

Bootcamp '20

SESIONI I NJËMBËDHJETË



What will be covered?

- What is Sass?
- How to install Sass?
- How to compile Sass?
- Variables
- Operators
- Nesting
- Partials
- Modules
- Mixins
- Inheritance
- Functions
- Conditions
- @each

What is Sass?

- Sass është gjuhë skriptuese e kompilueshme
- Sass mundëson dy lloje sintaksash
 - Me kllapa gjarpërore (hapëse dhe mbyllëse)
 - Me shtyrje (të shtyrit e kodit anësh - njejtë sikurse në Python)
- Prapashtesa e skedave të tipit Sass mund të jetë `.scss` ose `.sass`
- Kompilohet në skeda të tipit `.css`



CSS Compatible

Sass is completely compatible with all versions of CSS. We take this compatibility seriously, so that you can seamlessly use any available CSS libraries.

Industry Approved

Over and over again, the industry is choosing Sass as the premier CSS extension language.

Feature Rich

Sass boasts more features and abilities than any other CSS extension language out there. The Sass Core Team has worked endlessly to not only keep up, but stay ahead.

Large Community

Sass is actively supported and developed by a consortium of several tech companies and hundreds of developers.

Mature

Sass has been actively supported for over 13 years by its loving Core Team.

Frameworks

There are an endless number of frameworks built with Sass. [Compass](#), [Bourbon](#), and [Susy](#) just to name a few.

How to install Sass?

Sass mund të instalohet në disa mënyra

- Nga repository zyrtar i Sass
- Përmes npm (Node Package Manager)
- Përmes Brew

<https://www.npmjs.com/>

<https://brew.sh/>

How to compile Sass?

Sass mund të kompilohet në disa mënyra

- Përmes terminalit

> sass input.scss output.css

> sass folder/scss:folder/css

- Përmes desktop aplikacioneve të posatshme
 - Koala <http://koala-app.com/>
 - Scout App <https://scout-app.io/>
- Në VS Code përmes shtesës [Live Sass Compiler](#)

Visual Studio Code – Sass Extension

- Pasi të kemi instaluar shtesën **Live Sass Compiler** duhet ta konfigurojmë

```
{
  "explorer.confirmDelete": false,
  "window.zoomLevel": 2,
  "liveSassCompile.settings.formats": [
    {
      "format" : "compressed",
      // "extensionName" : ".min.css",
      "savePath" : "/assets/css"
    }
  ],
  "liveSassCompile.settings.target": "css",
  "liveSassCompile.settings.mode": "write",
  "liveSassCompile.settings.excludeMap": false,
  "kite.showWelcomeNotificationOnStartup": false
}
```

Unknown Configuration Setting

No quick fixes available

Për më tepër rreth mënyrës së kmund të mësoni në vegëzën në vijim

<https://github.com/ritwickdey/vscode-live-sass-compiler/blob/master/docs/settings.md>

Variables

- Para çdo emri të variables duhet të qëndrojë simboli \$
- Emri i variablës ndahet nga vlera përkatëse përmes simbolit :

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

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


Operators

- Sass ofron edhe mundësinë e përdorimit të operatorëve
- Operatorët: +, -, *, /, dhe %

```
.container {  
  width: 100%;  
}  
  
article[role="main"] {  
  float: left;  
  width: 600px / 960px * 100%;  
}  
  
aside[role="complementary"] {  
  float: right;  
  width: 300px / 960px * 100%;  
}
```

```
.container {  
  width: 100%;  
}  
  
article[role="main"] {  
  float: left;  
  width: 62.5%;  
}  
  
aside[role="complementary"] {  
  float: right;  
  width: 31.25%;  
}
```

Nesting

```
nav {  
  ul {  
    margin: 0;  
    padding: 0;  
    list-style: none;  
  }  
  
  li { display: inline-block; }  
  
  a {  
    display: block;  
    padding: 6px 12px;  
    text-decoration: none;  
  }  
}
```

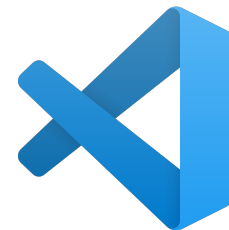
```
nav ul {  
  margin: 0;  
  padding: 0;  
  list-style: none;  
}  
nav li {  
  display: inline-block;  
}  
nav a {  
  display: block;  
  padding: 6px 12px;  
  text-decoration: none;  
}
```

Nuk është e lavdëruar ndërthurja e tepërt andaj edhe duhet të anashkalohet kjo praktikë.

Partials

- Partials janë skeda të posatshme emri i të cilave fillon me _ (underline)
- Mundësojnë ripërdorim të kodit
- Nuk kompilohen

Le të shohim nga një shembull



Modules

- Përmes përdorimit të partials ne mund ta ndajm codebase tonë në disa module logjike

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

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

```
// styles.scss
@use 'base';

.inverse {
  background-color: base.$primary-color;
  color: white;
}
```

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

.inverse {
  background-color: #333;
  color: white;
}
```

Le të shohim nga një shembull



Mixins

- Kemi raste kur veti të njejta duhet të përsëriten në stilizimin e më shumë elementeve vetëm se që pranojnë vlera të ndryshme.
- Këtë përsëritje Sass e eleminon përmes @mixin
- Një @mixin bëhet i qasshëm për përdorim përmes komandës @include

```
@mixin transform($property) {  
  -webkit-transform: $property;  
  -ms-transform: $property;  
  transform: $property;  
}  
.box { @include transform(rotate(30deg)); }
```

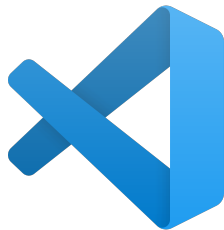
```
.box {  
  -webkit-transform: rotate(30deg);  
  -ms-transform: rotate(30deg);  
  transform: rotate(30deg);  
}
```

Inheritance

- Ka raste kur disa elemente ndajnë një numër të konsiderueshëm të karakteristikave të përbashkëta, e dallojnë vetëm në karakteristika specifike.
- Që të mund të arrijmë benefitin e ripërdorimit të disa karakteristikave dhe më pasë të shtohen karakteristika të reja ne mund të përdorim trashëgiminë që e ofron Sass

Inheritance

Le të shohim nga një shembull



```
/* This CSS will print because %message-shared is extended. */
%message-shared {
  border: 1px solid #ccc;
  padding: 10px;
  color: #333;
}

// This CSS won't print because %equal-heights is never extended.
%equal-heights {
  display: flex;
  flex-wrap: wrap;
}

.message {
  @extend %message-shared;
}

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

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

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

Functions

Funksionet gjithmonë kthejnë vlerë - brenda vetit kanë @return

```
@function pow($base, $exponent) {  
  $result: 1;  
  @for $_ from 1 through $exponent {  
    $result: $result * $base;  
  }  
  @return $result;  
}  
  
.sidebar {  
  float: left;  
  margin-left: pow(4, 3) * 1px;  
}
```

Kthehet në CSS:

```
.sidebar {  
  float: left;  
  margin-left: 64px;  
}
```


Conditionals

```
$light-background: #f2ece4;
$light-text: #036;
$dark-background: #6b717f;
$dark-text: #d2e1dd;

@mixin theme-colors($light-theme: true) {
  @if $light-theme {
    background-color: $light-background;
    color: $light-text;
  } @else {
    background-color: $dark-background;
    color: $dark-text;
  }
}

.banner {
  @include theme-colors($light-theme: true);
  body.dark & {
    @include theme-colors($light-theme: false);
  }
}
```

Kthehet në CSS:

```
.banner {
  background-color: #f2ece4;
  color: #036;
}
body.dark .banner {
  background-color: #6b717f;
  color: #d2e1dd;
}
```

@each

```
$sizes: 40px, 50px, 80px;

@each $size in $sizes {
  .icon-#{ $size } {
    font-size: $size;
    height: $size;
    width: $size;
  }
}
```

Kthehet në CSS:

```
.icon-40px {
  font-size: 40px;
  height: 40px;
  width: 40px;
}

.icon-50px {
  font-size: 50px;
  height: 50px;
  width: 50px;
}

.icon-80px {
  font-size: 80px;
  height: 80px;
  width: 80px;
}
```

@each with maps

```
$icons: ("eye": "\f112", "start": "\f12e", "stop": "\f12f");

@each $name, $glyph in $icons {
  .icon-#{$name}:before {
    display: inline-block;
    font-family: "Icon Font";
    content: $glyph;
  }
}
```

Kthehet në CSS:

```
@charset "UTF-8";

.icon-eye:before {
  display: inline-block;
  font-family: "Icon Font";
  content: "\f112";
}

.icon-start:before {
  display: inline-block;
  font-family: "Icon Font";
  content: "\f12e";
}

.icon-stop:before {
  display: inline-block;
  font-family: "Icon Font";
  content: "\f12f";
}
```

@each - Destructuring

```
$icons:  
  "eye" "\f112" 12px,  
  "start" "\f12e" 16px,  
  "stop" "\f12f" 10px;  
  
@each $name, $glyph, $size in $icons {  
  .icon-#{ $name }:before {  
    display: inline-block;  
    font-family: "Icon Font";  
    content: $glyph;  
    font-size: $size;  
  }  
}
```

Kthehet në CSS:

```
@charset "UTF-8";  
.icon-eye:before {  
  display: inline-block;  
  font-family: "Icon Font";  
  content: "\f112";  
  font-size: 12px;  
}  
  
.icon-start:before {  
  display: inline-block;  
  font-family: "Icon Font";  
  content: "\f12e";  
  font-size: 16px;  
}  
  
.icon-stop:before {  
  display: inline-block;  
  font-family: "Icon Font";  
  content: "\f12f";  
  font-size: 10px;  
}
```

@for

```
$base-color: #036;  
  
@for $i from 1 through 3 {  
  ul:nth-child(3n + #{ $i }) {  
    background-color: lighten($base-color, $i * 5%);  
  }  
}
```

Kthehet në CSS:

```
ul:nth-child(3n + 1) {  
  background-color: #004080;  
}  
  
ul:nth-child(3n + 2) {  
  background-color: #004d99;  
}  
  
ul:nth-child(3n + 3) {  
  background-color: #0059b3;  
}
```

@while

```
/// Divides `$value` by `$ratio` until it's below `$base`.
@function scale-below($value, $base, $ratio: 1.618) {
  @while $value > $base {
    $value: $value / $ratio;
  }
  @return $value;
}

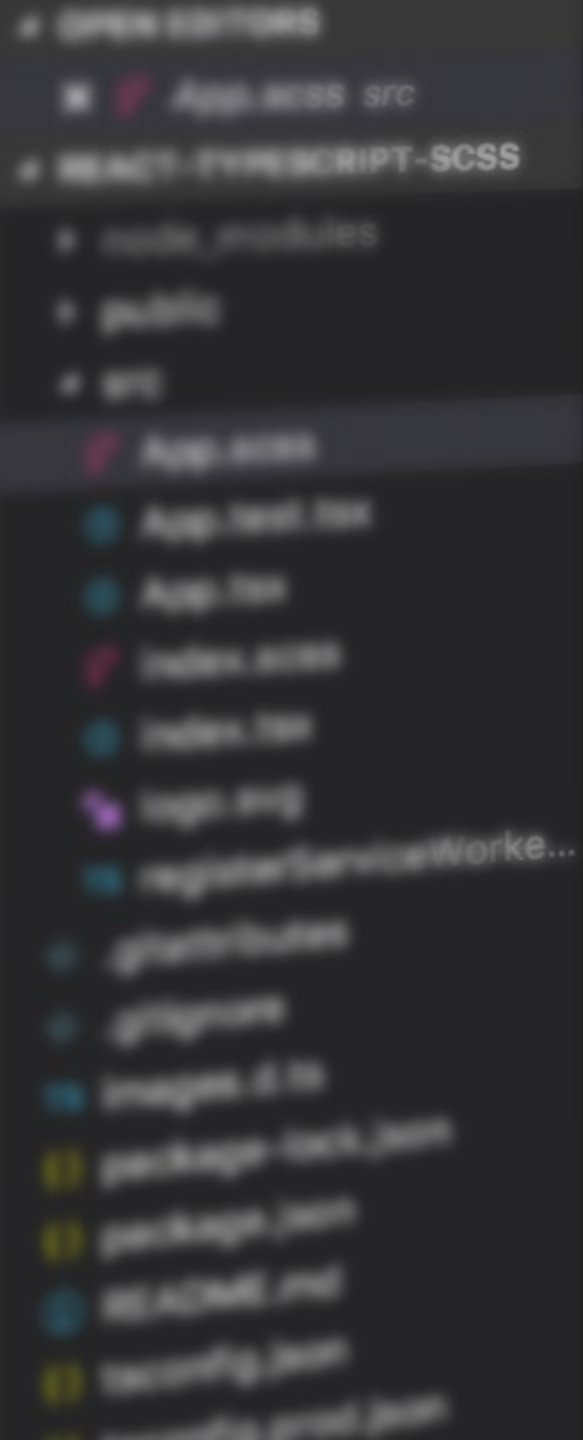
$normal-font-size: 16px;
sup {
  font-size: scale-below(20px, 16px);
}
```

Kthehet në CSS:

```
sup {
  font-size: 12.36094px;
}
```

QUESTIONS





```
1  .App {
2    text-align: center;
3  }
4
5  .App-logo {
6    animation: App-logo-spin infinite 20s linear;
7    height: 80px;
8  }
9
10 .App-header {
11   background-color: #222;
12   height: 150px;
13   padding: 20px;
14   color: white;
15 }
16
17 .App-title {
18   font-size: 1.5em;
19 }
20
21 .App-intro {
22   font-size: large;
23 }
24
25 @keyframes App-logo-spin {
26   from { transform: rotate(0deg); }
27   to { transform: rotate(360deg); }
```

EXAMPLES