

CSS Preprocessors (SASS/LESS) - Session 8

Table of Contents

- [Introduction](#)
- [Why Use a CSS Preprocessor?](#)
- [SASS vs LESS](#)
- [Setting Up](#)
- [SASS Fundamentals](#)
- [LESS Fundamentals](#)
- [Best Practices](#)
- [Build Tools](#)
- [Homework](#)

Introduction

CSS preprocessors extend CSS with features like variables, nesting, mixins, and functions, making your CSS more maintainable and powerful.

Why Use a CSS Preprocessor?

- **Variables:** Reuse values like colors and fonts
- **Nesting:** Match HTML structure in CSS
- **Mixins:** Reuse CSS declarations
- **Functions:** Perform calculations and manipulate values
- **Modularity:** Split code into multiple files

SASS vs LESS

Feature	SASS/SCSS	LESS
Syntax	.scss or .sass	.less
Variables	<code>\$variable</code>	<code>@variable</code>
Mixins	<code>@mixin/@include</code>	<code>.mixin()</code>
Community	Larger	Large

Setting Up

Install SASS

```
npm install -g sass
sass --watch input.scss:output.css
```

Install LESS

```
npm install -g less
lessc styles.less styles.css
```

SASS Fundamentals

Variables & Nesting

```
// Variables
$primary: #3498db;
$font-stack: Arial, sans-serif;

// Nesting
nav {
  ul {
    margin: 0;
    li { display: inline-block; }
  }
}
```

Mixins & Extend

```
// Mixin
@mixin border-radius($radius) {
  -webkit-border-radius: $radius;
  border-radius: $radius;
}

// Extend
%message {
  border: 1px solid #ccc;
  padding: 10px;
}

.button {
  @include border-radius(5px);
  @extend %message;
  background: $primary;
}
```

LESS Fundamentals

Variables & Mixins

```
@primary: #3498db;
```

```
.border-radius(@radius: 5px) {
  -webkit-border-radius: @radius;
  border-radius: @radius;
}

.button {
  background: @primary;
  .border-radius();
}
```

Best Practices

1. Organize code into logical partials
2. Use meaningful variable names
3. Limit nesting to 3 levels
4. Create reusable mixins
5. Document your code

Build Tools

Gulp Example

```
const gulp = require('gulp');
const sass = require('gulp-sass')(require('sass'));

gulp.task('sass', function() {
  return gulp.src('src/scss/**/*.scss')
    .pipe(sass().on('error', sass.logError))
    .pipe(gulp.dest('dist/css'));
});
```

Homework

1. Convert a CSS file to SASS/LESS
2. Create a theme with variables
3. Build a responsive grid
4. Document your components

Resources

- [SASS Docs](#)
- [LESS Docs](#)
- [SASS Guidelines](#)