

Podsumowanie

Podsumowanie

W tej prezentacji znajdziesz podsumowanie najważniejszych rzeczy, które dzisiaj omówiliśmy.

Konfiguracja środowiska - Gulp

Aby pracować z Sass, wymagane jest użycie specjalnego środowiska programistycznego, ze względu na specyfikę tej technologii. Sama przeglądarka nie rozumie składni Sass dlatego potrzebujemy jakiegoś rozwiązania które kod Sass zmieni na CSS - język który rozumie przeglądarka.

Do takiej automatyzacji zadań służy nam Gulp. Poznaliśmy jego podstawową konfigurację dzięki której po każdej aktualizacji pliku `scss` czy `html` zostają przeprowadzone odpowiednie zadania, zapisywane pliki końcowe a także odświeżana przeglądarka.

Podsumowanie

Sass

Sass jest preprocesorem CSS.

```
nav {  
  ul {  
    margin: 0;  
    padding: 0;  
  }  
  
  li {  
    display: block;  
  }  
}
```

Zmienne

Zmienne w Sassie przechowują dowolną wartość CSS

```
$font-stack: Helvetica, sans-serif;  
$color-primary: #333;  
  
body {  
  font: 100% $font-stack;  
  color: $color-primary;  
}
```

Podsumowanie

Zagnieżdżanie

Zagnieżdżanie selektorów w Sass jest największą przewagą w stosunku do standardowego CSS. Pozwala ono na umieszczanie selektorów wewnątrz siebie - dokładnie tak jak to robimy z elementami HTML!

Dodatkowo dzięki selektorowi rodzica `&` możemy w łatwy sposób dołączać pseudoklasy, pseudoelementy czy inne klasy do głównego elementu.

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

Podsumowanie

Partials i importowanie

Sass pozwala na rozdzielenie kodu na wiele plików i zaimportowanie ich do jednego pliku zbiorczego metodą `@import`.

Aby uniknąć tworzenia niepotrzebnego pliku **css** wystarczy na początku nazwy wpisać znak **podkreślnika**.

Dobłą praktyką jest stworzenie jednego pliku `scss` składającego się jedynie z metod importowania.

`_reset.scss`

```
* {  
  box-sizing: border-box;  
  margin: 0;  
  padding: 0;  
}
```

`style.scss`

```
@import "reset";  
  
body {  
  font: sans-serif;  
  color: #fff;  
}
```

Podsumowanie

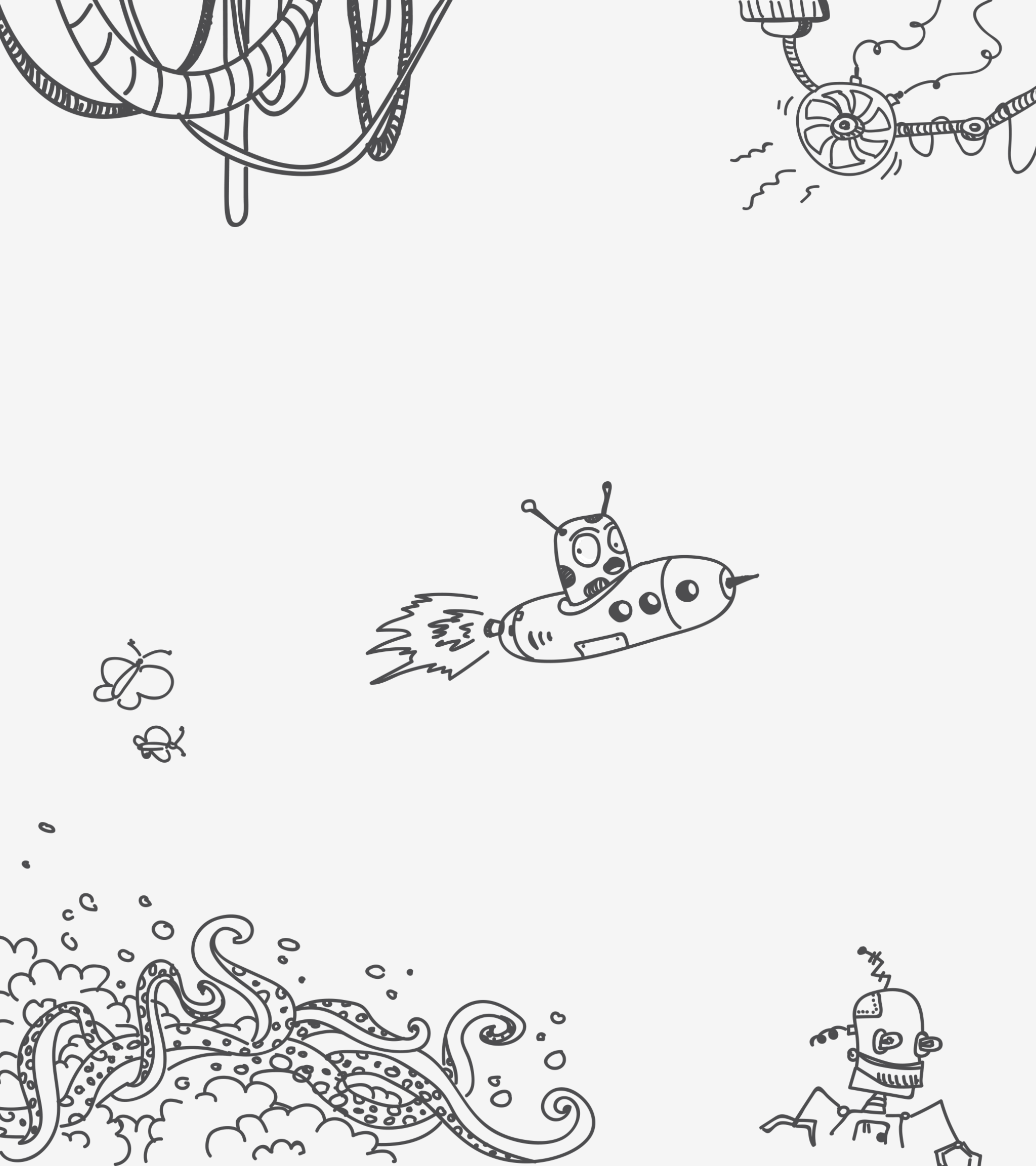
Mixins

Mixin pozwala na zapisanie deklaracji CSS (właściwość i jej wartość) w postaci pewnego rodzaju zmiennej.

Mixin może również przekazywać argumenty.

```
@mixin boxShadow($color) {  
    -webkit-box-shadow: 10px 10px 5px 0 $color;  
    -moz-box-shadow: 10px 10px 5px 0 $color;  
    box-shadow: 10px 10px 5px 0 $color;  
}  
  
.box {  
    @include boxShadow(red);  
}
```

@include – używamy mixina



Omówienie pracy domowej

Omówcie teraz treści przygotowanych zadań domowych.