Artykuł

Dzielenie projektu

Idź w górę Poprzedni Nastepny

Aby wykorzystać jak najwięcej rzeczy które daje nam Sass, podzielimy teraz cały nasz projekt na mniejsze pliki. Każda sekcja, przyciski czy style generyczne będą umieszczone w osobnych plikach tak aby łatwo dało się nawigować po całym projekcie i go rozwijać.

Wiemy też, że Sass oferuje możliwość zagnieżdżania. Cały projekt BestShop napisany jest według założeń metodologii BEM (`Block__Element--Modifier`). Tzn. że taki kod CSS:

Kopiuj

```
.form {
  display: flex;
  flex-direction: column
}

.form__field {
  display: flex;
  flex-direction: column
}

.form__field:first-of-type {
  margin-bottom: 42px
}
```

w Sass będzie wyglądał tak:

Kopiuj

```
.form {
  display: flex;
  flex-direction: column;

  &__field {
    display: flex;
    flex-direction: column;

    &:first-of-type {
      margin-bottom: 42px;
    }
  }
}
```

Podczas omówienia rozwiązania tego warsztatu, będziemy przerabiać kod CSS tak aby jak najlepiej odpowiadał on standardom języka Sass.

Podział projektu na pliki

Pierwszym krokiem będzie stworzenie folderu `scss` w **głównym katalogu** repozytorium z projektem. W folderze tym tworzymy plik `main.scss` a także foldery:

- `elements` - sekcje, przyciski,
- `generic` - pliki resetu, grid,
- `settings` - zmienne, mixiny.

Settings

scss/settings/_colors.scss

Rozpocznijmy od folderu `settings`. Stwórz w nim plik `_color.scss`. Umieścimy tam zmienne z kolorami które wykorzystujemy w projekcie. Podczas 1 modułu korzystaliśmy ze zmiennych CSS3, teraz przeróbmy to na zmienne Sass.

Kopiuj

```
$color-primary: #08A6E4;
$color-primary-shadow: rgba(7, 172, 230, .16);
$color-redish: #FB3B64;
$color-green: #55DFB4;
$color-black: #000000;
$color-white: #ffffff;
$color-grey: #A5A5A5;
$color-light-grey: #F7F7F7;
```

scss/settings/_fonts.scss

Kolejny plik to `_fonts.scss`. Będzie on zawierał deklaracje `font-family` dla dwóch fontów z których korzystamy.

Kopiuj

```
$font-raleway: "Raleway", sans-serif;
$font-open-sans: "Open Sans", sans-serif;
```

scss/settings/_mixins.scss

W naszym projekcie będziemy również potrzebować mixinów odpowiedzialnych za nastawianie odpowiednich reguł `@media`. Stwórzmy plik `_mixins.scss` który na razie będzie pusty. Uzupełnimy go w kolejnych krokach.

scss/settings/_index.scss

Pozostał nam do stworzenia plik `_index.scss` w którym zaimportujemy wszystkie pliki z tego folderu i utworzymy zmienną określającą zaokrąglenie przycisków.

Kopiuj

```
@import "fonts";
@import "colors";
@import "mixins";

$button-radius: 35px;
```

Generic

scss/generic/_reset.scss

Kolejny folder to `generics`. Umieścimy tam plik `_reset.scss` do którego wklej treść zawartą tutaj: https://gist.githubusercontent.com/DavidWells/18e73022e723037a50d6/raw/ead7e72d11c847ad9b81f094686543!

Elements

Folder `elements` będziemy wypełniać wraz z kolejnymi krokami naszego projektu.

scss/main.scss

Wróćmy do pliku `main.scss` i zaimportujmy pliki które do tej pory stworzyliśmy. Na początek importujemy plik resetu z folderu `generic`, a następnie plik `_index.scss` z folderu `settings`. Możemy ten import uprościć, zapisując po prostu `settings`.

Kopiuj

```
@import "generic/reset";

@import "settings";
```

Jak oceniasz artykuł ?