

PREPROCESORY

CSS

Arkusze CSS

2

- Statyczne reguły stylów
- Płaska struktura
 - selektor { atrybuty }
 - selektor { atrybuty }
 - selektor { atrybuty }
 - Nie odzwierciedla struktury dokumentu HTML
- Brak możliwości definiowania zbiorów atrybutów, które można wykorzystać w wielu miejscach arkusza stylów CSS
 - np. kolor, rozmiar czcionki, wysokość linii

Arkusze CSS

3

- Brak możliwości definiowania stałych do użycia w wielu miejscach arkusza CSS
 - ▣ np. kolorów wykorzystywanych w wielu miejscach
 - ▣ Jeśli kolor ma zostać zmieniony należy poprawić go w każdym miejscu gdzie został użyty
 - Dużo pracy
 - Łatwo przeoczyć jakieś wystąpienie
- Brak arytmetyki na kolorach
 - ▣ np. kolor jaśniejszy o X od koloru bazowego strony

Arkusze CSS

4

- Arkusze mogą rozrastać się do dużych rozmiarów
- Trudne utrzymywanie i rozszerzanie stylów w dużych projektach

Preprocesory CSS

5

- Najbardziej popularne:
 - Sass (<http://sass-lang.com/>)
 - Less (<http://lesscss.org/>)
 - Stylus (<http://learnboost.github.io/stylus/>)
- W większości zbliżona funkcjonalność

Preprocessor Less

6

- Instalacja:

 - \$ npm install -g less

- Wygenerowanie standardowego arkusza CSS na podstawie arkusza Less:

 - \$ lessc styles.less > styles.css

 - ▣ Środowiska programistyczne mogą robić to automatycznie w momencie modyfikacji pliku .less
 - ▣ Do przeglądarki dystrybuowany jest statyczny plik CSS

Preprocesor Less

7

- W czasie wytwarzania aplikacji wygodne może być przetwarzanie stylów w locie:

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

```
<script src="less.js" type="text/javascript"></script>
```

- Wynikowe reguły CSS generowane po załadowaniu strony w przeglądarce
 - ▣ Nie należy stosować tego mechanizmu w wersji dystrybucyjnej strony
- Każdy poprawny arkusz CSS jest też poprawnym arkuszem less

Zmienne i arytmetyka na kolorach

8

```
@base-color: #2233AD;  
/* arytmetyka na kolorach: */  
@light-blue: @base-color + #222222;  
@dark-blue: @base-color - #222222;
```

```
#main-container {  
  background-color: @base-color;  
}
```

```
#footer {  
  background-color: @dark-blue;  
}
```


Wbudowane funkcje

9

```
.header {  
  color: saturate(@base-color, 5%);  
  background-color: lighten(@base-color, 25%);  
}
```

- Przykłady dostępnych funkcji:
 - `image-size("image.png");`
 - `convert(5cm, mm)`
 - `background-image: svg-gradient(to left, red, blue);`
 - Matematyczne: `ceil`, `floor`, `round`, `sqrt`, `abs`, `sin`...

Domieszki

10

```
/*definicja domieszki: */  
.rounded-corners{  
  border-radius: 4px;  
  -webkit-border-radius: 4px;  
  -moz-border-radius: 4px;  
}  
  
div.highlight {  
  color: red;  
  .rounded-corners; /* użycie domieszki */  
}
```

Domieszki z parametrami

11

```
.rounded-corners(@radius: 4px) {  
  border-radius: @radius;  
  -webkit-border-radius: @radius;  
  -moz-border-radius: @radius;  
}
```

```
div.highlight {  
  color: red;  
  .rounded-corners(8px); /* użycie domieszki */  
}
```

„Dziedziczenie”

12

```
div.post{
```

```
  /* atrybuty */
```

```
}
```

```
div.news:extend(div.post) {
```

```
  /* dziedziczy atrybuty div.post */
```

```
}
```

Zagnieżdżanie reguł

13

```
#header {  
  color: black;  
  
  nav { /* zamiast #header nav */  
    color: blue;  
  }  
  
  .logo { /* zamiast #header .logo */  
    width: 300px;  
  }  
}
```

Pętle

14

```
.generate-columns(@n, @i: 1) when (@i =< @n) {  
  .col-@{i} {  
    width: (@i * 100% / @n);  
  }  
  .generate-columns(@n, (@i + 1));  
}  
  
.generate-columns(12);
```

Wynikowy CSS:

```
.col-1 {  
  width: 8.3%;  
}  
.col-2 {  
  width: 16.6%;  
}  
.col-3 {  
  width: 25%;  
}  
/* ... */
```

15

Pytania?

Dziękuję za uwagę