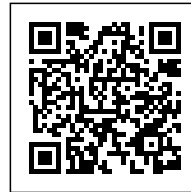




# MOTYW POTOMNY I CSS3

Posted on 10 kwietnia 2017 by Maciej Cybulski

Jeśli czytasz ten wpis, to znaczy że chcesz uatrakcyjnić swoją stronę w WordPress'ie. CSS (wraz z najnowszymi bibliotekami w wersji 3) bardzo nam w tym pomoże.



---

**Categories:** [dla początkujących](#), [HTML5 i CSS3](#), [motywy](#), [porady](#)

**Tag:** [css](#), [css3](#), [html](#), [kod](#), [motyw potomny](#), [szablon](#), [WordPress](#)

Jeśli czytasz ten wpis, to znaczy, że chcesz uatrakcyjnić swoją stronę w WordPress'ie. CSS (wraz z najnowszymi bibliotekami w wersji 3) bardzo nam w tym pomoże.

## Czym jest CSS?

CSS czyli Kaskadowy Arkusz Stylu, to język skryptowy dzięki któremu możemy wpływać na wygląd naszej strony. Jeśli chcesz, aby dany element był innego koloru? A może bardziej lub mniej zaokrąglony? A może, gdy na niego najedziemy, to ma się pojawiać wizualny efekt? To wszystko umożliwia nam CSS. **Jak język HTML odpowiada za strukturę strony, tak CSS określa jej wygląd i wzajemne położenie jej elementów.**

Standardowo pliki arkusza stylów to pliki tekstowe z rozszerzeniem `.css`. Każdy motyw zawiera

określone reguły dotyczące wyglądu poszczególnych elementów strony, które znajdziemy właśnie w plikach .css. W motywie podstawowym plikiem opisującym styl jest style.css

Ok więc pobawmy się stylem

## Co to motyw potomny i jakie daje mi on możliwości?

Aby rozpocząć nasze eksperymenty, powinniśmy utworzyć motyw potomny. Dlaczego to takie ważne? WordPress i motywy dla niego przeznaczone mają pewną właściwość: każda aktualizacja motywu nadpisuje (zastępuje) wszystkie jego dotychczasowe pliki. Oznacza to, że w momencie aktualizacji motywu, tracimy wszystkie zmiany wprowadzone przez nas w plikach z tego motywu. Jako, że będziemy pracować na plikach .css zawartych w motywie, ten problem dotyczy również ich.

Tworząc motyw potomny skutecznie zapobiegamy utracie naszych zmian, zatem po aktualizacjach strona w dalszym ciągu wygląda tak, jak sobie tego zażyczyliśmy.

## Ok, utwórzmy motyw potomny "step by step":

1. W naszym folderze motywów (domyślnie jest to : /wp-content/themes/) tworzymy nowy folder o dowolnej nazwie. Najczęściej stosuje się nazwę motywu rodzica z dodatkiem "child", np: DIVI-child, wtedy od razu wiadomo, że jest to motyw potomny i jakiego motywu.

Otwieramy najprostszy notatnik jaki mamy w komputerze i tworzymy nowy plik. Dla komputerów z systemem Windows dobrym rozwiązaniem jest program Notepad ++, ale śmiało też można edytować pliki txt/css w systemowym Notatniku. Komputery spod znaku nadgryzionego jabłka mają więcej ciekawych alternatyw np. Atom, TextEdit, TextWrangler, czy komercyjna Coda. Ale w temacie narzędzi programistycznych można by napisać oddzielny artykuł, więc wystarczy tej dygresji, wróćmy do naszego motywu potomnego.

W edytorze tekstu wpisujemy

```
/*  
Theme Name: Nazwa Motywu Potomnego  
Theme URI: adres WWW motywu rodzica  
Version: 1.0.0 - 10.04.2017  
Description: Motyw potomny testowy  
Author: M.Cybulski
```

```
Author URI: adres WWW autora  
Template: Divi  
*/  
@import url('../Divi-child/style.css');
```

## Co znaczą poszczególne elementy nagłówka?

Nagłówek w tym przypadku to treść zawarta pomiędzy znakami `/* i */`. W języku PHP również są oznaczeniem bloku komentarza. I tak:

**Theme Name** - nazwa naszego motywu potomnego

**Theme URI** - adres strony naszego motywu lub motywu rodzica - w sumie to może być też jakikolwiek tekst, czy link. Nie jest on istotny ze względu na poprawność działania naszego motywu potomnego.

**Version** - wersja naszego motywu - tutaj dobrym rozwiązaniem jest takie oznaczenie wersji by przy dużej ilości modyfikacji, nie pogubić się. Ja osobiście stosuję zarówno notację zaczerpniętą z wersjonowania systemu linux, ale też uzupełniam ją datą ostatniej aktualizacji. Stąd jestem w stanie określić jak dawno przeprowadzałem edycję arkusza.

**Description** - opis naszego motywu potomnego

**Author** - informacja, kto wykonał motyw potomny

**Author URI** - adres strony WWW autora

**Template** - nazwa motywu rodzica (i zarówno nazwa katalogu w którym się on znajduje). Wpisujemy ją również w ścieżce dostępu do pliku style.css, przy `@import url!`

Właściwie, to potrzebujemy w opisie tylko Theme Name oraz Template, ale sugeruję również posługiwać się pozostałymi elementami opisu motywu potomnego - dzięki temu możemy przekazać więcej informacji dla innych osób, które będą korzystać z naszego motywu potomnego.

Tak utworzony plik zapisujemy jako: style.css i umieszczamy w folderze z motywem potomnym.

By umieścić motyw na stronie, należy z folderu w którym znajduje się plik style.css (czyli w naszym przypadku /Divi-child/) utworzyć archiwum .zip - spakować np. WinZip'em.

Taki plik umieszczamy na naszej stronie, dodając go jako nowy motyw na stronie. Wygląd -> Motywy,

Dodaj nowy, Wyślij na serwer.

2. Po wprowadzeniu powyższych działań wchodzimy do kokpitu. W menu wybieramy "Wygląd" -> "Motywy". Wybieramy motyw oznaczony naszą nazwą motywu potomnego. Po tak wprowadzonych zmianach, możemy zacząć zabawę z wyglądem strony.

3. Wchodzimy w "Wygląd" -> "Edytor" i pojawia nam się takie okno:



Upewnij się, że masz ustawiony motyw potomny w prawym górnym rogu. Jak widzisz, mamy tu podgląd naszego pliku style.css. W nowej linijce pod spodem możemy wprowadzać wszelkie zmiany wizualne. Po skończeniu, klikamy "Zaktualizuj plik" i podziwiamy efekty na naszej stronie.

W przypadku problemów warto zwrócić uwagę na to czy ścieżki są poprawnie wprowadzone. Należy pamiętać o tym że system rozróżnia wielkość liter zarówno w plikach jak i ścieżkach. Zatem /Divi/ to nie to samo co /divi/

Ok, mamy przygotowany motyw potomny, ale jak wprowadzać zmiany?

## CSS w akcji

Przedstawię Ci najważniejsze informacje na temat języka CSS, oraz kilka prostych przykładów, które możesz zastosować u siebie na stronie.

Język CSS składa się z kilku elementów, o których warto pamiętać. Ogólna reguła została przedstawiona poniżej:

```
selektor {
```

```
wlasciwosc: wartosc;
```

}

**selektor** - czyli element (lub grupa elementów) strony, któremu zamierzamy przypisać konkretny styl CSS

**nawiasy klamrowe** - w których umieszczamy właściwości i określamy ich wartości

**właściwość** - czyli co chcemy zmienić w danym selektorze

**wartość** - przypisujemy tu, w jaki sposób chcemy zmienić nasz element

Możemy również do jednego lub kilku selektorów określić kilka właściwości i ich wartości. W takim przypadku oddzielamy je średnikiem ";". Wygląda to tak:

```
selektor1, selektor2 {
```

```
wlasciwosc: wartosc;
```

```
wlasciwosc: wartosc;
```

```
wlasciwosc: wartosc;
```

}

Ostatnia para właściwość :wartość nie musi się kończyć znakiem ;.

Zapis w ten sposób powoduje większą czytelność kodu i polecam posługiwać się taką formą.

## Co zatem możemy zmieniać na naszej stronie?

Elementów jest wiele, poznasz na początek elementy tekstowe: h1 i p. Nagłówków rozróżniamy w sumie 6 - od h1 do h6.

h1- nagłówek poziomy pierwszego

p - akapit

Oczywiście możemy zmieniać wygląd zarówno tekstu, jak i obrazu czy sekcji.

Dla przykładu zmienimy kolor tła nagłówków h1. W tym przypadku "h1" to nasz selektor, właściwością

będzie "background-color", a wartość przypiszemy tzw. kodem HEX (czyli szesnastkowym systemie liczbowym. Więcej informacji o nim [tutaj](#).)

Nasz kod będzie wyglądał tak:

```
h1 {  
  
    background-color: #FF530D;  
  
}
```

W miejsce wartości możesz wpisać angielską nazwę koloru, np: black, white, green. Jeśli potrzebujesz dokładniej określić kolor, posłuż się kodem HEX. Zadanie wyboru koloru z pewnością ułatwi Ci narzędzie, udostępniane przez Adobe- Kuler. Znajdziesz go pod adresem: <https://color.adobe.com/pl/>

Wpisz powyższą formułę w edytorze pliku style.css i podziwiał efekt.

Po zaktualizowaniu pliku nic się nie zmieniło na stronie? Spokojnie - potrzebujemy zastosować jeszcze jedną rzecz.

W kodzie CSS mamy "hierarchię ważności", czyli kaskadowość poszczególnych reguł. Oznacza to, że jeśli w motywie głównym mamy kolor body określony - to właśnie on będzie użyty w pierwszej kolejności i "nadpisuje" wpisany przez nas. Aby tego uniknąć i wymusić zastosowanie naszej wartości stosujemy **deklarację !important** (koniecznie z wykrzyknikiem na początku), która "łamie" zasady i sprawia, że nasza reguła staje się najważniejsza i to właśnie ona zostanie zastosowana na stronie.

Tak więc aktualizujemy powyższy kod i w ostateczności wygląda on tak:

```
h1 {  
  
    background-color: #FF530D !important;  
  
}
```

W ten sposób otrzymujemy nowy kolor nagłówków naszej strony.

Jeśli samodzielnie chcesz zagłębić temat CSS, polecam Ci książki na Helionie lub ebookpoint.pl

Masz pytania? Pisz śmiało w komentarzach.

**There are no comments yet.**