



Zagnieżdżanie

Zagnieżdżanie

Jak wiemy, w zwykłym CSS możemy pisać tylko selektor pod selektorem tak jak pokazuje to poniższy kod:

```
.module {}  
.module .title {}  
.module .content {}  
.module .content .link: hover  
.module .content .link: hover {}
```

W SCSS możemy pisać tak samo jak w klasycznym SCSS, ale także możemy stosować zagnieżdżanie:

```
.module {  
  .title {  
  }  
  
  .content {  
    .link {  
      &: hover {}  
    }  
  }  
}
```

Oba zapisy są jednoznaczne, a ich zastosowanie zależy w zasadzie od preferencji piszącego kod.

Zagnieżdżanie - Przykład

HTML

```
<div class="main-content">
  <h3>Title</h3>
  <span>23.05.2016</span>
</div>
```

Składnia SCSS

```
.main-content {
  border: 1px solid red;

  h3 {
    margin: 10px 5px;
    font-size: 3em;
  }

  span {
    margin: 5px 0;
    font-size: 1em;
  }
}
```

Rezultat w CSS

```
.main-content {
  border: 1px solid red;
}

.main-content h3 {
  margin: 10px 5px;
  font-size: 3em;
}

.main-content span {
  margin: 5px 0;
  font-size: 1em;
}
```

Zagnieżdżanie - Przykład

HTML

```
<nav>
  <ul>
    <li><a href="#about">O nas</a></li>
    <li><a href="#team">Zespół</a></li>
    <li><a href="#contact">Kontakt</a></li>
  </ul>
</nav>
```

Składnia SCSS

```
nav {
  ul {
    margin: 0;
    padding: 0;
    list-style: none;

    > li {
      display: inline-block;
    }
  }

  a {
    display: block;
    padding: 6px 12px;
    text-decoration: none;
  }
}
```

Rezultat w CSS

```
nav ul {
  margin: 0;
  padding: 0;
  list-style: none;
}

nav ul > li {
  display: inline-block;
}

nav a {
  display: block;
  padding: 6px 12px;
  text-decoration: none;
}
```

Zagnieżdżanie - Selektor rodzica

Ampersand – &

W przypadku pseudoklas i pseudoelementów przydatny jest operator &.

Jest on odwołaniem się do rodzica danego selektora.

Składnia SCSS

```
a {
  &:hover {
    color: red;
  }
  &:visited {
    color: black;
  }
}
```

& w wygenerowanym pliku, jest powtórzeniem jego nazwy.

Rezultat w CSS

```
a:hover {
  color: red;
}
a:visited {
  color: black;
}
```

Dobre rady



Dobłą praktyką jest unikanie zbyt wielu poziomów zagnieźdżeń, gdyż taki kod jest trudny do utrzymania.

Przyjmijmy zasadę, że im mniej poziomów zagnieźdżeń, tym lepiej. Starajcie się ograniczać zagnieźdżanie do max 3 poziomów.