Wprowadzenie do SASS

Spis treści

- SASS – podstawy
- Zmienne
- Mixin
- Extend
- Funkcje
- Warunek if
- Pętle – each, for, while
- Funkcje matematyczne

Czym jest SASS?

- SASS (*Syntactical Awesome Stylesheet*) jest językiem skryptowym, którego kod przetwarzany jest do plików wynikowych kaskadowych arkuszy stylów - CSS.
- Dwie składnie kodowania:
  - Sassy CSS (.scss) – wywodzi się z CSS
  - Wywodzi się z języka skryptowego HAML, gdzie pomijane są klamry oraz średniki (.sass)

Instalacja SASS

- http://rubyinstaller.org/downloads/
- Wiersz poleceń z obsługą Ruby - **Start Command Prompt with Ruby** (dla Windows)
- Instalacja przy użyciu komendy
  `gem install sass`
- `sass -v`
- Kompilacja
  `sass plik.scss:plik.css`

SASS - komentarze

- SASS dodaje opcję komentarzy jednej linii, nie są one jednak widoczne w plikach .css

main.scss

```
// komentarz niewidoczny
// dla *.css


/* komentarz widoczny dla
   *.css
*/
```

main.css

```
/*
   komentarz widoczny dla
   *.css
*/
```

SASS – importowanie

- W CSS rzadko korzysta się z `@import`
- `@import` w .scss oraz .sass wykonywany jest w trakcie kompilacji i zapisywany do jednego pliku .css
- Dopisywanie rozszerzeń plików są opcjonalne
- `@import "buttons";`

Nesting – zagnieżdżanie

main.scss

```
.content {
    font-size: 12px;
    color: green;
    p {
        margin: 15px 0;
    }
    h1 {
        margin: 30px 15px;
        border: 2px solid red;
    }
}
```

main.css

```
.content {
    font-size: 12px;
    color: green;
}
.content p {
    margin: 15px 0;
}
.content h1 {
    margin: 30px 15px;
    border: 2px solid red;
}
```

Parent Selector

- W trakcie zagnieżdżania można wykorzystać operator &, który odpowiada za selektor rodzicielski.

main.scss

```
.content {
   font-size: 12px;
   p {
      margin: 15px 0;
   }
   h1 {
      margin: 30px 15px;
   }
   .callout {
      color: red;
   }
   &.callout {
      color: green;
   }
}
```

main.css

```
.content {
   font-size: 12px;
}
.content p {
   margin: 15px 0;
}
.content h1 {
   margin: 30px 15px;
}
.content .callout {
   color: red;
}
.content.callout {
   color: red;
}
```

Parent Selector

- Bardzo często używany w połączeniu z pseudo klasami.

main.scss

```
a {
    color: blue;
    &:hover {
        color: red;
    }
    &:active {
        color: green;
    }
}
```

main.css

```
a {
    color: blue;
}
a:hover {
    color: red;
}
a:active {
    color: green;
}
```

Parent Selector

- Selektory mogą być również dodawane **przed** znacznikiem &.

main.scss

```
.contact {
    float: left;
    width: 300px;
    .footer & {
        width: 400px;
    }
}
```

main.css

```
.contact {
    float: left;
    width: 300px;
}
.footer .contact {
    width: 400px;
}
```

Zagnieżdżanie

main.scss

```
.content {
   color: blue;
   .callout {
      h2 {
         a {
            &:hover {
               color: red;
            }
         }
      }
   }
}
```

main.css

```
.content {
   color: blue;
}
.content .callout h2 a:hover {
   color: red;
}
```

Zmienne – deklaracja i korzystanie

- Zmienne w SASS deklarujemy za pomocą znacznika $, np. `$zmienna`.

main.scss

```
$color: #232323;

.contact {
   border: 1px solid $color;
   li {
      color: $color;
   }
}
```

main.css

```
.contact {
   border: 1px solid #232323;
}
.contact li {
   color: #232323;
}
```

Zmienne - typy

- Boolean

```
$radius: false;
$shadow: true;
```

- Numbers – nie trzeba podawać jednostek

```
$font-size: 1.5em;
$line-height: 1.2;
$border: 3px;
```

Zmienne - typy

- Colors

```
$color: red;
$border: #rgba(0, 255, 0, 0.5);
$shadow: #333;
```

- Strings – można deklarować z lub bez „”

```
$header: ‘Helvetica’;
$font-family: Arial;
$message: ”Loading...”;
```

Zmienne - typy

- Lists

```
$authors: pawel, mirek, andrzej, krzysztof;
$margin: 30px 0 20px 80px;
```

Zmienne - zasięg

- Zmienne ustawiane wewnątrz deklaracji (pomiędzy { }) nie mogą być używane poza tym blokiem!

main.scss

```
p {
    $color: #ccc;
    border: 2px solid $color;
}
h1 {
    border: 2px solid $color;
}
```

main.css

```
Syntax error: Undefined
variable: „$color”.
```

Zmienne - zasięg

- Ustawiając nowe wartości dla zmiennych zadeklarowanych poza blokiem deklaracji, zmienia wartość na stałe.

main.scss

```
$color: #232323;

.contact {
    $color: #555555;
    background: $color;
}
h1 {
    color: $color;
}
```

main.css

```
.contact {
    background: #555555;
}
h1 {
    color: #555555;
}
```

Zmienne - Interpolacja

- Używając znacznika `#{$zmienna}` możemy używać zmiennych w selektorach, nazwach właściwości czy stringach.

main.scss

```
$side: top;

body {
    position: relative;
    #{$side}: -0.5em;
}
.callout-#{$side} {
    background: blue;
}
```

main.css

```
body {
    position: relative;
    top: -0.5em;
}
.callout-top {
    background: blue;
}
```

Mixin

main.css

```
.btn-a {
   background: #777;
   border: 1px solid #ccc;
   font-size: 1em;
   text-transform: uppercase;
}
.btn-b {
   background: #ff0;
   border: 1px solid #ccc;
   font-size: 1em;
   text-transform: uppercase;
}
```

Mixin - deklaracja

main.scss

```
@mixin button {
   border: 1px solid #ccc;
   font-size: 1em;
   text-transform: uppercase;
}
.btn-a {
   @include button;
   background: #777;
}
.btn-b {
   @include button;
   backgroudn: #ff0;
}
```

main.css

```
.btn-a {
   border: 1px solid #ccc;
   font-size: 1em;
   text-transform: uppercase;
   background: #777;
}
.btn-b {
   border: 1px solid #ccc;
   font-size: 1em;
   text-transform: uppercase;
   background: #ff0;
}
```

Mixin – argumenty

main.scss

```
@mixin box-sizing($x) {
   -webkit-box-sizing: $x;
   -mox-box-sizing: $x;
   box-sizing: $x;
}
.content {
   @include box-sizing(border-box);
   border: 1px solid #ccc;
   padding: 20px;
}
.callout {
   @include box-sizing(content-box);
}
```

main.css

```
.content {
   -webkit-box-sizing: border-box;
   -moz-box-sizing: border-box;
   box-sizing: border-box;
   border: 1px solid #ccc;
   padding: 20px;
}
.callout {
   -webkit-box-sizing: content-box;
   -moz-box-sizing: content-box;
   box-sizing: content-box;

}
```

Mixin – argumenty domyślne

main.scss

```
@mixin box-sizing($x: border-box) {
   -webkit-box-sizing: $x;
   -mox-box-sizing: $x;
   box-sizing: $x;
}
.content {
   @include box-sizing;
   border: 1px solid #ccc;
   padding: 20px;
}
.callout {
   @include box-sizing(content-box);
}
```

main.css

```
.content {
   -webkit-box-sizing: border-box;
   -moz-box-sizing: border-box;
   box-sizing: border-box;
   border: 1px solid #ccc;
   padding: 20px;
}
.callout {
   -webkit-box-sizing: content-box;
   -moz-box-sizing: content-box;
   box-sizing: content-box;

}
```

Mixin – więcej argumentów

main.scss

```
@mixin button($radius, $color)
{
   border-radius: $radius;
   color: $color;
}
.btn-a {
   @include button(4px, #000);
}
```

main.css

```
.btn-a {
   border-radius: 4px;
   color: #000;
}
```

Mixin – więcej argumentów

main.scss

```
@mixin button($radius, $color: #000)
{
   border-radius: $radius;
   color: $color;
}
.btn-a {
   @include button(4px);
}
```

main.css

```
.btn-a {
   border-radius: 4px;
   color: #000;
}
```

Mixin – interpolacja

main.scss

```
@mixin highlight($color, $side) {
   border-#{$side}-color: $color;
}
.btn-a {
   @include highlight(#ff0, right);
}
```

main.css

```
.btn-a {
   border-right-color: #ff0
}
```

Extend

main.scss

```
.btn-a {
   background: #777;
   border: 1px solid #ccc;
   font-size: 1em;
   text-transform: uppercase;
}
.btn—b {
   @extend btn-a;
   background: #ff0;
}
```

main.css

```
.btn-a,
.btn-b {
   background: #777;
   border: 1px solid #ccc;
   font-size: 1em;
   text-transform: uppercase;
}
.btn—b {
   background: #ff0;
}
```

Zagnieżdżanie + Extend

main.scss

```
.content {
   border: 1px solid #ccc;
   padding: 20px;
   h2 {
      font-size: 3em;
      margin: 20px 0;
   }
}
.callout {
   @extend .content;
   background: #ddd;
}
```

main.css

```
.content,
.callout {
   border: 1px solid #ccc;
   padding: 20px;
}
.content h2,
.callout h2 {
   font-size: 3em;
   margin: 20px 0;
}
.callout {
   background: #ddd;
}
```

Extend – problemy

main.scss

```
.btn-a {
   background: #777;
   border: 1px solid #ccc;
   font-size: 1em;
   text-transform: uppercase;
}
.btn—b {
   @extend btn-a;
   background: #ff0;
}
.sidebar .btn-a {
   text-transform: lowercase;
}
```

main.css

```
.btn-a,
.btn-b {
   background: #777;
   border: 1px solid #ccc;
   font-size: 1em;
   text-transform: uppercase;
}
.btn—b {
   background: #ff0;
}
.sidebar .btn-a,
.sidebar .btn-b {
   text-transform: lowercase;
}
```

Selektor zastępczy

- Dopóki `.btn-b` rozszerza `.btn-a` każde wystąpienie, które modyfikuje `.btn-a` modyfikuje także `.btn-b`
- Możemy wykorzystać selektory zastępcze
- Deklarujemy je używając `%`
- Mogą być rozszerzane, ale nigdy nie są selektorem same w sobie

Selektor zastępczy

main.scss

```
%btn {
   background: #777;
   border: 1px solid #ccc;
   font-size: 1em;
   text-transform: uppercase;
}
.btn-a {
   @extend %btn;
}
.btn—b {
   @extend %btn;
   background: #ff0;
}
.sidebar .btn-a {
   text-transform: lowercase;
}
```

main.css

```
.btn-a,
.btn-b {
   background: #777;
   border: 1px solid #ccc;
   font-size: 1em;
   text-transform: uppercase;
}
.btn—b {
   background: #ff0;
}
.sidebar .btn-a {
   text-transform: lowercase;
}
```

Funkcje

main.scss

```
@function fluidize($target, $context) {
   @return ($target / $context) * 100%;
}
.sidebar {
   width: fluidize(350px, 1000px);
}
```

main.css

```
.sidebar {
   width: 35%;
}
```

If, else if, else

main.scss

```
$theme: pink;

header {
    @if $theme == dark {
        background: #000;
    } @else if $theme == pink {
        background: pink;
    } @else {
        background: #fff;
    }
}
```

main.css

```
header {
    background: pink;
}
```

Porównania

- == równe
- != nie równe
- > większe niż
- >= większe niż lub równe
- < mniejsze niż
- <= mniejsze niż lub równe

Each

- Za pomocą `@each` możemy przejść po całej liście.

main.scss

```
$authors: maciej pawel michal;

@each $author in $authors {
    .author-#{$author} {
        background: url(author-
#{$author}.jpg)
    }
}
```

main.css

```
.author-maciej {
    background: url(author-
maciej.jpg);
}
.author-pawel {
    background: url(author-
pawel.jpg);
}
.author-michal {
    background: url(author-
michal.jpg);
}
```

Pętla for

main.scss

```
$i: 1;

.item {
   position: absolute;
   right: 0;
   @for $i from 1 through 4 {
      &.item-#{$i} {
         top: $i * 30px;
      }
   }
}
```

main.css

```
.item {
   postion: absoulte;
   right: 0;
}
.item.item-1 {
   top; 30px;
}
.item.item-2 {
   top; 60px;
}
.item.item-3 {
   top; 90px;
}
.item.item-4 {
   top; 120px;
}
```

Pętla While

main.scss

```
$i: 1;

.item {
   position: absolute;
   right: 0;
   @while $i < 4 {
      &.item-#{$i} {
         top: $i * 30px;
      }
      $i: $i + 1;
   }
}
```

main.css

```
.item {
   postion: absoulte;
   right: 0;
}
.item-1 {
   top; 30px;
}
.item-2 {
   top; 60px;
}
.item-3 {
   top; 90px;
}
```

Funkcje matematyczne

- W SASS możemy korzystać z wszystkich operacji numerycznych:
  - Dodawanie +
  - Odejmowanie -
  - Mnożenie *
  - Dzielenie /
  - Dzielenie modulo %

Funkcje matematyczne

- `round($number)` – zaokrąglenie do liczby całkowitej
- `ceil($number)` – zaokrąglenie w górę
- `floor($number)` – zaokrąglenie w dół
- `abs($number)` – wartość bezwzględna
- `min($list)` – minimalna wartość z listy
- `max($list)` – maksymalna wartość z listy
- `percentage($number)` – konwertowanie na wartość procentową

Funkcje matematyczne

main.scss

```
h2 {
    line-height: ceil(1.2);
}
```

main.css

```
h2 {
    line-height: 2;
}
```

main.scss

```
$context: 1000px;

.sidebar {
    width: percentage(450px/
$context);
}
```

main.css

```
.sidebar {
    width: 45%;
}
```

Dziękuję za uwagę