



PODSTAWOWE POJĘCIA

OSADZANIE STYLÓW W DOKUMENCIE

Styl wewnętrzny

Styl lokalny

```
<p style="color:green;font-size:15pt;
">Treść dokumentu</p>
```

Styl zagnieźdżony

```
<html>
<head>
<style type="text/css">
P
{
color:green;
font-size:15pt;
}
</style>
</head>
<body>
<p>Tekst akapitu</p>
</body>
</html>
```

Styl zewnętrzny

Zewnętrzny arkusz stylów

Umieszczany jest w nagłówku <head> dokumentu. Styl zewnętrzny obejmuje działaniem całą zawartość strony i może być wykorzystany również w podstronach.

```
<html>
<head>
<link href="arkusz.css" rel="stylesheet"
type="text/css" />
</head>
<body>
<p>Tekst akapitu</p>
</body>
</html>
```

W przypadku języka XHTML polecenie odpowiedziałne za podłączenie zewnętrznego arkusza stylów ma postać:

```
<?xml-stylesheet type="text/css"
href="arkusz.css" ?>
Dla zachowania zgodności dokumentów XHTML zaleca się umieszczenie podwójnej deklaracji wywołującej arkusz stylów. Oczywiście oba wpisy muszą znaleźć się w nagłówku kodu witryny.
```

```
<link rel="stylesheet" type="text/css"
href="arkusz.css" />
<?xml-stylesheet type="text/css"
href="arkusz.css" ?>
```

Styl importowany

Jest pobierany z oddzielnego pliku znajdującego się pod wskazanym adresem. Podobnie jak styl zewnętrzny, działaniem obejmuje całą treść strony, na której został użyty.

```
<html>
<head>
<style type="text/css">
@import url("arkusz.css");
P
{
color:green;
}
</style>
</head>
<body>
<p>Tekst akapitu</p>
</body>
</html>
```

Budowa zewnętrznego arkusza stylów

Zewnętrzny arkusz jest zwykłym plikiem tekstowym, a jego zawartość może wyglądać tak jak na poniższym przykładzie. Ważne jest tylko, aby plik miał rozszerzenie .css, np. arkusz.css.

```
(* To jest przykład zewnętrznego arkusza stylów *)
P
{
color:green;
font-size:15pt;
}
```

Trzeci z selektorów wykorzystujących ciągi znaków pozwala na podpięcie stylu tekstu, który występuje w dowolnym miejscu ciągu znaków. Zwróć uwagę na znak " w zapisie selektora. Poniższy przykład dotyczy wszystkich znaczników H1, w których występuje atrybut title zawierający ciąg znaków nagłówek.

```
H1[title="nagłówek"]
{
font-style:italic;
}
```

SELEKTOR OGÓLNEGO RODZIEŃSTWA

Selektor ogólnego rodzeństwa składa się z dwóch prostych selektorów rozdzielonych znakiem ~ (tyldy) i pasuje do tych elementów selektora drugiego (w poniższym przykładzie jest to P), które są poprzedzone elementami pierwszego selektora (w poniższym przykładzie jest to H1). Oba elementy muszą mieć wspólnego rodzica (w naszym przypadku jest to DIV) i dodatkowo drugi element nie musi być bezpośrednio poprzedzony pierwszym elementem.

```
<div>
<h1></h1>
<p></p>
<h1></h1>
<p></p>
<h1></h1>
<p></p>
</div>
```

Przykład selektora ogólnego rodzeństwa wygląda następująco:

```
H1 ~ P
{
color:red;
font-weight:bolder;
}
```

GRUPOWANIE SELEKTORÓW

Selektory można dowolnie grupować, jeżeli pewne wartości mają obowiązywać dla każdego z nich. Dzięki zbiorczemu zapisowi możemy znacznie uprościć strukturę arkusza i pracę z kodem witryny. Poniżej przedstawiono przykład zgrupowanych selektorów H1, H2 oraz P.

```
H1, H2, P
{
color:green;
}
```

IDENTYFIKATORY

Identyfikator pozwala na przypisanie formatowania do wybranego znacznika mającego unikalny atrybut id. Dzięki temu możemy zróżnicować sposób prezentacji elementów na stronie.

```
#pochyly
{
font-style:italic;
font-size:20pt;
}
```

Aby skorzystać z przykładowego stylu, wybrany znacznik musi mieć dodatkowy atrybut id.

```
<p id="pochyly">Treść akapitu</p>
```

Możliwy jest również inny zapis, który wymusza przypisanie identyfikatora wyłącznie do określonego znacznika.

```
H1#pochyly
{
font-style:italic;
font-size:20pt;
}
```

KLASY

Klasa umożliwia zróżnicowanie formatowania wybranych elementów na stronie w zależności od atrybutu class umieszczonego w znaczniku HTML. W odróżnieniu od identyfikatora klasa może być wykorzystywana wielokrotnie.

```
.moja
{
color:green;
font-size:14pt;
}
```

Aby skorzystać z przykładowej klasy, znacznik musi mieć zdefiniowany atrybut class z nazwą klasy.

```
<p class="moja">Treść akapitu</p>
```

Klasa może zostać również powiązana z konkretnym znacznikiem HTML.

```
P.moja
{
color:green;
font-size:14pt;
}
```

PSEUDOKLASY

Specyfikacja CSS przewiduje kilka ściśle określonych klas pozwalających na formatowanie niektórych elementów dokumentu. Do najpopularniejszych pseudoklas z całą pewnością zaliczają się pseudoklasy odsyłaczy.

Pseudoklasa :link

Dotyczy formatowania odnośnika w stanie nienaruszonym.

```
A:link
{
color:navy;
}
```

Pseudoklasa :visited

Dotyczy formatowania odnośnika, który został już odwiedzony. Pozwala na wyróżnienie go spośród innych odnośników znajdujących się na stronie.

```
A:visited
{
color:green;
}
```

Pseudoklasa :hover

Pozwala na ustalenie formatowania elementu, nad którym w danej chwili znajduje się kursor myszy.

```
A:hover
{
color:red;
text-decoration:underline;
}
```

Pseudoklasa :active

Odpowiada za ustalenie wyglądu aktywnego elementu na stronie.

```
A:active
{
color:silver;
}
```

Pseudoklasa :focus

Pozwala na wyróżnienie aktualnie wyświetlanego elementu, np. odsyłacza.

```
A:focus
{
color:braun;
}
```

Pseudoklasa :lang

Jest wykorzystywana do określania formatowania dokumentów wielojęzycznych.

```
P:lang(pl)
{
font-weight:bold;
color:black;
}
```

Aby skorzystać z przykładowej pseudoklasy :lang zdefiniowanej powyżej, kod HTML musi wyglądać następująco:

```
<p lang="pl">Tekst w języku polskim.</p>
```

Pseudoklasa :target

Stosując pseudoklasę :target, możemy wymusić formatowanie elementów będących kotwicami na stronie. Formatowanie będzie miało miejsce tylko wtedy, gdy odnośnik prowadzący do kotwicy zostanie wybrany, a adres będzie miał postać: <http://adres/pl/plik.html#kotwica>. Odpowiedni arkusz stylów może mieć następującą postać:

```
*:target
{
text-decoration:none;
color:green;
}
```

Pseudoklasa :enabled i :disabled

Stosując pseudoklasy :enabled oraz :disabled, możemy kontrolować wygląd elementów interfejsu użytkownika (formularzy), w zależności od tego, czy są one dostępne, czy też nie.

```
input[type="text"]:enabled
{
background:#C0FFC0;
}
input[type="text"]:disabled
{
background:#FFC0C0;
}
```

Pseudoklasa :checked

Dzięki pseudoklasie :checked możemy kontrolować wygląd aktywnych pól formularza typu radio lub checkbox.

```
input:checked
{
border:1px solid red;
color:green;
width:25px;
height:25px;
background-color:red;
}
```

Pseudoklasa :root

Pseudoklasa :root należy do grupy pseudoklas strukturalnych, za pomocą których możemy odwołać się do struktury dokumentu XHTML/HTML; ta struktura jest poza zasięgiem klasycznych selektorów. Omawiana pseudoklasa odnosi się do korzenia struktury, którym w przypadku dokumentów XHTML/HTML jest element HTML, i zawsze działa w całym dokumencie.

```
:root
{
background-color:#C0FFC0;
color:navy;
}
```

Pseudoklasa :nth-child()

Dzięki pseudoklasie :nth-child() możemy odwołać się do elementu, który ma przed sobą określoną w nawiasie liczbę rodzeństwa w strukturze dokumentu. Wartość umieszczona w nawiasie może być numerem, słowem kluczowym lub formułą.

```
p:nth-child(3)
{
color:red;
}
UL LI:nth-child(even)
{
color:red;
font-style:italic;
}
OL LI:nth-child(odd)
{
color:green;
font-weight:bolder;
}
LI:nth-child(2n+5)
{
color:red;
font-style:italic;
font-weight:bolder;
}
```

SELEKTORY

GRUPOWANIE SELEKTORÓW

Wyobraź sobie, że w Twoim dokumencie tekst ma mieć kolor czarny, a wszystkie użyte nagłówki powinny być czerwone. Odpowiedni arkusz stylów może mieć postać:

```
P
{
color:black;
}
H1
{
color:red;
}
H2
{
color:red;
}
H3
{
color:red;
}
```

Powyższy kod możemy skrócić, grupując selektory. Uproszczona wersja może mieć postać:

```
P
{
color:black;
}
H1, H2, H3
{
color:red;
}
```

SELEKTOR PROSTY

Selektorem prostym jest znacznik języka HTML przypisany do elementu, który chcemy w danym momencie formatować. Styl zdefiniowany dla przykładowego selektora P będzie się odnosił wyłącznie do akapitów zamkniętych w znaczniku <p></p>.

```
P
{
color:red;
font-size:14pt;
}
```

SELEKTOR UNIWERSALNY

Selektor uniwersalny pozwala na ustalenie formatowania dla wszystkich elementów strony WWW. Obecnie dostępne są dwa selektory uniwersalne: BODY oraz *.

```
*
{
color:red;
}
BODY
{
color:red;
}
```

SELEKTOR POTOMKA

Selektor potomka pozwala na ustawienie formatowania dla elementu występującego we wnętrzu innego znacznika. Przykład wymusza nadanie formatowania fragmentowi tekstu zamkniętego w i znajdującego się w znaczniku <h1></h1>.

H1 SPAN

```
{
font-size:10pt;
}
```

SELEKTOR DZIECKA

Selektor dziecka jest wynikiem zależności panujących pomiędzy poszczególnymi znacznikami języka HTML. Przykładowy styl zadziała, gdy wewnątrz bloku tekstu zostanie umieszczony znacznik .

```
P > SPAN
{
font-style:italic;
text-decoration:underline;
}
```

SELEKTOR RODZIEŃSTWA

Na podstawie tego rodzaju selektora możemy przypisać styl elementowi bezpośrednio sąsiadującemu z innymi. By warunek został spełniony, oba selektory muszą mieć wspólnego rodzica.

```
P + SPAN
{
font-style:italic;
text-decoration:underline;
}
```

SELEKTOR ATRYBUTU

Selektor atrybutu odnosi się do wybranego znacznika HTML mającego ściśle określony atrybut. Przykładowy styl zadziała w przypadku nagłówka stopnia pierwszego o określonym atrybucie title.

```
H1[title]
{
color:red;
}
```

SELEKTOR ATRYBUTU Z MOŻLIWOŚCIĄ DOPASOWANIA CIĄGU ZNAKÓW

Selektor atrybutu z możliwością dopasowania ciągu znaków daje możliwość odwołania się do wybranego elementu strony na podstawie fragmentu ciągu znaków występujących jako atrybut. Dostępne są trzy sposoby odwołania się do ciągu znaków. Pierwszy z selektorów wykorzystujących ciągi znaków pozwala na podpięcie stylu tekstu, od którego zaczyna się dany ciąg znaków. Zwróć uwagę na znak ~ w zapisie selektora. Poniższy przykład dotyczy wszystkich znaczników H1, w których występuje atrybut title zaczynający się od znaków pie.

```
H1[title~="pie"]
{
color:red;
}
```

Drugi z selektorów wykorzystujących ciągi znaków pozwala na podpięcie stylu tekstu, którym kończy się dany ciąg znaków. Zwróć uwagę na znak \$ w zapisie selektora. Poniższy przykład dotyczy wszystkich znaczników H1, w których występuje atrybut title kończący się na ony.

```
H1[title$="ony"]
{
text-decoration:underline;
}
```

Pseudoklasa :nth-last-child()
Pseudoklasa :nth-last-child() działa podobnie jak opisana powyżej pseudoklasa :nth-child(), z tą różnicą, że liczenie elementów odbywa się od końca, nie zaś od początku, jak było w pseudoklasie :nth-child().

```
LI:nth-last-child(2n+5)
{
  color:red;
  font-style:italic;
  font-weight:bold;
}
```

Pseudoklasa :nth-of-type()
Stosując pseudoklasę :nth-of-type(), możemy się odwołać do ściśle określonego elementu, który ma przed sobą określoną w nawiasie liczbę rodzeństwa w strukturze dokumentu. Wartość umieszczona w nawiasie może być: numerem, słowem kluczowym lub formułą. Innymi słowy, :nth-of-type() działa podobnie jak pseudoklasa :nth-child(), ale odnosi się wyłącznie do elementów określonego typu, np. do akapitów.

```
P:nth-of-type(3)
{
  color:red;
}
```

Pseudoklasa :nth-last-of-type()
Pseudoklasa :nth-last-of-type() działa podobnie jak opisana powyżej pseudoklasa :nth-of-type(), z tą różnicą, że liczenie ściśle określonych elementów odbywa się od końca, nie zaś od początku, jak było w pseudoklasie :nth-of-type(). Oczywiście również tutaj pomiędzy nawiasami możemy stosować numery, słowa kluczowe lub formuły.

```
HI:nth-last-of-type(odd)
{
  color:red;
  font-style:italic;
  font-weight:bold;
}
```

Pseudoklasa :last-child
Kolejną strukturalną pseudoklasą jest :last-child, dzięki której możemy odnieść się z formatowaniem do elementu będącego ostatnim dzieckiem swojego rodzica. Przykładowy arkusz stylów wykorzystujący pseudoklasę :last-child może mieć postać:

```
LI:last-child
{
  color:red;
  font-style:italic;
  font-weight:bold;
}
```

Pseudoklasa :first-child
Pseudoklasa :first-child odpowiada za formatowanie pierwszego dziecka elementu nadrzędnego. Dokładniej mówiąc, jeżeli np. znacznik <div> zawiera w sobie trzy akapity określone za pomocą znacznika <p>, to formatowaniu zostanie poddany tylko pierwszy z nich:

```
P:first-child
{
  font-size:10pt;
  font-style:italic;
}
```

Pseudoklasa :first-of-type
Pseudoklasa :first-of-type to kolejny selektor strukturalny, dzięki któremu możemy przypisać formatowanie do pierwszego ściśle określonego elementu mającego wspólnego rodzica. Mówiąc prościej: Kiedy korzystamy z prezentowanej pseudoklas, możemy na przykład poddać formatowaniu pierwszą pozycję listy.

```
LI:first-of-type
{
  color:red;
  font-style:italic;
  font-weight:bold;
}
```

Pseudoklasa :last-of-type
Pseudoklasa :last-of-type to następny selektor strukturalny, dzięki któremu możemy przypisać formatowanie do ostatniego ściśle określonego elementu mającego wspólnego rodzica. Innymi słowy, korzystając z prezentowanej pseudoklas, możemy na przykład poddać formatowaniu ostatnią pozycję listy.

```
LI:last-of-type
{
  color:red;
  font-style:italic;
  font-weight:bold;
}
```

Pseudoklasa :only-child
Pseudoklasa :only-child pozwala poddać formatowaniu tylko te elementy, które są pierwszym i jedynym dzieckiem swojego rodzica. Przykładowy arkusz korzystający z prezentowanej pseudoklas może mieć postać:

```
LI:only-child
{
  color:red;
}
```

JEDNOSTKI I WARTOŚCI

JEDNOSTKI WZGLĘDNE

px – piksele

```
P {margin-left:20px;}
```

Jednostka ta opiera się na pojedynczych punktach widocznych na ekranie monitora – pikselach.

em – określa proporcję wysokości do domyślnej czcionki danego elementu

```
P {margin-left:2em;}
```

Zasada działania tej jednostki jest stosunkowo prosta i polega na określaniu zależności pomiędzy poszczególnymi wielkościami czcionki.

ex – określa proporcję do wysokości litery x

```
P {margin-left:1ex;}
```

Stosowanie jednostki ex wiąże się z rodzajem użytej czcionki. Jak zapewne wiesz, każda czcionka inaczej wygląda, a co za tym

```
font-style:italic;
font-weight:bold;
}
```

Pseudoklasa :only-of-type
Pseudoklasa :only-of-type działa podobnie jak opisana nieco wcześniej :only-child, z tą różnicą, że dotyczy tylko tych elementów, które są jedynymi dziećmi danego typu swojego rodzica.

```
P:only-of-type
{
  color:red;
  font-style:italic;
  font-weight:bold;
}
```

Pseudoklasa :empty
Ostatnią strukturalną pseudoklasą jest :empty, która odpowiada za formatowanie elementów niemających swoich dzieci i niezawierających innych elementów. Dlatego z trzech poniższych przykładów tylko pierwszy element możemy uznać za pusty.

```
<div><div>
<div>Lorem ipsum dolor sit amet, consectetur
adipiscing elit.</div>
<div></div>
</div>
```

Odpowiedni arkusz wykorzystujący omawianą pseudoklasę może mieć następującą postać:

```
DIV:empty
{
  width:80%;
  height:200px;
  background-color:silver;
  margin:auto;
}
```

Pseudoklasa :not()
Pseudoklasa :not() zawiera w nawiasie argument, który jest selektorem prostym, np. może to być p. Następnie wymusza formatowanie tych elementów, które nie są argumentem zapisanym w nawiasie.

```
:not(p)
{
  color:red;
}
```

PSEUDOELEMENTY

Ciekawą grupę selektorów stanowią pseudoelementy, które pozwalają nam na przypisanie kolejnych wymyślnych stylów do elementów strony WWW.

::first-letter
Pozwala na formatowanie pierwszej litery w bloku tekstu. Przedstawiony poniżej arkusz spowoduje powiększenie pierwszej litery do wielkości 100 punktów i nadanie jej koloru czerwonego.

```
P::first-letter
{
  color:red;
  font-size:100pt;
}
```

::first-line
W przeciwieństwie do poprzednika obejmuje formatowaniem całą pierwszą linię, a nie tylko pierwszą literę.

```
P::first-line
{
  color:red;
  font-size:20pt;
  font-weight:bold;
}
```

::after oraz ::before
Służą do umieszczania tekstu przed formatowanym elementem strony i za nim.

```
P::before
{
  content:"WAŻNA WIADOMOŚĆ";
  font-size:15pt;
  font-style:italic;
  font-weight:bold;
}
```

P::after

```
{
  content:"AUTOR Bartosz Danowski";
  font-size:15pt;
  font-style:italic;
  font-weight:bold;
}
```

::selection
Pseudoelement ::selection służy do formatowania tekstu zaznaczonego przez użytkownika strony. Wraz z tym elementem możemy używać następujących stylów: color, background, cursor, outline.

```
P::selection
{
  color:red;
}
```

idzie, ma inną wielkość podstawową, dlatego czcionka Arial o wielkości 2ex nie będzie równa czcionce Times New Roman o wielkości 2ex.

% – procenty

```
P {font-size:10%;}
```

Procenty służą do określania wielkości względem wartości domyślnej.

JEDNOSTKI BEZWZGLĘDNE

in – cal

```
P {margin-left:2in;}
```

Cale wywodzi się z amerykańskiego systemu miarowego i głównie tam są wykorzystywane. W Polsce ta jednostka raczej nie będzie potrzebna.

pt – punkt

```
P {font-size:12pt;}
```

Punkty wywodzi się z typografii, gdzie są standardową jednostką miary. W praktyce 72 punkty odpowiadają jednemu calowi, a ten z kolei równa się 2,54 cm.

cm – centymetr

```
P {font-size:1cm;}
```

Centymetr jest miarą stosowaną w większości krajów na świecie. Jeden centymetr odpowiada 0,39 cala.

mm – milimetr

```
P {font-size:12mm;}
```

Milimetr jest jednostką mniejszą niż centymetr. Występuje w systemie metrycznym; na jeden centymetr składa się 10 mm.

pc – pica

```
P {font-size:12pc;}
```

Jest jednostką typograficzną, podobnie jak punkt; jeden pica równa się 12 punktom.

JEDNOSTKI CZASU I JEDNOSTKI KĄTOWE

Jednostki służące do określania czasu oraz kątów zostały wprowadzone w drugiej sekcji – CSS2. Mają zastosowanie przy projektowaniu stron wzbogacających o możliwość czytania dokumentu przez komputer oraz obsługi mediów osadzanych na stronach WWW.

Jednostki czasu to:

```
ms – milisekunda
HS {pause-after:100ms;}
s – sekunda
HS {pause-after:10s;}
```

Obie jednostki są używane do określania odstępów pomiędzy czytanyimi fragmentami.

Jednostki kątowe to:

```
deg – stopień
HS {elevation:10deg;}
grad – gradian
HS {elevation:10grad;}
rad – radian
HS {elevation:10rad;}
```

Jednostki te używane są do określania pozycji źródła dźwięku.

WARTOŚCI INHERIT, INITIAL ORAZ UNSET

Wartość inherit wymusza dziedziczenie właściwości danego elementu bezpośrednio od jego rodzica. Dokładniej widać to na przykładzie:

DZIEDZICZENIE I KASKADOWOŚĆ

Zrozumienie pojęcia dziedziczenia wymaga od nas zapoznania się z hierarchią ważności poszczególnych znaczników wewnątrz dokumentu. Kaskadowe arkusze stylów wprowadzają pojęcie drzewa i na jego przykładzie doskonale widać te zależności. Poniżej znajduje się przykładowy listing prostego kodu strony oraz rozrysowane dla niego drzewo.

```
<html>
<head>
<title>przykład drzewa</title>
</head>
<body>
<p>przykład tekstu.</p>
<table>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</table>
</body>
</html>
```

Analizując rysunek, widzimy, że określenie np. koloru tekstu (niebieski) bezpośrednio dla pary znaczników <body></body> spowoduje jego użycie dla każdego elementu strony znajdującego się niżej w hierarchii. Jeżeli dodatkowo zdefiniujemy kolor listy jako zielony, to tekst całego dokumentu będzie niebieski, poza listą, która zostanie sformatowana kolorem zielonym.

Kolejnym bardzo ważnym pojęciem stosowanym w kaskadowych arkuszach stylów, a przy tym występującym w samej nazwie, jest kaskadowość. Funkcja ta odpowiada za określenie

```
<div style="border: 9px double red;
padding:10px; margin:5px">
<div style="border:inherit; padding:inherit;">

Dziecko elementu nadrzędnego. Posiada obramowanie, ponieważ
wykorzystał wartość inherit.

</div>

<div style="border: padding:10px;
margin:5px">

Dziecko elementu nadrzędnego. Nie posiada obramowania, ponieważ
nie użyto wartości inherit.
</div>
</div>
```

Dziecko elementu nadrzędnego. Posiada obramowanie, ponieważ wykorzystał wartość inherit.

Dziecko elementu nadrzędnego. Nie posiada obramowania, ponieważ nie użył wartości inherit.

Rysunek 1. Przykład działania wartości inherit

Parametr initial umożliwia zresetowanie ustawień danego elementu do jego domyślnej wartości określonej przez specyfikację CSS.

Ostatnia z uniwersalnych wartości to unset. Pozwala ona na zresetowanie ustawień danego elementu do jego domyślnych wartości określonych w specyfikacji CSS. Dodatkowo automatycznie dziedziczy wartość od swojego rodzica.

WARTOŚCI POZYCJONUJĄCE

Specyfikacja CSS przewiduje następujące wartości pozycjonujące, które mogą zostać wykorzystane przy określaniu właściwości wybranego elementu.

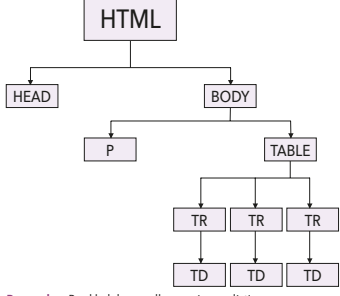
- top – pozycjonuje dany element do góry,
- bottom – pozycjonuje dany element do dołu,
- right – pozycjonuje dany element do prawej strony,
- left – pozycjonuje dany element do lewej strony,
- center – ustawia dany element na środku.

P {text-align:center;}

Należy pamiętać o tym, że pozycje można określać w sposób złożony. Na przykład dany element można pozycjonować do górnego prawego narożnika.

DIV {background-position: right top;}

hierarchii stosowanych stylów w dokumencie. Wiemy już, że style możemy wstawiać do dokumentu na kilka sposobów (bezpośrednio w kodzie strony jako atrybut dowolnego znacznika, w nagłówku <head></head>, globalnie dla całego dokumentu oraz przez dołączenie zewnętrznego arkusza). Mieszanie zastosowanych stylów jest możliwe i często spotykane, dlatego konieczne stało się określenie ważności poszczególnych metod. Zasada kaskadowości przyjęta przez twórców specyfikacji CSS wygląda następująco: najpierw ładowane i uwzględniane są zewnętrzne arkusze, następnie style wpisane do nagłówka <head></head>, a na samym końcu style wpisane bezpośrednio do znacznika. Takie rozwiązanie umożliwia pełną kontrolę nad dokumentem, a w przypadku sprzeczności zdefiniowanych stylów użyty zostanie ten styl, który jest najbliższy formatowanemu dokumentowi.



Rysunek 2. Przykład drzewa dla powyższego listingu

REGUŁY SPECJALNE I DEKLARACJE

REGUŁY

@charset
Służy do deklarowania strony kodowej zewnętrznego arkusza stylów. Reguła powinna zostać zawarta na początku arkusza, a zapis ma postać:

```
/*Zewnętrzny arkusz stylów/*
@charset "UTF-8";
P {color: red;}
```

@media
Pozwala na deklarowanie stylów odpowiedzialnych za wygląd strony na konkretnych typach urządzeń. Mówiąc precyzyjniej, możemy zadeklarować odmienne formatowanie dla strony wyświetlanej na ekranie komputera czy wydruku. Dostępne typy mediów to:

- all – wszystkie media;
- print – formatowanie dla wydruków, przydatne, gdy chcesz stworzyć ładnie wyglądające wydruki Twojej strony;
- screen – formatowanie dla ekranu komputera;
- speech – formatowanie dla oprogramowania czytającego daną stronę na głos – syntezatorów mowy.

Przykład widoczny poniżej wymusza czerwone tło strony, gdy zostanie ona uruchomiona na monitorze z rozdzielczością poziomą 1920px.

```
@media screen and (min-width: 1920px) {
  body {
    background-color:red;
  }
}
```

@import
Umożliwia podłączenie jednego arkusza stylów do drugiego. Możesz stworzyć jeden główny arkusz, a następnie podłączyć do niego drugi dedykowany np. tylko do wydruków. Przykład takiego rozwiązania znajduje się poniżej.

```
/*Zewnętrzny arkusz stylów - domyślny/*
@charset "UTF-8";
@import url("wdruki.css") print;

P {color: red;}
```

@font-face
Służy do podłączania własnego pliku czcionki do dokumentu HTML. Zapis w arkuszu stylów ma postać:

```
@font-face {
  font-family:MojaKypasionaCzcionka;
  src:url (fonts/plikczcionki.ttf);
}
```

Aby skorzystać z czcionki podłączonej za pomocą reguły @font-face, wystarczy odwołać się do przypisanej do niej nazwy.

```
P {font-family:MojaKypasionaCzcionka;}
```

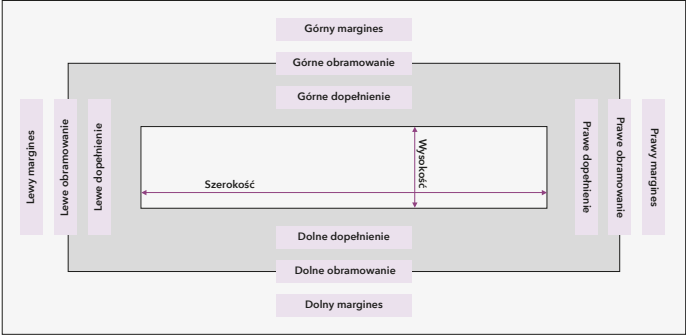
DEKLARACJA !IMPORTANT

Reguła !important pozwala na łamanie reguł rządzących światem kaskadowych arkuszy stylów. Mówiąc prościej, dzięki wprowadzeniu deklaracji !important możemy np. „wylączyć” nadziedzenie i narzucić z góry wybraną wartość dla formatowanego elementu.

```
<style>
P {
  color: red !important;
}
</style>
```

PUDEŁKOWY MODEL FORMATOWANIA ELEMENTÓW

Kaskadowe arkusze stylów do formatowania wszystkich elementów wykorzystują model pudełka. W praktyce polega to na tym, że wszystkie elementy umieszczają się w dodatkowej przestrzeni o kształcie prostokątów.



Rysunek 3. Schemat pudełka

MARGINESY	
Lewy margines	
margin-left	
Dostępne wartości:	wartość liczbową wartość procentową auto initial inherit
Wartość domyślna:	0
Przykład:	p {margin-left:1cm;}
Dziedziczenie:	nie

Prawy margines	
margin-right	
Dostępne wartości:	wartość liczbową wartość procentową auto initial inherit
Wartość domyślna:	0
Przykład:	p {margin-right:1cm;}
Dziedziczenie:	nie

Górny margines	
margin-top	
Dostępne wartości:	wartość liczbową wartość procentową auto initial inherit
Wartość domyślna:	0
Przykład:	p {margin-top:1cm;}
Dziedziczenie:	nie

Dolny margines	
margin-bottom	
Dostępne wartości:	wartość liczbową wartość procentową auto initial inherit
Wartość domyślna:	0
Przykład:	p {margin-bottom:1cm;}
Dziedziczenie:	nie

Zbiórny zapis właściwości marginesów	
margin	
Dostępne wartości:	<margin-top> <margin-right> <margin-bottom> <margin-left>
Wartość domyślna:	zależy od indywidualnych wartości poszczególnych stylów
Przykład:	p {margin:1cm 3cm;}
Dziedziczenie:	nie

Replikacja wartości marginesów
Schemat przedstawia zasady, z których korzystają kaskadowe arkusze stylów przy zbiorczym zapisie wartości marginesów. Powielanie wartości marginesów w zbiorczym zapisie nazywa się replikacją wartości. Wartość górnego marginesu może zostać pobrana przez prawy i dolny margines. Natomiast prawy margines jest wykorzystywany przez lewy.



Rysunek 4. Schemat powielania wartości marginesów w zbiorczym zapisie

DOPEŁNIENIA	
Lewe dopełnienie	
padding-left	
Dostępne wartości:	wartość liczbową initial inherit
Wartość domyślna:	0
Przykład:	p {padding-left:1cm;}
Dziedziczenie:	nie

Prawe dopełnienie	
padding-right	
Dostępne wartości:	wartość liczbową initial inherit
Wartość domyślna:	0
Przykład:	p {padding-right:1cm;}
Dziedziczenie:	nie

Górne dopełnienie	
padding-top	
Dostępne wartości:	wartość liczbową initial inherit
Wartość domyślna:	0
Przykład:	p {padding-top:1cm;}
Dziedziczenie:	nie

Dolne dopełnienie	
padding-bottom	
Dostępne wartości:	wartość liczbową initial inherit
Wartość domyślna:	0
Przykład:	p {padding-bottom:1cm;}
Dziedziczenie:	nie

Zbiórny zapis właściwości dopełnienia	
padding	
Dostępne wartości:	<padding-top> <padding-right> <padding-bottom> <padding-left>
Wartość domyślna:	zależy od indywidualnych wartości poszczególnych stylów
Przykład:	p {margin:1cm 3cm;}
Dziedziczenie:	nie

OBRAMOWANIE

Styl obramowania	
Górne obramowanie	
border-top-style	
Dostępne wartości:	none hidden dotted dashed solid double dot-dash dot-dot-dash wave groove ridge inset outset initial inherit
Wartość domyślna:	none
Przykład:	p {border-top-style:dotted;}
Dziedziczenie:	nie

Prawe obramowanie	
border-right-style	
Dostępne wartości:	none hidden dotted dashed solid double dot-dash dot-dot-dash wave groove ridge inset outset initial inherit
Wartość domyślna:	none
Przykład:	p {border-right-style:dotted;}
Dziedziczenie:	nie

Dolne obramowanie	
border-bottom-style	
Dostępne wartości:	none hidden dotted dashed solid double dot-dash dot-dot-dash wave groove ridge inset outset initial inherit
Wartość domyślna:	none
Przykład:	p {border-bottom-style:dotted;}
Dziedziczenie:	nie

Lewe obramowanie	
border-left-style	
Dostępne wartości:	none hidden dotted dashed solid double dot-dash dot-dot-dash wave groove ridge inset outset initial inherit
Wartość domyślna:	none
Przykład:	p {border-left-style:dotted;}
Dziedziczenie:	nie

Zbiórny zapis stylu obramowania	
border-style	
Dostępne wartości:	none hidden dotted dashed solid double dot-dash dot-dot-dash wave groove ridge inset outset initial inherit
Wartość domyślna:	zależy od indywidualnych wartości poszczególnych stylów
Przykład:	p {border-style:dotted solid;}
Dziedziczenie:	nie

Szerokość obramowania	
Górne obramowanie	
border-top-width	
Dostępne wartości:	none wartości liczbowe wartości procentowe thin medium thick initial inherit
Wartość domyślna:	none
Przykład:	p {border-top-width:5px;}
Dziedziczenie:	nie

Prawe obramowanie	
border-right-width	
Dostępne wartości:	none wartości liczbowe wartości procentowe thin medium thick initial inherit
Wartość domyślna:	none
Przykład:	p {border-right-width:5px;}
Dziedziczenie:	nie

Dolne obramowanie	
border-bottom-width	
Dostępne wartości:	none wartości liczbowe wartości procentowe thin medium thick initial inherit
Wartość domyślna:	none
Przykład:	p {border-bottom-width:5px;}
Dziedziczenie:	nie

Lewe obramowanie	
border-left-width	
Dostępne wartości:	none wartości liczbowe wartości procentowe thin medium thick initial inherit
Wartość domyślna:	none
Przykład:	p {border-left-width:5px;}
Dziedziczenie:	nie

Zbiórny zapis szerokości obramowania	
border-width	
Dostępne wartości:	none wartości liczbowe wartości procentowe thin medium thick initial inherit
Wartość domyślna:	none
Przykład:	p {border-width:5px 2px;}
Dziedziczenie:	nie

Kolor obramowania	
Górne obramowanie	
border-top-color	
Dostępne wartości:	nazwa lub wartość koloru transparent initial inherit
Wartość domyślna:	według ustawień domyślnego koloru dla dokumentu
Przykład:	p {border-top-color:red;}
Dziedziczenie:	nie

Prawe obramowanie	
border-right-color	
Dostępne wartości:	nazwa lub wartość koloru transparent initial inherit
Wartość domyślna:	według ustawień domyślnego koloru dla dokumentu
Przykład:	p {border-right-color:red;}
Dziedziczenie:	nie

Dolne obramowanie	
border-bottom-color	
Dostępne wartości:	nazwa lub wartość koloru transparent initial inherit
Wartość domyślna:	według ustawień domyślnego koloru dla dokumentu
Przykład:	p {border-bottom-color:red;}
Dziedziczenie:	nie

Lewe obramowanie	
border-left-color	
Dostępne wartości:	nazwa lub wartość koloru transparent initial inherit
Wartość domyślna:	według ustawień domyślnego koloru dla dokumentu
Przykład:	p {border-left-color:red;}
Dziedziczenie:	nie
<ul style="list-style-type: none">• nazwa lub wartość koloru – nazwa własna, zapis RGB lub wartość szesnastkowa koloru;• transparent – niewidoczne obramowanie;• inherit – pobiera wartości z elementu nadrzędnego – rodzica.	

Zbiórny zapis koloru obramowania	
border-color	
Dostępne wartości:	nazwa lub wartość koloru transparent initial inherit
Wartość domyślna:	według ustawień domyślnego koloru dla dokumentu
Przykład:	p {border-color:red;}
Dziedziczenie:	nie

Zaokrąglanie narożników obramowania	
Lewy górny narożnik	
border-top-left-radius	
Dostępne wartości:	długość procent initial inherit
Wartość domyślna:	0
Przykład:	div { border-top-left-radius: 5px 10px;}
Dziedziczenie:	nie

Prawy górny narożnik	
border-top-right-radius	
Dostępne wartości:	długość procent initial inherit
Wartość domyślna:	0
Przykład:	div { border-top-right-radius: 5px 10px;}
Dziedziczenie:	nie

Lewy dolny narożnik	
border-bottom-left-radius	
Dostępne wartości:	długość procent initial inherit
Wartość domyślna:	0
Przykład:	div { border-bottom-left-radius: 5px 10px;}
Dziedziczenie:	nie

Prawy dolny narożnik	
border-bottom-right-radius	
Dostępne wartości:	długość procent initial inherit
Wartość domyślna:	0
Przykład:	div { border-bottom-right-radius: 5px 10px;}
Dziedziczenie:	nie

Zapis zbiórny	
border-radius	
Dostępne wartości:	liczba procent initial inherit
Wartość domyślna:	0
Przykład:	div { border-radius: 5px 10px;}
Dziedziczenie:	nie

Graficzne obramowanie	
Definiowane grafiki obramowania	
border-image-source	
Dostępne wartości:	none plik obrazka initial inherit
Wartość domyślna:	none
Przykład:	div {border-image-source: url(ramka.png)}
Dziedziczenie:	nie

Powielanie graficznego obramowania	
border-image-repeat	
Dostępne wartości:	stretch repeat round initial inherit
Wartość domyślna:	stretch
Przykład:	div {border-image-repeat: repeat;}
Dziedziczenie:	nie

Odległość obramowania od elementu	
border-image-outset	
Dostępne wartości:	długość wartość liczbową initial inherit
Wartość domyślna:	0
Przykład:	div {border-image-outset: 15px;}
Dziedziczenie:	nie

Podział obramowania na plastry	
border-image-slice	
Dostępne wartości:	wartość liczbową procent fill initial inherit
Wartość domyślna:	0
Przykład:	div {border-image-slice:20%;}
Dziedziczenie:	nie

Szerokość graficznego obramowania

border-image-width	
Dostępne wartości:	<i>wartość liczbową procent auto initial inherit</i>
Wartość domyślna:	0
Przykład:	<code>div {border-image-width: 10px;}</code>
Dziedziczenie:	nie

Zbiorzcy zapis graficznego obramowania

border-image	
Dostępne wartości:	<code><border-image-source> <border-image-slice> <border-image-width> <border-image-outset> <border-image-repeat></code>
Wartość domyślna:	none
Przykład:	<code>div {border-image:url (obrazek.png) 30 round;}</code>
Dziedziczenie:	nie

Zbiorzcy zapis właściwości obramowania

Górne obramowanie

border-top	
Dostępne wartości:	<code><border-style> <border-width> <border-color></code>
Wartość domyślna:	zależy od indywidualnych wartości poszczególnych stylów
Przykład:	<code>p {border-top:solid thick red;}</code>
Dziedziczenie:	nie

Prawe obramowanie

border-right	
Dostępne wartości:	<code><border-style> <border-width> <border-color></code>
Wartość domyślna:	zależy od indywidualnych wartości poszczególnych stylów
Przykład:	<code>p {border-right:solid thick red;}</code>
Dziedziczenie:	nie

Dolne obramowanie

border-bottom	
Dostępne wartości:	<code><border-style> <border-width> <border-color></code>
Wartość domyślna:	zależy od indywidualnych wartości poszczególnych stylów
Przykład:	<code>p {border-bottom:solid thick red;}</code>
Dziedziczenie:	nie

Lewe obramowanie

border-left	
Dostępne wartości:	<code><border-style> <border-width> <border-color></code>
Wartość domyślna:	zależy od indywidualnych wartości poszczególnych stylów
Przykład:	<code>p {border-left:solid thick red;}</code>
Dziedziczenie:	nie

Zbiorzcy zapis właściwości obramowania

border	
Dostępne wartości:	<code><border-style> <border-width> <border-color></code>

KOLOR I TŁO

KOLOR

Nazwa własna koloru

Do określania kolorów, podobnie jak w języku HTML, możemy użyć nazw kolorów, np. red, green czy blue. Niestety, nazwy zostały przypisane jedynie kilkunastu podstawowym kolorom.

Tabela 1. Wybrane kolory z przypisanymi nazwami wchodzące w skład bezpiecznej palety 216 kolorów

Nazwa koloru	Zapis szesnastkowy	Nazwa koloru	Zapis szesnastkowy
Black	#000000	Green	#008000
Silver	#C0C0C0	Lime	#90FF00
Gray	#808080	Olive	#808000
White	#FFFFFF	Yellow	#FFFF00
Maroon	#800000	Navy	#000080
Red	#FF0000	Blue	#0000FF
Purple	#800080	Teal	#008080
Fuchsia	#FF00FF	Aqua	#00FFFF

Przykładowy zapis definiujący kolor może przybrać następującą postać:

```
DIV {color:red;}
DIV {color:#FF0000;}
```

Wartość RGB

Zapis kolorów opierający się na modelu RGB przypisuje liczby całkowite z przedziału od 0 do 255 każdej wartości składowej palety RGB (ang. red, green, blue), stąd liczb może być aż dziewięć.

Zapis w postaci RGB może przybierać następującą postać:

```
DIV {color:rgb(255,0,0);}
```

Oczywiście możemy jeszcze zapisać kolor w postaci RGB, definiując procent nasycenia danej barwy. Niezbędne tutaj jest użycie wartości procentowych.

```
DIV {color:rgb(100%,0%,0);}
```

Wartość domyślna:	zależy od indywidualnych wartości poszczególnych stylów
Przykład:	<code>p {border:solid thick red;}</code>
Dziedziczenie:	nie

WYMIARY

Szerokość

width	
Dostępne wartości:	<i>wartość liczbową wartość procentową auto initial inherit</i>
Wartość domyślna:	auto
Przykład:	<code>p {width:30%;}</code>
Dziedziczenie:	nie

Minimalna szerokość

min-width	
Dostępne wartości:	<i>wartość liczbową wartość procentową initial inherit</i>
Wartość domyślna:	0
Przykład:	<code>p {min-width:90px;}</code>
Dziedziczenie:	nie

Maksymalna szerokość

max-width	
Dostępne wartości:	<i>wartość liczbową wartość procentową none initial inherit</i>
Wartość domyślna:	none
Przykład:	<code>p {max-width:100px;}</code>
Dziedziczenie:	nie

Wysokość

height	
Dostępne wartości:	<i>wartość liczbową wartość procentową auto initial inherit</i>
Wartość domyślna:	auto
Przykład:	<code>p {height:30%;}</code>
Dziedziczenie:	nie

Minimalna wysokość

min-height	
Dostępne wartości:	<i>wartość liczbową wartość procentową initial inherit</i>
Wartość domyślna:	0
Przykład:	<code>p {min-height:90px;}</code>
Dziedziczenie:	nie

Maksymalna wysokość

max-height	
Dostępne wartości:	<i>wartość liczbową wartość procentową none initial inherit</i>
Wartość domyślna:	none
Przykład:	<code>p {max-height:100px;}</code>
Dziedziczenie:	nie

Zmiana sposobu liczenia wymiarów pudełka

box-sizing	
Dostępne wartości:	content-box border-box initial inherit
Wartość domyślna:	content-box
Przykład:	<code>#przykład {box-sizing:border-box;}</code>
Dziedziczenie:	nie

Wartość szesnastkowa

Kolejnym sposobem zapisu kolorów jest metoda szesnastkowa, będąca tym samym co zapis szesnastkowy doskonale znany każdemu, kto miał do czynienia z komputerami.

DIV {color:#FF0000;}

Obsługa modelu barw HSL

Projekt trzeciej specyfikacji CSS wprowadza obsługę zapisu barwy według modelu HSL. Według definicji zamieszczonej w Wikipedii HSL to:

... jeden z modeli opisowych dla kolorów postrzeganych przez ludzi. Ten sposób opisowy miał polegać na tym, że każdej barwie postrzeganej przez człowieka jest przyporządkowany jeden punkt w przestrzeni trójwymiarowej identyfikowany przez trzy składowe: (h, s, l). Model pojawił się w okresie startu telewizji – pierwsze demonstracje w latach 1926 – 1930.

Znaczenie i zakres współrzędnych:

- *H: Hue* – (z ang. *odcien, barwa*) o wartościach z przedziału od 0 do 360 *stopni*,
- *S: Saturation* – (z ang. *nasycenie koloru*) z przedziału 0-1 albo 0–100%,
- *L: Lightness* – (z ang. *średnie światło białe*) z przedziału 0-1 albo 0–100%.

Zapis koloru czerwonego w modelu HSL stosowany w trzeciej specyfikacji CSS wygląda następująco:

```
BODY
{
  background-color:hsl(0,100%, 50%);
}
```

Obsługa modelu barw HSLA

HSLA jest rozwinięciem omawianego nieco wcześniej modelu barw HSL, który dodatkowo uzupełniono o obsługę przezroczystości.

Zapis koloru czerwonego w modelu HSL stosowany w trzeciej specyfikacji CSS wygląda następująco:

```
BODY
{
  background-color:hsla(0,100%,50%,0.2);
}
```

Jak widzisz, poza zapisem pierwszych trzech właściwości koloru na końcu dodano informację o stopniu przezroczystości. Stopień przezroczystości może przybierać wartość od 0 do 1 z krokiem co 0.1.

Obsługa modelu barw RGBA

RGBA jest rozwinięciem dobrze znanego modelu barw RGB, który dodatkowo uzupełniono o obsługę przezroczystości. Zapis koloru zielonego w modelu RGBA stosowany w trzeciej specyfikacji CSS wygląda następująco:

```
BODY
{
  background-color:rgba(0,255,0,0.2);
}
```

Jak widzisz, poza zapisem pierwszych trzech właściwości koloru na końcu dodano informację o stopniu przezroczystości. Stopień przezroczystości może przybierać wartość od 0 do 1 z krokiem co 0.1.

Kolor elementu

color	
Dostępne wartości:	<i>nazwa lub wartość koloru initial inherit</i>
Wartość domyślna:	zależy od domyślnych ustawień przeglądarki
Przykład:	<code>p {color:#FF0000;}</code>
Dziedziczenie:	tak

Przezroczystość elementu

opacity	
Dostępne wartości:	0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1 initial inherit
Wartość domyślna:	1
Przykład:	<code>p {opacity:0.2;}</code>
Dziedziczenie:	tak

Zmiana kolorystyki

filter	
Dostępne wartości:	none blur() brightness() contrast() drop-shadow() grayscale() hue-rotate() invert() opacity() saturate() sepia() url()
Wartość domyślna:	none
Przykład:	<code>img {filter:grayscale(100%);}</code>
Dziedziczenie:	nie

TŁO

Kolor tła

background-color	
Dostępne wartości:	<i>nazwa lub wartość koloru transparent initial inherit</i>
Wartość domyślna:	transparent
Przykład:	<code>p {background-color:#FF0000;}</code>
Dziedziczenie:	nie

Element graficzny jako tło

background-image	
Dostępne wartości:	url none initial inherit
Wartość domyślna:	none
Przykład:	<code>body {background-image:url (tlo.jpg);}</code>
Dziedziczenie:	nie

Zatrzymanie graficznego tła

background-attachment	
Dostępne wartości:	scroll fixed initial inherit
Wartość domyślna:	scroll
Przykład:	<code>body {background-attachment: fixed;}</code>
Dziedziczenie:	nie

Kontrola powieliania graficznego tła

background-repeat	
Dostępne wartości:	repeat repeat-x repeat-y no-repeat initial inherit
Wartość domyślna:	repeat
Przykład:	<code>body {background-repeat: repeat-x;}</code>
Dziedziczenie:	nie

Pozycjonowanie graficznego tła

background-position	
Dostępne wartości:	left right center top bottom <i>wartości liczbowe wartości procentowe initial inherit</i>
Wartość domyślna:	0% 0%
Przykład:	<code>body {background-position:center center;}</code>
Dziedziczenie:	nie

➔ Rozmiar tła

background-size	
Dostępne wartości:	<i>wartość bezwzględna wartość względna auto contain cover initial inherit</i>
Wartość domyślna:	auto
Przykład:	<code>body {background-size:50% 100%;}</code>
Dziedziczenie:	nie

Kontrola styku tła i obramowania

background-origin	
Dostępne wartości:	padding-box border-box content-box initial inherit
Wartość domyślna:	padding-box
Przykład:	<code>div {background: url (papier.gif); background-repeat: no-repeat; background-origin: content-box;}</code>
Dziedziczenie:	nie

Kontrola przycięcia tła

background-clip	
Dostępne wartości:	padding-box border-box content-box box initial inherit
Wartość domyślna:	border-box
Przykład:	<code>div {background-clip: border-box;}</code>
Dziedziczenie:	nie

Zbiorzcy zapis właściwości tła

background	
Dostępne wartości:	<code><background-color> <background-image> <background-repeat> <background-attachment> <background-position> <background-origin> <background-clip> <background-size></code>
Wartość domyślna:	zależna od wartości poszczególnych opcji
Przykład:	<code>body {background:#E7E7E7 url(tlo1.jpg) no-repeat 15% 15%;}</code>
Dziedziczenie:	nie

Tło wieloelementowe

Jedną z ciekawszych nowości dostępnych w trzeciej specyfikacji CSS jest możliwość umieszczenia kilku fragmentów tła graficznego dla tego samego elementu. Operacja nie jest skomplikowana i może wyglądać tak:

```
DIV
{
  width:500px;
  height:500px;
  background-color:#C0FFC0;
  background-image: url(kolo.jpg),
    url(kula.jpg);
  background-position: top left,
    bottom right;
  background-repeat: repeat-x, no-repeat;
}
```

Zwróć uwagę na pogrubione elementy listingu. Korzystamy z tych samych stylów, które omawiałem przy okazji wstawiania pojedynczego tła, ale po przecinku dopisujemy kolejne wartości.

Cień

box-shadow	
Dostępne wartości:	none h-offset v-offset blur spread color inset initial inherit
Wartość domyślna:	none
Przykład:	<code>#cien {box-shadow:5px 10px #888888;}</code>
Dziedziczenie:	nie

Przenikanie warstw tła

background-blend-mode	
Dostępne wartości:	normal multiply screen overlay darken lighten color-dodge saturation color luminosity initial inherit
Wartość domyślna:	normal
Przykład:	<code>div {background-blend-mode:lighten;}</code>
Dziedziczenie:	nie

TEKST, DEKORACJA I KIERUNEK ZAPISU

POZIOME WYRÓWNANIE TEKSTU	
text-align	
Dostępne wartości:	left right center justify initial inherit
Wartość domyślna:	left
Przykład:	p {text-align:left;}
Dziedziczenie:	tak
WCIĘCIE TEKSTU	
text-indent	
Dostępne wartości:	wartość liczbową wartość procentową initial inherit
Wartość domyślna:	0
Przykład:	p {text-indent:3pt;}
Dziedziczenie:	tak
PRZEKSZTAŁCANIE TEKSTU	
text-transform	
Dostępne wartości:	capitalize uppercase lowercase none initial inherit
Wartość domyślna:	none
Przykład:	p {text-transform:uppercase;}
Dziedziczenie:	tak
DEKORACJA TEKSTU	
text-decoration	
Dostępne wartości:	none underline overline line-through blink initial inherit
Wartość domyślna:	none
Przykład:	p {text-decoration:underline;}
Dziedziczenie:	nie
ODSTĘPY POMIĘDZY LITERAMI	
letter-spacing	
Dostępne wartości:	normal długość initial inherit
Wartość domyślna:	normal
Przykład:	p {letter-spacing:3pt;}
Dziedziczenie:	tak
ODSTĘPY POMIĘDZY WYRAZAMI	
word-spacing	
Dostępne wartości:	normal długość initial inherit
Wartość domyślna:	normal
Przykład:	p {word-spacing:3pt;}
Dziedziczenie:	tak
ODSTĘP MIĘDZY LINIAMI	
line-height	
Dostępne wartości:	normal wielokrotność wysokości czcionki procent wysokości czcionki initial inherit
Wartość domyślna:	normal
Przykład:	p {line-height:100%;}
Dziedziczenie:	tak
PUSTA PRZESTRZEŃ	
white-space	
Dostępne wartości:	normal pre nowrap pre-wrap pre-line initial inherit
Wartość domyślna:	normal
Przykład:	p {white-space:nowrap;}
Dziedziczenie:	tak
KIERUNEK TEKSTU	
direction	
Dostępne wartości:	ltr rtl initial inherit
Wartość domyślna:	ltr
Przykład:	p {direction:rtl;}
Dziedziczenie:	tak
ORIENTACJA TEKSTU DLA ELEMENTÓW ZAGNIEŻDŻONYCH	
unicode-bidi	
Dostępne wartości:	normal embed bidi-override initial inherit
Wartość domyślna:	normal
Przykład:	p {unicode-bidi:embed;}
Dziedziczenie:	nie
ŁAMANIE DŁUGICH CIĄGÓW ZNAKÓW	
word-wrap	
Dostępne wartości:	normal break-word initial inherit

Wartość domyślna:	normal
Przykład:	p {word-wrap:break-word;}
Dziedziczenie:	tak
REGUŁY ŁAMANIA WYRAZÓW	
word-break	
Dostępne wartości:	normal break-all keep-all initial inherit
Wartość domyślna:	normal
Przykład:	p {word-break:break-all;}
Dziedziczenie:	tak
PODZIAŁ W MIEJSCU WYSTĘPOWANIA DWYWIZU	
hyphens	
Dostępne wartości:	manual auto none
Wartość domyślna:	manual
Przykład:	p {hyphens:auto;}
Dziedziczenie:	tak
PIONOWE WYRÓWNANIE TEKSTU	
vertical-align	
Dostępne wartości:	baseline sub super bottom text-bottom middle top text-top wartość liczbową wartość procentową initial inherit
Wartość domyślna:	baseline
Przykład:	p {vertical-align:super;}
Dziedziczenie:	nie
WYRÓWNANIE OSTATNIEGO WIESZA TEKSTU	
text-align-last	
Dostępne wartości:	auto left right center justify start end initial inherit
Wartość domyślna:	auto
Przykład:	p {text-align-last:right;}
Dziedziczenie:	tak
KOMBINACJA WIELU ZNAKÓW W PRZESTRZENI JEDNEGO ZNAKU TYPOGRAFICZNEGO	
text-combine-upright	
Dostępne wartości:	none all
Wartość domyślna:	none
Przykład:	p {text-combine-upright:all;}
Dziedziczenie:	tak
KOLOR DEKORACJI TEKSTU	
text-decoration-color	
Dostępne wartości:	kolor initial inherit
Wartość domyślna:	domyślny kolor
Przykład:	p {text-decoration-color:green;}
Dziedziczenie:	nie
POŁOŻONE LINII DEKORACJI	
text-decoration-line	
Dostępne wartości:	none underline overline line-through initial inherit
Wartość domyślna:	none
Przykład:	p {text-decoration-line:overline;}
Dziedziczenie:	nie
WYGLĄD LINII DEKORACJI	
text-decoration-style	
Dostępne wartości:	solid double dotted dashed wavy initial inherit
Wartość domyślna:	solid
Przykład:	p {text-decoration-style:overline;}
Dziedziczenie:	nie
SPOSÓB JUSTOWANIA TEKSTU	
text-justify	
Dostępne wartości:	auto inter-word inter-ideograph inter-cluster distribute kashida trim initial inherit
Wartość domyślna:	auto
Przykład:	p {text-justify:inter-word;}
Dziedziczenie:	tak
KIERUNEK TEKSTU	
text-orientation	
Dostępne wartości:	mixed upright sideways
Wartość domyślna:	mixed
Przykład:	p {text-orientation:sideways;}
Dziedziczenie:	tak

CIĘN TEKSTU	
text-shadow	
Dostępne wartości:	h-shadow v-shadow blur-radius color none initial inherit
Wartość domyślna:	none
Przykład:	p {text-shadow: 2px 2px 5px #ff0000;}
Dziedziczenie:	tak
ROZMIAR ZNAKU TABULACJI	
tab-size	
Dostępne wartości:	liczba długość initial inherit
Wartość domyślna:	8
Przykład:	p {tab-size:12px;}
Dziedziczenie:	tak
DODATKOWY SYMBOL PRZY TEKŚCIE	
text-emphasis-style	
Dostępne wartości:	none [[filled open] [dot circle double-circle triangle sesame]] znak lub symbol
Wartość domyślna:	none
Przykład:	p {text-emphasis-style:'^';}
Dziedziczenie:	tak
CZCIONKA	
RODZAJ CZCIONKI	
font-family	
Dostępne wartości:	nazwa czcionki serif sans-serif monospace fantasy cursive initial inherit
Wartość domyślna:	domyślne ustawienie przeglądarki użytkownika oglądającego witrynę
Przykład:	p {font-family:arial, helvetica, verdana, sans-serif;}
Dziedziczenie:	tak
ROZMIAR CZCIONKI	
font-size	
Dostępne wartości:	rozmiar absolutny rozmiar względny xx-small x-small small medium large x-large xx-large smaller larger initial inherit
Wartość domyślna:	medium
Przykład:	p {font-size:20pt;}
Dziedziczenie:	tak
WAGA CZCIONKI	
font-weight	
Dostępne wartości:	normal bold 100 200 300 400 500 600 700 800 900 bolder lighter initial inherit
Wartość domyślna:	normal
Przykład:	p {font-weight:bold;}
Dziedziczenie:	tak
STYL CZCIONKI	
font-style	
Dostępne wartości:	normal italic oblique initial inherit
Wartość domyślna:	normal
Przykład:	p {font-style:italic;}
Dziedziczenie:	tak
WARIANTY CZCIONKI	
font-variant	
Dostępne wartości:	normal small-caps initial inherit
Wartość domyślna:	normal
Przykład:	p {font-variant:small-caps;}
Dziedziczenie:	tak

KOLOR DODATKOWEGO SYMBOLU PRZY TEKŚCIE	
text-emphasis-color	
Dostępne wartości:	kolor
Wartość domyślna:	domyślny kolor
Przykład:	p {text-emphasis-color:red;}
Dziedziczenie:	tak
POZYCJA DODATKOWEGO SYMBOLU PRZY TEKŚCIE	
text-emphasis-position	
Dostępne wartości:	[over under] && [right left]
Wartość domyślna:	over right
Przykład:	p {text-emphasis-position:under left;}
Dziedziczenie:	tak
ZBIORCZY ZAPIS USTAWIEŃ DODATKOWEGO SYMBOLU PRZY TEKŚCIE	
text-emphasis	
Dostępne wartości:	<text-emphasis-style> <text-emphasis-color>
Wartość domyślna:	-
Przykład:	p {text-emphasis: filled sesame #555;}
Dziedziczenie:	tak
ROZMIESZCZENIE LITER	
font-kerining	
Dostępne wartości:	auto normal none
Wartość domyślna:	Auto
Przykład:	p {font-kerining: normal;}
Dziedziczenie:	tak
DOPASOWANIE ROZMIARU CZCIONKI NIEZALEŻNIE OD JEJ RODZAJU	
font-size-adjust	
Dostępne wartości:	wartość liczbową none initial inherit
Wartość domyślna:	none
Przykład:	p {font-size-adjust:0.58;}
Dziedziczenie:	tak
STOPIEŃ KONDENSACJI CZCIONKI	
font-stretch	
Dostępne wartości:	ultra-condensed extra-condensed condensed semi-condensed normal semi-expanded expanded extra-expanded ultra-expanded initial inherit
Wartość domyślna:	normal
Przykład:	p {font-stretch:expanded;}
Dziedziczenie:	tak
ZBIORCZY ZAPIS WŁAŚCIWOŚCI CZCIONEK	
font	
Dostępne wartości:	styl> <font-variant> <font-weight> <font-size> <line-height> <font-family>
Wartość domyślna:	patrz na właściwości poszczególnych stylów
Przykład:	p {font:italic small-caps 600 14pt/2 arial;}
Dziedziczenie:	tak
Gdy analizuje się przykładowy zbiorczy styl, od razu widać, że poszczególnych wartości nie rozdzielają żadne dodatkowe znaki – przecinki, myślniki itp. Jedynym wyjątkiem są wielkość czcionki oraz odstęp linii; pomiędzy tymi wartościami występuje /.	
ZAWARTOŚĆ GENEROWANA I LICZNIKI	
REGUŁA CONTENT	
content	
Dostępne wartości:	normal none counter attr ['ciąg znaków' open-quote close-quote no-open-quote no-close-quote 'url' initial inherit
Wartość domyślna:	normal
Przykład:	h1:after {content:"informacja widoczna przed nagłówkiem";}
Dziedziczenie:	nie
CYTOWANIE	
quotes	
Dostępne wartości:	none "znak" "znak" initial inherit
Wartość domyślna:	wg ustawień przeglądarki
Przykład:	q:before {quotes:"<<" ">>";}
Dziedziczenie:	tak

LICZNIKI	
Resetowanie licznika	
counter-reset	
Dostępne wartości:	none <i>nazwa licznika</i> initial inherit
Wartość domyślna:	none
Przykład:	BODY {counter-reset:licznik;}
Dziedziczenie:	nie

Określanie wielkości wzrostu wartości

counter-increment	
Dostępne wartości:	none nazwa licznika wartość wzrostu initial inherit
Wartość domyślna:	none
Przykład:	h1:before {counter-increment: licznik 3;}
Dziedziczenie:	nie

LISTY

TYP LISTY	
list-style-type	
Dostępne wartości:	disc circle square decimal decimal-leading-zero lower-roman upper-roman lower-greek lower-latin upper-latin armenian georgian lower-alpha upper-alpha none initial inherit
Wartość domyślna:	disc
Przykład:	ul {list-style-type:circle;}
Dziedziczenie:	tak

POZYCJA LISTY WZGLĘDEM WYPUNKTOWANIA	
list-style-position	
Dostępne wartości:	inside outside inherit
Wartość domyślna:	outside
Przykład:	ul {list-style-position: inside;}
Dziedziczenie:	tak

ZBIORCZY ZAPIS WŁAŚCIWOŚCI LIST	
list-style	
Dostępne wartości:	<list-style-type> <list-style-position> <list-style-image>
Wartość domyślna:	–
Przykład:	ul {list-style:url (punktor.gif) inside;}
Dziedziczenie:	–

SPOSÓB WYPUNKTOWANIE LISTY	
list-style-image	
Dostępne wartości:	url none inherit
Wartość domyślna:	none
Przykład:	ul {list-style-image: url(punkto.gif);}
Dziedziczenie:	tak

TABELE

PODPIS TABELI	
caption-side	
Dostępne wartości:	top bottom initial inherit
Wartość domyślna:	top
Przykład:	table {caption-side:bottom;}
Dziedziczenie:	tak

Wartość domyślna:	separate
Przykład:	table {border-collapse: collapse;}
Dziedziczenie:	tak

GENEROWANIE TABELI	
table-layout	
Dostępne wartości:	auto fixed initial inherit
Wartość domyślna:	auto
Przykład:	table {table-layout:fixed;}
Dziedziczenie:	nie

ODSTĘPY WE WNĘTRZU TABELI

border-spacing	
Dostępne wartości:	wartość liczbową initial inherit
Wartość domyślna:	0
Przykład:	table {border-spacing:1cm 2cm;}
Dziedziczenie:	tak

POJEDYNCZE OBRAMOWANIE	
border-collapse	
Dostępne wartości:	collapse separate initial inherit

KONTROLA PUSTYCH KOMÓREK	
empty-cells	
Dostępne wartości:	show hide initial inherit
Wartość domyślna:	show
Przykład:	table {empty-cells:hide;}
Dziedziczenie:	tak

KOLUMNY

LICZBA KOLUMN	
column-count	
Dostępne wartości:	wartość liczbową auto initial inherit
Wartość domyślna:	auto
Przykład:	div {column-count:3;}
Dziedziczenie:	nie

STYL LINII ROZDZIAŁAJĄCEJ KOLUMNY	
column-rule-style	
Dostępne wartości:	none hidden dotted dashed solid double groove ridge inset outset initial inherit
Wartość domyślna:	none
Przykład:	div {column-rule-style: dotted;}
Dziedziczenie:	nie

WYPEŁNIANIE TREŚCIĄ	
column-fill	
Dostępne wartości:	balance auto initial inherit
Wartość domyślna:	balance
Przykład:	div {column-fill:auto;}
Dziedziczenie:	nie

GRUBOŚĆ LINII ROZDZIAŁAJĄCEJ KOLUMNY	
column-rule-width	
Dostępne wartości:	medium thin thick wartość liczbową initial inherit
Wartość domyślna:	medium
Przykład:	div {column-rule-width:10px;}
Dziedziczenie:	nie

ODSTĘP POMIĘDZY KOLUMNAMI	
column-gap	
Dostępne wartości:	długość normal initial inherit
Wartość domyślna:	normal
Przykład:	div {column-gap:15px}
Dziedziczenie:	nie

ZBIORCZY ZAPIS WŁAŚCIWOŚCI LINII ROZDZIAŁAJĄCYCH KOLUMNY	
column-rule	
Dostępne wartości:	<column-rule-width> <column-rule-style> <column-rule-color> initial inherit
Wartość domyślna:	medium none color
Przykład:	div {column-rule: 4px outset #ff00ff;}
Dziedziczenie:	nie

KOLOR LINII ROZDZIAŁAJĄCYCH KOLUMNY	
column-rule-color	
Dostępne wartości:	<i>kolor</i> initial inherit
Wartość domyślna:	-
Przykład:	div {column-rule-color:red;}
Dziedziczenie:	nie

ROZCIĄGANIE ELEMENTU Z KOLUMNAMI	
column-span	
Dostępne wartości:	none all initial inherit
Wartość domyślna:	none
Przykład:	div {column-span:all;}
Dziedziczenie:	nie

ZBIORCZY ZAPIS WŁAŚCIWOŚCI KOLUMN	
columns	
Dostępne wartości:	auto <column-width> <column-count> initial inherit
Wartość domyślna:	auto auto
Przykład:	div {columns:150px 4;}
Dziedziczenie:	nie

SUGEROWANA SZEROKOŚĆ KOLUMNY	
column-width	
Dostępne wartości:	auto wartość liczbową initial inherit
Wartość domyślna:	auto
Przykład:	div {column-width:100px;}
Dziedziczenie:	nie

WYMUSZENIE PODZIAŁU PRZED	
break-before	
Dostępne wartości:	auto avoid avoid-page page left right recto verso avoid-column column avoid-region region
Wartość domyślna:	auto
Przykład:	div {break-before: avoid-region;}
Dziedziczenie:	nie

WYMUSZENIE PODZIAŁU PO	
break-after	
Dostępne wartości:	auto avoid avoid-page page left right recto verso avoid-column column avoid-region region
Wartość domyślna:	auto
Przykład:	div {break-after: avoid-region;}
Dziedziczenie:	nie

WYMUSZENIE PODZIAŁU WEWNĄTRZ	
break-inside	
Dostępne wartości:	auto avoid avoid-page avoid-column avoid-region
Wartość domyślna:	auto
Przykład:	div {break-inside: avoid-region;}
Dziedziczenie:	nie

PRZEKSZTAŁCANIE

ZMIANA POŁOŻENIA PRZEKSZTAŁCANYCH ELEMENTÓW	
transform-origin	
Dostępne wartości:	x-axis y-axis z-axis initial inherit
Wartość domyślna:	50% 50% 0
Przykład:	div {transform-origin: 20% 40%;}
Dziedziczenie:	nie

Dostępne wartości:	skew(x-angle,y-angle) skew(angle) skewY(angle) perspective(n) initial inherit
Wartość domyślna:	none
Przykład:	div {transform: rotate(7deg);}
Dziedziczenie:	nie

OKREŚLANIE SPOSOBU RENDEROWANIA ELEMENTÓW W PRZESTRZENI 3D	
transform-style	
Dostępne wartości:	flat preserve-3d initial inherit
Wartość domyślna:	flat
Przykład:	div {transform-style: preserve-3d;}
Dziedziczenie:	nie

WIDOCZNOŚĆ PRZEKSZTAŁCANEGO ELEMENTU OD TYŁU	
backface-visibility	
Dostępne wartości:	visible hidden initial inherit
Wartość domyślna:	visible
Przykład:	div {backface-visibility: hidden;}
Dziedziczenie:	nie

STEROWANIE ZACHOWANIEM PRZEKSZTAŁCANYCH ELEMENTÓW

PERSPEKTYWA WIDOKU	
perspective	
Dostępne wartości:	Liczba pikseli none
Wartość domyślna:	none
Przykład:	div {perspective: 200px;}
Dziedziczenie:	nie

transform	
Dostępne wartości:	none matrix(n,n,n,n,n,n) matrix3d (n,n,n,n,n,n,n,n,n,n,n,n,n,n) translate(x,y) translate3d(x,y,z) translateX(x) translateY(y) translateZ(z) scale(x,y) scale3d(x,y,z) scaleX(x) scaleY(y) scaleZ(z) rotate(angle) rotate3d(x- y,z,angle) rotateX(angle) rotateY(angle) rotateZ(angle)

PUNKT OPARCIA ELEMENTU NA OSI X I Y	
backface-visibility	
Dostępne wartości:	x-axis y-axis initial inherit
Wartość domyślna:	50% 50%
Przykład:	div {perspective-origin: 8% 8%;}
Dziedziczenie:	nie

PRZEJŚCIA

OPÓŹNIENIE ROZPOCZĘCIA PRZEJŚCIA	
transition-delay	
Dostępne wartości:	czas initial inherit
Wartość domyślna:	0s
Przykład:	div {transition: width 2s;}
Dziedziczenie:	nie

TEMPO PRZEJŚCIA	
transition-timing-function	
Dostępne wartości:	linear ease ease-in ease-out ease-in-out step-start step-end steps(int,start end) cubic-bezier(<i>n,n,n,n</i>) initial inherit
Wartość domyślna:	easy
Przykład:	div {transition-timing- function: linear;}
Dziedziczenie:	nie

CZAS TRWANIA PRZEJŚCIA	
transition-duration	
Dostępne wartości:	czas initial inherit
Wartość domyślna:	0s
Przykład:	div {transition-duration: 5s;}
Dziedziczenie:	nie

ZBIORCZY ZAPIS CECH PRZEJŚCIA	
transition	
Dostępne wartości:	<transition-property> <transition-duration> <transition-timing-function> <transition-delay> initial inherit
Wartość domyślna:	all 0s ease 0s
Przykład:	div {transition: width 2s;}
Dziedziczenie:	nie

WŁAŚCIWOŚCI PRZEJŚCIA	
transition-property	
Dostępne wartości:	none all <i>właściwość</i> initial inherit
Wartość domyślna:	all
Przykład:	div {transition-property: width;}
Dziedziczenie:	nie

UKŁAD ELASTYCZNY

WYMIAR ELEMENTU ELASTYCZNEGO	
flex-basis	
Dostępne wartości:	numer auto initial inherit
Wartość domyślna:	auto
Przykład:	div {flex-basis: 100px;}
Dziedziczenie:	nie
KIERUNEK UGIĘCIA ELEMENTU ELASTYCZNEGO	
flex-direction	
Dostępne wartości:	row row-reverse column column-reverse initial inherit
Wartość domyślna:	row
Przykład:	div {flex-direction: row-reverse;}
Dziedziczenie:	nie
ZAWIJANIE ELASTYCZNEGO ELEMENTU	
flex-wrap	
Dostępne wartości:	nowrap wrap wrap-reverse initial inherit
Wartość domyślna:	nowrap
Przykład:	div {flex-wrap: wrap;}
Dziedziczenie:	nie
SKRÓCONY ZAPIS WŁAŚCIWOŚCI FLEX-DIRECTION ORAZ FLEX-WRAP	
flex-flow	
Dostępne wartości:	<flex-direction> <flex-wrap> initial inherit
Wartość domyślna:	row nowrap

Przykład:	div {flex-flow: row-reverse wrap;}
Dziedziczenie:	nie

STOPIEŃ POWIĘKSZENIA WZGLĘDEM INNYCH ELEMENTÓW TEGO SAMEGO KONTENERA

flex-grow	
Dostępne wartości:	Wartość liczbową initial inherit
Wartość domyślna:	0
Przykład:	div {flex-grow: 1;}
Dziedziczenie:	nie

STOPIEŃ ZMNIEJSZENIA WZGLĘDEM INNYCH ELEMENTÓW TEGO SAMEGO KONTENERA

flex-shrink	
Dostępne wartości:	Wartość liczbową initial inherit
Wartość domyślna:	1
Przykład:	div {flex-shrink: 2;}
Dziedziczenie:	nie

ZBIORCZY ZAPIS WŁAŚCIWOŚCI ELASTYCZNEGO ELEMENTU

flex	
Dostępne wartości:	<flex-grow> <flex-shrink> <flex-basis> auto initial inherit
Wartość domyślna:	0 1 auto
Przykład:	div {flex:auto;}
Dziedziczenie:	nie

ANIMACJA

OPÓŹNIANIE STARTU	
animation-delay	
Dostępne wartości:	time initial inherit
Wartość domyślna:	0s
Przykład:	div {animation-delay: 5s;}
Dziedziczenie:	nie
OKREŚLANIE KIERUNKU	
animation-direction	
Dostępne wartości:	normal reverse alternate alternate-reverse initial inherit
Wartość domyślna:	normal
Przykład:	div {animation-direction: reverse;}
Dziedziczenie:	nie
CZAS TRWANIA	
animation-duration	
Dostępne wartości:	time initial inherit
Wartość domyślna:	0
Przykład:	div {animation-duration: 5s;}
Dziedziczenie:	nie
OKREŚLANIE DODATKOWYCH WŁAŚCIWOŚCI	
animation-fill-mode	
Dostępne wartości:	none forwards backwards both initial inherit
Wartość domyślna:	None
Przykład:	div {animation-fill-mode: 5s;}
Dziedziczenie:	nie
LICZBA POWTÓRZEŃ ANIMACJI	
animation-iteration-count	
Dostępne wartości:	liczba infinite initial inherit
Wartość domyślna:	1
Przykład:	div {animation-iteration-count: 10;}
Dziedziczenie:	nie

NAZWA ANIMACJI	
animation-name	
Dostępne wartości:	keyframeName none initial inherit
Wartość domyślna:	none
Przykład:	div {animation-name:mojaanimacja;}
Dziedziczenie:	nie

URUCHAMIANIE LUB ZATRZYMYWANIE ANIMACJI

animation-play-state	
Dostępne wartości:	paused running initial inherit
Wartość domyślna:	running
Przykład:	div {animation-play-state: paused;}
Dziedziczenie:	nie

OKREŚLANIE TEMP

animation-timing-function	
Dostępne wartości:	linear ease ease-in ease-out ease-in-out step-start step-end steps(int,start end) cubic-bezier(<i>n</i> , <i>n</i> , <i>n</i> , <i>n</i>) initial inherit
Wartość domyślna:	easy
Przykład:	div {animation-timing-function: linear;}
Dziedziczenie:	nie

ZBIORCZY ZAPIS

animation	
Dostępne wartości:	<animation-delay> <animation-direction> <animation-duration> <animation-fill-mode> <animation-iteration-count> <animation-name> <animation-play-state> <animation-timing-function>
Wartość domyślna:	-
Przykład:	div {animation: mojaanimacja 4s 10}
Dziedziczenie:	nie

SIATKA

ROZMIAR WIERZSY SIATKI	
grid-template-rows	
Dostępne wartości:	none auto max-content min-content length initial inherit
Wartość domyślna:	none
Przykład:	.siatka {grid-template-rows: 100px 300px;}
Dziedziczenie:	nie

LICZBA I ROZMIAR KOLUMN

grid-template-columns	
Dostępne wartości:	none auto max-content min-content długość initial inherit
Wartość domyślna:	none
Przykład:	.siatka {grid-template-columns:auto auto auto auto;}
Dziedziczenie:	nie

NAZWY OBSZARÓW NA SIATCE

grid-template-areas	
Dostępne wartości:	none nazwa własna
Wartość domyślna:	none
Przykład:	.siatka {grid-template-areas:nazwa nazwa;}
Dziedziczenie:	nie

ZBIORCZY ZAPIS GRID-TEMPLATE

grid-template	
Dostępne wartości:	none <grid-template-rows> / <grid-template-columns> <grid-template-areas> initial inherit
Wartość domyślna:	none
Przykład:	.siatka {grid-template: 150px / auto auto auto;}
Dziedziczenie:	nie

DOMYŚLNA WYSOKOŚĆ WIERZSY

grid-auto-rows	
Dostępne wartości:	none auto max-content min-content wysokość initial inherit
Wartość domyślna:	auto
Przykład:	.siatka {grid-auto-rows:300px;}
Dziedziczenie:	nie

DOMYŚLNA SZEROKOŚĆ KOLUMN

grid-auto-columns	
Dostępne wartości:	none auto max-content min-content szerokość initial inherit
Wartość domyślna:	auto
Przykład:	.siatka {grid-auto-columns:100px;}
Dziedziczenie:	nie

DOMYŚLNY SPOŚÓB UKŁADANIA SIĘ ELEMENTÓW SIATKI

grid-auto-flow	
Dostępne wartości:	row columns row dense column dense
Wartość domyślna:	row
Przykład:	.siatka {grid-auto-flow: column;}
Dziedziczenie:	nie

OKREŚLANIE PIERWSZEGO WIERZSY SIATKI

grid-row-start	
Dostępne wartości:	auto numer wiersza
Wartość domyślna:	auto
Przykład:	.siatka {grid-row-start:2;}
Dziedziczenie:	nie

OKREŚLANIE OSTATNIEGO WIERZSY SIATKI

grid-row-end	
Dostępne wartości:	auto numer wiersza
Wartość domyślna:	auto
Przykład:	.siatka {grid-row-start:4;}
Dziedziczenie:	nie

POZYCJONOWANIE

RODZAJ POZYCJONOWANIA	
position	
Dostępne wartości:	static relative absolute fixed initial inherit
Wartość domyślna:	static
Przykład:	p {position:absolute;}
Dziedziczenie:	nie

DODATKOWY ODSTĘP POMIĘDZY WIERSZAMI	
grid-row-gap	
Dostępne wartości:	odstęp
Wartość domyślna:	0
Przykład:	.siatka {grid-row-gap:30px;}
Dziedziczenie:	nie

ZBIORCZY ZAPIS WŁAŚCIWOŚCI WIERZSY

grid-row	
Dostępne wartości:	<grid-row-start> / <grid-row-end>
Wartość domyślna:	auto / auto
Przykład:	.siatka {grid-row:3 /r9;}
Dziedziczenie:	nie

OKREŚLANIE PIERWSZEJ KOLUMNY SIATKI

grid-column-start	
Dostępne wartości:	auto numer kolumny
Wartość domyślna:	auto
Przykład:	.siatka {grid-column-start:4;}
Dziedziczenie:	nie

OKREŚLANIE OSTATNIEJ KOLUMNY SIATKI

grid-column-end	
Dostępne wartości:	auto numer kolumny
Wartość domyślna:	auto
Przykład:	.siatka {grid-column-end:8;}
Dziedziczenie:	nie

DODATKOWY ODSTĘP POMIĘDZY KOLUMNAMI

grid-column-gap	
Dostępne wartości:	odstęp
Wartość domyślna:	0
Przykład:	.siatka {grid-column-gap:20px;}
Dziedziczenie:	nie

ZBIORCZY ZAPIS USTAWIEŃ KOLUMN

grid-column	
Dostępne wartości:	<grid-column-start> / <grid-column-end>
Wartość domyślna:	auto / auto
Przykład:	.siatka {grid-column:1 / span 2 }
Dziedziczenie:	nie

ZBIORCZA DEFINICJA PIERWSZEGO I OSTATNIEGO WIERZSA LUB KOLUMNY

grid-area	
Dostępne wartości:	<grid-row-start> / <grid-columns-start> / <grid-row-end> / <grid-column-end>
Wartość domyślna:	auto / auto / auto / auto
Przykład:	.siatka {grid-area: 2 / 1 / span 2 / span 3 }
Dziedziczenie:	nie

DODATKOWY ODSTĘP POMIĘDZY WIERSZAMI I KOLUMNAMI

grid-gap	
Dostępne wartości:	odstęp
Wartość domyślna:	0 0
Przykład:	.siatka {grid-gap:20px 50px;}
Dziedziczenie:	nie

ZBIORCZY ZAPIS WŁAŚCIWOŚCI SIATKI

grid	
Dostępne wartości:	none <grid-template-rows> / <grid-template-columns> <grid-template-areas> <grid-template-rows> / <grid-auto-flow> <grid-auto-columns> <grid-auto-flow> <grid-auto-rows> / <grid-template-columns> initial inherit
Wartość domyślna:	-
Przykład:	.siatka {grid:150px / auto auto auto;}
Dziedziczenie:	nie

OKREŚLANIE POZYCJI

Względem górnej krawędzi

top	
Dostępne wartości:	auto wartość liczbową wartość procentową initial inherit
Wartość domyślna:	auto

Przykład:	p {position:absolute; top:10px;}
Dziedziczenie:	nie

Względem prawej krawędzi

right	
Dostępne wartości:	auto wartość liczbową wartość procentową initial inherit
Wartość domyślna:	auto
Przykład:	p {position:absolute; right:10px;}
Dziedziczenie:	nie

Względem dolnej krawędzi

bottom	
Dostępne wartości:	auto wartość liczbową wartość procentową initial inherit
Wartość domyślna:	auto
Przykład:	p {position:absolute; bottom:10px;}
Dziedziczenie:	nie

Względem lewej krawędzi

left	
Dostępne wartości:	auto wartość liczbową wartość procentową initial inherit
Wartość domyślna:	auto

Przykład:	p {position:absolute; left:10px;}
Dziedziczenie:	nie

PŁYWIANIE ELEMENTÓW

float	
Dostępne wartości:	left right none initial inherit
Wartość domyślna:	none
Przykład:	img {float:right;}
Dziedziczenie:	nie

TAMOWANIE ELEMENTÓW

clear	
Dostępne wartości:	left right none both initial inherit
Wartość domyślna:	none
Przykład:	img {clear:right;}
Dziedziczenie:	nie

KOLEJNOŚĆ NAKŁADANYCH ELEMENTÓW

z-index	
Dostępne wartości:	auto <i>wartość liczbową określającą pozycję w stosie</i> initial inherit
Wartość domyślna:	auto
Przykład:	p {z-index:2;}
Dziedziczenie:	nie

PUDEŁKOWY MODEL FORMATOWANIA ELEMENTÓW

STEROWANIE WYMIAROWANYMI ELEMENTAMI

overflow	
Dostępne wartości:	visible hidden scroll auto initial inherit
Wartość domyślna:	visible
Przykład:	p {overflow:scroll;}
Dziedziczenie:	nie

STEROWANIE WYMIAROWANYMI ELEMENTAMI – PRAWA KRAWĘDŹ

overflow-x	
Dostępne wartości:	visible hidden scroll auto initial inherit
Wartość domyślna:	visible
Przykład:	p {overflow-x:scroll;}
Dziedziczenie:	nie

EFEKTY WIZUALNE

KADROWANIE

clip	
Dostępne wartości:	rect (top, right, bottom, left) auto initial inherit
Wartość domyślna:	auto
Przykład:	p {clip:rect(10px, 150px, 150px, 10px,)} p {clip:rect(10px, 1

UKRYWANIE ELEMENTÓW

visibility				
Dostępne wartości:	visible	hidden	collapse	
	initial	inherit		
Wartość domyślna:	visible			
Przykład:	p {visibility:hidden;}			
Dziedziczenie:	tak			

ZMIANA ROZMIARU WYMIAROWANEGO ELEMENTU

resize	
Dostępne wartości:	none both horizontal vertical initial inherit
Wartość domyślna:	none
Przykład:	div {resize:both;}
Dziedziczenie:	nie

STRONA

ŁAMANIE STRONY PRZED ELEMENTEM

page-break-before	
Dostępne wartości:	auto always avoid left right initial inherit
Wartość domyślna:	auto

Przykład:	h1 {page-break-before: always;}
Dziedziczenie:	nie

ŁAMANIE STRONY PO ELEMENTIE

page-break-after	
Dostępne wartości:	auto always avoid left right initial inherit
Wartość domyślna:	auto
Przykład:	h1 {page-break-after:always;}
Dziedziczenie:	nie

ŁAMANIE STRONY WENWĄTRZ ELEMENTU

page-break-inside	
Dostępne wartości:	avoid auto initial inherit
Wartość domyślna:	auto
Przykład:	h1 {page-break-inside:avoid;}
Dziedziczenie:	nie

KONTURY

GRUBUŚĆ KONTURU

outline-width	
Dostępne wartości:	none wartości liczbowe wartości procentowe thin medium thick initial inherit
Wartość domyślna:	medium
Przykład:	a {outline-width:5px;}
Dziedziczenie:	nie

STYL KONTURU

outline-style			
Dostępne wartości:	none	hidden	dotted
	dashed	solid	double
	groove	ridge	inset
	outset	initial	inherit
Wartość domyślna:	none		
Przykład:	a {outline-style:dashed;}		
Dziedziczenie:	nie		

WDOWY

widows	
Dostępne wartości:	wartość liczbową initial inherit
Wartość domyślna:	2
Przykład:	P {widows:4;}
Dziedziczenie:	tak

SIEROTY

orphans	
Dostępne wartości:	wartość liczbową initial inherit
Wartość domyślna:	2
Przykład:	P {orphans:4;}
Dziedziczenie:	tak

KOLOR KONTURU

outline-color	
Dostępne wartości:	definicja koloru invert inherit
Wartość domyślna:	–
Przykład:	a {outline-color:green;}
Dziedziczenie:	nie

OFFSET KONTURU

outline-offset	
Dostępne wartości:	długość initial initial inherit
Wartość domyślna:	0
Przykład:	div {outline-offset:1px;}
Dziedziczenie:	nie

ZBIORCZY ZAPIS WŁAŚCIWOŚCI KONTURU

outline	
Dostępne wartości:	<outline-color> <outline-style> <outline-width> initial inherit
Wartość domyślna:	Sprawdź indywidualne wartości
Przykład:	a {outline:green dashed 5px;}
Dziedziczenie:	nie

INTERFEJS UŻYTKOWNIKA

KURSORY

cursor	
Dostępne wartości:	auto crosshair default pointer move e-resize ne-resize nw-resize n-resize se-resize sw-resize s-resize w-resize text wait help progress
Wartość domyślna:	auto
Przykład:	a {cursor:help;}
Dziedziczenie:	tak

KOLOR KURSORA W POLU FORMULARZA

caret-color	
Dostępne wartości:	auto <i>kolor</i>
Wartość domyślna:	auto
Przykład:	<code>input {caret-color: red;}</code>
Dziedziczenie:	tak

SPOSÓB ZAZNACZANIA TREŚCI NA STRONIE

user-select	
Dostępne wartości:	auto none text all
Wartość domyślna:	auto
Przykład:	div { user-select: none; }
Dziedziczenie:	tak

RESETOWANIE ELEMENTU WBUDOWANEGO W PRZEGŁĄDARKĘ

appearance		
Dostępne wartości:	auto	none
Wartość domyślna:	auto	
Przykład:	input {appearance:none;}	
Dziedziczenie:	nie	

ZMIANA SPOSOBU PRZEWIJANIA

scroll-behavior	
Dostępne wartości:	auto smooth
Wartość domyślna:	auto
Przykład:	div {scroll-behavior:smooth;}
Dziedziczenie:	nie

POZOSTAŁE

RESET WSZYSTKICH WŁAŚCIWOŚCI CSS DLA WYBRANEGO ELEMENTU HTML

all	
Dostępne wartości:	initial inherit unset
Wartość domyślna:	-
Przykład:	div {all: initial;}
Dziedziczenie:	nie

INFORMACJA O CHĘCI ZMIANY CECHY WYBRANEGO ELEMENTU HTML

will-change	
Dostępne wartości:	auto scroll-position contents transform opacity left top initial inherit unset
Wartość domyślna:	auto
Przykład:	div {will-change: transform;}
Dziedziczenie:	nie