

CSS Preprocessor

AMRUTHA & SABBU



CONTENT

1. INTRODUCTION

2. FEATURES

3. WHY USE CSS PREPROCESSOR?

4. COMMONLY USED CSS PREPROCESSORS

- SCSS/SASS
- LESS
- STYLUS

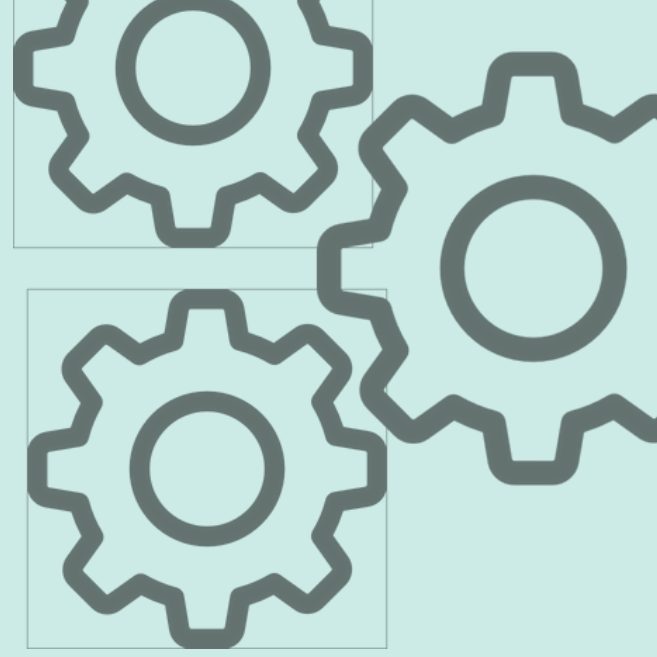
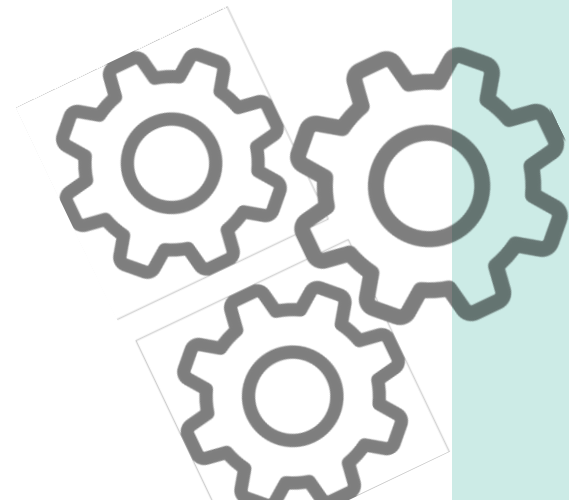
5. COMPARING THE TYPES

6. CSS PREPROCESSOR TODAY!

7. CONS OF CSS PREPROCESSOR

8. SHOULD WE USE CSS PREPROCESSOR?

Introduction to CSS Preprocessor



- Scripting language that extends the capabilities of CSS then compiles it into standard CSS
- Advanced version of CSS

Features

REUSABILITY

NESTING

FUNCTIONS AND MIXINS

MATHEMATICAL OPERATIONS

IMPROVED MAINTENANCE

05



Why Use?

**EASY TO AUTOMATE
REPETITIVE TASKS**

**CREATE REUSABLE CODE
SNIPPETS**

ORGANISED FILES

BROWSER COMPATIBILITY

The Sass logo is written in a pink, cursive script font.The Less logo features the word "less" in a white, lowercase, sans-serif font, enclosed within a dark blue rounded rectangle with white curly braces on the left and right sides.The Stylus logo is written in a green, cursive script font.

Most Common CSS Pre-processor

06



Sass/Scss

- Syntactically Awesome Style Sheet
- Released in 2006
- Two syntax: SASS & SCSS
- **SASS** : Strict Indentation
- **SCSS** : Use of semicolons and braces



.scss / .sass

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

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

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

  li { display: inline-block; }

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




Installation

09

NPM

```
npm install -g sass
```

Choco(Windows)

```
choco install sass
```

Brew(Mac)

```
brew install sass/sass/sass
```

Terminal

```
sass main_file.scss output_file.css
```

Less

- Leaner Style Sheets
- Released in 2009
- Originally build with Ruby, now Less is rewritten in JavaScript.





.less

```
#colors() {  
  primary: blue;  
  secondary: green;  
}  
  
.button {  
  color: #colors[primary];  
  border: 1px solid #colors[secondary];  
}
```

```
.component {  
  width: 300px;  
  @media (min-width: 768px) {  
    width: 600px;  
    @media (min-resolution: 192dpi) {  
      background-image: url(/img/retina2x.png);  
    }  
  }  
  @media (min-width: 1280px) {  
    width: 800px;  
  }  
}
```



Installation

12

With Node Js

```
npm install -g less
```

With Browser

```
<link rel="stylesheet/less" type="text/css" href="styles.less" />  
<script src="https://cdn.jsdelivr.net/npm/less" ></script>
```

Terminal

```
lessc main_file.scss output_file.css
```



Stylus

- Minimalist Pre-processor
- Released in 2010
- Optional colons, semi-colons and braces
- Use Stylus if
 - Comfortable with Jade and CoffeeScript
 - Work in a Node.js environment

Stylus

.styl

```
$ stylus --prefix web- sh
```

used with this code:

```
.class styl  
width: 10px
```

Processed css

```
.web-class css {  
width: 10px;  
}
```

```
body  
  color: red  
ul  
  li  
    color: blue  
  a  
    background-color: @color
```

```
font-size = 14px  
font-stack = "Lucida Grande", Arial, sans-serif  
  
body  
  font font-size font-stack  
  
//Compiles to:  
body {  
  font: 14px "Lucida Grande", Arial, sans-serif;  
}
```

14

Stylus

15

NPM

```
$ npm install stylus -g
```

PNPM

```
$ pnpm add -g stylus
```

Terminal

```
$ stylus --compress < main_file.styl > output_file.css
```

Comparison



- Variables starting with \$
- Largest User Base and Active Community



- Variables starting with @
- Easiest to Compile and Integrate



- Popular choice for Node JS developers
- Combines the extensive logical prowess of Sass with the simplicity of LESS

CSS Preprocessors Today

Survey Report

18

SASS

LESS

STYLUS

Would Use Again

75.9%

18.8%

7.9%

Would Not Use

14.5%

36.2%

8.8%

Interested

4.5%

8.6%

13.7%

Not Interested

4.7%

34.1%

42%

Never Heard

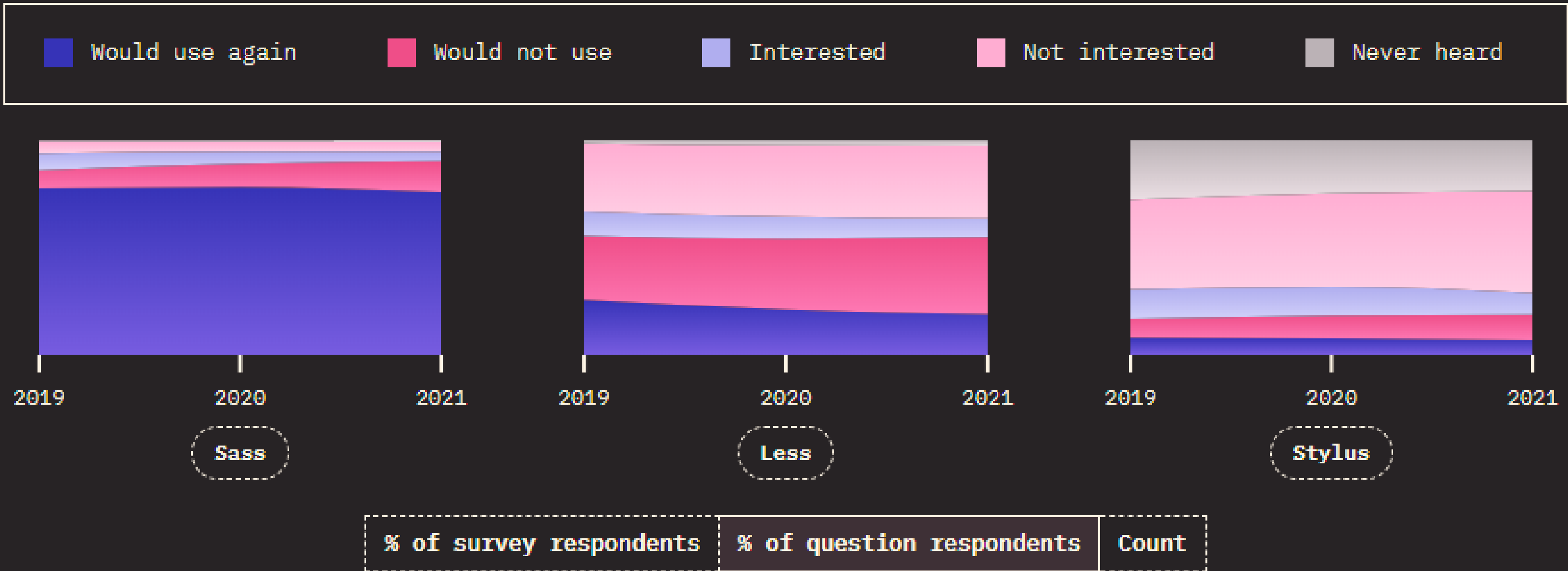
0.6%

2.5%

27.8%

EXPERIENCE OVER TIME 💰

Overview of opinions on the technologies surveyed over time.



Technologies with only one year of data are not included.



Any Cons?

19

- **Debugging is harder.**
- **Compilation slows down development.**
- **Very large CSS files.**
- **Maintenance and over-engineering.**
- **Capability and understanding.**

Should We Use Css Pre-processors



Yes

You Should

But

As a newbie

No



No...! Why?

Difficult to Debug

Chance to get mix it up

Programming knowledge needed

Increases file size

If you want to use it.. Then ensure

- Good at CSS**
- Basic programming knowledge**
- Confidence in fixing a bug**

THANK YOU

Content Management
MA, Web Design and Content Planning

AMRUTHA & SABBU