



Dokumentacja techniczna szablonów sklepów Sky-Shop.pl

UWAGA: Należy wykonać kopie wszystkich plików graficznych. Błędna edycja lub złe nadpisanie plików może spowodować awarię w działaniu Kreatora Szablonów.

Dostęp do plików HTML/CSS/JS na FTP jest równoznaczne z utratą gwarancji Sky-Shop na poprawne działanie szablonu sklepu oraz Kreatora Szablonu sklepu.

Wszelkie realizację zgłoszeń usterek i błędnego działania szablonu sklepu oraz Kreatora Szablonu które nie występują w platformie co jednoznacznie wskazuje na usterki spowodowane ingerencją w kod, będą realizowane odpłatnie.

1. Podstawowe założenia

- Skórka podzielona jest na moduły które mogą być wyświetlane w kolejności wybranej przez developera, jedynymi modułami które nie mogą być ustawiane w dowolnej kolejności to nagłówek (header) i stopka (footer)
- Skórki wykorzystują system szablonów (PHP) który jest używany w plikach HTML
- Do budowy wyglądu wizualnego może być używany CSS który będzie wykonywany po wszystkich bazowych plikach albo SCSS który musi zostać przekompilowany w Kreatorze
- Pliki JavaScript są również kompilowane i minifikowane w jedną całość w celu lepszego działania całego systemu

2. Struktura plików

- Pliki odpowiedzialne za moduły znajdują się w folderze /modules i są zawarte w folderach których nazwy określają rodzaj modułu
- W folderach z modułami znajdują się kolejne podfoldery /one, /two, /three itd... w których są umieszczone pliki odpowiedzialne za ten konkretny moduł
- Plikami niezbędnymi do poprawnego działania modułu są pliki o rozszerzeniu .html oraz .scss o takiej samej nazwie jaką ma folder - rodzic
- Plik .scss jest plikiem w którym może być wykonywany kod SCSS jak i tradycyjny CSS, szablon musi być jednak zapisany w Kreatorze po edycji kodu by móc zobaczyć zmiany
- Plik .js o takiej samej nazwie jak folder - rodzic służy do wykonywania logiki dla tego jednego określonego modułu, inne globalne funkcjonalności w JavaScript znajdują się w pliku app.js
- Pliki full.png, mockup.png oraz small.png są plikami (screenami) poglądowymi wykorzystywanymi w Kreatorze do prezentacji na polu roboczym

3. Zmienne PHP

- {\$LANG} - aktualny język sklepu
- {\$LANGS} - (array) - pozostałe języki sklepu
- {\$CURRENCY} - waluta klienta (EUR, PLN, USD itd.)

- `{$CONTROLLER}` - sekcja sklepu (index, product, category, news, cart itd.)
- `{$CAT_ID}` - ID kategorii jeżeli istnieje, w przeciwnym razie 0
- `{$CAT_NAME}` - nazwa kategorii jeżeli istnieje, w przeciwnym razie pusty string
- `{$PROD_ID}` - ID produktu jeżeli istnieje, w przeciwnym razie 0,
- `{$PROD_PRICE}` - cena produktu jeżeli istnieje, w przeciwnym razie 0
- `{$PROD_NAME}` - nazwa produktu jeżeli istnieje, w przeciwnym razie pusty string
- `{$PROD_AMOUNT}` - ilość produktu jeżeli istnieje, w przeciwnym razie 0
- `{$NEWS_ID}` - ID artykułu jeżeli istnieje, w przeciwnym razie 0

4. Zmienne PHP - teksty statyczne

- Wszystkie treści statyczne są przechowywane w bazie danych i są wyświetlane na stronie sklepu poprzez zmienne PHP, struktura tej zmiennej jest zawsze taka sama i wygląda następująco: `{$_L_*}` gdzie gwiazdka to klucz danej treści statycznej którą chcemy użyć np. `{$_L_PRICE}` (Cena, Price itd.)

5. Debugowanie PHP

- W celu debugowania PHP należy upewnić się, że sklep posiada aktywny moduł API i utworzony klucz, można to zrobić w Konfiguracja -> Web API sklepu
- Aby móc debugować zmienne PHP należy utworzyć ciastko o nazwie "apikey" oraz jako wartość należy podać wygenerowany klucz
- Najłatwiej można utworzyć ciastko sklejając w konsoli następujący kod `cookies.create(apikey,'klucz_web_api_sklepu',60*60*24*7*30)`
- Aby wyświetlać zawartość zmiennej PHP należy wkleić w dowolnym miejscu `{$_|@debug_print_var}` gdzie gwiazdka to nazwa zmiennej
- Do adresu URL dowolnej podstrony dokleić parametr: `?show_tpl_variables=1`
Zmienne przesyłane do szablonu różnych podstron sklepu (strona główna, kategoria, produkt) mogą być inne, szczególnie w niektórych podstronach mogą być dodatkowe zmienne, których nie ma w innej części sklepu. Na przykład w podstronie produktu będzie dostępna jego cena, nazwa, opis itd. W podstronie zamówienia będzie suma, rabaty itd.

6. Tworzenie stylu szablonu przy pomocy SCSS

- Style SCSS są to prekompilowane style CSS pozwalające szybciej oraz wydajniej pracować z arkuszami CSS, dodatkowo zapewniają porządek w kodzie i łatwość we wprowadzaniu zmian
- Oficjalna dokumentacja SCSS znajduje się tutaj (http://sass-lang.com/documentation/file.SASS_REFERENCE.html)
- Właściwości CSS które do poprawnego działania na wszystkich przeglądarkach wymagają prefixów, można używać poprzez funkcje Mixins w dostępnej bibliotece
- Zmienne SCSS (`$variable`) służą do podpinania kolorów oraz rodzajów fontów i wielkości tekstu



7. Zmienne SCSS

- \$mainColor - kolor główny sklepu
- \$additionalColor - dodatkowy kolor sklepu (kontrast)
- \$priceColor - kolor ceny produktu
- \$textColorOnButtons - kolor czcionki na buttonach
- \$backgroundColorFooter - kolor tła stopki
- \$backgroundColorTopMenu - kolor tła menu górnego
- \$topMenuTextColor - kolor czcionki w menu górnym
- \$topMenuTextColorHover - kolor czcionki w menu górnym po najechaniu
- \$footerTextColor - kolor czcionki w stopce
- \$footerTextColorHover - kolor czcionki w stopce po najechaniu

8. JavaScript i akcje wykonywane dynamicznie

- Dużo akcji wykonywanych na sklepie np. dodawanie do koszyka, wykonywanych jest dynamicznie w technologii AJAX
- Dane do wysłania są przechowywane w data atrybutach, który jest pobierany i wysyłany w AJAX'ie
- Nazewnictwo dla akcji wykonywanych dynamicznie są to klasy .core_* gdzie gwiazdka to nazwa wykonywanej akcji np. .core_addToCart