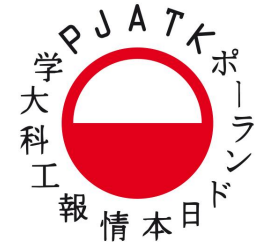

Technologie Internetu - wykład

Preprocesory CSS



Czym jest preprocesor

- Narzędzie, które wygeneruje docelowy kod we właściwym języku programowania.
- Kod będzie generowany z kodu napisanego przez nas w innym języku.
- W tym wypadku, docelowy kod to CSS, rozumiany przez przeglądarkę.

Po co preprocesor CSS

- Łatwość pisania kodu.
- Automatyzacja regułek i wartości w CSS.
- Elementy programowania w CSS - większa dynamika.
- Dobra organizacja kodu.

Najpopularniejsze preprocesory CSS

- SASS
- LESS
- Stylus



SASS

- Powstał w 2006 roku, aby uczynić kod CSS ładniejszym.
- sass-lang.com
- Oryginalna składnia, SASS, różni się od CSS.
Nie ma `}`; tylko wcięcia i nowe linie.
- Nowsza składnia, SCSS, bardziej przypomina CSS.



SASS

SASS

```
.item
  padding: 10px 1%
  flex-basis: 30%

  img
    width: 43%
    margin: 5px 1%
```

SCSS

```
.item {
  padding: 10px 1%;
  flex-basis: 30%;

  img {
    width: 43%;
    margin: 5px 1%;
  }
}
```

Jak użyć SASS

Opcje do wyboru

- Wtyczka do edytora, np. Live Sass.
- Kompilator online, np:
sass-lang.com/playground/
- Biblioteka sass w node, kompilująca plik podczas edycji.

Wynikowy CSS trafia na stronę www

```
npm install sass
```

```
sass --watch styl.scss styl.css
```

```
<link rel="stylesheet/less"  
type="text/css" href="plik.less"  
>
```

```
<script src="ADRES_PLIKU_JS" >  
</script>
```

SASS - zmienne

CSS

```
body {  
  background: #B30011;  
}
```

SCSS

```
$colo: #B30011;  
  
body {  
  background: $colo;  
}
```

SASS - mixiny

CSS

```
body {  
  margin: 0;  
  padding: 0;  
}
```

SCSS

```
@mixin x {  
  margin: 0;  
  padding: 0;  
}  
  
body {  
  @include x;  
}
```

SASS - zagnieżdżanie selektorów

CSS

```
.item {  
  padding: 10px 1%;  
  flex-basis: 30%;  
}  
  
.item img {  
  width: 43%;  
  margin: 5px 1%;  
}
```

SCSS

```
.item {  
  padding: 10px 1%;  
  flex-basis: 30%;  
  
  img {  
    width: 43%;  
    margin: 5px 1%;  
  }  
}
```

SASS - pseudoklasy

CSS

```
.item {  
  color: red;  
}  
  
.item:hover {  
  color: blue;  
}
```

SCSS

```
.item {  
  color: red;  
  
  &:hover {  
    color: blue;  
  }  
}
```

SASS - media queries

CSS

```
@media screen {  
  .klasa {color: green;}  
}  
  
@media screen and (min-width:  
768px) {  
  .klasa {color: blue;}  
}
```

SCSS

```
.klasa{  
  
  @media screen {  
    color: green;  
    @media (min-width:768px) {  
      color: blue;  
    }  
  }  
}
```

SASS - operator

CSS

```
body {  
  padding: 20px;  
}
```

SCSS

```
$val: 10px * 2;  
  
body {  
  padding: $val;  
}
```

SASS - funkcje

CSS

```
body {  
  color: #e60016;  
}
```

SCSS

```
$colo: #B30011;  
  
body {  
  color: lighten($colo, 10%)  
}
```

SASS - namespace

CSS

```
.klasa {  
    padding: 0  
}
```

SCSS

```
//plik _sub.scss  
.klasa {  
    padding: 0  
}
```

```
//główny plik scss  
@use 'sub';
```

SASS - zasięg zmiennych

CSS

```
#page #header {  
  color: red;  
}  
  
#page2 #header {  
  color: white;  
}
```

SCSS

```
$var: red;  
#page {  
  #header { color: $var; }  
}  
  
#page2 {  
  $var: white;  
  #header { color: $var; }  
}
```

SASS - komentarze

SCSS

```
/* Więcej  
niż jedna linijka */  
@var: xxx;  
  
// jedna linijka  
@var: white;
```

LESS

- Inspirowany przez SASS ale lżejszy.
- Oryginalnie napisany w Ruby, obecnie działa w JavaScript.
- Kompiluje się do CSS, który jest używany przez stronę www.
- lesscss.org

Jak użyć LESS

Opcje do wyboru

- Wtyczka do edytora kodu, np. Easy LESS.
- Kompilator online, np:
lesscss.org/less-preview
- Biblioteka less w node, kompilująca plik podczas edycji.
- Biblioteka JavaScript, kompilująca plik bezpośrednio na stronie.

```
npm install less
```

```
lessc styles.less styles.css
```

```
<link rel="stylesheet/less"  
type="text/css" href="plik.less"  
>
```

```
<script src="ADRES_PLIKU_JS" >  
</script>
```

LESS - zmienne

CSS

```
#div1 {  
  color: #554433;  
}  
#div2 {  
  color: #554433;  
}  
#div3 {  
  color: #554433;  
}
```

LESS

```
@kolor: #554433;  
  
#div1 {color: @kolor;}  
#div2 {color: @kolor;}  
#div3 {color: @kolor;}
```

LESS - mixiny

CSS

```
.klasa {  
    width: 100px; height: 200px;  
}  
#item1 {  
    color: black;  
    width: 100px; height: 200px;  
}  
#item2 {  
    color: red;  
    width: 100px; height: 200px;  
}
```

LESS

```
.klasa {  
    width: 100px; height: 200px;  
}  
#item1 {  
    color: black;  
    .klasa;  
}  
#item2 {  
    color: red;  
    .klasa;  
}
```

LESS - zagnieżdżanie selektorów

CSS

```
#div {  
    color: black;  
}  
#div #item1 {  
    width: 200px;  
}  
#div #item2 {  
    width: 100px;  
}
```

LESS

```
#div {  
    color: black;  
    #item1 {  
        width: 200px;  
    }  
    #item2 {  
        width: 100px;  
    }  
}
```

LESS - pseudoklasy

CSS

```
a {  
  font-size: 12px;  
}  
a:hover {  
  color: red;  
}
```

LESS

```
a {  
  font-size: 12px;  
  &:hover {  
    color: red;  
  }  
}
```

LESS - media queries

CSS

```
@media screen {  
  .klasa {color: green;}  
}  
  
@media screen and (min-width:  
768px) {  
  .klasa {color: blue;}  
}
```

LESS

```
.klasa{  
  
  @media screen {  
    color: green;  
    @media (min-width:768px) {  
      color: blue;  
    }  
  }  
}
```

LESS - operator

CSS

```
.myclass {  
  font-size: 20px;  
  line-height: 12px;  
}
```

LESS

```
@fSize: 10px;  
@two: 2px;  
  
.myclass {  
  font-size: @fSize * 2;  
  line-height: @fSize + @two;  
}
```

LESS - funkcje

CSS

```
.class {  
  width: 50%;  
  color: #f6430f;  
  background-color: #f8b38d;  
}
```

LESS

```
@base: #f04615;  
@width: 0.5;  
  
.class {  
  width: percentage(@width);  
  color: saturate(@base, 5%);  
  background-color:  
    spin(lighten(@base, 25%), 8);  
}
```

LESS - namespace

CSS

```
#div1 p {  
  color: red;  
}  
#div2 {  
  color: red;  
}
```

LESS

```
#div1{  
  p{color: red}  
}  
  
#div2{  
  #div1 > p  
}
```

LESS - zasięg zmiennych

CSS

```
#page #header {  
  color: red;  
}  
#page2 #header {  
  color: white;  
}
```

LESS

```
@var: red;  
#page {  
  #header { color: @var; }  
}  
  
#page2 {  
  #header { color: @var; }  
  @var: white;  
}
```

LESS - komentarze, importy itp

LESS

```
/* Więcej  
niż jedna linijka */  
@var: xxx;  
  
// jedna linijka  
@var: white;  
  
@import "plik"; // importujemy plik.less
```

Zmienne w CSS vs Zmienne w SASS/LESS

- Zmienne w CSS są dynamiczne.
- Przeglądarka przelicza wartości w momencie otwarcia strony www.
- Zmienne w preprocesorach, przestają być zmiennymi po kompilacji do CSS.
- Na stronie www są to po prostu wartości reguł CSS.