Jak stylować by dobrze wyglądać

SASS / BEM / Flexbox

Paweł Sołtys

Front-End Developer @ Straal

Plan zajęć

1. SASS - wprowadzenie, przydatne ficzery
2. BEM - zasady i jak sensownie je wdrażać
3. Flexbox - idea działania i jak uniknąć podstawowych błędów

SASS - Syntactically Awesome Style Sheets

- Preprocessor CSS

  Pliki, na których pracujemy są kompilowane do klasycznego CSS, który dopiero jest ładowany w przeglądarce. Sam SASS jest obecny jedynie w fazie developmentu

- Przejrzysta organizacja

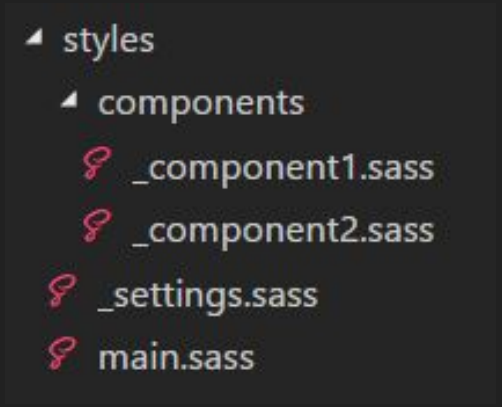
  Stosowanie partiali i mixinów pozwala na czytelną modularyzację kodu. Style można dzielić na kategorie i komponenty

- Parametryzacja kodu

  Zmienne pozwalają na trzymanie powtarzających się wartości w jednym miejscu oraz budować wygodne pliki konfiguracyjne

SASS - Partial

Pliki



Import

main.sass

```
1   @import 'settings'
2
3   @import 'componenets/component1'
4   @import 'componenets/component2'
5
```

Źródło: https://sass-lang.com/documentation/file.SASS_REFERENCE.html#partials

SASS - Zmienna

CSS

```css
.button--disabled {
    background: #bbb;
}

.link--inactive {
    color: #bbb;
}
```

SASS

```sass
$muted-color: #bbb

.button--disabled
    background: $muted-color

.link--inactive
    color: $muted-color
```

Nie należy przesadzać!

Uwspólniamy jedynie te wartości, które mają takie samo bądź bardzo zbliżone przeznaczenie

Źródło: https://sass-lang.com/documentation/file.SASS_REFERENCE.html#variables_

SASS - Mixin

- Predefiniowany, reużywalny zestaw właściwości. Pozwala umieścić powtarzający się w projekcie kod w jednym miejscu

- Przy wywoływaniu można przekazać do niego parametry, które następnie używane są do określenia stylu wynikowego

- Uwaga! Każde użycie mixina tworzy kopię przypisanych mu właściwości (nadmierne użycie == znacznie powiększony rozmiar pliku .css

Źródło: https://sass-lang.com/documentation/file.SASS_REFERENCE.html#mixins

SASS - Extend

- Podobnie jak mixin, pozwala wielokrotnie używać raz napisany kod
- Extend użyty pod danym selektorem powoduje, że selektor ten dopisany zostaje do oryginalnie zdefiniowanego (DRY w pliku wynikowym)
- ALE nie można przekazywać do niego parametrów

Źródło: https://sass-lang.com/documentation/file.SASS_REFERENCE.html#extend

SASS - Nesting

SASS

```sass
.component
    padding: 10px 20px
    border: 1px solid $component-border-color

    &__element
        font-size: 20px

        &--modifier
            color: $component-element-modifier-color

    &__another-element
        padding: 10px 0
```

CSS

```css
.component {
    padding: 10px 20px;
    border: 1px solid #55ffbb;
}

.component__element {
    font-size: 20px;
}

.component__element--modifier {
    color: #ff0000;
}

.component__another-element {
    padding: 10px 0;
}
```

Bardzo przydatne przy stosowaniu BEMa!

Źródło: https://sass-lang.com/documentation/file.SASS_REFERENCE.html#nested_rules

BEM - Block Element Modifier

- Metodologia nazewnictwa klas CSS
- Komponentyzacja
  Głównym celem BEMa jest łatwy podział kodu na moduły, które ze sobą nie konfliktują
- Uproszczenie selektorów
  Stosowanie BEM sprawia, że selektory są unikalne, co ogranicza stosowanie głębokich selektorów (i jednocześnie zmniejsza prawdopodobieństwo wystąpienia konfliktów)

Źródło: http://getbem.com/

BEM - Struktura

```html
<button class="button">
    <span class="button__icon">x</span>
    <span class="button__text button__text--important">
        Close
    </span>
</button>
```

```scss
.button
    display: inline-block
    background: $button-background

    &--alert
        background: $button-background-alert

    &__icon
        display: inline-block
        margin-right: 5px
        vertical-align: middle

    &__text
        display: inline-block
        vertical-align: middle

        &--important
            text-transform: uppercase
```

BEM - Nie zagnieżdżamy elementów!

```html
<div class="panel">
    <div class="panel__body">
        <form class="panel__body__form">
            [...]
        </form>
    </div>
</div>
```

```scss
.panel
    border: 1px solid $panel-border-color

    &__body
        padding: 20px

        &__form
            display: block
            background-color: $form-background-color
```

Nazewnictwo BEMowe nie jest ściśle powiązane ze strukturą HTML

Klasy możemy podpinać niezależnie od poziomu zagłębienia w drzewie DOM i nadal powinny one stosować nazewnictwo niezagnieżdżone

Flexbox

- Dedykowana konstrukcja do układania layoutów

  Zastępuje stare podejścia, które nie były do tego przeznaczone ale używano ich z braku dedykowanego rozwiązania

- Zależność kontener-element

  Kontener flexowy wpływa na rozkład bezpośrednio podlegających mu elementów

- Kontener a jego otoczenie

  Kontener flexowy zachowuje się jak typowy element blokowy (div) gdy nadamy mu “display: flex” albo jak element inline’owy gdy nadamy mu “display: inline-flex”

Flexbox - zależność kontener-element

- Fundamentalna koncepcja w Flexboxie

```
<body>
    <div id="centered">Cześć, jestem na środku</div>
</body>
```

```
body {
    display: flex;
    align-items: center;
    justify-content: center;
}
```

```
#centered {
    display: flex;
    align-items: center;
    justify-content: center;
}
```

Flexbox - właściwości kontenera/elementu

Kontener

- display (flex | inline-flex)
- flex-direction (row | column | [...])
- Justify-content (flex-start | flex-end | [...])
- flex-wrap
- align-content

Element

- flex-grow
- flex-shrink
- Flex-basis
- flex
- align-self

Flexbox - właściwość ‘flex’

Właściwość ‘flex’ jest złączeniem właściwości ‘flex-grow’, ‘flex-shrink’ oraz ‘flex-basis’.

```
flex: auto;                     /* flex: 1 1 auto */
flex: initial;                  /* flex: 0 1 auto */
flex: none;                     /* flex: 0 0 auto */
flex: <positive-number>;        /* flex: <positive-number> 1 0 */

flex: 2;                  /* flex-grow */
flex: 20px;                     /* flex-basis */
flex: 2 1;                      /* flex-grow, flex-shrink */
flex: 2 10px;                   /* flex-grow, flex-basis */
flex: 2 1 10px;                 /* flex-grow, flex-shrink, flex-basis */
```

Gdy jako wartość zadamy jedną lub dwie wartości bez jednostki, ‘flex-basis’ustawiane jest na 0!

Źródło: https://developer.mozilla.org/en-US/docs/Web/CSS/flex

Materiały

- https://sass-lang.com/guide
- https://sass-lang.com/documentation/file.SASS_REFERENCE.html
- http://getbem.com/
- https://css-tricks.com/snippets/css/a-guide-to-flexbox/
- https://developer.mozilla.org/en-US/docs/Glossary/Flexbox

Pytania?

Dziękuję!

Kontakt:
pawel.soltys@straal.com