



Dziedziczenie Metoda @extend

Czym jest extend?

Jeśli chcemy utworzyć styl zawierający wszystkie deklaracje innego stylu plus kilka dodatkowych możemy użyć dyrektywy `@extend`.

Składnia scss

```
.message {  
  border: 1px solid #ccc;  
  padding: 10px;  
  color: #333;  
}  
.success {  
  @extend .message;  
  border-color: green;  
}  
.error {  
  @extend .message;  
  border-color: red;  
}
```

Rezultat css

```
.message, .success, .error {  
  border: 1px solid #ccc;  
  padding: 10px;  
  color: #333;  
}  
.success {  
  border-color: green;  
}  
.error {  
  border-color: red;  
}
```

@extend – niezamierzone style

Niezamierzone dodatkowe style

Składnia scss

```
a {  
  color: red;  
}  
.error {  
  @extend a;  
  border-color: red;  
}  
a:hover {  
  color: #333;  
  text-decoration: underline;  
}
```

Rezultat css

```
a:hover, .error:hover {  
  color: #333;  
  text-decoration: underline;  
}  
a, .error {  
  color: red;  
}  
.error {  
  border-color: red;  
}
```

Efekt? Każdy **.error** będzie podkreślał się i zmieniał kolor przy najechaniu kursorem.

@extend – zbędne selektory

Wiele niezamierzonych i zbędnych selektorów

W najlepszym przypadku

Plik wynikowy będzie zawierał dużo niepotrzebnego kodu i selektorów.

Składnia scss

```
#admin .nav a {  
  color: #333;  
}  
a {  
  color: #666;  
}  
#demo .body .link {  
  @extend a;  
}
```

W najgorszym przypadku

Nieoczekiwane nadpisanie stylów różnych elementów, trudne do wykrycia.

Rezultat css

```
#admin .nav a,  
#admin .nav #demo .body .link,  
#demo .body #admin .nav .link {  
  color: #333;  
}  
a, #demo .body .link {  
  color: #666;  
}
```

Różnica między @extend a @mixin

`@extend` i `@mixin` używane są w innych celach.

- `@extend` – służy do tego, aby wybrane selektory miały te same właściwości. Nie ma tutaj możliwości użycia argumentów.
- `@mixin` – pozwala na zagnieżdżenie dowolnego zestawu właściwości w różnych selektorach.
- `@extend` – może wykorzystać `@mixin`, jeśli został wstawiony dla selektora, który jest kopiowany.



`@extend`
vs
`@mixin`

@extend i placeholder %

- W przypadku metody `@extend` przydatny jest tzw. `placeholder selector %`.
- Jeśli selektor z placeholderem nie zostanie nigdzie użyty za pomocą `@extend`, to nie zostanie wygenerowany w CSS jako właściwość klasy lub identyfikatora.

@extend i placeholder %

Składnia scss

```
%message {  
  border: 1px solid #ccc;  
  padding: 10px;  
  color: #333;  
}  
.success {  
  @extend %message;  
  border-color: green;  
}  
.error {  
  @extend %message;  
  border-color: red;  
}
```

Rezultat css

```
.success, .error {  
  border: 1px solid #ccc;  
  padding: 10px;  
  color: #333;  
}  
.success {  
  border-color: green;  
}  
.error {  
  border-color: red;  
}
```

Dziedziczenie

- W praktyce w Sassie właściwości zostaną wpisane tylko raz.
- Po skompilowaniu oba selektory będą miały współdzielone właściwości.
- Te, które się różnią, wygenerują osobne, specyficzne selektory.

