

Tworzenie Stron Internetowych

odcinek 6

CSS – kaskadowe arkusze stylów

CSS (*Cascading Style Sheets*) – Kaskadowe Arkusze Stylów, język służący do opisanie warstwy prezentacyjnej

stylów

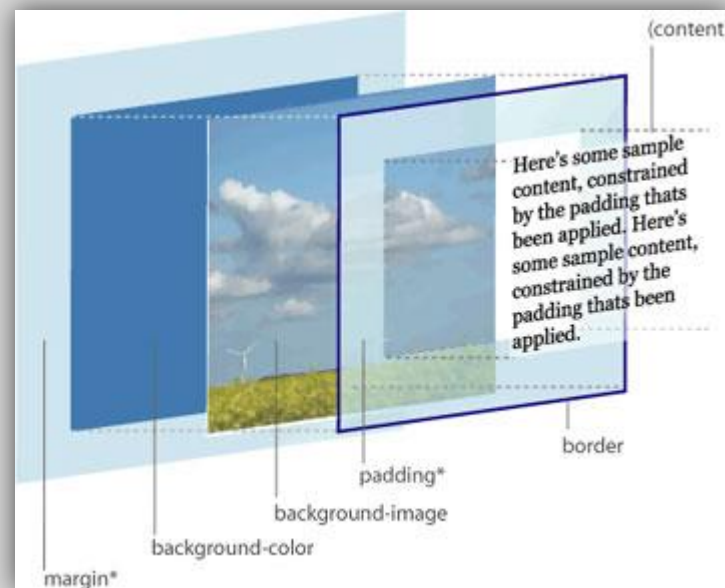
Reguły definiujące wygląd dokumentu opisanego za pomocą jakiegoś języka znaczników, np. HTML

arkusze

Reguły CSS gromadzi się w plikach nazywanych arkuszami stylów. Jeden arkusz można dołączyć do wielu dokumentów HTML i tym samym oszczędzić dużo pracy na powtarzaniu wszystkich cech wyglądu w każdym dokumencie osobno.

kaskadowe

W przypadku, gdy do jednego elementu HTML odnosi się więcej niż jedna reguła CSS, specyfikacja języka pozwala jednoznacznie określić, którą z nich jest ważniejsza i ma stanowić o wyglądzie tego elementu. Cechą CSS jest kaskadowe (hierarchicznie) nakładanie się reguł.



treść – struktura – prezentacja – zachowanie

Końcowy wygląd strony internetowej to efekt współdziałania kilku technologii, pełniących różne zadania. Stronę tę tworzą:

- *treść*, czyli to co chcemy opublikować w sieci
- *struktura*, czyli podział tekstu na części o różnym znaczeniu (tzw. *semantyka dokumentu*); za strukturę odpowiada **HTML**
- *prezentacja/układ*, czyli definiowanie wyglądu dokumentu (wyglądu jego elementów strukturalnych) za pomocą **CSS**
- *zachowanie*, czyli kontrola zachowania zawartości i interakcja z użytkownikiem przy wykorzystaniu **Javascript**.

Dawniej

W starszych wersjach HTML opisywał zarówno strukturę, jak i prezentację.

ścieżka ewolucyjna CSS

- różne formy **arkuszy stylów** istnieją od lat '80
- wraz z rozwojem Internetu strony WWW stają się coraz bardziej rozbudowane graficznie; bardziej elegancki wygląd oznacza większą złożoność kodu HTML; rozwiązaniem jest **oddzielenie warstwy prezentacyjnej od strukturalnej**, co pozwala mieć prosty, przejrzysty kod HTML i daje jednocześnie duże możliwości sterowania wyglądem stron WWW
- Cascading HTML Style Sheets (**CHSS**) – **protoplasta CSS** – pojawia się w 1994 roku
- w 1996 pojawia się pierwsza oficjalna wersja CSS (**CSS1**), wydana przez W3C.
- udoskonalona wersja, **CSS2**, pojawia się już w 1998 roku
- dopiero w 2011 pojawia się poprawiona wersja **CSS2.1** rekomendowana przez W3C
- rozwijana jest wersja **CSS3**, która została podzielona na moduły, część modułów poziomu 3 jest już rekomendowana przez W3C
- w 2009 rozpoczęły się prace nad CSS4; podział na moduły został utrzymany; obecnie tylko niektóre moduły poziomu 4 są wspierane przez przeglądarki (niekoniecznie wszystkie)

Za tworzenie specyfikacji języka CSS odpowiada World Wide Web Consortium (W3C, www.w3.org).

W przypadku nowych funkcjonalności CSS warto sprawdzić ich wsparcie przez przeglądarki:

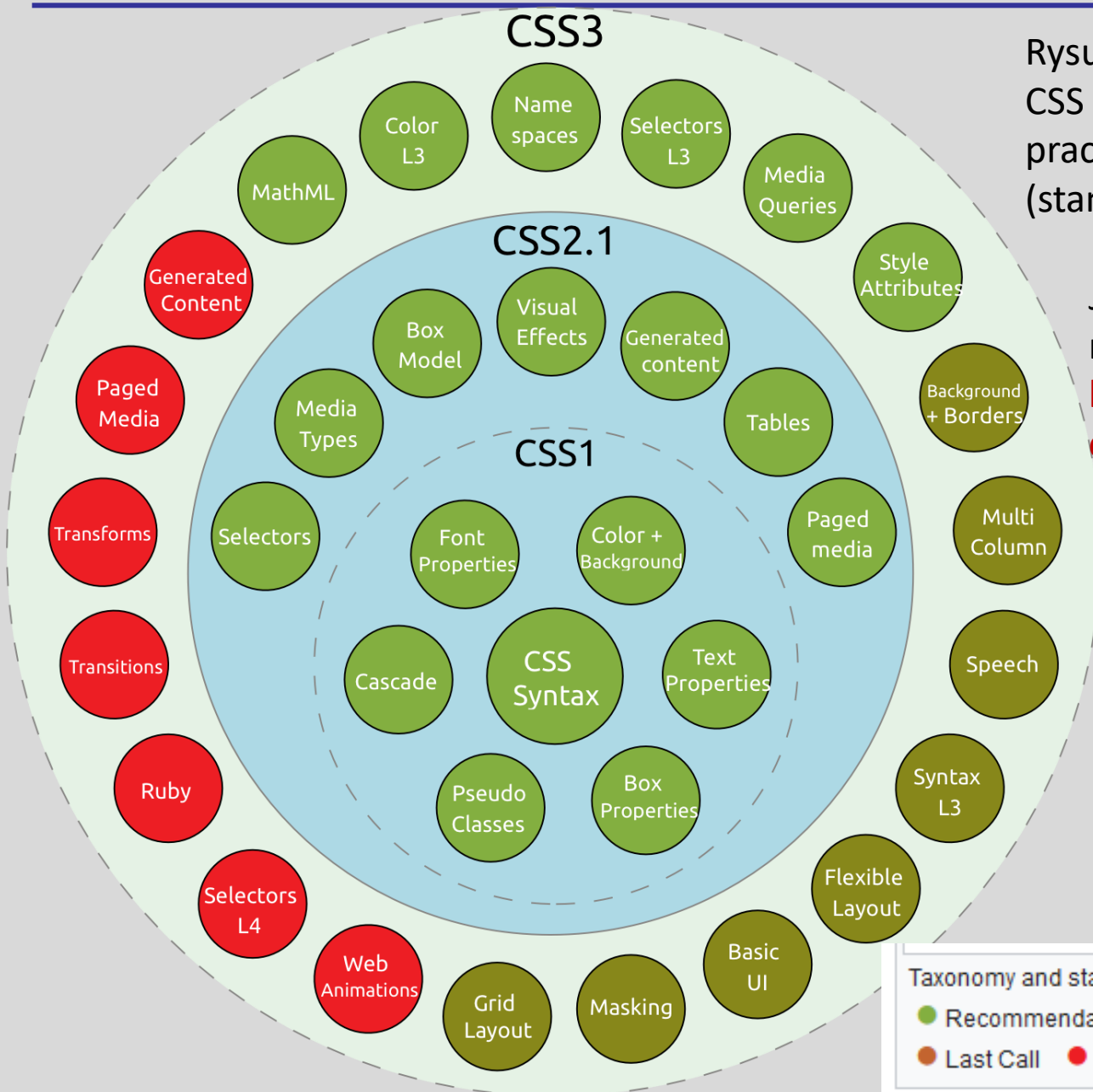
www.w3schools.com/cssref/css3_browsersupport.php lub <https://caniuse.com>

ścieżka ewolucyjna CSS

Rysunek pokazujący modułowość CSS oraz poziom zaawansowania prac na różnych modułach (stan na 8.2017).

Jeśli chcesz śledzić rozwój CSS na bieżąco, zajrzyj tu:

<https://www.w3.org/Style/CSS/current-work.en.html>



Taxonomy and status of CSS3 modules.

- Recommendation
- Candidate Recommendation
- Last Call
- Working Draft.

8.2017

CSS – zalety

Dlaczego używać CSS:

- Arkusze stylów to podstawowe narzędzie do budowania warstwy prezentacyjnej stron WWW (obowiązkowa wiedza dla webmastera).
- Przeszarzałe atrybuty i znaczniki formatujące wygląd zostały usunięte ze składni HTML i przeniesione do CSS.
- Dokumenty pisane z wykorzystaniem arkuszy stylów, czyli z oddzieleniem warstwy prezentacyjnej od strukturalnej, są zwykle bardziej przejrzyste i krótsze.
- CSS daje twórcom stron WWW więcej swobody i kontroli dla uzyskania pożądanego efektu wizualnego.
- Style pozwalają w łatwy sposób zarządzać całą serią dokumentów.
- Możliwość stosowania różnych selektorów ułatwia i przyspiesza tworzenie wyglądu stron WWW a także późniejsze jego modyfikacje.
- Style dają możliwości, które były i są nadal praktycznie niemożliwe do osiągnięcia w samym HTML.

CSS – edytory

Edycja CSS

Arkusze CSS to zwykłe pliki tekstowe, można więc go pisać w dowolnym edytorze tekstowym. Warto jednak używać edytorów programistycznych, które ułatwiają pisanie kodu. Dobry edytor CSS posiada:

- podświetlanie składni (unikanie błędów w poleceniach),
- zaawansowana edycja (wielopoziomowe cofanie, znajdź-zastąp,...),
- generatory elementów CSS (cechy, wartości).

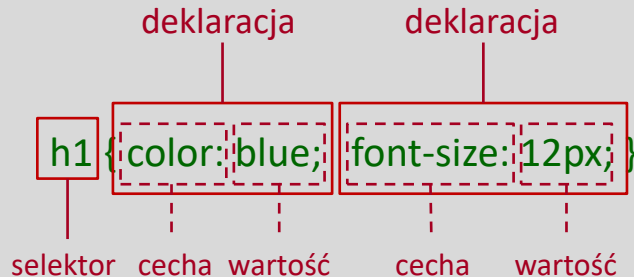
Często edytory do tworzenia stron internetowych obsługują edycję dokumentów HTML i CSS (i innych).

CSS – składnia

Język CSS składa się z reguł

Każda reguła składa się ze zbioru **cech**, którym nadaje **wartości** (*np. kolor czcionki ma być czerwony, wielkość pisma ma być duża*) oraz określenia (tzw. **selektora**) wskazującego, do których elementów w dokumencie HTML ma ta reguła być zastosowana (*np. zastosuj to tylko do pierwszego akapitu*). Każda para cecha-wartość to tzw. **deklaracja**. Przy jednym selektorze można umieścić wiele deklaracji oddzielając je średnikiem.

`selektor {cecha: wartość; inna-cecha: jakieś_wartości;}` - ogólna postać reguły CSS



- przykład reguły opisującej wygląd nagłówków h1

Białe znaki i komentarze

Składnia CSS pozwala umieszczać odstępy, przejścia do nowej linii prawie w każdym miejscu:

```
selektor
{cecha: wartość;
 inna-cecha: jakieś_wartości;}
```

Komentarze mają następującą postać:

```
/* treść komentarza */
```

Komentarz może zajmować wiele linii.

CSS – składnia

Wskazanie elementu HTML, który ma być opisany daną regułą CSS, odbywa się przy pomocy **selektora**. Do tego wskazania można wykorzystać **nazwę elementu**, jego **atrybuty lub wartości tych atrybutów**, **położenie w drzewie hierarchii dokumentu**, itd.

Duża liczba typów selektorów umożliwia precyzyjne wskazanie elementów w kodzie źródłowym dokumentu HTML i jest wielką zaletą CSS.

Każda grupa elementów ma określony zespół cech CSS, które można jej przypisać.

Każda cecha ma ściśle wyszczególnioną listę wartości, które może przyjąć.

Na przykład:

- cecha *text-align* (wyrównanie tekstu) może być przypisana tylko i wyłącznie do elementów blokowych; podanie jej dla elementów liniowych nie ma sensu.
- cecha *text-align* może przyjmować tylko wartości takie jak: *left*, *right*, *center*, *justify*; przypisanie do niej np. wartości koloru nie ma sensu.

CSS – osadzanie

Sposoby dołączenia (osadzania) stylów CSS do gotowych dokumentów są różne. Każdy sposób jest przydatny w innych sytuacjach. Często stosuje jednocześnie wszystkie z przedstawionych metod osadzania CSS, dobierając metodę w zależności od konkretnej potrzeby.

Styl lokalny

Styl ten polega na umieszczeniu formatowania CSS w dokumencie HTML.

Pozwala na nadanie formatowania konkretnemu pojedynczemu elementowi w kodzie HTML. Inną nazwą tego stylu to "w linii" (**inline**), ponieważ jest wstawiany w tej samej linii, w której znajduje się element formatowany. **Stylu tego nie należy używać często, ponieważ miesza on warstwę prezentacyjną i strukturalną dokumentu.**

```
<selektor style="cecha: wartość; cecha2: wartość2...">...</selektor>
```

np.:

```
<p style="color: red">tekst akapitu</p>
```

- przykład reguły opisującej wygląd akapitów (kolor tekstu)

CSS – osadzanie

Rozciąganie stylu

Znaczniem `...` możemy objąć kilka różnych elementów **liniowych** i nadać im wspólne formatowanie. Znacznik ten dobrze nadaje się do nadawania stylów, ponieważ sam w sobie nie ma określonego żadnego znaczenia semantycznego.

```
<span style="cecha1: wartość1; cecha2: wartość2..."> elementy liniowe </span>
```

np.: ``

```
    <p><i>jakiś fragment tekstu</i> i jego ciąg dalszy</p>
```

```
</span>
```

Wydzielone bloki

Znaczniem `<div>...</div>` możemy objąć kilka różnych elementów **liniowych** i **blokowych**, oraz nadać im wspólne formatowanie. Tą metodą możemy formatować większe fragmenty dokumentu. Podobnie jak ``, ten znacznik nie ma określonego żadnego znaczenia semantycznego, stąd jego przydatność jako „kontenera” do formatowania.

```
<div style="cecha1: wartość1; cecha2: wartość2..."> elementy liniowe i blokowe </div>
```

np.: `<div style="background-color: yellow">`

```
    <h1>tytuł</h1>
```

```
    <p>akapit</p>
```

```
    lista, grafika, tabela, ...
```

```
</div>
```

Uwaga. Powyższe dwa sposoby osadzania CSS są wariantem stylu lokalnego i mają jego wady.

CSS – osadzanie

Wewnętrzny arkusz stylów

Wewnętrzny arkusz stylów wstawiamy w części nagłówkowej dokumentu **<head>**. Używany jest do nadania stylu jednemu dokumentowi HTML. Zawartość takiego arkusza stylów wpisujemy do elementu **<style>...</style>** następująco (linie oznaczone na szaro występują tylko w XHTML):

```
<head>
...
<style>
  /* <![CDATA[ */
  selektor1 { cecha1: wartość1; cecha2: wartość2... }
  selektor2 { cecha3: wartość3; cecha4: wartość4... }
  ....
  /* ]]> */
</style>
...
</head>
```

Przykład:

```
<style>
  body {background-color: black;}
  h6 {color: red;}
</style>
```

Taki zestaw deklaracji spowoduje, że wszystkie nagłówki h6 będą miały kolor czerwony, a tło całego dokumentu będzie czarne.

CSS – osadzanie

Zewnętrzny arkusz stylów

Zewnętrzny arkusz stylów ujawnia najlepszą cechę CSS. Za pomocą jednego arkusza możemy sformatować nawet wszystkie strony tworzące serwis internetowy. Zmiana wyglądu jakiegoś elementu na wszystkich stronach wymaga zmodyfikowania kodu tylko w tym jednym arkuszu stylów.

Reguły opisujące wygląd stron(y) umieszczamy w pliku z rozszerzeniem **.css**, a w dokumencie HTML, w części nagłówkowej, podajemy:

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

przykładowy plik CSS:

```
body  
  {font-family: Verdana, Arial, Helvetica, sans-serif;  
  font-size: 10pt;  
  color: #003868;  
  background-color: #80B8E8;  
  margin: 6mm;}  
h1  
  {color: navy;  
  margin-left: 20px;}  
p  
  {text-align: justify;}
```

- krój czcionki
- rozmiar czcionki
- kolor czcionki
- kolor tła
- rozmiar marginesów

- rozmiar marginesu lewego

- wyrównanie tekstu

CSS – osadzanie

Alternatywny arkusz stylów

Alternatywny arkusz stylów pozwala udostępnić użytkownikowi kilka wersji wyglądu strony do wyboru. Aby uzyskać taki efekt należy zaprojektować kilka zewnętrznych arkuszy CSS i osadzić je w dokumencie HTML następująco:

```
<head>
  <link rel="stylesheet" type="text/css" href="styl.css" title="Nazwa domyślna" />
  <link rel="alternate" type="text/css" href="styl1.css" title="Nazwa1" />
  <link rel="alternate" type="text/css" href="styl2.css" title="Nazwa2" />
  ...
</head>
```

Użytkownik będzie mógł w przeglądarce wybrać jeden podanych przez nas stylów.
(W przeglądarce FF: *widok > styl strony > nazwy stylów podane atrybutem title*)

CSS – osadzanie

Import arkusza stylów

Używając polecenia:

```
@import url(adres zewnętrznego arkusza stylów);
```

możemy w naszym dokumencie osadzić arkusz CSS z dowolnego miejsca w Internecie. Powyższe polecenie możemy umieścić zarówno w wewnętrznym jak i zewnętrznym arkuszu stylów.

Wszystkie polecenia importu (może ich być kilka) muszą się znajdować zaraz na początku arkusza stylów, do którego następuje import (przed właściwymi regułami CSS, znajdującymi się w arkuszu).

Uwaga na poprawne podanie adresu importowanego arkusza.

Przykład:

```
@import url(http://www.astro.uni.wroc.pl/astroizery/pi\_styl1.css);
```

CSS – kaskadowość

Kaskadowość stylów

Kaskadowość stylów określa pierwszeństwo w nadawaniu stylów, jeśli do dokumentu HTML odnosi się więcej niż jeden arkusz CSS. Pierwszeństwo mają te, które znajdują się bliżej formatowanego elementu. Kolejność kaskadowości (hierarchii) jest następująca:

1. **Styl lokalny (inline)**
2. Rozciąganie stylu (SPAN)
3. Wydzielone bloki (DIV)
4. **Wewnętrzny arkusz stylów**
5. Import stylów do wewnętrznego arkusza
6. **Zewnętrzny arkusz stylów**
7. Import stylów do zewnętrznego arkusza
8. **Domyślny styl przeglądarki**

Style o wyższym priorytecie (wyżej na liście) mają pierwszeństwo w modyfikowaniu elementów dokumentu. Style niżej w hierarchii mogą zmienić formatowanie jakiegoś elementu, tylko jeśli style o wyższym priorytecie nie dotyczą tego konkretnego elementu. Dzięki kaskadowości nie występują konflikty pomiędzy regułami odnoszącymi się do tego samego elementu.

UWAGA! Polecenie dołączenia zewnętrznego arkusza powinno znajdować się w dokumencie wcześniej niż wewnętrzny arkusz. Odwrotna kolejność złamie zasady kaskadowości!

Zasady kaskadowości wewnątrz tego samego arkusza CSS (kilka różnych reguł odnoszących się do tej samej cechy tego samego elementu): **ostateczny wygląd określa reguła znajdująca się bliżej końca arkusza.**

CSS – kaskadowość

Kaskadowość stylów

Przykład działania kaskadowości CSS.

Do strony WWW dołączony jest zewnętrzny i wewnętrzny arkusz CSS. Reguła umieszczona w zewnętrznym arkuszu CSS odnosząca się do elementu <h1> jest następująca:

```
h1
  {color: navy;
  margin-left: 20px;}
```

W wewnętrznym arkuszu CSS podana jest natomiast reguła:

```
h1
  {color: orange;}
```

Końcowy efekt działania kaskadowości:

- kolor <h1> – orange (decyduje reguła ze stylu wewnętrznego, będąca wyżej w hierarchii),
- margines lewy – 20 px (decyduje reguła ze stylu zewnętrznego, będąca jedyną).

CSS – kaskadowość

Łamanie kaskadowości

Poleceniem **!important** możemy złamać zasady kaskadowości, jeżeli tego potrzebujemy. Polecenie to należy umieścić zaraz po wartości cechy, dla której chcemy złamać kaskadowość. **Nie należy nadużywać polecenia !important, choćby dlatego, że może utrudnić analizę sposobu formatowania elementów i późniejszą jego zmianę.**

Przykładowo, umieszczenie w zewnętrznym arkuszu CSS następującej reguły:

```
h6 { color: blue !important; background-color: green; }
```

spowoduje, że pomimo, podania takiej reguły w stylu lokalnym:

```
<h6 style="color: red; background-color: yellow"> Tytuł </h6>
```

Kolor nagłówka h6 będzie niebieski a nie czerwony. Natomiast kolor tła będzie zgodny z kaskadowością (tu: żółty).