

Wprowadzenie

CSS (ang. Cascading Style Sheets, kaskadowe arkusze stylów) są językiem opisu stylów używanym do opisu prezentacji dokumentu napisanego w HTML-u lub XML-u (włączając w to różne języki XML-owe takie jak XHTML czy SVG - Scalable Vector Graphics). W produktach bazowanych na XUL-u, takich jak Mozilla Suite, Firefox czy Thunderbird, CSS jest także używany do opisu stylu interfejsu aplikacji użytkownika (GUI). Na przykład Motywy w ogromnym stopniu korzystają z CSS do określenia wyglądu aplikacji. CSS jest specyfikacją W3C

Znaczniki HTML zostały zaprojektowane do określania struktury dokumentu. Tak by określać: "To jest nagłówek", "To jest paragraf", "To jest tabela", wykorzystując do tego odpowiednie znaczniki takie jak `<h1>`, `<p>`, `<table>` i im podobne. Przedstawienie zawartości dokumentu miało być zadaniem przeglądarki a nie znaczników. Wtedy dwie najczęściej wykorzystywane przeglądarki czyli Netscape Navigator i Internet Explorer dodawały do oryginalnej specyfikacji HTML dodatkowe znaczniki i atrybuty takie jak `` lub atrybut `color`, co powodowało że pisanie stron WWW, których treść była oddzielona od prezentacji, stawało się coraz trudniejsze. W celu rozwiązania tego problemu konsorcjum W3C, które jest odpowiedzialne za standaryzację HTML, stworzyło Style jako dodatek do HTML. Obecnie praktycznie każda przeglądarka w jakimś stopniu obsługuje Kaskadowe Arkusze Stylów.

Technologia CSS stopniowo wypiera inne metody definiowania układu i wyglądu witryn WWW. Uniwersalność i elastyczność kaskadowych arkuszy stylów sprawiają, że zmiana kolorystyki lub konstrukcji strony sprowadza się do zmodyfikowania arkusza definiującego wygląd określonych elementów.

Podstawy CSS

Kiedy podstawowa struktura dokumentu zostanie utworzona, można zacząć formatować jego wygląd za pomocą arkuszy stylów stylów.

Tworzenie stylów

Każda reguła tworząca arkusz stylów składa się z trzech części: selektora, właściwości i wartości, którą dana właściwość przyjmuje.

```
selektor {właściwość: wartość; ...}
```

Selektor określa do jakiego elementu HTML będzie odnosić się dana reguła stylu. Wszystkie właściwości selektora umieszczane są wewnątrz nawiasów klamrowych. **Właściwość** i **wartość** oddzielone są od siebie dwukropkiem, a od kolejnych par właściwość:wartość oddzielone są średnikiem. Właściwości atrybutu oraz ich wartości określa specyfikacja W3C.

Przykład: Aby wyświetlić wszystkie akapity czcionką Arial o kolorze czerwonym należy stworzyć selektor, który wybierze jedynie elementy `<p>` nie zmieniając zadnego innego elementu w dokumencie:

```
p {font-family: Arial; color:#FF0000;}
```

Specyfikacja CSS pozwala również na grupowanie selektorów wewnątrz jednej reguły.

```
h1, h2, h3, h4, {font-family: Arial; color:#FF0000;}
```

Wybór elementu

Najczęstszym kryterium wyboru elementów, jakie mają być formatowane jest ich nazwa. W ten sposób powyżej zostały wybrane wszystkie elementy <p> oraz grupa znaczników nagłówkowych.

Równie często stosowanymi sposobami są:

■ wybór elementu na podstawie klasy,

Jeżeli elementy zostały przydzielone do pewnych klas:

```
<div class="klasa1">
  Do danej klasy może należeć dowolna ilość elementów.
</div/>
```

To nazwy tych klas można podać w selektorze reguły stylów.

```
.klasa1 {color: red;}
```

Aby wybrać elementy na podstawie klasy należy w arkuszu stylów użyć znaku kropki (.)

■ wybór elementu na podstawie identyfikatora,

Jeżeli zostały określone identyfikatory elementów:

```
<div id="ident_1">
  Każdy atrybut id w dokumencie HTML musi mieć unikalną wartość.
  Żadne dwa elementy nie mogą mieć tego samego identyfikatora.
</div/>
```

To nazwy tych identyfikatorów można podać w selektorze reguły stylów.

```
#ident_1 {color: red;}
```

Aby wybrać element na podstawie identyfikatora należy w arkuszu stylów użyć znaku krzyżyka (#)

■ wybór elementu na podstawie kontekstu.

W CSS można także wybierać elementy na podstawie ich elementów nadrzędnych:

```
<div class="klasa1">
  <p>
    Element nadrzędny to taki, w którym jest umieszczony
    interesujący nas element.
  </p>
</div/>
```

A zatem aby określić dowolny element <p> umieszczony wewnątrz elementu <div> należącego do klasy "klasa1", należy użyć takiego selektora:

```
div.klasa1 p {color: red;}
```

Powyższy zapis oznacza, że każdy element <p> zawarty wewnątrz bloku <div> klasy "klasa1", będzie wyświetlany w kolorze czerwonym.

Stosowanie stylów

Powyżej zostały opisane podstawowe zagadnienia związane z tworzeniem reguł stylów. Najlepszym sposobem ich wykorzystania jest łączenie tych reguł w arkusze stylów. Dzięki temu możliwe jest określanie wyglądu pojedynczych elementów, całych dokumentów, jak również całych witryn WWW.

■ Tworzenie zewnętrznego arkusza stylów

Czynność ta sprowadza się do stworzenia pliku tekstowego o rozszerzeniu *.css i umieszczeniu w nim dowolnej ilości reguł.

Po stworzeniu zewnętrznego arkusza stylów konieczne jest jego dołączenie lub też zaimportowanie do dokumentu.

Zatem aby dołączyć zewnętrzny arkusz stylów do dokumentu HTML, należy w sekcji <head> dokumentu wpisać następujący kod:

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

Gdzie text/css oznacza, że arkusz stylów został stworzony w technologii CSS, a style.css jest nazwą pliku zawierającego reguły stylów.

Możliwe jest dołączenie kilku arkuszy stylów do jednego dokumentu. Daje to użytkownikowi dokumentu możliwość wyboru jak używany dokument ma być wyświetlany.

Specyfikacja pozwala na tworzenie bazy tzw. "trwałych stylów", które będą wykorzystywane niezależnie od wyboru użytkownika.

Tworzyć można także bazę tzw. "stylów preferowanych" wykorzystywanych gdy użytkownik nie dokona żadnego wyboru, oraz "stylów alternatywnych", które będą aktywne, jeśli użytkownik je wybierze.

```
<head>
<link rel="stylesheet" type="text/css" href="style.css" />
<link rel="stylesheet" type="text/css" href="style1.css"
      title="etykieta"/>
<link rel="alternate stylesheet" type="text/css" href="style2.css"
      title="etykieta"/>
</head>
```

W ten sposób w odpowiedniej kolejności zostały zdefiniowane: **trwały arkusz** tzw. bazowy, **preferowany arkusz**, który będzie wykorzystywany automatycznie oraz **alternatywny arkusz stylów**. Każdy z nich należy dołączyć do dokumentu przy pomocy osobnego elementu <link>

■ Tworzenie wewnętrznego arkusza stylów

Wewnętrzne arkusze stylów są definiowane na samym początku dokumentu HTML (w sekcji HEAD).

Aby stworzyć wewnętrzny arkusz stylów należy w sekcji HEAD dokumentu wstawić następujący kod:

```
<head>
<style type="text/css">
  W tym miejscu należy zdefiniować dowolną ilość reguł stylów.
</style>
```

```
</head>
```

■ Importowanie zewnętrznego arkusza stylów

Aby zaimportować zewnętrzny arkusz stylów należy wewnątrz elementu <style>, ale przed jakąkolwiek regułą wpisać następujący kod:

```
<head>
<style type="text/css">
    @import "arkusz_zewnetrzny.css"
</style>
</head>
```

■ Lokalne stosowanie stylów

Lokalne stosowanie stylów jest prostym rozwiązaniem jednak nie pozwala ono na zgrupowanie stylów w celu ich łatwiejszej edycji lub aktualizacji.

```
<p style="color: #FF0000; font-family: Arial;"></p>
```

■ Dodawanie komentarzy do arkuszy stylów

Dodawanie komentarzy do arkuszy stylów jest bardzo dobrym pomysłem, dzięki któremu można zapamiętać, w jakim celu zostały stworzone wyjątkowo skomplikowane reguły.

Aby dodać komentarz do arkusza stylów należy użyć następującej notacji:

```
/* treść komentarza */
```

Komentarze mogą zawierać znaki nowej linii, a zatem można je zapisywać w kilku wierszach. Znaki komentarza można również umieszczać wokół reguł stylów przez co przeglądarka ich nie zinterpretuje - jest to dobra metoda testowania arkuszy stylów.

Nie można natomiast zagnieżdżać komentarzy, czyli komentarz nie może zawierać kombinacji znaków */.

CSS3

Trwają również prace nad CSS3. Język ten wzbogaci się o wiele nowych elementów w stosunku do poprzedników. Zmiany nie ograniczą się tylko do dodania nowych właściwości. Nowością jest modułowy charakter CSS3 - nie będzie to już jednolita rekomendacja, lecz kilkadziesiąt rekomendacji (plannowane 36 modułów), co pozwoli na włączanie lub wyłączanie odpowiednich modułów w przeglądarkach, w zależności od potrzeb. Zredukują całkowity rozmiar specyfikacji jako dokumentu oraz umożliwi łatwe uaktualnianie i dopracowywanie poszczególnych modułów z osobna a nie wydawania kolejnej wersji specyfikacji.

CSS3 obecnie jest obsługiwany szczątkowo przez większość najnowszych przeglądarek (opartych o silnik Gecko np. Mozilla Firefox). Niektóre z nich posiadają eksperymentalne wsparcie dla wybranych właściwości CSS3 - w takim przypadku są one opatrywane odpowiednim prefiksem (np. -moz- dla Gecko, czy -o- dla Opery).

Mozilla wprowadziła w swoich przeglądarkach blisko 170 eksperymentalnych właściwości. Jednym z takich eksperymentalnych modułów wprowadzonych przez Mozillę jest moduł Multi-column layout .

■ Stosowanie kolumn

Własności CSS wprowadzone w przeglądarkach organizacji Mozilla pozwalają określić:

■ Liczbę kolumn

```
<div class="sample" style="-moz-column-count: 2;">  
    Przykładowy tekst zostanie wyświetlony w dwóch kolumnach.  
</div/>
```

■ Szerokość kolumn

```
<div class="sample" style="-moz-column-width: 200px;">  
    Jeśli własność -moz-column-count nie jest ustalona, przeglądarka automatycznie  
    wyświetli tyle kolumn, ile zmieści się w jej oknie.  
</div/>
```

■ Odstęp pomiędzy kolumnami

```
<div class="sample" style="-moz-column-gap: 15px;">  
    Przeglądarka wyświetla kolumny przylegające do siebie. Nie jest to zachowanie pożądané.  
    Aby poprawić tę sytuację, można za pomocą CSS ustawić dla kolumn odpowiednie wartości  
    marginesów wewnętrznych lub zastosować jest własność -moz-column-gap  
</div/>
```

Przykład:

Czytanie tekstu wyświetlanego w długich wierszach jest trudne i męczące dla oczu - przenoszenie wzroku z końca jednej linii na początek drugiej zajmuje zbyt wiele czasu, łatwo jest zgubić się w tekście i rozpocząć czytanie niewłaściwego wiersza. Z tego powodu i aby w pełni wykorzystać szerokość ekranu, tekst na stronach internetowych - tak samo jak w gazetach - powinien być

układany w sąsiadujących kolumnach o stałej szerokości.

Niestety osiągnięcie tego efektu nie jest możliwe przy użyciu języków HTML i obecnego CSS bez wymuszania sztywnej wysokości kolumn, bądź też bez stosowania skomplikowanych skryptów.

Przydatne linki i tytuły

Linki:

Organizacja W3C - <http://www.w3.org/Style/CSS/>

Wikipedia Polska - http://pl.wikipedia.org/wiki/Cascading_Style_Sheet

w3schools - <http://www.w3schools.com/css/default.asp>

Mozilla Project - <http://developer.mozilla.org/pl/docs/Kategoria:CSS>

Tytuły:

Eric A. Mayer - CSS według Erica Mayera

Elizabeth Castro - Po prostu HTML 4