SASS, STYLE & HOW TO SIP SASSPERELLA

"YOU SASSING ME?"

Josh "Design Daddy" Tregenza

WHAT IS SASS

- Sass is a pre-processor for your CSS Style sheets

- Created by Hampton Catlin

- Gay Rights Advocating Programmer extraordinaire

- He also created HAML

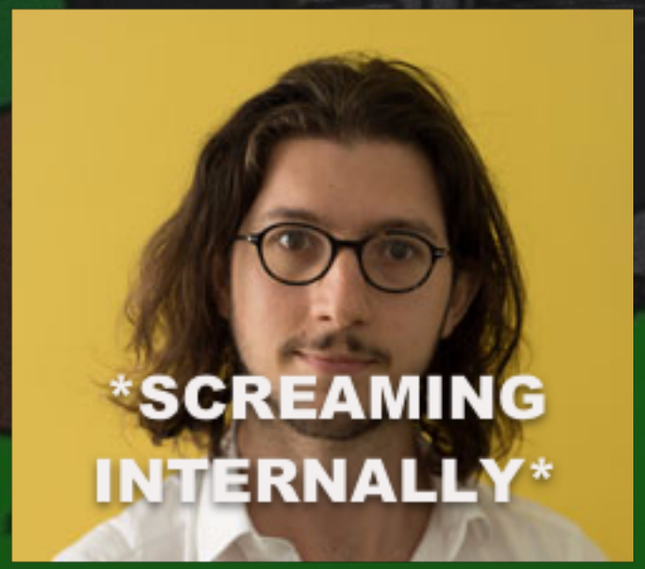
- He and his partner and collaborator, founded RareBit

WHAT IS SASS

- Stylesheets are getting larger and larger, with repeating elements

- With the advent of the media queries, elements are getting called in more places all over your stylesheet

- These cannot scale and are not readable whatsoever



WHAT IS SASS

- Sass helps make stylesheets readable through the cunning use of:

- Variables
- Nesting
- Import
- Mixins
- Extend/Inheritance
- Operators

VARIABLES

- Like JS, variables store information that can be reused throughout the stylesheet

SCSS INPUT

```scss
$font-stack:    Helvetica, sans-serif;
$primary-color: #333;

body {
  font: 100% $font-stack;
  color: $primary-color;
}
```

CSS OUTPUT

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

VARIABLES

- Common uses for variables are color, number and fonts

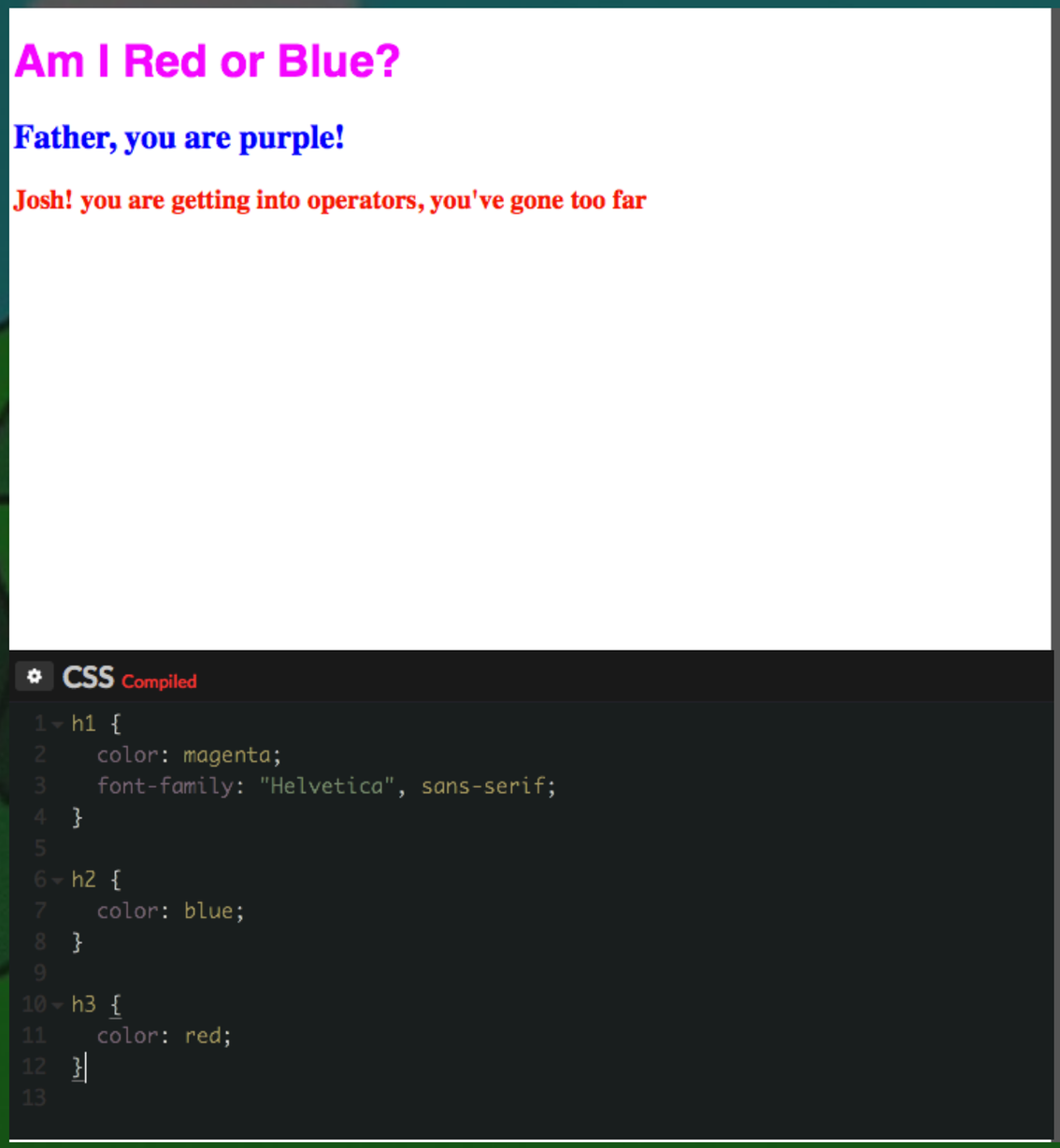
- Uncommon but powerful uses are, naming conventions, conditional activators

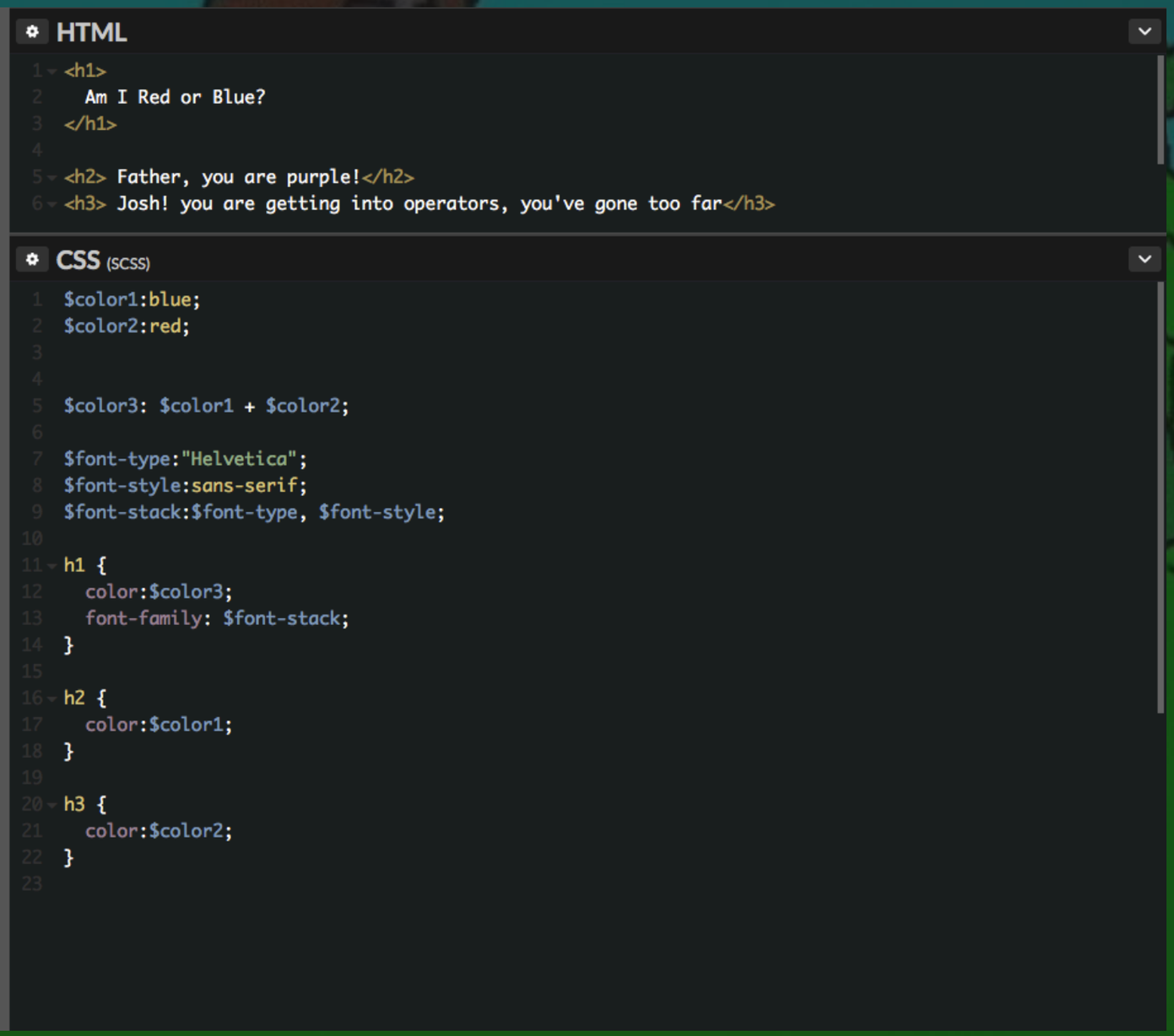
- If you are using a value more than 3 times in your styling, consider a variable

- As a rule of thumb if you are using a value more than 3 times in your styling, consider a variable

VARIABLES

- Variables can also call on other variables, creating a variabl-ception





MIXINS

- Mixins are the next logical step in variables

- Writing vendor prefixes for instance are tedious to write out again and again -o-and again -moz-and again -ms-and again

- Mixins become your shorthand for repetition

Input SCSS

```scss
@mixin border-radius($radius) {
  -webkit-border-radius: $radius;
     -moz-border-radius: $radius;
      -ms-border-radius: $radius;
          border-radius: $radius;
}

.box { @include border-radius(10px); }
```

Output CSS

```css
.box {
  -webkit-border-radius: 10px;
  -moz-border-radius: 10px;
  -ms-border-radius: 10px;
  border-radius: 10px;
}
```

MIXINS

- Use mixins for elements you are going to reuse, like border, shadows, flexbox, background gradients style

- Using Mixins and Variables together will allow you to write DRY (Don’t Repeat Yourself) first time, every time

NESTING

- In HTML, nesting is clear and concise, showing you the hierarchy of elements

- Sass allows you to nest your styling

Input SCSS

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

  li { display: inline-block; }

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

Output CSS

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

nav li {
  display: inline-block;
}

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

NESTING

- Nesting allows you to keep your elements together in a readable manner

- Nesting really comes in handy when declaring media queries and using pseudo selectors like :active or :hover

- Rule of threes applies to nesting as well, if you are three levels deep in your nesting you lose the benefits of readability

IMPORTS

- Allows you to segment your stylesheets into a modular folder structure

- Your buttons can be in one file, while your inputs are in another

- Utilising import will let you compile a robust stylesheet whilst having readable, scalable and modular Sass Files.

```
@charset 'UTF-8';

// 1. Configuration and helpers
@import
  'abstracts/variables',
  'abstracts/functions',
  'abstracts/mixins';

// 2. Vendors
@import
  'vendor/normalize';

// 3. Base stuff
@import
  'base/base',
  'base/fonts',
  'base/typography',
  'base/helpers';

// 4. Layout-related sections
@import
  'layout/header',
  'layout/footer';
```

IMPORTS

- Imports gives you the ability to call fewer style files on load, saving you sweet sweet milliseconds

EXTENDING / INHERITANCE

- Pop Quiz : You have a message class name that has several modifiers, how are you going to have the same styling for all of them but also the flexibility to change change certain style values. How to do?

- You could nest them, you could create a mixin to repeat the properties or...

Input SCSS

```scss
.message {
  border: 1px solid #ccc;
  padding: 10px;
  color: #333;
}

.success {
  @extend .message;
  border-color: green;
}

.error {
  @extend .message;
  border-color: red;
}

.warning {
  @extend .message;
  border-color: yellow;
}
```

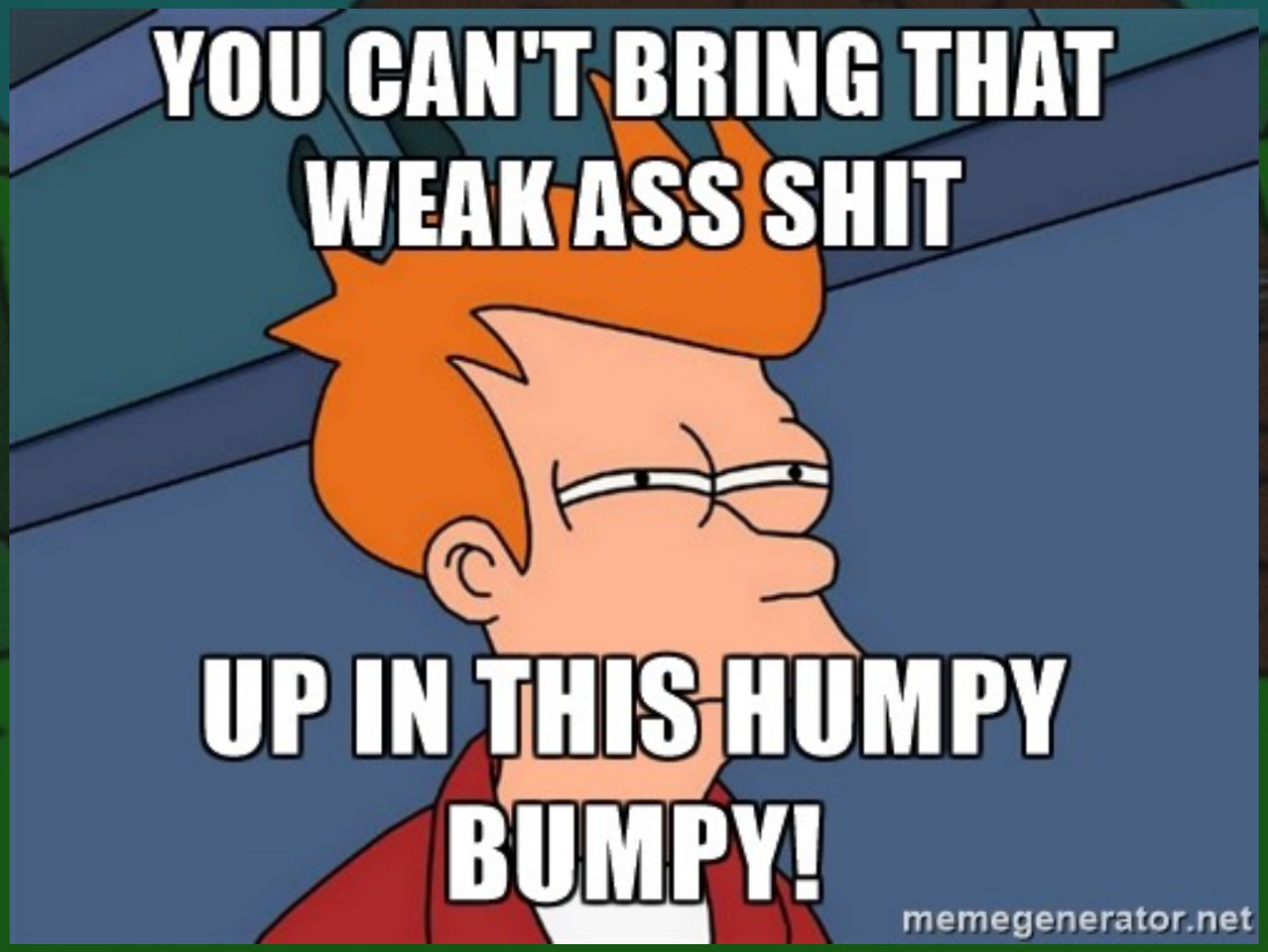
Output CSS

```css
.message, .success, .error, .warning {
  border: 1px solid #cccccc;
  padding: 10px;
  color: #333;
}

.success {
  border-color: green;
}

.error {
  border-color: red;
}

.warning {
  border-color: yellow;
}
```

EXTENDING / INHERITANCE

- Use extends when you don’t want to call multiple classnames in your HTML

- Sass is keeping you DRY all over this humpy bumpy



OPERATORS

- Operators is math

- Sometimes it’s math for numbers, sometimes it’s for colors

CSS (SCSS)

```scss
$color1:red;
$color2:blue;

$color3: $color1 + $color2;
// REMEMBER ME, I WAS AN OPERATOR WITHIN A VARIABLE ALL ALONG, I'M MIXING RED AND BLUE !!
```

- Follows the same logic as algebra, with hierarchy and the like

COMMENTING

- In Sass you can also make comments that only those working in the Sass files can see

- They aren’t compiled so you can write a novella and you’ll be swell

- Use Sass commenting to help others understand your elements or mixing or variables.

CSS (SCSS)

```
1  // WOODY HAS EYES EVERYWHERE, PLEASE SEND HELP, BIG BROTHER IS WATCHING US ALL, WE AREN'T SAFE
2  /* Everything is fine, Woody is a great and kind leader */
```

CSS Compiled

```
1  /* Everything is fine, Woody is a great and kind leader */
2
```

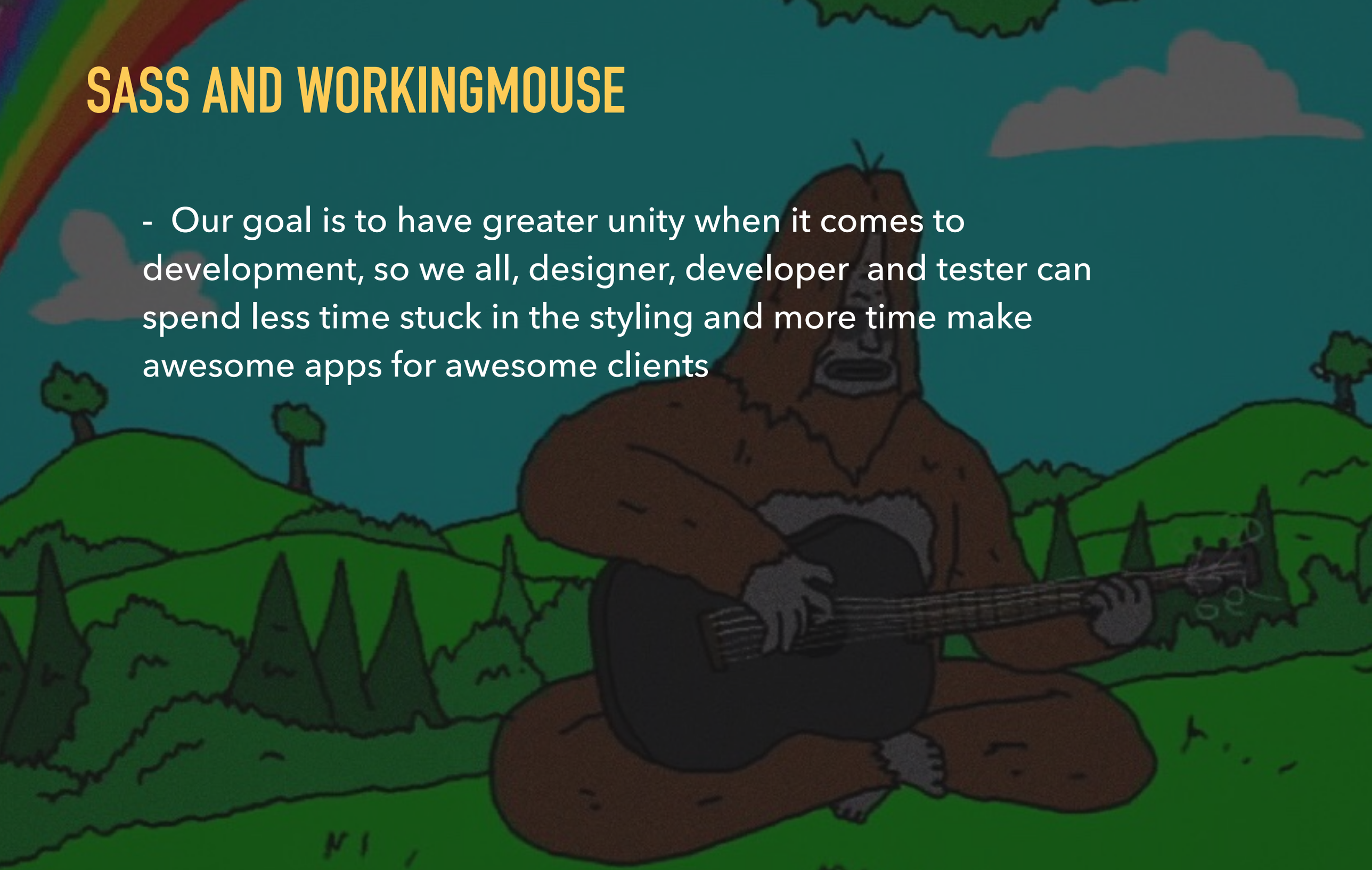
SASS AND WORKINGMOUSE

- The Design Team of Livin La Vida Loca is currently structuring out a Sass file structure, that is modular, scalable and suited for WorkingMouse now and into the future.

- It’s going to be the Designer’s very own generator, A SassBot, if you will

SASS AND WORKINGMOUSE

- Our goal is to have greater unity when it comes to development, so we all, designer, developer and tester can spend less time stuck in the styling and more time make awesome apps for awesome clients




- FIN -