-- DALSZĄ CZĘŚĆ STRONY -->
</body>
</html>

```

```

//plik omnie.html
<!DOCTYPE html>
<html lang="pl">
<head>
    <link rel="stylesheet" href="styl.css" type="text/css"/>
    <meta charset="UTF-8"/>
    <title>Pierwsza strona w HTML5</title>
</head>
<body>
<div id="tytul">
    <p>Paragraf tytułu</p>
    <p>Drugi paragraf tytułu</p>
    <span>Element blokowy tekstu</span>
</div>
<!-- DALSZĄ CZĘŚĆ STRONY -->
</body>
</html>

```

### c) klasa

W przeciwieństwie do identyfikatora, klasa może oddziaływać na całą grupę elementów strony. Klasy mogą być zarówno ogólnego przeznaczenia (daną klasę można przypisać do dowolnego elementu na stronie WWW) jak i specjalnie dostosowane tylko dla wybranych elementów (przykładowo tylko dla nagłówków h3).

#### 1) klasa ogólnego przeznaczenia

klasę ogólnego przeznaczenia w pliku CSS rozpoczyna się od kropki (w przeciwieństwie do identyfikatora)

```

//plik css
.prawo {
    text-align: right;
}

```

```

//plik index.html

```

```

<!DOCTYPE html>
<html lang="pl">
<head>
    <link rel="stylesheet" href="styl.css" type="text/css"/>
    <meta charset="UTF-8"/>
    <title>Pierwsza strona w HTML5</title>
</head>
<body>
<p class="prawo">Ten element będzie równany do prawej. Element poniżej też</p> <!--p1-->

```

```

<div>
  <div class="prawo">
    
  </div>
</div>
<p>jednak ten już nie</p>
<!-- DALSZĄ CZĘŚĆ STRONY -->
</body>
</html>

```

Jak widać zarówno paragraf p1 jak i obraz będą wyrównane do prawej. Dla obrazu zastosowany został pewien trik – gdybyśmy dla samego znacznika img ustawili klasę „prawo” nie uzyskalibyśmy odpowiedniego efektu gdyż obraz nie jest tekstem. Jednak w elemencie blokowym HTML nie wie z czym ma do czynienia – obraz traktowany jest jako element potencjalnie tekstowy. Identyfikacja uzyskalibyśmy zapisując powyższy kod w taki sposób:

```

<p class="prawo">
  
</p>

```

Ostatni paragraf p2 nie zawiera odnośnika do klasy, a przez to nie będzie wyrównany do prawej strony.

## 2) klasa dla konkretnego elementu

Podobnie do klasy ogólnego przeznaczenia, nazwę tej klasy rozpoczyna się od kropki. Jednak w jej przypadku przed kropką trzeba podać nazwę konkretnego znacznika, którego ma ona dotyczyć. Przykład:

```
//plik styl.css
```

```
p.wiekszeZnaki {
  font-size: 17pt;
}
```

```
//plik index.html
```

```

<!DOCTYPE html>
<html lang="pl">
<head>
  <link rel="stylesheet" href="styl.css" type="text/css"/>
  <meta charset="UTF-8"/>
  <title>Pierwsza strona w HTML5</title>
</head>
<body>
<p class="wiekszeZnaki">Paragraf dostanie większe znaki</p>
<span class="wiekszeZnaki">Ponieważ element nie jest paragrafem to klasa nie zostanie dla niego
zastosowana!</span>
<p>Paragraf nie ma wywołanej klasy CSS więc czcionka nie zostanie powiększona</p>
</body>
</html>

```

### 3. Wybrane właściwości CSS

#### a) właściwości tła

Każdy z elementów posiada swoje tło. Za pomocą CSS możemy ustawić jego właściwości:

1) background-color – ustawia wskazany kolor jako tło dla danego elementu; kolor można podawać za pomocą nazw bądź np. kodu szesnastkowego

```
<section style="background-color: red;">  
<p>Paragraf wewnątrz section</p>  
<p>Kolejny; nadal ma kolor tła</p>  
<h1>Także nagłówek będzie miał tło</h1>  
</section>
```

2) background-image – ustawia jako tło wskazany obraz

```
<section style="background-image: url('http://jaguscik.pl/images/css_logo.png');">  
<p>Paragraf wewnątrz section</p>  
<p>Kolejny; nadal ma kolor tła</p>  
<h1>Także nagłówek będzie miał tło</h1>  
</section>
```

Dobrym zwyczajem jest ustawiać tło z obrazem wraz z kolorem tła – niektóre przeglądarki mogą mieć zablokowane tło graficzne.

3) background-repeat – jeżeli obraz jest mniejszy od wielkości elementu, dla którego został wybrany na tło, właściwość decyduje w jaki sposób ma się zachować:

- repeat – powtarza obraz w poziomie i pionie; domyślnie zachowanie
- no-repeat – nie powtarza obrazu
- repeat-x – powiela obraz tylko w poziomie (szerokość)
- repeat-y – powiela obraz tylko w pionie (wysokość)

```
<section style="background-image: url('http://jaguscik.pl/images/css_logo.png'); background-repeat: no-repeat;">
```

```
<p>Paragraf wewnątrz section</p>  
<p>Kolejny; nadal ma kolor tła</p>  
<h1>Także nagłówek będzie miał tło</h1>  
</section>
```

4) background-origin – określa, od której części ma być rysowane tło

- padding-box – rysowanie tła rozpoczyna się od zakończenia RYSOWANEGO obramowania elementu (zależny od ustawionej wartości border)
- border-box – rysowanie tła rozpoczyna się od sztywnej granicy elementu (od momentu, gdy się rozpoczyna element)
- content-box – rysowanie tła rozpoczyna się od początku zawartości elementu

5) background-size – określa wielkość obrazu ustawionego jako tło danego elementu. Możliwe opcje:

- 100px 50px – ustawia wielkość obrazu – w podanym przykładzie 100 pikseli szerokości i 50 pikseli wysokości; tylko taki obszar zostanie nim wypełniony (jeżeli obraz jest większy/mniejszy to zostanie przeskalowany). Jeżeli podamy tylko jedną wartość, druga zostanie ustawiona na 'auto'
- 50% 50% - ustawia obszar pokrycia obrazem procentowo – w podanym przykładzie będzie to 50% wielkości elementu (szerokości i wysokości); jeżeli obraz będzie większy/mniejszy to zostanie przeskalowany. Jeżeli podamy tylko jedną wartość, druga zostanie ustawiona na 'auto'
- cover – pokrywa cały dostępny obszar elementu; niekiedy obraz nie wyświetli się jeżeli zostanie dodatkowo ustawiona właściwość background-position
- contain – pokrywa cały obszar z zawartością danego elementu

6) background-position – ustawia początek, od którego zaczyna być rysowany obrazek tła (domyślna wartość 0px 0px)

- left top, left center, left bottom, right top, right center, right bottom, center top, center center,

center bottom – możliwe pary określające start wypełniania obrazem tła; jeżeli wpisujemy tylko jedno słowo, drugie automatycznie będzie potraktowane jako center

- xpos ypos – ustawia punkt startowy rysowania obrazu tła; jeżeli jeden z parametrów nie zostanie ustawiony, drugi automatycznie otrzyma wartość 50%; jako pos ustawiamy jednostkę miary (wy tłumaczone w dalszej części instrukcji)

7) background-attachment – określa jak ma zachowywać się obraz tła przy przewijaniu treści strony

- scroll – obraz przewija się wraz z resztą strony; wartość domyślna

- fixed – obraz wyświetlany jest bez przewijania (respektując punkt widoku)

- local – obraz przewijany jest wraz z zawartością elementu, dla którego jest ustawiony

8) background-clip – ustawia ucięcie tła względem wybranej opcji:

- padding-box – tło wyświetla się na całym elemencie, jednak dopiero od zakończenia jego obramowania

- border-box – tło wyświetla się na całym elemencie uwzględniając sztywną granicę elementu, dla którego zostało ustawione (domyślnie)

- content-box – tło dopasowuje się do zawartej treści wewnątrz elementu, którego dotyczy

9) background – ogólna właściwość tła; dzięki niej możemy ustawić w jednym miejscu wszystkie cechy, jakie ma ono posiadać wedle schematu

background: color position size repeat origin clip attachment image;

PRZYKŁAD:

```
<section style="background:black url('http://jaguscik.pl/images/css_logo.png') no-repeat center;">
```

```
<p>Paragraf wewnątrz section</p>
```

```
<p>Kolejny; nadal ma kolor tła</p>
```

```
<h1>Także nagłówek będzie miał tło</h1>
```

```
</section>
```

b) właściwości tekstu

Przedstawione poniżej właściwości oddziałują tylko na tekst umieszczony na stronie/w elemencie

1)color – ustawia wybrany kolor dla czcionki

```
<p style="color: blue;">Tekst przykład</p>
```

Proszę pamiętać, że wedle standardu CSS, jeżeli ustawiamy właściwość color to powinniśmy także ustawić background-color.

2) text-align – ustawia poziome wyrównanie tekstu względem wielkości (szerokości) zawierającego go elementu; możliwe wartości: left (do lewej), right (do prawej), center (wyśrodkowany), justify (justowany, rozłożony równomiernie)

```
<p style="text-align: center;">Tekst przykład</p>
```

3) text-decoration – ustawia dekorację tekstu zawartego w elemencie; możliwe wartości: underline (podkreślenie dolne), overline (linia nad tekstem), line-through (przekreślony tekst), none (żadna dekoracja nie zostanie ustawiona)

```
<p style="text-decoration: line-through;">Tekst przykład</p>
```

4) text-transform – dzięki tej właściwości możemy mieć kontrolę nad wielkością liter w tekście; możliwe wartości: none (tekst nie zostanie zmieniony), capitalize (pierwsze litery w tekście będą pisane wielką literą), uppercase (cały tekst zostanie napisany dużymi literami), lowercase (cały tekst będzie napisany małymi literami)

```
<p style="text-transform: uppercase;">Tekst przykład</p>
```

5) text-indent – określa wcięcie tekstu przy rozpoczęciu wypisywania go (określane w odpowiednich jednostkach miary)

```
<p style="text-indent:2cm;">Tekst przykład Tekst przykład Tekst przykład Tekst przykład Tekst  
przykład Tekst przykład Tekst przykład Tekst przykład Tekst przykład Tekst przykład Tekst  
przykład Tekst przykład Tekst przykład Tekst przykład Tekst przykład Tekst przykład Tekst
```

przykład Tekst przykład Tekst przykład </p>

6) text-overflow – określa jak przeglądarka ma potraktować ciąg znakowy, który przekroczy ramy elementu, w który został wpisany; możliwe wartości: clip (tekst zostanie na sztywno ucięty), ellipsis (jeżeli tekst jest za długi to zostaje ucięty, a na jego końcu zostają dodane trzy kropki - '...', symbolizujące iż tekst nie został zakończony) lub dowolna wartość tekstowa która ma zostać wyświetlona przed ucięciem tekstu (nieobsługiwane przez większość przeglądarek).

```
<div style="white-space:nowrap;width:150px;border:1px solid black;text-overflow:ellipsis;">Ten tekst nie zostanie w całości wyświetlony ponieważ jego ogólna długość przekracza wartość szerokość</div>
```

Dodatkowe właściwości, jak white-space, width oraz border ustawione zostały po to, by zmusić element do nie zawijania wierszy, wyświetlenia jego obramowania oraz ustawienia na sztywno jego szerokości – aby zobaczyć efekt ucinania tekstu. Niestety, pomimo ustawienia obcięcia tekst nadal wyświetli się cały, wyłamując się poza ramy rodzica. Żeby tekst był poprawnie ucinany trzeba dodać jeszcze jedną właściwość:

- overflow – definiuje zachowanie się zawartości elementu, gdy ta przekroczy jego granice; możliwe wartości: visible (zawartość nie zostanie obcięta; domyślna akcja), hidden (część zawartości, która znajdzie się poza granicą elementu nadrzędnego zostanie obcięta i nie będzie pokazywana), scroll (dodaje suwaki przy granicach elementu w celu przewinięcia nadmiarowej zawartości), auto (jeżeli treść nie wychodzi poza ramy elementu suwaki są niewidoczne; jeżeli wyjdzie poza granicę to suwaki się pojawiają).

Teraz, znając dodatkową właściwość należy zmodyfikować poprzedni przykład :

```
<div style="white-space:nowrap;width:150px;border:1px solid black;overflow:hidden;text-overflow:ellipsis;">Ten tekst nie zostanie w całości wyświetlony ponieważ jego ogólna długość przekracza wartość szerokość</div>
```

Tekst powinien zostać przycięty, a przed końcem elementu (warstwy) powinny wyświetlić się trzy kropki.

7) text-shadow – dodaje efekt cienia do ciągu znakowego. Składnia polecenia:

```
text-shadow: h-shadow v-shadow blur color;
```

gdzie:

h-shadow - oznacza przesunięcie cienia względem oryginalnego punktu długości (oś x); określany w pikselach (px)

v-shadow - oznacza przesunięcie cienia względem oryginalnej wysokości (oś y); określany w pikselach (px)

blur – oznacza tzw. rozmycie o określoną ilość pikseli (im więcej tym bardziej gładki będzie wydawać się cień)

color – parametr opcjonalny; określa kolor cienia

```
<p style="text-shadow:4px 4px;">Cień tekstu</p>
```

c) właściwości czcionki:

1) font-family – poprzez nazwę określa czcionkę, która ma zostać użyta w wybranym elemencie dla tekstu; aby przygotować się na ewentualność, że klient (użytkownik docelowy) nie posiada pliku danej czcionki, można po przecinku podać kroje alternatywne np.:

```
p<style="font-family:'Times New Roman', Times, serif;">Ten paragraf zostanie napisany czcionką Times New Roman; jeżeli nie ma jej zainstalowanej w systemie zostanie wybrana inna, alternatywna</p>
```

2) font-style – określa styl, w jakim tekst zostanie napisany; możliwe wartości: normal (tekst

zachowa swoje proporcje; wartość domyślna), italic/oblique (tekst zostanie napisany kursywą/ukośnie)

3) font-size – umożliwia zmianę wielkości czcionki dla danego elementu HTML.

4) font-variant – umożliwia napisanie całego tekstu dużymi literami; oryginalne duże litery (np. rozpoczynające zdanie) zostaną zamienione na nieco większe od reszty litery (dla wartości small-caps; dla wartości normal tekst wyświetli się bez żadnej zmiany)

5) font-weight – umożliwia zmianę szerokości czcionki. Dzięki temu parametrowi możemy ustawić np. wartość grubą (bold) lub cieńszą niż normalnie (lighter); można też ustawić jedną z wartości numerycznych (z przedziału 100-900).

#### 4. Kolory HTML

Kolory w CSS można podawać na trzy różne sposoby.

1) poprzez nazwę – CSS posiada kilkadziesiąt zdefiniowanych wartości kolorów. Wszystkie te wartości zapisane są pod określonymi nazwami, które można podawać przy określaniu kolorów poszczególnych elementów strony, np. red, green, blue, aqua, black, white, brown, cyan, magenta, firebrick, gold i inne (pełną listę kolorów można znaleźć tutaj - [http://www.w3schools.com/cssref/css\\_colornames.asp](http://www.w3schools.com/cssref/css_colornames.asp) ).

2) poprzez funkcję mieszającą rgb() - wszystkie kolory jakie możemy ujrzeć na ekranie stanowią mieszaninę trzech podstawowych kolorów – R (red; czerwony) G (green; zielony) B (blue; niebieski); chociaż niektóre z systemów mieszania kolorów wydają się być odpowiedniejsze (istnieje kilka innych kombinacji barw) to właśnie RGB jest najpopularniejszą paletą kolorów przy tworzeniu stron WWW bądź grafiki na nią. Funkcja mieszania kolorów ma postać:

rgb(red,green,blue)

gdzie za nazwy kolorów można podstawić wartość 0-255; wartość 0 oznacza, że dany kolor jest wygaszony (nie bierze udziału w tworzeniu barwy), 255 oznacza natomiast jego maksymalne nasycenie (jego najsilniejszy odcień). Za pomocą odpowiednich kombinacji można uzyskać pożądane efekty barw.

3) poprzez podanie wartości heksadecymalnej danego koloru – najpopularniejsze podawanie wartości barw wśród projektantów stron WWW. Wartość barwy w tym zapisie może wyglądać np. tak:

#1AFF00

jakakolwiek wartość barwy w tym zapisie zawsze musi rozpoczynać się od znaku #. Kolejne znaki można interpretować w następujący sposób:

1A – wartość koloru czerwonego (R)

FF – wartość koloru zielonego (G)

00 – wartość koloru niebieskiego (B)

Zapis ten jest krótszą formą zapisu RGB – zajmuje mniej miejsca, a efekt jest taki sam. Oczywiście różni się zapisem wartości minimalnych (00) i maksymalnych (FF). Niezmienny pozostaje zapis barwy na 8 bitach (jeden bajt; stąd maksymalna wartość 255).

## 5. Jednostki miar w CSS

CSS obsługuje następujące jednostki, w których można określać rozmiary elementów (i inne właściwości wymagające jednostek):

% - wartość procentowa; zmienia się wraz z dostępną przestrzenią roboczą strony (miara relatywna; użyteczna gdy chcemy aby strona adoptowała się pod różne rozdzielczości w różnych urządzeniach)

in – wartość podawana w calach (absolutna)

cm – wartość podawana w centymetrach (absolutna)

mm – wartość podawana w milimetrach (absolutna)

em – wartość w tej mierze zależy od aktualnej wielkości użytej czcionki. Przykładowo 1 em będzie dokładnie równa wielkości (wysokości) czcionki, 2em będzie dwa razy większa niż czcionka (dla czcionki 12pt wartość wyniesie 24pt) , dla 3em będzie to wielkość czcionki x 3 itd.; wartość relatywna

ex – wartość 1ex jest wartością x-wielkości danej czcionki (przeważnie wartość x-wielkość stanowi połowę rozmiaru czcionki); wartość relatywna

pt – wartość podawana w punktach (jeden punkt odpowiada 1/72 cala); wartość absolutna

pc – wartość podawana w pikach (jeden pika odpowiada 12 punktom); wartość absolutna

px – wartość podawana w pikselach (wielkość punktu ekranu; wartość pośrednia ze względu na fakt, że każdy ekran może mieć różną wielkość punktu)

Zadania do samodzielnego wykonania:

1. Utworzone poprzedni tabele należy odpowiednio sformatować:

-tabela ma wyglądać jak w standardowym programie tekstowym – posiadać obrys każdej komórki oraz całej tabeli; dobrze byłoby, aby obrys tabeli powinien mieć inny kolor, inny nagłówek, a jeszcze innych dzielący poszczególne komórki.

- wszystkie pola nagłówek tabeli powinny mieć kolor tła; proszę jednak ustawić inny kolor tła dla całej kolumny Id

- nazwy samochodów mogą być pogrubione

- opisy powinny być pisane trochę mniejszą czcionką, kursywą. Jej kolor proszę ustawić na jeden z odcieni szarości

2. Mając utworzoną listę oferty proszę odnaleźć style pozwalające na jej modyfikację:

- proszę zamienić punkty listy nienumerowanej na graficzne (najlepiej grafika przedstawiająca koło); proszę tak utworzyć styl by działał na przeglądarkach: Firefox, Chrome oraz Internet Explorer

- proszę zmienić ustawienie punktu listy nienumerowanej na wewnętrzne (w efekcie elementy listy powinny być bardziej wcięte niż przed zastosowaniem właściwości)

- proszę zmienić (za pomocą stylu) numerację listy numerowanej na rzymską dużą.

3. Proszę utworzyć plik ze stylami. W pliku należy zdefiniować wygląd każdego elementu strony (tytułu, menu, treści właściwej oraz stopki) – koloru tła, tekstu oraz wielkości tekstu. Proszę znaleźć właściwość i za jej pomocą zmienić domyślną czcionkę w poszczególnych elementach. Tytuł strony proszę przyozdobić cieniem liter.

4. Stworzyć tło z obrazu w taki sposób, by nawet przy przewijaniu treści było stale ustawione w tym samym miejscu (opcja fixed). Proszę znaleźć w jaki sposób ustawić częściową przezroczystość pozostałych elementów strony – tak, aby grafika była widoczna nawet w przypadku ustawionego tła dla któregośkolwiek z nich.

5. Odsyłacze stron także posiadają swoje właściwości. Proszę zdefiniować je dla każdego z ich stanów (inny wygląd odsyłacza nie otwieranego, inne dla otwartego, a jeszcze inne dla aktualnie wybranego). Proszę dopasować ostateczny efekt do swoich preferencji. Można pokusić się o modyfikację paska nawigacyjnego w taki sposób by nie wyglądał tradycyjnie – można zmienić

wygląd samych odsyłaczy, ich czcionkę, kolor, udekorowanie; Ponadto proszę każdy z nich wstawić w osobne pola, które będą posiadały odpowiednie właściwości swoich ramek (granic) oraz tła. Ostateczny efekt powinien przypominać przycisk graficzny, chociaż nim nie będzie (będzie natomiast stworzony z kilku elementów HTML).

6. Proszę spróbować zmienić wygląd pól formularza ze strony kontaktowej – zastosować półprzezroczystość, zmienić krój czcionki wewnątrz elementu, zmienić domyślne kolory oraz obramowanie.

Zadania dodatkowe:

1. Proszę znaleźć sposób (inny niż znacznik pre) na wyświetlenie tekstu z zachowaniem białych spacji oraz linii w przeglądarce internetowej (można tego dokonać za pomocą stylów).
2. Proszę znaleźć co oznacza dyrektywa !important

## **MATERIAŁ DODATKOWY!!!!!!**

Operacje na obrazach w CSS.

Arkusze CSS umożliwiają zmianę wszystkich parametrów wyświetlanych obrazów ze znaczników `<img>`. Chociaż HTML pozwala na ustawienie ich podstawowych właściwości poprzez atrybuty (jak szerokość czy wysokość) to jednak efektywniej jest ustawiać je za pomocą stylów.

Przeważnie obrazy „opakowuje się” dodatkowo w element zbiorczy, przykładowo w znacznik `<div></div>` (lub wedle specyfikacji HTML5 w `<figure></figure>`). Dzięki temu można je wydajniej rozmieszczać i mieć pewność, że np. nie wprowadzą zamętu na stronie (nie zastawią innej zawartości strony).

Każdemu obrazowi można zmieniać parametr przezroczystości. Pozwala to uzyskać na stronie bardzo ciekawe efekty. Jeżeli chcemy, aby miniatury obrazów w galerii „podświetlały się” przy wybraniu ich kursorem myszy wystarczy:

```
<!DOCTYPE html>
<html>
<head>
<style>
div.img
{
margin: 5px;
padding: 5px;
border: 1px solid #0000ff;
height: auto;
width: auto;
float: left;
text-align: center;
}
div.img img
{
display: inline;
margin: 5px;
border: 1px solid #ffffff;
}
div.img a:hover img {border: 1px solid #0000ff;}
img:hover {
opacity: 0.4;
}
```

```

    filter:alpha(40);
}
div.desc
{
    text-align: center;
    font-weight: normal;
    width: 120px;
    margin: 5px;
}
</style>
</head>
<body>

<div class="img">
  <a target="_blank" href="http://www.w3schools.com/css/klematis_big.jpg"></a>
  <div class="desc">Opis obrazu</div>
</div>
<div class="img">
  <a target="_blank" href="http://www.w3schools.com/css/klematis2_big.htm"></a>
  <div class="desc">Opis obrazu</div>
</div>
<div class="img">
  <a target="_blank" href="http://www.w3schools.com/css/klematis3_big.htm"></a>
  <div class="desc">Opis obrazu</div>
</div>
<div class="img">
  <a target="_blank" href="http://www.w3schools.com/css/klematis4_big.htm"></a>
  <div class="desc">Opis obrazu</div>
</div>

</body>
</html>

```

Powyższy kod stworzy na stronie minigalerię z 4 obrazkami. Po najechaniu na dowolny z nich zostanie ustanowiona dla niego ramka, a sam obrazek stanie się częściowo przezroczysty mieszając swoje barwy z barwami obiektów będących pod obrazem (w tym wypadku kolorem białym) dzięki czemu obraz „podświetla się”. (przykład inspirowany przykładem ze strony [www.w3schools.com](http://www.w3schools.com)).

Małe pliki graficzne przeważnie powinny pobierać się szybko nawet na słabym/wolnym łączu internetowym (np. dane pakietowe GPRS). Jednak problemem może być spora ilość zapytań do serwera (każdy plik musi zostać „zażądany” od klienta, być wysłany i potwierdzony) co przy sporym ruchu na stronie może doprowadzić do niepożądanych opóźnień w ostatecznym wyświetleniu strony.

CSS umożliwia wykorzystywanie tzw. sprite'ów obrazów. Czym są sprites? To po prostu kolekcja

wielu obrazów (np. kilku bądź kilkudziesięciu) w jednym, pojedynczym pliku.

Jako przykład niech posłuży chęć utworzenia własnego paska nawigacyjnego strony; aby przyciski nie wyglądały jak zwykłe kotwice (znaczniki <a>) można użyć następujących ustawień:

```
<!DOCTYPE html>
<html>
<head>
<style>
a.home
{
width:46px;
height:44px;
float: left;
display: block;
margin-right: 5px;
background:url(http://www.w3schools.com/css/img_navsprites.gif) 0 0;
}
a.next
{
width:43px;
float: left;
display: block;
height:44px;
background:url(http://www.w3schools.com/css/img_navsprites.gif) -91px 0;
}
</style>
</head>

<body>
<a class="home" href="http://www.onet.pl"></a>
<a class="next" href="http://www.wp.pl"></a>
</body>
</html>
```

Dzięki powyższemu zabiegowi wyświetlą się dwa przyciski (obok siebie). Jeden z nich (domek) prowadzi na portal onet.pl natomiast strzałka na stronę wp.pl. Oryginalny obraz użyty w przykładzie znajduje się pod adresem [http://www.w3schools.com/css/img\\_navsprites.gif](http://www.w3schools.com/css/img_navsprites.gif).

Przy pierwszym odnośniku jego tło pobierane jest z załadowanego obrazu. Ponieważ <a> ma ograniczoną wielkość pobiera się tylko część obrazu (dokładnie dom). W drugim wypadku musimy wskazać która część obrazu ma zostać pobrana. Uczynione to zostało w linii:

```
background:url(http://www.w3schools.com/css/img_navsprites.gif) -91px 0;
```

gdzie -91px określa przesunięcie od lewego brzegu (left), od którego ma być czytany obraz. Drugi parametr (0) byłby brany pod uwagę gdybyśmy chcieli przesunąć czytanie obrazu od górnego brzegu (top).

## 2. Animacje, przejścia i transformacje elementów HTML w CSS3.

Wraz z pojawieniem się 3 wersji CSS przy pomocy stylów możliwe stało się tworzenie efektownych animacji elementów na stronie WWW. Można przykładowo tworzyć efekt obracającego elementu w przestrzeni z, x bądź y, powiększać i pomniejszać czy też rozciągać

go/kurczyć.

#### a) przejście (transition)

Dzięki przejściom dowolny element strony internetowej można płynnie zmienić wartość danej właściwości z jednej na drugą - przykładowo przy najechaniu na niego myszą rozciągnie się bądź skurczy. Dla tej operacji dostępne są następujące cechy (właściwości):

- transition\_property – określa dla której/których właściwości (np. szerokość, wysokość, przezroczystość) ma zostać zastosowany efekt przejścia. Możliwe wartości: none (brak), all (dla wszystkich możliwych właściwości danego elementu), property (wskazana właściwość; można wymienić kilka właściwości oddzielając je przecinkami).
- transition-duration – określa jak długi ma być efekt przejścia pomiędzy stanami elementu. Czas podawany jest w odpowiednich miarach (przeważnie sekundy)
- transition-timing-function – określa krzywą czasu wedle której następuje efekt przejścia. Istnieje 6 sposobów definicji tego parametru:

1) linear – efekt przejścia będzie jednostajny (przez cały okres trwania identyczny sposób przerysowania)

2) ease – domyślne ustawienie; początkowo wolny start, później szybsza animacja i na końcu spowolniona

3) ease-in – tylko wolny start (w dalszej części animacja jednostajnie przyspieszona)

4) ease-out – tylko wolny koniec

5) ease-in-out – wolny start i koniec

6) cubic-bezier (n,n,n,n) – własne ustawienie prędkości animacji (każdy z 4 parametrów może przyjąć wartości pomiędzy 0.0 a 1.0)

- transition-delay – określa opóźnienie rozpoczęcia efektu przejścia (w sekundach)

- transition – właściwość zbiorcza dzięki której, po spacji, możemy określić jej wszystkie wartości.

Definicja posiada następującą postać:

transition cecha czas\_trwania funkcja\_przejscia opoznienie

Przykład:

Proszę w przykładzie do galerii podmienić style na poniższe:

```
<style>
div.img
{
  margin: 5px;
  padding: 5px;
  border: 1px solid #0000ff;
  height: auto;
  width: auto;
  float: left;
  text-align: center;
}
div.img img
{
  display: inline;
  margin: 5px;
  border: 1px solid #ffffff;
}
img {
  transition: opacity 2s ease-in 2s;
```

```

-webkit-transition: opacity 2s ease-in 2s;
}
div.img a:hover img {border: 1px solid #0000ff;}
img:hover {
  opacity: 0.4;
  filter:alpha(40);
}
div.desc
{
  text-align: center;
  font-weight: normal;
  width: 120px;
  margin: 5px;
}
</style>

```

Od tego momentu będzie widoczny ładny efekt stopniowego rozjaśniania obrazu.

**WAŻNE!** Dla przeglądarek opartych o biblioteki webkit (m. in. Chrome oraz Safari) należy osobno zdefiniować każdą z wymienionych właściwości z przedrostkiem -webkit- (tak jak w przykładzie). W przeciwnym wypadku efekt na tych przeglądarkach może nie zadziałać.

## b) transformacja 2D/3D

Pozwala na zmienianie właściwości wyświetlania wskazanego obiektu strony WWW.

- transform – właściwość określa jaka modyfikacja ma zostać wykonana na wskazanym elemencie/elementach. Jako wartość przyjmuje jedną z podanych funkcji:

- 1) none – żaden z efektów nie zostanie zastosowany
- 2) translate(x,y), translate3d(x,y,z) – przenosi wskazany obiekt względem oryginalnego umiejscowienia o podane wartości xy (transformacja 2D) lub dodatkowo o parametr osi z (transformacja 3D)
- 3) translateX(x), translateY(y), translateZ(z) – podobnie jak w punkcie 2) jednak z tą różnicą, że można podać wartość tylko dla jednej, konkretnej osi
- 4) scale(x,y), scale3d(x,y,z) – wskazany element jest skalowany (w zależności od podanych parametrów pomniejszany/powiększany względem oryginalnej wielkości)
- 5) scaleX(x), scaleY(y), scaleZ(z) – jak w 4) z tym, że można podać wartość tylko dla jednej, konkretnej osi
- 6) rotate(deg), rotate3d(x,y,z,deg) – obraca element o wskazaną wartość kąta deg. W przypadku obrotu 3D wskazuje się względem której osi ma nastąpić obrót (0 – dana oś nie bierze udziału w obrocie, 1 – względem niej ma nastąpić obrót).
- 7) rotateX(deg), rotateY(deg), rotateZ(deg) – obraca obiekt o wskazany kąt tylko względem wybranej osi
- 8) skew(deg-x, deg-y) – pochyla obiekt o wskazaną wartość kąta dla osi x i y
- 9) skewX(deg), skewY(deg) – pochyla obiekt tylko na wskazanej osi
- 10) perspective(n) – określa perspektywę widoku na przetransformowany element.
- 11) matrix(n,n,n,n,n) – za pomocą tej funkcji można ustalić parametry transformacji obiektu na płaszczyźnie 2D (kolejno: obrót, skalowanie, przeniesienie oraz pochylenie)
- 12) matrix3d(n,n,n,n,n,n,n,n,n,n,n,n,n) – podobnie jak poprzednio za pomocą tej funkcji można ustalić parametry wszystkich możliwych transformacji na elemencie po płaszczyźnie 3D (dochozą wartości z)

- transform-origin – właściwość decyduje z jakiego punktu obiektu mają nastąpić przekształcenia (domyślnie jest to jego środkowy punkt). Składnia:

transform-origin: osX, osY, osZ

- 1) osX – może przyjąć wartości: left (od lewej strony), center (domyślna wartość), right (od prawej strony) lub konkretną wartość liczbową (w miarach CSS)
- 2) osY – może przyjąć wartości: top (od góry), center, bottom (od dołu) lub konkretną wartość liczbową (w miarach CSS)
- 3) osZ – można podawać jedynie wartości liczbowe

WAŻNE:

- dla przeglądarki Internet Explorer 9 transformacje zadziałają jedynie wtedy gdy ich deklaracja pozostanie poprzedzona przedrostkiem -ms-
- dla przeglądarek opartych o webkit należy dodać przedrostek -webkit-
- niektóre wersje Opery nie obsługują transformacji 3D

PRZYKŁAD:

W poprzednim przykładzie proszę zamienić następujące linie kodu na poniższe:

```
img {
transition: opacity 2s ease-in 2s, transform 1s;
-webkit-transition: opacity 2s ease-in 2s, transform 1s;
}
div.img a:hover img {border: 1px solid #0000ff;}
img:hover {
opacity: 0.4;
filter:alpha(40);
transform: scale(1.5, 1.5);
-ms-transform: scale(1.5, 1.5);
-webkit-transform: scale(1.5, 1.5);
}
```

c) animacje

Do niedawna wszelkie animacje na stronach internetowych tworzone były przeważnie w technologii Adobe Flash. Jednak standard HTML5, wraz ze stylami CSS3 umożliwia tworzenie animacji w prostszy, a przede wszystkim darmowy, sposób. Tworzenie animacji przy pomocy arkusza CSS należy rozpocząć od wybrania elementu na stronie, który ma zostać animowany. W stylach dla tego elementu należy określić wszystkie opcje rodzaju animacji oraz czasu trwania. Właściwości animacji:

- animation-name – określa nazwę animacji (sekwencji) która ma zostać wykonana na obiekcie
- animation-duration – określa czas trwania ustawionej animacji (w sekundach); jeżeli nie zostanie ustawiona lub ustawiona zostanie wartość 0 (domyślna) to animacja nie zostanie wykonana (zostanie wyświetlony tylko wynik)
- animation-timing-function – określa krzywą czasu wedle której następuje efekt przejścia. Możliwe opcje zostały opisane w podpunkcie a)
- animation-delay – określa czas opóźnienia, po którym animacja się rozpocznie (w sekundach)
- animation-iteration-count – określa ile razy animacja ma zostać powtórzona (domyślnie 1 – tylko jeden raz); animację można zapętlić w nieskończoność poprzez podanie wartości 'infinite'

- animation-direction – określa kierunek animowania obiektu. Możliwe wartości:

- 1) normal – animacja odtwarza się w sekwencji zaplanowanej przez projektanta
- 2) reverse – animacja odtwarza się w sekwencji odwrotnej do zaplanowanej przez projektanta
- 3) alternate – kolejne części animacji wyświetlają się jak przy wartości normal w każdej sekundzie niepodzielnej przez dwa (np. 1, 3, 5, 7 itd.), a w każdej sekundzie podzielnej przez dwa jako reverse
- 4) alternate-reverse – zachowanie odwrotne do opisanego w podpunkcie 3)

- animation-play-state – określa czy animacja ma być odtwarzana (running – domyślna wartość) czy zostać zatrzymana (paused)

-animation – poprzez tą właściwość można określić kolejno wszystkie wartości tworzonej animacji poza opcją animation-play-state.

Powyżej opisane własności określają wszystkie cechy animacji obiektów. Do animacji elementu należy stworzyć tzw. ramkę animacji w której określa się właściwości elementu do zmieniony.

Przykładowa ramka:

```
@keyframes ramka {  
  from {width: 50px; height: 10px;}  
  to {width: 200px; height: 100px;}  
}
```

Użycie tak stworzonej ramki spowoduje poszerzenie się obiektu z którym zostanie skojarzona. Ramkę można używać do animacji dowolnej ilości obiektów na stronie.

Prócz zaprezentowanego ustawiania from... to można użyć określenia procentowego:

```
@keyframes ramka_p {  
  0% {width: 50px; height: 10px;}  
  20% {width: 200px; height: 100px;}  
  35% {width: 500px; height: 500px;}  
  50% {width: 550px; height: 550px;}  
  60% {width: 400px; height: 300px;}  
  75% {width: 200px; height: 100px;}  
  90% {width: 50px; height: 10px;}  
  100% {opacity: 0.5; filter:alpha(50);}  
}
```

Animacja z tej ramki przez połowę animacji będzie się rozszerzać w nierównomiernych odstępach, po czym ponownie zacznie się kurczyć otrzymując jednocześnie nową wartość właściwości nieprzezroczystości (ostatecznie dany obiekt staje się półprzezroczysty).

**WAŻNE:** aby przeglądarki oparte o technologię webkit poprawnie wykonały animację należy stworzyć dla nich osobne ustawienia animacji oraz ramek poprzedzone przedrostkiem -webkit-, np.

-webkit-animation: <wartosc>;

@-webkit-keyframes <nazwa\_ramki> {}

Internet Explorer dopiero od wersji 10 obsługuje animacje CSS3.

Przykładowa animacja:

```
<!DOCTYPE html>  
<html>  
<head>  
<style>
```

```

div
{
width:50px;
height:10px;
background:red;
position:relative;
animation: ramka_p 5s;
-webkit-animation:ramka_p 5s;
}
@keyframes ramka_p {
0% {width: 50px; height: 10px;}
20% {width: 200px; height: 100px;}
35% {width: 500px; height: 500px;}
50% {width: 550px; height: 550px;}
60% {width: 400px; height: 300px;}
75% {width: 200px; height: 100px;}
90% {width: 50px; height: 10px;}
100% {opacity: 0.5; filter:alpha(50);}
}
@-webkit-keyframes ramka_p {
0% {width: 50px; height: 10px;}
20% {width: 200px; height: 100px;}
35% {width: 500px; height: 500px;}
50% {width: 550px; height: 550px;}
60% {width: 400px; height: 300px;}
75% {width: 200px; height: 100px;}
90% {width: 50px; height: 10px;}
100% {opacity: 0.5; filter:alpha(50);}
}
@keyframes ramka {
from {width: 50px; height: 10px;}
to {width: 200px; height: 100px;}
}
@-webkit-keyframes ramka {
from {width: 50px; height: 10px;}
to {width: 200px; height: 100px;}
}
</style>
</head>
<body>
<div></div>
</body>
</html>

```

Przykład zawiera dwa rodzaje ramek – rozbudowanej (procentowej) oraz prostej (from to), by zmienić ramkę wystarczy podać jej odpowiednią nazwę we właściwości animation.

### 3. Nowe efekty graficzne w CSS3

1) border-radius – jeżeli element posiada utworzone obramowanie to ta właściwość umożliwia jego zaokrąglenie. Wartość określa się w pikselach bądź procentach. Właściwość zbiorcza (można ustawiać każdy róg osobno np. border-top-left-radius, border-top-right-radius itp...).

2) box-shadow – tworzy efekt cienia dla wskazanego elementu HTML na stronie. Jego składnia wygląda następująco:

box-shadow cien\_poziomy cien\_pionowy rozmycie rozmiar kolor inset

Pierwszy parametr pozwala na określenie pozycji cienia w poziomie, drugi określa pozycję cienia w pionie; poprzez rozmycie określa się jak daleko cień będzie „widoczny” (efekt smużenia/rozmycia sztywnej granicy cienia obiektu); pod rozmiar podstawia się wartość całkowitego rozmiaru cienia (jego grubość); kolor to parametr opcjonalny pozwalający zmienić kolor cienia. Ostatni parametr, jeżeli zostanie ustawiony, określi sposób „przechowywania” tworzonego cienia: na zewnątrz elementu (poza jego brzegiem) bądź wewnątrz (zmniejsza pozostałe części elementu).

WAŻNE: przeglądarka Opera (starsze wersje) wymaga w deklaracji przedrostka -o- ; niektóre przeglądarki oparte o Webkit wymagają deklaracji przedrostka -webkit-

3) border-image – tworzy obramowanie ze wskazanego obrazu graficznego. Posiada następujące właściwości:

- border-image-source – ścieżka do obrazu, który ma stać się obramowaniem. Podawany przez funkcję url('ścieżka');
- border-image-slice – określa wewnętrzne przesunięcie treści obramowanego elementu od ramki (odpowiednik padding)
- border-image-width – określa szerokość ramki
- border-image-outset – określa przesunięcie ramki od brzegu elementu (odpowiednik margin)
- border-image-repeat – określa w jaki sposób obrazek ma zostać dopasowany jako obramowanie.

Dostępne wartości:

- stretch – obraz zostanie rozciągnięty (domyślne zachowanie)
- repeat – obraz zostanie powtórzony do czasu osiągnięcia odpowiedniej długości (dopasuje się do elementu)
- round – obraz zostanie powtórzony do czasu osiągnięcia odpowiedniej długości (dopasuje się do elementu). Jeżeli jednak obramowanie miałyby wyglądać nieestetycznie (powtórzenie kończy się w połowie obrazu) to zostanie ono odpowiednio „zaokrąglone” w celu eliminacji niepożądanego efektu

WAŻNE: wszystkie przedstawione powyżej właściwości nie są jeszcze obsługiwane przez większość dostępnych przeglądarek. Jediną możliwością obsłużenia opisanej właściwości jest ustawienie wszystkich parametrów w jednym parametrze:

- border-image – właściwość zbiorcza. Jej deklaracja powinna wyglądać następująco:

border-image: source slice width outset repeat

WAŻNE: Opera wymaga osobnej deklaracji z przedrostkiem -o-  
Przeglądarki oparte na Webkit wymagają przedrostka -webkit-  
Internet Explorer w żadnej wersji nie obsługuje tego parametru.

4) nowe operacje na tłach graficznych elementów strony WWW

- background-clip – dzięki niemu można określić do której części obiektu ma zostać przycięte tło. Możliwe wartości: border-box (do brzegu elementu), padding-box (do obicia elementu), content-box (tylko do zawartości elementu)
- background-origin – określa od której części elementu ma być rozpoczęte rysowanie tła. Dostępne wartości: padding-box (od obicia zawartości), border-box (od brzegu elementu), content-box (od zawartości elementu)
- background-size – określa wielkość obrazu tła. Dostępne wartości: liczbowe (wraz z określeniem

miary CSS), cover (skaluje tło tak by zajęło całą dostępną przestrzeń; może zostać ucięte w niektórych rejonach), contain (dopasowuje wielkość tła do elementu).

Przykład:background-size: 80px 80px; //pierwsze ustawienie to szerokość, drugie to wysokość

5) ustawianie gradientów poprzez style

CSS3 posiada funkcje, dzięki którym bez wykorzystywania plików graficznych można uzyskać efekt gradientu (ładne przechodzenie jednej barwy w inną). Arkusze umożliwiają uzyskanie 2 efektów: liniowego oraz kąтового.

a) liniowy

linear-gradient(color1, color2, color3, ..., color N) – funkcja tworzy liniowy gradient z podanych kolorów (dowolna ilość)

WAŻNE: Przeglądarki oparte o Webkit wymagają użycia przed nazwą funkcji przedrostka -webkit-  
Starsze przeglądarki oparte o Gecko wymagają przedrostka -moz-  
Niektóre przeglądarki Opera wymagają przedrostka -o-  
Internet Explorer dopiero od wersji 10 poprawnie obsługuje tę funkcję.

Kierunek przechodzenia gradientu można zmienić (domyślnie poziomy). Wystarczy przed deklaracjami kolorów dodać odpowiednie słowo-klucz:

I)

dla -webkit: left  
dla funkcji bez przedrostka: to right  
dla pozostałych: right

gradient będzie przechodził z lewa do prawa (pionowo)

II)

dla -webkit: left top  
dla funkcji bez przedrostka: to bottom right  
dla pozostałych: bottom right

gradient będzie przechodził z lewego górnego rogu do prawego dolnego

UWAGA: wartości tego parametru można „miksować” - można np. wpisać wartość left bottom (to top right, top right), right top (to bottom left, bottom left) itd. itp. otrzymując nowe efekty graficzne.

Przykładowa deklaracja:

```
body
{
background: -webkit-linear-gradient(left bottom, red , blue);
background: -o-linear-gradient(top right, red, blue);
background: -moz-linear-gradient(top right, red, blue);
background: linear-gradient(to top right, red , blue); /* ta deklaracja MUSI być na końcu*/
}
```

III) można także użyć kątów

```
body
{
background: -webkit-linear-gradient(75deg, red , blue); /* For Safari */
background: -o-linear-gradient(75deg, red, blue); /* For Opera 11.1 to 12.0 */
background: -moz-linear-gradient(75deg, red, blue); /* For Firefox 3.6 to 15 */
background: linear-gradient(75deg, red , blue); /* Standard syntax (must be last) */
}
```

Wartość kąta w tym przypadku może wynosić od 0 do 360 (z przyrostkiem deg)

Inną funkcją gradientów liniowych jest:

repeat-linear-gradient(color1, color2 val%, color3 val%,..., colorN val%) - kopiuje ustawiony gradient tyle razy, by zappełnić element, dla którego został ustawiony. Wartości val% określają procentową zawartość danego koloru w gradiencie.

Przykład użycia:

```
body
{
background: -webkit-repeating-linear-gradient(red , blue 15%);
background: -o-repeating-linear-gradient(red, blue 15%);
background: -moz-repeating-linear-gradient(red, blue 15%);
background: repeating-linear-gradient(red , blue 15%);
}
```

b) radialny

radial-gradient(colorStart, color1, color2, ..., colorEnd) – tworzy elipsę ze środkiem wypełnionym kolorem startu. Następnie, w postaci pierścieni dodaje kolejne kolory by jako tło wykorzystać kolor końcowy.

WAŻNE: gradient radialny wymaga identycznych przedrostków co gradient liniowy!

W powyżej zaprezentowanej funkcji każdy z kolorów zajmie identyczną ilość miejsca w gradiencie. Można oczywiście zdecydować ile procent dany kolor będzie zajmować poprzez wartości procentowe, np.

```
body
{
background: -webkit-radial-gradient(red 5%, green 15%, blue 60%);
background: -o-radial-gradient(red 5%, green 15%, blue 60%);
background: -moz-radial-gradient(red 5%, green 15%, blue 60%);
background: radial-gradient(red 5%, green 15%, blue 60%); /*ta funkcja musi wystapic na koncu*/
}
```

Domyślnie funkcja radialna używa kształtu elipsy (chyba, że gradient nakładany jest na kwadrat). Jeżeli jednak chcemy wykorzystać kształt koła wystarczy przed definicją kolorów napisać parametr circle, np.

```
body
{
background: -webkit-radial-gradient(circle, red 5%, green 15%, blue 60%);
background: -o-radial-gradient(circle, red 5%, green 15%, blue 60%);
background: -moz-radial-gradient(circle, red 5%, green 15%, blue 60%);
```

```
background: radial-gradient(circle, red 5%, green 15%, blue 60%);
}
```

Samo rysowanie gradientu można również zmienić. Możliwe jest przestawienie środka rysowania i/lub sposób wypełniania tła przez kręgi. Istnieją 4 sposoby wypełnienia:

- closest-side – gradient rozchodzi się do najbliższej krawędzi
- farthest-side – gradient rozchodzi się do najdalszej krawędzi
- closest-corner – gradient rozchodzi się do najbliższego narożnika
- farthest-corner (domyślne) – gradient rozchodzi się do najdalszego narożnika

prócz tego można decydować o położeniu centrum rozchodzenia się kolorów (poprzez procentowe wartości), np.

```
body
{
background: -webkit-radial-gradient(60% 55%, farthest-side,blue,green,yellow,black);
background: -o-radial-gradient(60% 55%, farthest-side,blue,green,yellow,black);
background: -moz-radial-gradient(60% 55%, farthest-side,blue,green,yellow,black);
background: radial-gradient(60% 55%, farthest-side,blue,green,yellow,black);
}
```

Gradient radialny także można powtarzać. Służy do tego funkcja:

repeating-radial-gradient(colorStart, color1 val%, color2 val%, ..., colorEnd val%) - powtarza gradient wedle ustawionego wzoru aż do zapelnienia całej dostepnej przestrzeni elementu

przykład:

```
body
{
background: -webkit-repeating-radial-gradient(red, yellow 10%, green 15%);
background: -o-repeating-radial-gradient(red, yellow 10%, green 15%);
background: -moz-repeating-radial-gradient(red, yellow 10%, green 15%);
background: repeating-radial-gradient(red, yellow 10%, green 15%);
}
```

#### 4. Przystosowywanie interfejsu użytkownika poprzez style CSS3

- resize – umożliwia użytkownikowi zmianę wielkości wskazanego elementu. Możliwe wartości:

- 1) none – zmiana rozmiaru nie jest możliwa
- 2) both – zmiana rozmiaru jest możliwa zarówno w pionie jak i poziomie
- 3) horizontal – zmiana rozmiaru jest możliwa tylko w poziomie
- 4) vertical – zmiana rozmiaru jest możliwa tylko w pionie

**WŁAŚCIWOŚĆ NIE DZIAŁA Z INTERNET EXPLORER ORAZ OPERA**

- box-sizing – pozwala ustalić w jaki sposób wskazane elementy strony będą wypełniały element nadrzędny. Dostępne wartości:

- 1) content-box – szerokość i wysokość elementu ustawiane są jako parametry zawartości elementu. Wszelkie pozostałe części obiektu (padding, border) znajdują się poza wskazanymi wartościami i rysowane są poza wskazaną szerokością i wysokością (przez co cały element staje się większy)
- 2) border-box – szerokość i wysokość elementu obejmuje także części padding oraz border. Oznacza to, że rozmiar zawartość jest odpowiednio pomniejszany o rozmiar obramowania i

obicia/otoczenia (domyślne ustawienie).

UWAGA: przeglądarki używające Gecko wymagają prefiksu -moz-

- outline-offset – ustawia obramowanie poza obiektem (poza jego granicą). Obramowanie poza obiektem nie zajmuje miejsca (nie wlicza się do szerokości oraz wysokości elementu) oraz obrysowuje obiekt (nie musi być prostokątne). Przyjmuje wartości liczbowe w dowolnej mierze CSS.

**WŁAŚCIWOŚĆ NIE DZIAŁA Z INTERNET EXPLORER**

- appearance – właściwość pozwala na zmianę wyglądu wskazanego obiektu tak, by przypominał część interfejsu użytkownika systemu, na którym strona jest aktualnie wyświetlana. Dostępne wartości:

- 1) normal – renderuje jako normalny element
- 2) icon – renderuje jako mały plik graficzny
- 3) window – renderuje jako okno programu
- 4) button – renderuje jako przycisk systemowy
- 5) menu – renderuje jako zestaw opcji wyboru
- 6) field – renderuje jako pole do wstawiania tekstu

UWAGA: Aby opcja działała na Webkit, należy dodać prefiks -webkit-

Aby opcja działała na Gecko należy dodać prefiks -moz-

**NIE DZIAŁA NA OPERA ORAZ INTERNET EXPLORER**

W większości właściwości CSS można ustawiać „inherit”. Wartość ta oznacza, że dany element odziedziczy wartość właściwości z elementu nadrzędnego (rodzica). Czyli jeżeli np. istnieje jeden element div, dla którego określono wielkość czcionki, po czym dodany zostanie element span, a w nim wielkość czcionki ustawiona będzie na inherit, to czcionka w elemencie span będzie miała wielkość czcionki z warstwy, która zawiera tworzony znacznik span.

Na tym etapie materiał zaplanowany na CSS został zakończony. Wszelkie dodatkowe informacje na temat stylów można odnaleźć w sieci internet.

Zadania:

- 1) Proszę na podstronie Galeria utworzyć schludną minigalerię – wszystko przy pomocy CSS. Dobrym rozwiązaniem byłoby, aby przy najechaniu kursorem na dany element obraz był np. powiększany bądź obramowywany.
- 2) Jeżeli użytkownik nakieruje kursor myszy na którąkolwiek pozycję z menu strony, ta powinna się okręcić o 45% i nieznacznie powiększyć.
- 3) Wyświetlane zdjęcie (po kliknięciu) może być lekko pochylone (skew).
- 4) Proszę stworzyć animowany tytuł strony – przykładowo obracający się o dany kąt bądź też przesuwany się z jednej strony w drugą. Animacja powinna być stała (przez cały czas wyświetlania strony).
- 5) W formularzu kontaktowym proszę uczynić pola tekstowe rozszerzalnymi. Przykładowo pole typu text powinno być rozszerzalne w poziomie podczas gdy pole textarea powinno mieć możliwość rozszerzania w pionie i poziomie.