



# CSS PREPROCESSORS

SASS, LESS

*dr Michał Kuciapski*  
Higher Banking School

# AGENDA

1. Introduction
2. SASS
3. LESS

Referenced materials:

1. Kianosh Pourian & Dale Sande, *Sass in the Real World*, GitBook, 2014
2. Wynn Netherland Nathan Weizenbaum, Chris Eppstein, Brandon Mathis , *Sass and Compas in Action*, MANNING 2013
3. Bass Jobsen, *Less Web Development Essentials*, Packt Publishing 2014
4. Keith Zantow, *Using LESS for more maintainable, semantic, and lean web sites*, Etix
5. <https://www.keycdn.com/blog/sass-vs-less>
6. <http://sass-lang.com>
7. <https://www.tutorialspoint.com/sass>

# Introduction

# THE PROBLEMS WITH CSS

- Repetitive & redundant
  - Time-consuming
- Cascading Changes = tons of busywork
- Organization:
  - Vertical placement limitations
  - Limited organizational tools

# THE PROBLEMS WITH CSS

```
/*////////////////////*/
Table of Contents
& Search Codes

@top Table of Contents

@msc Miscellaneous
@tol Inline Style Tools
@col Color Design Scheme
@typ Typography
@man Main Layout
@hed General Header
@nav Navigation Menu
@ind Index
@pcp Product & Category Pages
@cim Category Images
@phd Page Headers
@pds Products Pages
@dtl Product Detail Pages
@abt About Us
@frm Forms
@bat Bat Chart
@img Images
@btn Buttons
@hlp Helper Classes
@ftr Footer
@psy Print styles
@meq Media Queries
@404 404 Page
@car Carousel
@pri Privacy Policy
////////////////////*/

```

```
=====
body, #maincontent {
    background: ■#fff;
}

a, a:visited, .nav-list a:visited {
    color: ■#69BE28;
}

h1 {
    color: ■#69BE28;
}

/* Section Headings, etc. (Black bg, Green txt) */
.vertMenu, .sectionheading, .detaildescription h2, div.detailprice, .textblock
h3, .textblock h2, .contactInfo h3, .cstitle {
    background: linear-gradient(■rgb(70,70,70), ■black 50%);
    color: ■#69BE28;
}

.sectionheading p {
    color: ■#A5ACAF;
}

/* Page Headers: White/gray bg, green bottom border */
div.detailname, div.ecthreview, .prodname, .tiled h4, #bios h3 {
    background: linear-gradient(■#fff, ■#e9e9e9);
    border-bottom: 2px solid ■#69BE28;
}

/* Subtitles: White */
.category .catdesc, .catheader p, .proddescription > p {
    color: ■white;
}

/* Default Buttons: Green bg, black txt */
button, input[type="submit"], input[type="button"] {
    background: ■#69BE28 url(..../images/glass.png) repeat-x 0 50%;
    border-radius: 5px;
    color: ■white;
}

```

# WHAT ARE CSS PREPROCESSORS

- Tools for development:
  - Nested syntax
  - Ability to define variables
  - Ability to define mixins
  - Mathematical functions
  - Operational functions
  - Joining of multiple files
- Gives more functionality to CSS
- Dynamic style sheets
- Allow for fewer lines of code

# HOW THEY WORK

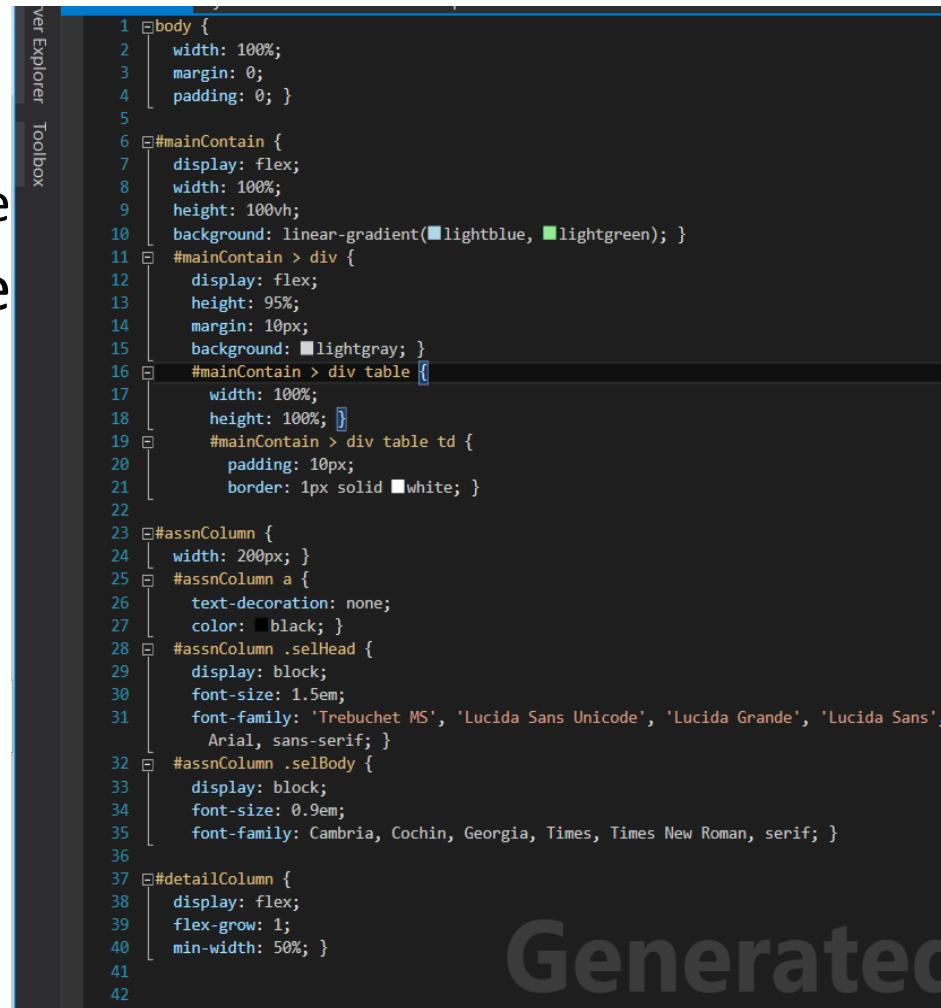
- Creation
- Compilation
- Integration

# WHY USE PREPROCESSORS?

- Developers turning to toolkits
- Create more manageable, reusable, compact style sheets
- Better syntax
- Calculations and logic
- Reduces repetition
- Cleaner code
- Easier maintenance - snippets and libraries
- Faster development time
- Saves from monotony
- More time for creativity and back-end development
- Programmatic CSS features

# DISADVANTAGES OF PREPROCESSORS?

- Generated CSS - possibly:
  - bloated
  - reader-unfriendly source code
- Debugging is harder - source Maps
- Barriers to entry - programming code
- If many people are working on the same site, then should use the same preprocessor.
- Can be slow to compile



The screenshot shows a browser's developer tools with the "CSS" tab selected. The left sidebar has "Server Explorer" and "Toolbox" options. The main area displays the following generated CSS code:

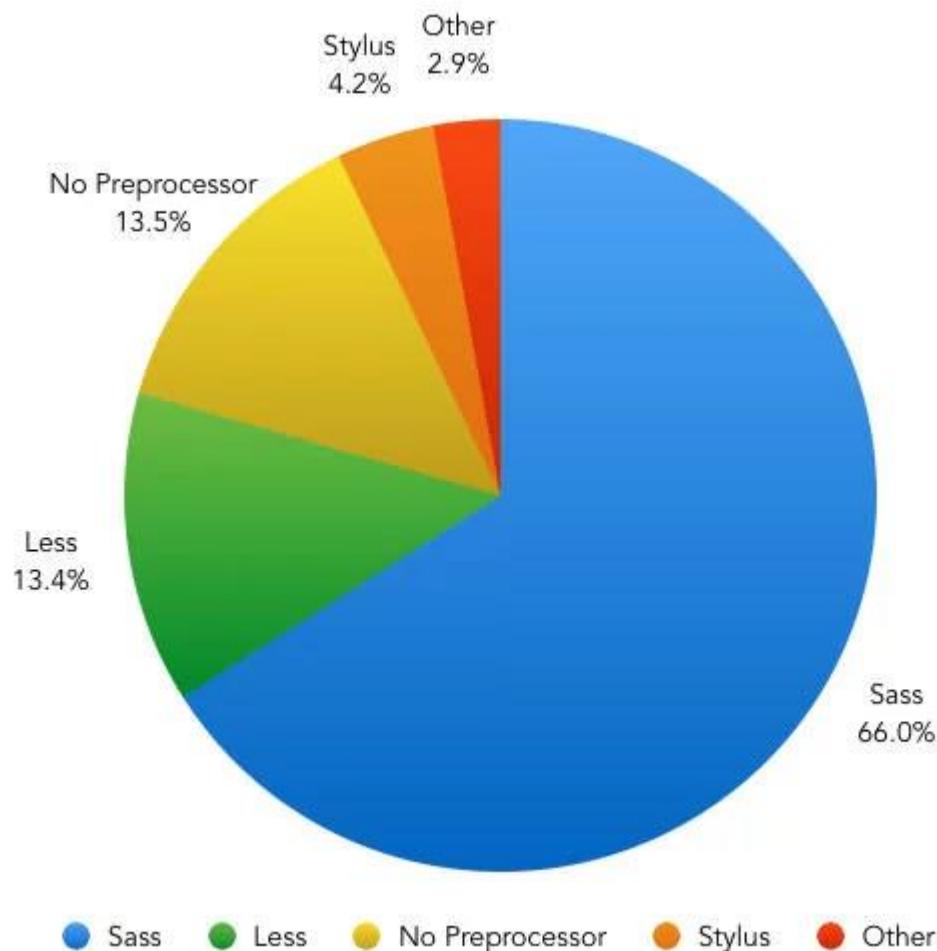
```
1 body {  
2     width: 100%;  
3     margin: 0;  
4     padding: 0; }  
5  
6 #mainContain {  
7     display: flex;  
8     width: 100%;  
9     height: 100vh;  
10    background: linear-gradient(■lightblue, ■lightgreen); }  
11   #mainContain > div {  
12       display: flex;  
13       height: 95%;  
14       margin: 10px;  
15       background: ■lightgray; }  
16     #mainContain > div table {  
17         width: 100%;  
18         height: 100%; }  
19     #mainContain > div table td {  
20         padding: 10px;  
21         border: 1px solid ■white; }  
22  
23   #assnColumn {  
24       width: 200px; }  
25   #assnColumn a {  
26       text-decoration: none;  
27       color: ■black; }  
28   #assnColumn .selHead {  
29       display: block;  
30       font-size: 1.5em;  
31       font-family: 'Trebuchet MS', 'Lucida Sans Unicode', 'Lucida Grande', 'Lucida Sans',  
32       Arial, sans-serif; }  
33   #assnColumn .selBody {  
34       display: block;  
35       font-size: 0.9em;  
36       font-family: Cambria, Cochin, Georgia, Times, Times New Roman, serif; }  
37  
38   #detailColumn {  
39       display: flex;  
40       flex-grow: 1;  
41       min-width: 50%; }  
42
```

Generated

# CSS STYLING SOLUTIONS

- CSS-Preprocessors:
  - SASS
  - LESS
  - StylusMyth
- CSS-Postprocessors:
  - Autoprefixer
  - Pleeease (node.js)
- CSS3/CSS4 (<http://css4.rocks>)

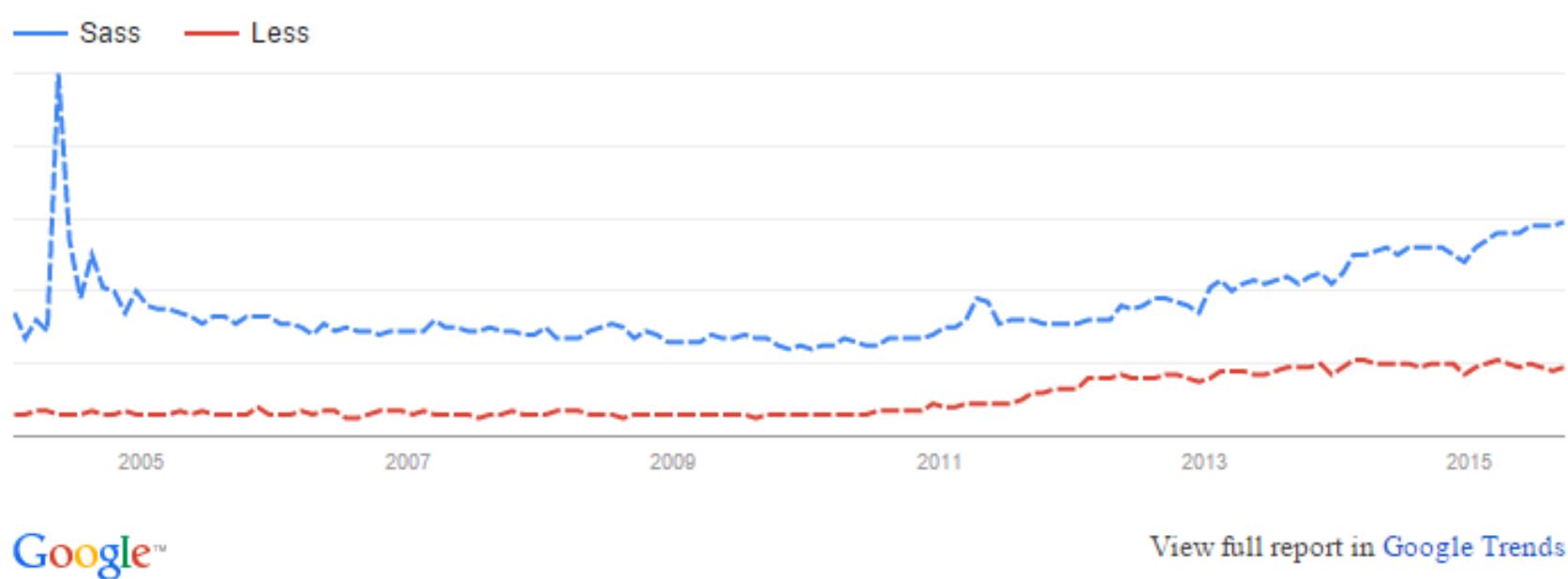
# PREPROCESSORS PREFERENCE



Source: 2015 Survey from [ashleynolan.co.uk](http://ashleynolan.co.uk)

# PREPROCESSORS TRENDS

Interest over time. Web Search. Worldwide, 2004 - present.



Google™

Source: Google Trends

# LESS vs SASS - DIFFERENCES

## LESS

Based in JavaScript  
JavaScript Evaluation  
Client-side operation

Less efficient logic

Fewer frameworks

Guarded mixins

Loops only allowed for  
numeric values

Easier to install?

## BOTH

Extensions for faster &  
easier web  
development

Nesting

Logic

Mixins & Selector  
Inheritance

Nesting

Error notifications

Documentation

## SASS

Based in Ruby  
Pure CSS. No  
compatibility issues

Compass

Better Nesting

Strange scope issues

Very PHP-ish

# TOOLS - COMPILERS

Compiler	</> Sass Language	</> Less Language	Mac	Windows
CodeKit	✓	✓	✓	
Koala	✓	✓	✓	✓
Hammer	✓		✓	
LiveReload	✓	✓	✓	✓
Prepros	✓	✓	✓	✓
Scout	✓		✓	✓
Crunch		✓	✓	✓

# TOOLS - EDITORS

Code Editor	</> Sass	</> Less
SublimeText	<a href="#">Sass Bundle</a>	<a href="#">Less-sublime</a>
Notepad++	<a href="#">Notepad-plus-plus</a>	<a href="#">Less-for-Notepad-plusplus</a>
VisualStudio	<a href="#">SassyStudio</a>	<a href="#">CSS is Less</a>
TextMate	<a href="#">SCSS.tmbundle</a>	<a href="#">Less.tmbundle</a>
Eclipse	<a href="#">Eclipse platform</a>	<a href="#">Eclipse for Less</a>
Coda	<a href="#">Coda Sass</a>	<a href="#">Coda 2 Less</a>

# SASS

# OVERVIEW

- SASS: „Syntactically Awesome Style Sheets“
- <http://sass-lang.com>
- Created by Hampton Catlin
- Further designed by Nathan Weizenbaum and Chris Eppstein
- Uses .SCSS or .SASS extensions
- Must be compiled into a CSS file
- Variables, if else statements, loops, mix ins, etc.
- 100% CSS3 compliant

# SASS - TWO SYNTAXES

- **Indented syntax (.SASS)**
  - Much different from traditional CSS syntax
  - No curly braces or semi colons
  - More strict
  - supports shorthand for directives like @mixin and @include. Instead of @mixin and @include you can use = and + characters, which require less typing and makes your code simpler, and easier to read.
- **Sassy CSS (SCSS)**
  - Superset of CSS
  - If you know css, already writing in SCSS
  - Easier to work with existing CSS
- Both are completely valid and neither will be deprecated

# SASS SYNTAXES VERSIONS

## SASS (indented syntax)

```
nav
  ul
    margin: 0
    padding: 0
    list-style: none

  li
    display: inline-block

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

## SCSS

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

  li { display: inline-block; }

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

# SASS - DEPRECATED SYNTAX

S. No.	Operator & Description
1	= It was used instead of : when setting variables and properties to values of SassScript.
2	= It was used instead of : whenever you are assigning default value of a variable.
3	! Instead of \$, ! was used as variable prefix. Functionality will not be changed when it is used instead of \$.

# INSTALLING SASS

- **General solution** – Rubby and console  
[https://www.tutorialspoint.com/sass/sass\\_installation.htm](https://www.tutorialspoint.com/sass/sass_installation.htm)
- **Third party:**
  - Visual studio – extensions:
    - Web Essentials - <http://vswebessentials.com>
    - Web Compiler -  
<https://marketplace.visualstudio.com/items?itemName=MadsKristensen.WebCompiler>
  - Visual Studio Code (<https://code.visualstudio.com/docs/languages/css>) – extensions:
    - <https://marketplace.visualstudio.com/items?itemName=spook.easysass>
  - Others: CodeKit, Koala ... - <https://graygrids.com/best-tools-resources-compile-manage-sass-less-stylus-css-preprocessors>

# ZADANIE – INSTALACJA ŚRODOWISKA

1. Zainstaluj Visual Studio Code (VSC) -  
<https://code.visualstudio.com>

2. W VSC zainstaluj rozszerzenia:

- Easy Sass -  
<https://marketplace.visualstudio.com/items?itemName=spook.easysass>
- Beautify css/sass/scss/less -  
<https://marketplace.visualstudio.com/items?itemName=michelemelluso.code-beautifier>
- SCSS IntelliSense -  
<https://marketplace.visualstudio.com/items?itemName=mrmlnc.vscode-scss>
- Live HTML Previewer -  
<https://marketplace.visualstudio.com/items?itemName=hdg.live-html-previewer>

# SASS – IMPORTANT FEATURES

- Comments
- Variables
- Nesting
- Partials
- Import
- Mixins
- Functions
- Extend/Inheritance
- Operators
- Scripts
- Directives
- Control Directives
- Output styles

# SCSS – COMMENTS

- Two kinds of comments:
  - single line comments (which do not exist in CSS)
  - block comments
- Single line comments will be deleted on compilation and block comments will remain in the compiled file

SCSS

```
$main-font-color: black;  
$main-background-color: white;  
  
body {  
  color: $main-font-color;  
}  
  
.content {  
  // color: $main-font-color;  
  /*  
   background-color: $main-background-color;  
   margin: 0;  
   */  
  margin: 0 auto;  
}
```

CSS

```
body {  
  color: black;  
}  
  
.content {  
  /*  
   background-color: $main-background-color;  
   margin: 0;  
   */  
  margin: 0 auto;  
}
```

## SCSS – COMMENTS

- **Interpolation** - within the multiline comments are resolved in the resulting CSS. You can specify variables or property names within the curly braces.

```
$version: "7.8";  
/* Framework version for the generated CSS is #{$version}. */
```

```
/* Framework version for the generated CSS is 7.8. */
```

# SCSS – VARIABLES

- A way to **store information that you want to reuse** throughout your stylesheet.
- You can store things like **colors**, **font stacks**, or any CSS value you think you'll want to reuse.
- Sass uses the **\$ symbol** to make something a variable.

SCSS

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

CSS

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

# VARIABLES - EXAMPLE

.scss

.sass

```
$blue: #3bbfce;  
$margin: 16px;  
  
.content-navigation {  
    border-color: $blue;  
    color:  
        darken($blue, 9%);  
}  
  
.border {  
    padding: $margin / 2;  
    margin: $margin / 2;  
    border-color: $blue;  
}
```

```
/* CSS */  
  
.content-navigation {  
    border-color: #3bbfce;  
    color: #2b9eab;  
}  
  
.border {  
    padding: 8px;  
    margin: 8px;  
    border-color: #3bbfce;  
}
```

# SCSS – VARIABLES

- Data types

S. No.	Data Type & Description	Example
1	<b>Numbers</b> It represents integer types.	2, 10.5
2	<b>Strings</b> It is sequence of characters defined within single or double quotes.	'Tutorialspoint', "Tutorialspoint"
3	<b>Colors</b> It is used for defining color value.	red, #008000, rgb(25,255,204)
4	<b>Booleans</b> It returns true or false boolean types.	10 > 9 specifies true
5	<b>Nulls</b> It specifies null value which is unknown data.	if(val==null){//statements}
6	<b>Space and Comma</b> Represents the values which are separated by spaces or commas.	1px solid #eeeeee, 0 0 0 1px
7	<b>Mapping</b> It maps from one value to another value.	FirsyKey: frstvalue, SecondKey: secvalue

## ZADANIE – ZMIENNE PODSTAWY

Dla dokumentu “index.html”:

1. Zdefiniuj poprzez **zmienną** przechowującą:
  - formatowanie tekstu (czcionka, rozmiar, kolor, ...)
  - paddingi
  - marginesy

Kolorystyka ma być zdefiniowania z tonacji niebieskiej.

2. Zastosuj zmienne dla stylowania dokumentu.
3. Zmodyfikuj kolorystykę dokumentu, aby była w tonacji zielonej.

# ZADANIE – ZMIENNE PODSTAWY (OPCJONALNE)

Dla dokumentu “cv.html”:

1. Zdefiniuj **zmienne** przechowujące:

- formatowanie tekstu (czcionka, rozmiar, kolor, ...)
- paddingi
- marginesy
- obramowania
- tła (np. sekcji)

Kolorystyka ma być zdefiniowania z tonacji niebieskiej.

2. Zastosuj zmienne dla stylowania dokumentu.
3. Zmodyfikuj kolorystykę dokumentu, aby była w tonacji zielonej.

# SCSS – VARIABLES

- **Data types – lists** - lists specify multiple values which are separated using spaces or commas. Even single value are considered as a list.

```
10px 11px, 15px 16px
```

If the inner list and the outer list have same separator, then you can use parentheses to specify where both the lists will start and stop. You can specify these lists as shown below –

```
{10px 11px} {15px 16px}
```

Use () instead of {}

# SCSS – VARIABLES

- **Data types – lists** - SASS uses some of the **list functions** such as:
  - **nth function** – It provides specific item of the list.
  - **join function** – It joins multiple lists into one.
  - **append function** – It appends the item to other end of the list.
  - **@each directive** – It provides styles for the each item in the list.
- **Getting element** - 

```
$list: "item-1", "item-2", "item-3";  
nth($list, 1) -> "item-1"
```

## ZADANIE – ZMIENNE LISTY

Dla dokumentu “index.html”:

1. Zdefiniuj **zmienną** przechowującą **paddingi** (wersje dla znaczących odstępów oraz niewielkich) jako **listę**.
2. Wykorzystaj zmienną dla zdefiniowania paddingów nagłówków (h1-h6).

## ZADANIE – ZMIENNE LISTY (*OPCJONALNE*)

Dla dokumentu “cv.html”:

1. Zdefiniuj **zmienną** przechowującą **paddingi** (wersje dla znaczących odstępów oraz niewielkich) jako **listę**.
2. Wykorzystaj zmienną dla zdefiniowania paddingów sekcji.

# SCSS – VARIABLES

- **Data types – Maps** - are combination of keys and values in which keys are used to represent the values. Maps define values into groups and can be accessed dynamically. It uses some of the functions such as:
  - **map-get** – It provides values of the map.
  - **map-merge** – It adds values to the map.
  - **@each directive** – It specifies styles for key/value pair in the map.
- Maps represent empty key/value pairs as ( ) with no elements and uses the **inspect (\$value)** function to display the output for the maps.

```
$map: (FirstKey: frstvalue, SecondKey: secvalue, Thirdkey: thdvalue);
```

## ZADANIE – ZMIENNE FUNKCJE MAPY

Dla dokumentu “index.html”:

1. Zdefiniuj **mapę** przechowującą rozmiary czcionek (bardzo mala, srednia, duza, bardzo duza).
2. Wykorzystaj mapę dla przypisania czcionek (map-get).

## ZADANIE – ZMIENNE FUNKCJE MAPY (OPCJONALNE)

Dla dokumentu “cv.html”:

1. Zdefiniuj **mapę** przechowującą rozmiary czcionek (bardzo mala, srednia, duza, bardzo duza).
2. Wykorzystaj mapę dla przypisania czcionek (map-get).

# SCSS – VARIABLES

- **Default** - You can set the default values for variables by adding *!default* flag to the end of the variable value. It will not re-assign the value, if it is already assigned to the variable.

```
$auroras-dress-color: blue;
```

```
$auroras-dress-color: red !default;
```

```
p:after {  
    color: blue; }  
}
```

```
p:after {  
    color: $auroras-dress-color;  
}
```

# ZADANIE – ZMIENNE FUNKCJE MAPY

Dla dokumentu “index.html”:

1. Przypisz **domyślny rozmiar czcionki** do stosowanej zmiennej.

# ZADANIE – ZMIENNE FUNKCJE MAPY (OPCJONALNE)

Dla dokumentu “cv.html”:

1. Przypisz **domyślny rozmiar czcionki** do stosowanej zmiennej.

# SCSS – NESTING

- **Nest your CSS selectors** in a way that follows the same visual hierarchy of your HTML.
- **Overly nested rules** will result in over-qualified CSS that could prove hard to maintain and is generally considered bad practice.

SCSS

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

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;  
}
```

## NESTED RULES - EXAMPLE

.scss

.sass

```
table.h1 {  
  margin: 2em 0;  
  td.ln {  
    text-align: right;  
  }  
}  
  
li {  
  font: {  
    family: serif;  
    weight: bold;  
    size: 1.2em;  
  }  
}
```

```
/* CSS */  
  
table.h1 {  
  margin: 2em 0;  
}  
table.h1 td.ln {  
  text-align: right;  
}  
  
li {  
  font-family: serif;  
  font-weight: bold;  
  font-size: 1.2em;  
}
```

# SCSS – NESTING

- Bad practice

SCSS (bad!)

```
.main-nav {  
  .nav-list {  
    padding-left: 0;  
    .nav-node {  
      display: inline-block;  
      .nav-link {  
        color: black;  
      }  
    }  
  }  
}
```

CSS

```
.main-nav .nav-list {  
  padding-left: 0;  
}  
.main-nav .nav-list .nav-node {  
  display: inline-block;  
}  
.main-nav .nav-list .nav-node .nav-link {  
  color: black;  
}
```

# SCSS – NESTING

- Good practice

SCSS (better!)

```
.main-nav {  
  .nav-list {  
    padding-left: 0;  
  }  
  .nav-node {  
    display: inline-block;  
  }  
  .nav-link {  
    color: black;  
  }  
}
```

CSS

```
.main-nav .nav-list {  
  padding-left: 0;  
}  
.main-nav .nav-node {  
  display: inline-block;  
}  
.main-nav .nav-link {  
  color: black;  
}
```

## ZADANIE – NESTING PODSTAWY

Dla dokumentu “index.html”:

1. Zmodyfikuj przypisanie stylu **stosując zagnieżdżenia** zgodnie z dobrymi praktykami.

# ZADANIE – NESTING PODSTAWY (OPCJONALNE)

Dla dokumentu “cv.html”:

1. Zmodyfikuj przypisanie stylu **stosując zagnieżdżenia** zgodnie z dobrymi praktykami.

# SCSS – NESTING

- **Nested Properties**

```
.line {  
  font: {  
    family: Lucida Sans Unicode;  
    size: 20px;  
    weight: bold;  
    variant: small-caps;  
  }  
}
```

```
.line {  
  font-family: Lucida Sans Unicode;  
  font-size: 20px;  
  font-weight: bold;  
  font-variant: small-caps;  
}
```

## ZADANIE – NESTING PODSTAWY

Dla dokumentu “index.html”:

1. Zmodyfikuj formatowanie tekstu stosując **zagnieżdżanie właściwości.**

# ZADANIE – NESTING PODSTAWY (OPCJONALNE)

Dla dokumentu “cv.html”:

1. Zmodyfikuj formatowanie tekstu stosując **zagnieżdżanie właściwości**.

# SCSS – NESTING

- **Placeholder** - a special type of selector. These look like class. They're meant to be used with the @extend directive.

```
.error {  
  border: 1px solid #f00;  
  background-color: #fdd;  
}
```

```
.seriousError {  
  @extend .error;  
  border-width: 3px;  
}
```

```
.error, .seriousError {  
  border: 1px solid #f00;  
  background-color: #fdd; }
```

```
.seriousError {  
  border-width: 3px; }
```

## ZADANIE – NESTING PLACEHOLDER

Dla dokumentu “index.html”:

1. Zdefiniuj dla header styl jako klasę definiując tło jako gradient.
2. Zdefiniuj dla footer styl jako klasę definiując jako **rozszerzenie stylu** dla headera dodatkowo definiując obramowanie.

## ZADANIE – NESTING PLACEHOLDER (OPCJONALNE)

Dla dokumentu “cv.html”:

1. Zdefiniuj klasę definiującą obramowanie dla elementów na podstawie wartości zmiennych.
2. Zdefiniuj klasę definiującą kolor wypełniania elementu **rozszerzającą** klasę definiującą obramowanie.

# SCSS – NESTING

- **Placeholder** - a special type of selector. These look like class and id selectors, except the # or . is replaced by %. They're meant to be used with the @extend directive.

```
%error {  
    border: 1px solid #f00;  
    background-color: #fdd;  
}  
  
.seriousError {  
    @extend %error;  
    border-width: 3px;  
}
```

```
.seriousError {  
    border: 1px solid #f00;  
    background-color: #fdd; }  
  
.seriousError {  
    border-width: 3px; }
```

## ZADANIE – NESTING PLACEHOLDER

Dla dokumentu “index.html”:

1. Zmodyfikuj klasę header, aby **nie była ona generowana do wynikowego pliku css**, ale mogła być częścią innych klas.

## ZADANIE – NESTING PLACEHOLDER (OPCJONALNE)

Dla dokumentu “cv.html”:

1. Zmodyfikuj klasę definującą obramowanie dla elementów, aby **nie była ona generowana do wynikowego pliku css**, ale mogła być częścią innych klas.

# SCSS – NESTING

- **Interpolation** - variables in selectors and property names using `#{}`  syntax.

```
p:after {  
  content: "I have #{8 + 2} books on SASS!";  
}
```

```
p:after {  
  content: "I have 10 books on SASS!";  
}
```

# ZADANIE – NESTING INTERPOLATION

Dla dokumentu “index.html”:

1. Zmodyfikuj klasę przypisaną do header, aby jej **nazwa była pobierana ze zmiennej**.
2. Zmodyfikuj klasę przypisaną do header, aby gradient był definiowany również dla starszych przeglądarek, gdzie **nazwa właściwości (bez przedrostka) pobierana jest ze zmiennej**. ([https://www.w3schools.com/css/css3\\_gradients.asp](https://www.w3schools.com/css/css3_gradients.asp))
3. Zmodyfikuj klasę przypisaną do header, definiując radius dla obramowania, działających również dla starszych przeglądarek, gdzie **nazwa właściwości (bez przedrostka) pobierana jest ze zmiennej**.

# ZADANIE – NESTING INTERPOLATION (OPCJONALNE)

Dla dokumentu “cv.html”:

1. Zdefiniuj klasę definiującą tło jako gradient, działający również dla starszych przeglądarek, gdzie **nazwa właściwości (bez przedrostka) pobierana jest ze zmiennej**.
2. Zdefiniuj klasę definiującą radius dla obramowania, działających również dla starszych przeglądarek, **gdzie nazwa właściwości (bez przedrostka) pobierana jest ze zmiennej**.

# SCSS – NESTING

- The Parent Selector „&“

SCSS

```
/* wrong */
.nav-link {
  color: black;
  .active { color: navy; }
  :hover { color: gray; }
}
/* correct */
.nav-link {
  color: black;
  &.active { color: navy; }
  &:hover { color: gray; }
}
```

CSS

```
/* wrong */
.nav-link { color: black; }
.nav-link .active { color: navy; }
.nav-link :hover { color: gray; }

/* correct */
.nav-link { color: black; }
.nav-link.active { color: navy; }
.nav-link:hover { color: gray; }
```

## ZADANIE – PARENT SELECTOR

Dla dokumentu “index.html”:

1. Za pomocą **parent selector** zdefinuj styl dla listy, aby element, na który nastąpi najechanie był podświetlony.

## ZADANIE – PARENT SELECTOR (OPCJONALNE)

Dla dokumentu “cv.html” za pomocą **parent selector**:

1. Zdefinuj styl, aby najechanie na poszczególne sekcje cv delikatnie je podświetlało.
2. Zdefiniuj proste menu u góry dokumentu pozwalające na przechodzenie do poszczególnych sekcji, ze stylami najechania na link oraz bycia aktywnym.

# SCSS – NESTING

- The Parent Selector „&“

SCSS (not easy to understand)

```
.social-link {  
  border: 0;  
  &-facebook {  
    background-image: url(facebook.png);  
  }  
  &-twitter {  
    background-image: url(twitter.png);  
  }  
}
```

CSS

```
.social-link {  
  border: 0;  
}  
.social-link-facebook {  
  background-image: url(facebook.png);  
}  
.social-link-twitter {  
  background-image: url(twitter.png);  
}
```

## ZADANIE – PARENT SELECTOR (OPCJONALNE)

Dla dokumentu “cv.html” za pomocą parent selector:

1. Zdefiniuj proste menu u góry dokumentu pozwalające na przechodzenie do poszczególnych sekcji ze stylami najechania na link oraz bycia aktywnym. (wcześniejsze zadanie)
2. Zdefiniuj różne kolory dla linków menu jako osobne klasy.

# SCSS – NESTING

- Redundant nesting

SCSS (redundant nesting)

```
.message {  
  span {  
    a {  
      border-color: gray;  
    }  
  }  
}  
body.error .message {  
  span {  
    a {  
      border-color: red;  
    }  
  }  
}
```

SCSS (optimized)

```
.message {  
  span {  
    a {  
      border-color: gray;  
    }  
  }  
}  
body.error & {  
  border-color: red;  
}
```

## ZADANIE – PARENT SELECTOR

Dla dokumentu “index.html” przy wykorzystaniu **parent selector**:

1. Zdefiniuj klasę wyroznienie, gdzie wszystkie paragrafy mają mieć czcionkę 1.5em.
2. Zdefiniuj klasę ważna, która ma być stosowana razem z klasą wyroznienie, gdzie wszystkie paragrafy mają mieć czcionkę 1.5em oraz kolor czerwony.

# SCSS – PARTIALS

- Partial Sass files contain **little snippets of CSS** that you can include in other Sass files.
- A partial is simply a Sass **file named with a leading underscore** – like `_partial.scss`.
- Sass partials are used with the **`@import` directive**.

## Partials

```
/*
Theme Name: My Theme
Theme URI: http://example.com/my-theme
Author: Me
Author URI: http://example.com/
Description: Just a random theme
*/
@import "_variables"; // Sass files with Leading underscore
@import "_normalize";
@import "_navigation";
```

# SCSS – IMPORT

- Each time you use **@import** in CSS it creates another HTTP request.
- Sass **@import** will **take the file that you want to import and combine it with the file you're importing into** so you can serve a single CSS file to the web browser.
- When you import a file you **don't need to include the file extension .SCSS.**

```
/*
Theme Name: Twentyfifteen Child
Theme URI: http://example.com/my-theme
Author: Me
Author URI: http://example.com/
Description: Just a random theme
Template: twentyfifteen
*/
// this is an ordinary CSS import
@import "../twentyfifteen/style.css";
// files without a extension in will merge in (they must have the extension .sass or .scss)
@import "_variables";
@import "_child_theme_styles";
```

# ZADANIE – PARTIALS, IMPORT

Dla dokumentu “index.html”:

1. Wszystkie zmienne **wstaw do pliku zmienne.scss.**
2. **Zimportuj plik zmienne.scss.**

## ZADANIE – IMPORT (OPCJONALNE)

Dla dokumentu “cv.html” :

1. Wszystkie zmienne **wstaw do pliku** zmienne.scss.
2. **Zimportuj plik** zmienne.scss.

# SCSS – MIXINS

- A **mixin** lets you **make groups of CSS declarations** that you want to reuse throughout your site.
- To **create a mixin** you use the **@mixin directive** and give it a name.
- You can **pass in values** to make your mixin more flexible. You can use a **variable inside the parentheses**.
- After you create your mixin, you can then **use it** as a CSS declaration starting with **@include followed by the name of the mixin**.

## 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); }
```

## CSS

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

# MIXINS - EXAMPLE

.scss

```
@mixin table-base {
  th {
    text-align: center;
    font-weight: bold;
  }
  td, th {padding: 2px}
}

@mixin left($dist) {
  float: left;
  margin-left: $dist;
}

#data {
  @include left(10px);
  @include table-base;
}
```

.sass

```
/* CSS */

#data {
  float: left;
  margin-left: 10px;
}
#data th {
  text-align: center;
  font-weight: bold;
}
#data td, #data th {
  padding: 2px;
}
```

## ZADANIE – MIXIN

Dla dokumentu “index.html” napisz mixin:

1. O nazwie **zaokrSekcja**, **definujący**: padding, obramowanie oraz radius. Napisz styl dla nagłówka (header), dla którego przygotowania zastosuj mixin.
2. Który definiuje border-radius **dla różnych przeglądarek**. Zastosuj go w mixinie o nazwie zaokrSekcja.
3. O nazwie **listaBezFormatowania**, **definujący** listę bez wypunktowania, paddingów, marginesów, itp.. Napisz styl dla list (ul), dla którego przygotowania zastosuj mixin.

## ZADANIE – MIXIN (OPCJONALNE)

Dla dokumentu “cv.html”:

1. Napisz **mixin(y)** przygotowujący(e) **poziome menu**, które zawsze jest u góry strony. Napisz klasę(y) korzystające z mixina dodatkowo stylujące menu (kolor tła, najechanie, czcionki, ...).
2. Mixiny z zadania dotyczącego dokumentu “index.html” wstaw do **osobnego pliku**, który **zimportuj**. Następnie wykorzystaj mixin o nazwie **zaokrSekcja**.

# SCSS – MIXINS

- **Mixin Arguments - Keyword Arguments** - the arguments, which are named can be passed in any order and the default values of argument can be omitted.

```
@mixin bordered($color, $width: 2px) {  
  color: #77C1EF;  
  border: $width solid black;  
  width: 450px;  
}  
  
.style {  
  @include bordered($color:#77C1EF, $width: 2px);  
}
```

```
.style {  
  color: #77C1EF;  
  border: 2px solid black;  
  width: 450px;  
}
```

## ZADANIE – MIXIN PARAMETERS

Dla dokumentu “index.html”:

1. Zmodyfikuj mixin o nazwie zaokrSekcja, aby **wartości pobierane były jako parametry** - część z nich niech posiada wartości domyślne. Zmodyfikuj styl dla nagłówka (header), dla którego zastosowano mixin, aby wywołanie mixina zawierało parametry.

## ZADANIE – MIXIN PARAMETERS (OPCJONALNE)

Dla dokumentu “cv.html”:

1. Zmodyfikuj mixin przygotowujący poziome menu, aby **wartości pobierane były jako parametry**. Zmodyfikuj klasę(y) korzystające z mixina, aby stosowały one parametry.

# SCSS – MIXINS

- **Mixin Arguments – Variable Arguments** - variable argument is used to pass any number of arguments to mixin. It contains keyword arguments passed to the function or mixin. Keyword arguments passed to the mixin can be accessed using *keywords(\$args) function* which return values mapped to String.

```
]}@mixin colors($background) {  
    background-color: $background;  
}  
  
$values: magenta, red, orange;  
  
.container {  
    @include colors($values);  
}
```

```
.container {  
    background-color: magenta, red, orange; }
```

# SCSS – MIXINS

- **Mixin Arguments – Content Blocks** - block of styles is passed to the mixin for the placement inside the styles. In @content directive location, styles gets included into the mixin.

```
@mixin element {  
  @content;  
}  
  
@include element {  
  .block {  
    color: green;  
  }  
}
```

```
.block {  
  color: green;  
}
```

## ZADANIE – MIXIN @CONTENT

Dla dokumentu “index.html”:

1. Zmodyfikuj mixin o nazwie zaokrSekcja i klasę go stosującą, aby zawartość styli była wprowadzana jako **@content**.

# SCSS – MIXINS - NESTING

- **Placeholder** - these are specified with "#".

```
@mixin firefox-message($selector) {  
  body.firefox #{$selector}:before {  
    content: "Hi, Firefox users!";  
  }  
}  
  
@include firefox-message(".header");
```

```
body.firefox .header:before {  
  content: "Hi, Firefox users!"; }
```

# ZADANIE – MIXIN NESTING

Dla dokumentu “index.html”:

1. Napisz mixin o nazwie **stylNaglowek** z parametrami: naglowek (h1-h6), klasa (np. “wazne”), kolor (np. “green”). Zgodnie z parametrami mixin ma przygotować klasę o formie **nagłówek.klasa** (np. h2.wazne).
2. Poprzez mixin przygotuj kilka klas oraz zastosuj je dla nagłówków.

## ZADANIE – MIXIN NESTING (OPCJONALNE)

Dla dokumentu “cv.html”:

1. Napisz mixin o nazwie **stylDiv** z parametrami: klasa (np. “zwrocUwage”), kolor (np. “green”), czcionka, rozmiar, padding, margines, float i ew. inne. Zgodnie z parametrami mixin ma **przygotować klasę** o formie ***div.klasa*** (np. div.wazne).
2. Poprzez mixin przygotuj kilka klas oraz zastosuj je dla sekcji.

# SCSS – FUNCTIONS

- SASS supports the use of **functions** by providing some **keyword arguments**, which are specified using normal CSS function syntax.

```
p {  
  color: hsl($hue: 0, $saturation: 50%, $lightness: 50%);  
}
```

```
p {  
  color: hsl(290,60%,70%);  
}
```

```
p {  
  color: #d185e0;  
}
```

## ZADANIE – FUNCTIONS

W dokumencie “index.html”:

1. Zmodyfikuj mixin o nazwie stylNaglowek, aby można było **wskazać przezroczystość**. Za pomocą stosownej funkcji (**lighten**, **darken**, **invert**, **hsl**) dokonaj modyfikacji sposobu przypisania koloru czcionku, aby brała pod uwagę przezroczystość.
2. Poprzez mixin przygotuj kilka klas oraz zastosuj je dla nagłówków.

## ZADANIE – FUNCTIONS (OPCJONALNE)

W dokumencie “cv.html”:

1. Zmodyfikuj mixin o nazwie stylDiv, aby można było wskazać przezroczystość. Za pomocą stosownej funkcji (**lighten**, **darken**, **invert**, **hsl**) dokonaj modyfikacji sposobu przypisania koloru czcionku, aby brała pod uwagę przezroczystość.
2. Poprzez mixin przygotuj kilka klas oraz zastosuj je dla sekcji.

# SCSS – FUNCTIONS

- You can create your **own function** and use them in your script context or can be used with any value. **Functions are called by using the function name** and with any parameters.

```
$first-width: 5px;  
$second-width: 5px;  
  
@function adjust_width($n) {  
    @return $n * $first-width + ($n - 1) * $second-width;  
}  
  
#set_width { padding-left: adjust_width(10); }
```

```
#set_width {  
    padding-left: 95px;  
}
```

## ZADANIE – FUNKCJE

Dla dokumentu “index.html”:

1. Zdeklaruj **zmienną** dla przechowywania podstawowej wartości paddingu jako 3px.
2. Napisz **funkcję**, która zwraca wartość paddingu jako wielokrotność wartości podstawowej parametru.
3. Wykorzystaj funkcję dla przypisania paddingu poszczególnych sekcji. Zmień wartość zmiennej przechowywania podstawowej wartości paddingu na 10px.

## ZADANIE – FUNKCJE (OPCJONALNE)

Dla dokumentu “cv.html”:

1. Zdeklaruj **zmienną** dla przechowywania podstawowej wartości szerokości sekcji jako 100%.
2. Napisz **funkcję**, która zwraca wartość szerokości, jako wartość podstawową pomniejszoną o wartość parametru.
3. Wykorzystaj funkcję dla przypisania szerokości poszczególnych sekcji. Zmień wartość zmiennej przechowywania podstawowej szerokości na 90%.

# SCSS – FUNCTIONS

- **Defining Custom SASS Functions** - You can define your own SASS functions while using Ruby API. You can add your custom functions by adding them to Ruby methods.

```
module Sass::Script::Functions
  def reverse(string)
    assert_type string, :String
    Sass::Script::Value::String.new(string.value.reverse)
  end
  declare :reverse, [:string]
end
```

# SCSS – MIXINS (EXAMPLES)

- Different contents per pixel ratio (Retina)

```
@mixin pixel-ratio($ratio){  
  @media (-webkit-min-device-pixel-ratio: $ratio), /* Webkit browsers */  
          (min--moz-device-pixel-ratio: $ratio),      /* old Firefox (prior to FF 16) */  
          (min-resolution: #{$ratio}dppx),            /* the standard using dppx */  
          (min-resolution: #{$ratio * 96}dpi){        /* fallback using dpi */  
    @content;  
  }  
}
```

# SCSS – MIXINS (EXAMPLES)

- Different background images per pixel ratio

```
@mixin background-image-hidpi($normalImage, $hdpiImage, $ratio){  
  // normal image  
  background-image: url('#{$normalImage}');  
  // high dpi image  
  @include pixel-ratio($ratio){  
    background-image: url('#{$hdpiImage}');  
  }  
}
```

# SCSS – MIXINS (EXAMPLES)

- Mobile First with IE<9

## Mobile First with IE<9

```
@mixin respond-to-ie($size){  
  @if $old-ie {  
    @content;  
  } @else {  
    @media all and (min-width: $size) {  
      @content;  
    }  
  }  
}  
  
<!--[if lte IE 8]>  
<link rel="stylesheet" href="old_ie.css" media="all" />  
<![endif]-->  
<!--[if gt IE 8]><!-->  
<link rel="stylesheet" href="styles.css" media="all" />  
<!--<![endif]-->
```

style.scss

```
$old-ie: false !default;  
@import "_respond-to-ie";  
  
@include respond-to-ie($breakMedium){  
  body {  
    margin: 0;  
  }  
}
```

old\_ie.scss

```
$old-ie: true;  
@import styles.scss
```

# SCSS – ALTERNATIVE STYLESHEETS

```
<link rel="stylesheet" title="(current)" href="topnav_current.css" />
<link rel="stylesheet alternate" title="red" href="topnav_red.css" />
<link rel="stylesheet alternate" title="blue" href="topnav_blue.css" />
<link rel="stylesheet alternate" title="green" href="topnav_green.css" />

// overwrite navigation Colors
$topnavBackgroundColor: transparent;
$topnavColor: $colorWhite;
$topnavHoverBackgroundColor: $colorWhite;
$topnavHoverColor: $colorWhite;
$topnavActiveBackgroundColor: #3F80E9;
$topnavActiveColor: $colorWhite;
$subnavBackgroundColor: #3F80E9;
$subnavColor: $colorWhite;
$subnavDividerColor: $colorWhite;

@import "topnav_medium";
```

## ZADANIE – ALTERNATIVE STYLES

Dla dokumentu “index.html”:

1. Przygotuj **alternatywny styl** w tonacji czerwonej.
2. Sprawdź w przeglądarce **przełączanie się pomiędzy stylami**

<https://www.w3.org/Style/Examples/007/alternatives.html>

## ZADANIE – ALTERNATIVE STYLES (OPCJONALNE)

Dla dokumentu “cv.html”:

1. Przygotuj **alternatywny styl** w tonacji czerwonej.
2. Sprawdź w przeglądarce **przełączanie się pomiędzy stylami**

<https://www.w3.org/Style/Examples/007/alternatives.html>

# SCSS – EXTEND/INHERITANCE

- Using @extend lets you **share a set of CSS properties** from one selector to another.
- Helps keep your Sass very **dry/consistent**.

## SCSS

```
.message {  
  border: 1px solid #ccc;  
  padding: 10px;  
  color: #333;  
}  
  
.success {  
  @extend .message;  
  border-color: green;  
}  
  
.error {  
  @extend .message;  
  border-color: red;  
}
```

## CSS

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

# INHERITANCE - EXAMPLE

.scss

.sass

```
.error {  
  border: 1px #f00;  
  background: #fdd;  
}  
.error.intrusion {  
  font-size: 1.3em;  
  font-weight: bold;  
}  
  
.badError {  
  @extend .error;  
  border-width: 3px;  
}
```

```
/* CSS */  
  
.error, .badError {  
  border: 1px #f00;  
  background: #fdd;  
}  
  
.error.intrusion,  
.badError.intrusion {  
  font-size: 1.3em;  
  font-weight: bold;  
}  
  
.badError {  
  border-width: 3px;  
}
```

# SCSS – EXTEND/INHERITANCE

- **Multiple Extends** - more than one selector can be extended by a single selector.

```
.style {  
  font-size: 25px;  
  font-style: italic;  
}  
  
h2 {  
  color: #61C8E1;  
}  
  
.container {  
  @extend .style;  
  @extend h2  
}
```

```
.style, .container {  
  font-size: 25px;  
  font-style: italic;  
}  
  
h2, .container {  
  color: #61C8E1;  
}
```

# SCSS – EXTEND/INHERITANCE

- **Chaining Extends** - the first selector extended by second selector and the second selector is extended by third selector, and so on.

```
.style {  
    font-size: 25px;  
    font-style: italic;  
}  
  
h2 {  
    color: #61C8E1;  
    @extend .style  
}  
  
.container {  
    @extend h2  
}
```

```
.style, h2, .container {  
    font-size: 25px;  
    font-style: italic;  
}  
  
h2, .container {  
    color: #61C8E1;  
}
```

# SCSS – EXTEND/INHERITANCE

- **Selector Sequences** - the nested selector can use @extend by themselves.

```
.style {  
  font-size: 25px;  
  font-style: italic;  
  color: #61C8E1;  
}  
  
h2 .container {  
  @extend .style  
}
```

```
.style, h2 .container {  
  font-size: 25px;  
  font-style: italic;  
  color: #61C8E1;  
}
```

# SCSS – EXTEND/INHERITANCE

- **Merging Selector Sequences** - it merges two sequences i.e. one sequence extend another sequence that is present in other sequence.

```
.container .style a {  
    font-weight: bold;  
}  
  
#id .example {  
    @extend a;  
}
```

```
.container .style a,  
.container .style #id  
.example, #id .container  
.style .example {  
    font-weight: bold;  
}
```

# ZADANIE – EXTEND/INHERITANCE

Dla dokumentu “index.html”:

1. Przygotuj **klasę** o nazwie sekcja z wartościami obramowania, czcionki, kolorów, padding, margines, itp.
2. Zdefiniuj style dla headera oraz footera **rozszerzając klasę** sekcja.

## ZADANIE – EXTEND/INHERITANCE (OPCJONALNE)

Dla dokumentu “cv.html”:

1. Przygotuj **klasę** o nazwie sekcja z wartościami obramowania, czcionki, kolorów, padding, margines, itp.
2. Zdefiniuj style dla sekcji na podstawie (rozszerzenie) klasy sekcja, uwzględniając również najechanie na nie (hover).

# SCSS – EXTEND/INHERITANCE

- **Only Selectors** - percent character (%) can be used anywhere a id or class, it prevents its own ruleset from being rendered to CSS.

```
.style a%extreme {  
    font-size: 25px;  
    font-style: italic;  
    color: #61C8E1;  
}  
  
.container {  
    @extend %extreme;  
}
```

```
.style a.container {  
    font-size: 25px;  
    font-style: italic;  
    color: #61C8E1;  
}
```

# ZADANIE – EXTEND/INHERITANCE ONLY SELECTORS

Dla dokumentu “index.html”:

1. Zmodyfikuj style, aby klasa bazowa (po której jest dziedziczenie) o nazwie sekcja **nie była częścią generowanego css**.

# ZADANIE – EXTEND/INHERITANCE ONLY SELECTORS (OPCJONALNE)

Dla dokumentu “cv.html”:

1. Zmodyfikuj style, aby klasa bazowa (po której jest dziedziczenie) o nazwie sekcja **nie była częścią generowanego css**.

# SCSS – EXTEND/INHERITANCE

- The **!optional Flag** - used to allow the `@extend` for not to create any new selector or fail.

```
h2.important {  
  @extend .style  
!optional;  
}
```

A blank compile page gets  
display.

# SCSS – EXTEND/INHERITANCE

- **Directives** - if @extend is used inside the @media then it can extend the selectors only that are present within the same directive blocks.

```
@media print {  
    .style {  
        font-size: 25px;  
        font-style: italic;  
    }  
  
    .container {  
        @extend .style;  
        color: #61C8E1;  
    }  
}
```

```
@media print {  
    .style, .container {  
        font-size: 25px;  
        font-style: italic;  
    }  
  
    .container {  
        color: #61C8E1;  
    }  
}
```

# ZADANIE – EXTEND/INHERITANCE ONLY

## SELECTORS

Dla dokumentu “index.html”:

1. Zdefiniuj odpowiednie style dla headera oraz footera na podstawie klasy sekcja dopasowane do trybu wydruku.

# ZADANIE – EXTEND/INHERITANCE ONLY SELECTORS (OPCJONALNE)

Dla dokumentu “cv.html”:

1. Zdefiniuj odpowiednie style dla headera oraz footera na podstawie klasy sekcja dopasowane do trybu wydruku.

# SCSS – OPERATORS

- Sass has a handful of standard **math operators** like:  
+, -, \*, /, and %.

SCSS

```
.container { width: 100%; }

article[role="main"] {
  float: left;
  width: 600px / 960px * 100%;
}

aside[role="complimentary"] {
  float: right;
  width: 300px / 960px * 100%;
}
```

CSS

```
.container {
  width: 100%;
}

article[role="main"] {
  float: left;
  width: 62.5%;
}

aside[role="complimentary"] {
  float: right;
  width: 31.25%;
}
```

# SCSS – OPERATORS

- You **cannot use incompatible units**, while adding number with px and em leads to produce invalid CSS. Therefore, SASS will display an error if you use invalid units in CSS.
- SASS **supports relational operators** like <, >, <=, >= and equality operators ==, !=.

# SCSS – OPERATORS

- **Color Operations** - SASS allows using color components along with the arithmetic operations and any color expression returns a SassScript color value.

```
$color1: #333399;  
$color2: #CC3399;  
  
p{  
  color: $color1 + $color2;  
}
```

```
p {  
  color: #ff66ff;  
}
```

# SCSS – OPERATORS

- **String Operations** - you can use + operation to concatenate strings.

```
p {  
  font-size: 5px + 10px;  
}
```

```
p {  
  font-size: 15px;  
}
```

# SCSS – OPERATORS

- **Parentheses** - are a pair of signs, which are usually marked off by round brackets ( ) or square brackets [], providing symbolic logic that affect the order of operations.

```
p {  
  font-size: 5px + (6px * 2);  
  color:#ff0000;  
}
```

```
p {  
  font-size: 17px;  
  color: #ff0000;  
}
```

# ZADANIE – OPERATORS

Dla dokumentu “index.html”:

1. Zdefiniuj, aby <div> z tekstem “Sample paragraph of text” oraz <div> z listą były obok siebie i miały **procentowy rozmiar**, gdzie:
  - Pierwszy z nich ma mieć **800px z 1024px**.
  - Drugi z nich ma mieć **pozostałą część**.

## ZADANIE – OPERATORS (OPCJONALNE)

Dla dokumentu “cv.html”:

1. Zdefiniuj, aby dane w sekcjach Education oraz Courses & Diploma miały formę

1995-1998 Bachelor degree. Accounting. University of Northcap.

Gdzie daty zajmują **200px** z **1024px**, a opis **pozostałą część**. Wartości szerokości mają być **wyrażone procentowo**.

# SCSS – OPERATORS

- **Boolean Operations** - you can perform Boolean operations on Sass script by using *and*, *or* and *not* operators.

```
$age:20;  
.bool {  
  @if ($age > 10 and $age < 25) {  
    color: green;  
  }  
}
```

```
.bool {  
  color: green;  
}
```

# ZADANIE – OPERATORS BOOLEAN

Dla dokumentu “index.html”:

1. Zdefiniuj **zmienną** o nazwie podstRozmiarTekst z wartością 1.1em.
2. Napisz **funkcję** zwracającą rozmiar tekstu na podstawie **dwóch parametrów**: tryb (powiększenie/pomniejszenie), wartość (np. 20%). Funkcja zwraca odpowiedni rozmiar czcionki na podstawie wartości zmiennej bazowej.
3. Wykorzystaj funkcję dla określenia wartości rozmiaru tekstu.

# ZADANIE – OPERATORS BOOLEAN (OPCJONALNE)

Dla dokumentu “cv.html”:

1. Zdefiniuj **zmienną** o nazwie `podstRozmiarTekst` z wartością `1em`.
2. Napisz **funkcję** zwracającą rozmiar tekstu na podstawie **dwoch parametrów**: tryb (powiększenie / pomniejszenie), wartość (np. `20%`). Funkcja zwraca odpowiedni rozmiar czcionki na podstawie wartości zmiennej bazowej.
3. Dodaj **warunek nie powalający** o zmniejszenie czcionki o więcej niż `50%`, oraz zwiększenie o więcej niż `300%`.
4. Wykorzystaj funkcję dla określenia wartości rozmiaru tekstu.

# SCSS – DIRECTIVES

S. No.	Directives & Description
1	<b>@import</b> It imports the SASS or SCSS files, it directly takes the filename to import.
2	<b>@media</b> It sets the style rule to different media types.
3	<b>@extend</b> @extend directive is used to share rules and relationships between selectors.
4	<b>@at-root</b> @at-root directive is a collection of nested rules, which is able to make style block at root of the document.
5	<b>@debug</b> @debug directive detects the errors and displays the SassScript expression values to the standard error output stream.
6	<b>@warn</b> @warn directive is used to give cautionary advice about the problem; it displays the SassScript expression values to the standard error output stream.
7	<b>@error</b> @error directive displays the SassScript expression value as fatal error.

# SCSS – DIRECTIVES

- **@media directive** sets the style rule to different media types.
- **@media directive can be nested inside the selector SASS but it is bubbled up to the top level of the stylesheet.**

```
h2 {  
  color: #77C1EF;  
}  
  
.style {  
  width: 900px;  
  
  @media screen and (orientation: portrait) {  
    width: 500px;  
    margin-left: 120px;  
  }  
}
```

```
h2 {  
  color: #77C1EF;  
}  
  
.style {  
  width: 900px;  
}  
@media screen and (orientation: portrait) {  
  .style {  
    width: 500px;  
    margin-left: 120px;  
  }  
}
```

# SCSS – DIRECTIVES

- **@at-root (without: ...)** and **@at-root (with: ...)** - we can move the style outside of nested directive.

```
@media print {  
  .style {  
    height: 8px;  
    @at-root (without: media) {  
      color: #808000;;  
    }  
  }  
}
```

```
@media print {  
  .style {  
    height: 8px;  
  }  
}  
  
.style {  
  color: #808000;  
}
```

# SCSS – DIRECTIVES

- **@at-root (without: ...)** and **@at-root (with: ...)** - selector excludes the selector by default.

```
h2 {  
  color: #808000;  
  background-color: #DB7093;  
  
  @at-root {  
    .style{  
      font-size: 20px;  
      font-style: bold;  
      color: #B8860B;  
    }  
  }  
}
```

```
h2 {  
  color: #808000;  
  background-color: #DB7093;  
}  
.style {  
  font-size: 20px;  
  font-style: bold;  
  color: #B8860B;  
}
```

## ZADANIE – DIRECTIVES MEDIA

Dla dokumentu “index.html”:

1. Zdefiniuj **poprzez style i media query** odpowiednie z punktu widzenia UX wyświetlanie witryny dla urządzeń o rozmiarze ekranu do 420px oraz powyżej 1024px.

## ZADANIE – DIRECTIVES MEDIA (OPCJONALNE)

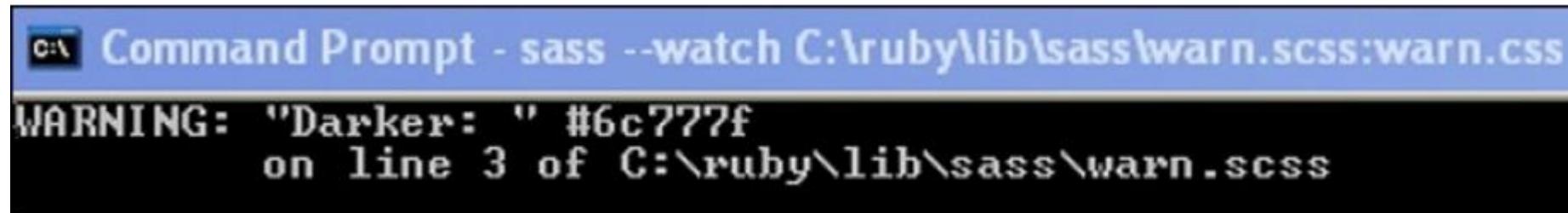
Dla dokumentu “cv.html”:

1. Zdefiniuj **poprzez style i media query** odpowiednie z punktu widzenia UX wyświetlanie witryny dla urządzeń o rozmiarze ekranu do 420px oraz powyżej 1024px.

## SCSS – DIRECTIVES

- **@warn** - directive is used to give cautionary advice about the problem. It displays the SassScript expression values to the standard error output stream.

```
$main-color: #bdc3c7;  
  
@warn "Darker: " darken($main-color, 30%);
```



The screenshot shows a Windows Command Prompt window. The title bar reads "Command Prompt - sass --watch C:\ruby\lib\sass\warn.scss:warn.css". The main area of the window displays the following text:

```
WARNING: "Darker: " #6c777f  
on line 3 of C:\ruby\lib\sass\warn.scss
```

# SCSS – DIRECTIVES

- **@error** - directive displays the SassScript expression value as fatal error.

```
$colors: (  
    blue: #c0392b,  
    black: #2980b9,  
  
);  
  
@function style-variation($style) {  
    @if map-has-key($colors, $style) {  
        @return map-get($colors, $style);  
    }  
  
    @error "Invalid color: '#{style}'.";  
}  
  
.container {  
    style: style-variation(white);  
}
```

```
Command Prompt - sass --watch C:\ruby\lib\sass\error.scss:error.css  
>>> Change detected to: lib/sass/error.scss  
      error C:\ruby\lib\sass\error.scss (Line 12: Invalid color: 'white'.)
```

# ZADANIE – OPERATORS BOOLEAN

Dla dokumentu “index.html”:

1. Dla funkcji zwracającej rozmiar tekstu na podstawie dwóch parametrów: tryb (powiększenie / zmniejszenie), wartość (np. 20%) dodaj **warunek** powodujący, że zmniejszenie czcionki o więcej niż 50% lub zwiększenie o więcej niż 300% zgłosi błąd.

# ZADANIE – OPERATORS BOOLEAN (OPCJONALNE)

Dla dokumentu “cv.html”:

1. Dla funkcji zwracającej rozmiar tekstu na podstawie dwóch parametrów: tryb (powiększenie / zmniejszenie), wartość (np. 20%) dodaj **warunek** powodujący, że zmniejszenie czcionki o więcej niż 50% lub zwiększenie o więcej niż 300% zgłosi błąd.

# SCSS – CONTROL DIRECTIVES

S. No.	Control Directive & Expression with Description
1	<b>if()</b>  Based on the condition, <i>if()</i> function returns only one result from two possible outcomes.
2	<b>@if</b>  The <i>@if</i> directive accepts SassScript expressions and uses the nested styles whenever the result of the expression is anything other than <i>false</i> or <i>null</i> .
3	<b>@for</b>  The <i>@for</i> directive allows you to generate styles in a loop.
4	<b>@each</b>  In <i>@each</i> directive, a variable is defined which contains the value of each item in a list.
5	<b>@while</b>  It takes SassScript expressions and until the statement evaluates to <i>false</i> it iteratively outputs nested styles.

# SCSS – CONTROL DIRECTIVES

- **if()** - function returns only one result from two possible outcomes. The result of the function can be referred to the variable that may not be defined or to have further calculations.
- **Syntax** - if(expression, value1, value2 ).

```
h2 {  
  color: if( 1 + 1 == 2 , green , red);  
}
```

```
h2 {  
  color: green;  
}
```

# SCSS – CONTROL DIRECTIVES

- **@if** - directive is used to selectively execute the code statements based on the result of evaluating an expression.
- **@if** statements can be followed by several **@else if** statements.
- **Syntax –**

```
@if expression { // CSS codes}  
@else if condition { // CSS codes}  
@else { // CSS codes}
```

```
$type: audi;  
p {  
  @if $type == benz {  
    color: red;  
  } @else if $type == mahindra {  
    color: blue;  
  } @else if $type == audi {  
    color: green;  
  } @else {  
    color: black;  
  }  
}
```

```
p {  
  color: green;  
}
```

# ZADANIE – @IF

Dla dokumentu “index.html”:

1. Zdefiniuj **mixin** zwracający różny kolor tła oraz rozmiar czcionki (zastosowanie wcześniejszej funkcji) w zależności od **typu nagłówka** (h1-h6), gdzie typ podawany jest jako **parametr**.
2. Wykorzystaj mixin dla stylowania nagłówków.

# SCSS – CONTROL DIRECTIVES

- **@for** - directive allows you to generate styles in a loop. The counter variable is used to set the output for each iteration.
- **Syntax:**
  - **@for \$var from <start> through <end>** - If the *<start>* is greater than *<end>* then the counter variable is decremented and when *<start>* is lesser than *<end>* the counter variable will be incremented.
  - **@for \$var from <start> to <end>** - If the *<start>* is greater than *<end>* then the counter variable is decremented and when *<start>* is lesser than *<end>* the counter variable will be incremented.

# SCSS – CONTROL DIRECTIVES

- **@for** - examples

```
@for $i from 1 through 4 {  
  .p#{$i} { padding-left : $i * 10px; }  
}
```

```
@for $i from 1 to 4 {  
  .p#{$i} { padding-left : $i * 10px; }  
}
```

```
.p1 {  
  padding-left: 10px;  
}  
  
.p2 {  
  padding-left: 20px;  
}  
  
.p3 {  
  padding-left: 30px;  
}  
  
.p4 {  
  padding-left: 40px;  
}
```

```
.p1 {  
  padding-left: 10px;  
}  
  
.p2 {  
  padding-left: 20px;  
}  
  
.p3 {  
  padding-left: 30px;  
}
```

## ZADANIE – @FOR

Dla dokumentu “index.html”:

1. Zdefiniuj **mixin** przy wykorzystaniu **pętli for** (numerowanie) style dla nagłówków (h1-h6) – określenie rozmiaru tekstu (każdy kolejny ma o 10% mniejszy tekst niż wcześniejszy).
2. Wykorzystaj przygotowane style.

## ZADANIE – @FOR (OPCJONALNE)

Dla dokumentu “cv.html”:

1. Zdefiniuj mixin przy wykorzystaniu **pętli for** (liczba kolumn) style dla sekcji zapewniające **responsywność** i rozmiar, analogicznie do **Bootstrapa** (<http://getbootstrap.com/css>, od sekcji Grid options).
2. Wykorzystaj przygotowane style (opcjonalne).
3. Zmodyfikuj mixin, aby liczba kolumn oraz rozmiary urządzeń były przekazywane jako parametry.

# SCSS – CONTROL DIRECTIVES

- **@each** - a variable is defined which contains the value of each item in a list
- **Syntax** - @each \$var in <list or map>

```
@each $color in red, green, yellow, blue {  
  .p_#${$color} {  
    background-color: #{$color};  
  }  
}
```

```
.p_red {  
  background-color: red;  
}  
  
.p_green {  
  background-color: green;  
}  
  
.p_yellow {  
  background-color: yellow;  
}  
  
.p_blue {  
  background-color: blue;  
}
```

# SCSS – CONTROL DIRECTIVES

- **@each - Multiple Assignment** - multiple values can also be used with `@each` directive like `$var1, $var2, $var3, ...` in `<list>`.
- **Syntax** - `@each $var1, $var2, $var3 ... in <list>`

```
@each $color, $border in (aqua, dotted), (red, solid), (green, double)
  .#$color {
    background-color : $color;
    border: $border;
  }
}
```

```
.aqua {
  background-color: aqua;
  border: dotted;
}

.red {
  background-color: red;
  border: solid;
}

.green {
  background-color: green;
  border: double;
}
```

# SCSS – CONTROL DIRECTIVES

- **@each - Multiple Assignment with Maps** - with maps and they are considered as lists of pairs. like \$var1, \$var2, \$var3, ... in <list>
- **Syntax** - @each \$var1, \$var2 in <map>

```
@each $header, $color in (h1: red, h2: green, h3: blue) {  
  #{$header} {  
    color: $color;  
  }  
}
```

```
h1 {  
  color: red;  
}  
  
h2 {  
  color: green;  
}  
  
h3 {  
  color: blue;  
}
```

## ZADANIE – @FOR, @EACH

Dla dokumentu “index.html”:

1. Zdefiniuj (zmodyfikuj) **mixin** przy wykorzystaniu pętli **for** (numerowanie) oraz **each** (kolory) style dla nagłówków – określenie rozmiaru tekstu oraz kolorów (każdy nagłówek ma mieć wszystkie wersje kolorystyczne).
2. Wykorzystaj przygotowane style.

## ZADANIE – @FOR, @EACH (OPCJONALNE)

Dla dokumentu “cv.html”:

1. Zdefiniuj **mixin** (zmodyfikuj istniejący) przy wykorzystaniu pętli **for** (liczba kolumn) oraz **each** (rozmiar urządzenia, np. sm czy md) style dla sekcji zapewniające responsywność i rozmiar, analogicznie do Bootstrapa (<http://getbootstrap.com/css>, od sekcji Grid options).
2. Wykorzystaj przygotowane style (opcjonalne).

# SCSS – CONTROL DIRECTIVES

- **@while** - directive takes SassScript expressions and until the statement evaluates to false, it iteratively outputs nested styles.
- **Syntax** - `while(condition) { // CSS codes}`

```
$i: 50;  
@while $i > 0 {  
  .padding-#{$i} { padding-left: 1px * $i; }  
  $i: $i - 10;  
}
```

```
.padding-50 {  
  padding-left: 50px;  
}  
  
.padding-40 {  
  padding-left: 40px;  
}  
  
.padding-30 {  
  padding-left: 30px;  
}  
  
.padding-20 {  
  padding-left: 20px;  
}  
  
.padding-10 {  
  padding-left: 10px;  
}
```

## ZADANIE – @WHILE

Dla dokumentu “index.html”:

1. Zdefiniuj **mixin** przy wykorzystaniu pętli **while** tworzący style z wartościami marginesów od 5px do 30px, co 5px (krok).
2. Wykorzystaj przygotowane style.
3. Zdefiniuj, aby przedział dla marginesów oraz krok były podawane jako parametry.

# SCSS – OUTPUT STYLE

- **:nested**

```
| #first {  
|   background-color: #00FFFF;  
|   color: #C0C0C0;  
| }  
  
| #first p {  
|   width: 10em;  
| }  
|
```

```
#first {  
  background-color: #00FFFF;  
  color: #C0C0C0;  
}  
|
```

```
#first p {  
  width: 10em;  
}
```

- **:expanded**

```
#first {  
  background-color: #00FFFF; color: #C0C0C0;  
}
```

- **:compact**

```
#first {  
  background-color:#00FFFF;color:#C0C0C0  
}
```

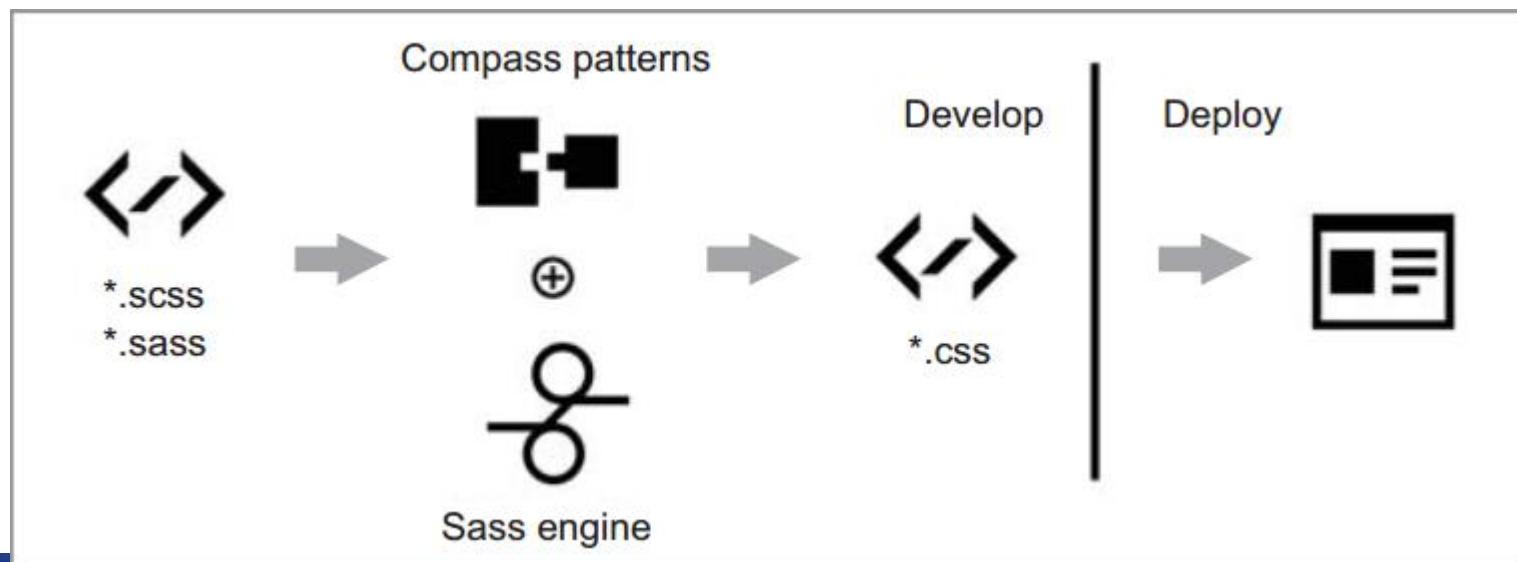
- **:compressed**

# SASS EXTENSIONS

- There are several extensions to add more features to SASS
- Probably the most used of those extensions is Compass
- It extends SASS with useful helpers and mixins
- Especially when using it for CSS3

# SASS EXTENSIONS - COMPASS

- Compass is an open-source CSS **Authoring Framework**.
- Compass is a **Sass framework** designed to make the work of styling the web smooth and efficient.
- Compass comes with a host of **Sass mixins and functions** that are organized into modules.



# SASS EXTENSIONS - COMPASS

## Installation:

- **General:** - <http://compass-style.org/install>
  - Downloading and running the Ruby installer
  - Installing Sass and Compass
    - gem install Sass
    - gem install Compass
  - Verification
    - sass -v
    - compass –v
- **Visual Studio:**
  - NuGet Package - Ruby.Compass
  - Bower Package – compass-mixins
- **Other – Scout App** - <http://scout-app.io>

## ZADANIE – COMPASS INSTALACJA

Zainstaluj bibliotekę dla SCSS - Compass – jednym z poniższych sposobów:

1. <https://github.com/Igosuki/compass-mixins>
2. <http://scout-app.io>
3. <http://compass-style.org/install>

# COMPASS

- Powerful mixins - Forget about browser suffixes
- Scaffolding - Blueprint (similar to Twitter Bootstrap)

```
#border-radius {  
  @include border-radius(25px);  
}
```

```
#border-radius {  
  -webkit-border-radius: 25px;  
  -moz-border-radius: 25px;  
  -ms-border-radius: 25px;  
  -o-border-radius: 25px;  
  border-radius: 25px;  
}
```

# COMPASS

- Mixins

```
1 <div class='ex'
2   <div id='pad'
3     Padding Bo
4   </div>
5   <div id='bor'
6     Border Bo
7   </div>
8 </div>
```

```
1 .example {
2   padding: 2em;
3 }
4 .example div {
5   background-color: blue;
6   border: 10px solid rgba(255, 0, 0, 0.5);
7   color: white;
8   text-shadow: black 2px 2px;
9   padding: 3px;
10  text-align: center;
11  margin-bottom: 2em;
12 }
13 .example #padding-box {
14   -moz-background-clip: padding;           , 0, 0, 0.5);
15   -o-background-clip: padding-box;
16   -webkit-background-clip: padding;        (blue, 50% 2px 2px);
17   background-clip: padding-box;
18 }
19 .example #border-box {
20   -moz-background-clip: border;           padding-box);
21   -o-background-clip: border-box;
22   -webkit-background-clip: border;        border-box);
23   background-clip: border-box;
24 }
```

[EXAMPLES](#)

Padding Box

Border Box

# COMPASS – EXAMPLES ([HTTP://COMPASS-STYLE.ORG/EXAMPLES](http://compass-style.org/examples))

- **Mixins**

## SCSS

```
@import "compass";  
  
.button {  
  @include border-radius(2px);  
  @include box-shadow(0 0 10px #666);  
  @include background(  
    linear-gradient(  
      to bottom, #ccc 0%,#eee 100%  
    )  
  );  
}
```

## CSS

```
.button {  
  -moz-border-radius: 2px;  
  -webkit-border-radius: 2px;  
  border-radius: 2px;  
  -moz-box-shadow: 0 0 10px #666666;  
  -webkit-box-shadow: 0 0 10px #666666;  
  box-shadow: 0 0 10px #666666;  
  background: url('data:image/svg+xml;base64,PD9...==');  
  background: -webkit-gradient(  
    linear, 50% 0%, 50% 100%,  
    color-stop(0%, #cccccc), color-stop(100%, #eee)  
  );  
  background: -moz-linear-gradient(top, #ccc 0%, #eee 100%);  
  background: -webkit-linear-gradient(top, #ccc 0%, #eee 100%);  
  background: linear-gradient(to bottom, #ccc 0%, #eee 100%);  
}
```

## EXAMPLES

```
1 .example table {  
2   border: 2px solid black;  
3 }  
4 .example table th, .example table td {  
5   border-right: 1px solid #7a98c6;  
6   border-bottom: 1px solid #7a98c6;  
7   border-left-width: 0;  
8   border-top-width: 0;  
9 }  
10 .example table th:last-child, .example table td:last-child {  
11   border-right-width: 0;  
12 }  
13 .example table th.last, .example table td.last {  
14   border-right-width: 0;  
15 }  
16 .example table tbody tr:last-child th, .example table tbody tr:last-child td,  
17   border-bottom-width: 0;  
18 }  
19 .example table tbody tr.last th, .example table tbody tr.last td, .example ta  
20   border-bottom-width: 0;  
21 }  
22 .example table thead th {  
23   border-bottom: 2px solid black;  
24 }  
25 .example table tfoot th, .example table tfoot td {  
26   border-top: 2px solid black;  
27 }  
28 .example table th:first-child {  
29   border-right: 2px solid black;  
30 }  
31 <th>Total</th>  
32 <td class='numeric'>11.0</td>  
33 <td class='numeric'>22.0</td>  
34 <td class='numeric'>33.0</td>  
35 <td class='numeric'>44.0</td>  
36 <td class='numeric'>110.0</td>
```

Title	One	Two	Three	Four	Row Total
Single	1.0	2.0	3.0	4.0	10.0
Tens	10.0	20.0	30.0	40.0	100.0
Total	11.0	22.0	33.0	44.0	110.0

```
1 @import "compass/ut  
2  
3 .example {  
4   table {  
5     @include inner-  
6     @include outer-  
7   }  
8 }
```

## ZADANIE – COMPASS MIXINS (OPCJONALNE)

Dla dokumentu “cv.html”:

1. Zdefiniuj styl dla sekcji Your Personal Information tło jako gradient oraz zaokrąglone obramowanie stostując odpowiednie **mixiny Compass**.
2. Zdefiniuj styl dla tabelek, aby miały one obramowanie wewnętrzne i zewnętrzne stosując odpowiednie **mixiny Compass**.

# COMPASS - EXAMPLES

- Spriting

Catalog

```
images/my-icons/new.png  
images/my-icons/edit.png  
images/my-icons/save.png  
images/my-icons/delete.png
```

SCSS

```
@import "compass/utilities/sprites";  
@import "my-icons/*.png";  
@include all-my-icons-sprites;
```

CSS

```
.my-icons-sprite,  
.my-icons-delete,  
.my-icons-edit,  
.my-icons-new,  
.my-icons-save { background: url('/images/my-icons-s34fe0604ab.png') no-repeat; }  
  
.my-icons-delete { background-position: 0 0; }  
.my-icons-edit { background-position: 0 -32px; }  
.my-icons-new { background-position: 0 -64px; }  
.my-icons-save { background-position: 0 -96px; }
```

## ZADANIE – COMPASS SPRITING (OPCJONALNE)

Dla dokumentu “cv.html”:

1. Zdefiniuj pasek narzędziowy z ikonami edycji, drukowania oraz zapisania poprzez mechanizm **Compass Spriting**.

# COMPASS - EXAMPLES

- **Sprites** – getting size of sprite

SCSS

```
@import "my-icons/*.png";
$box-padding: 5px;
$height: my-icons-sprite-height(some_icon);
$width: my-icons-sprite-width(some_icon);

.somediv {
    height:$height + $box-padding;
    width:$width + $box-padding;
}
```

LESS

# OVERVIEW

- <http://lesscss.org>
- Inspired by SASS
- Developed by Alexis Sellier in 2009
- Originally in Ruby on Rails, replaced by JavaScript
- Open source on GitHub
- Uses stylesheet.LESS extension
- First version of was written in Ruby but this was replaced by a JavaScript version.
- JavaScript compiles LESS code into regular CSS on the fly
- Variables, mix ins, operations and nested rules
- Nice debug feature

# MIXIN LIBRARIES

- LESSHAT
- Lots of Love for LESS
- LESS Elements

# INSTALLATION

- [NPM Node.JS](#) - npm install -g less
- WinLess - a simple windows installation and allows to easily compile your less files into CSS.
- [Homebrew](#) – for macOS.

# COMPILATION

- In Node.js projects
- Using a file watcher to compile on file change
- Compiling on page load with the rel="stylesheet" and type="text/less" or inline Use inline <style type="text/less">
- Most good IDE's have a built in LESS compiler, or a plugin available
- Command Line Compilation
- Browser Plugins
- Using online compilers

# COMPILATION

- **Precompile** - Compile LESS to CSS3, so only the single CSS file is loaded by your web application. This should be used for production.

```
$ lessc style.less > ../css/style.css
```

- **Post-compile** - You can use a less.js file, and include your less files in your HTML. This should NOT be used in production.

# COMPILATION

- **Post-compile - *in-browser*** - You can use a less.js file, and include your less files in your HTML. This should NOT be used in production.

```
<head>
    <meta charset="UTF-8">
    <style type="text/less">
        @h1color: red;
        body > h1 { color: @h1color; }
    </style>
    <script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/less.js/1.7.4/less.min.js">
    </script>
</head>

<!--LESS stylesheets first-->

<!--Post-compile LESS to CSS3-->
<script type="text/javascript" src="less.js"></script><script type="text/javascript">// <![CDATA[
less.watch();
// ]]></script>
```

# MINIFICATION AND COMPRESSION

- For production use, it is best to minimize and compress your CSS code.
- This makes your code lightweight, and therefore, your web site will be faster.

```
$ lessc -x styles.less > ../css/styles.css  
$ lessc -x --yui-compress styles.less > ../css/styles.css
```

# FRAMEWORKS BASE ON LESS

<a href="#">3L</a>	Mixins library
<a href="#">animate</a>	Library of CSS3 keyframe animations
<a href="#">Clearless</a>	Collection of mixins
<a href="#">Css3LessPlease</a>	Conversion of css3please.com to Less mixins
<a href="#">CssEffects</a>	Collection of CSS style effects
<a href="#">Cssowl</a>	Mixin library
<a href="#">cube.less</a>	Animated 3D cube using only CSS
<a href="#">est</a>	Mixin library
<a href="#">Hexagon</a>	Generate CSS hexagons with custom size and color
<a href="#">homeless</a>	Mixin library
<a href="#">LESS Elements</a>	Set of mixins
<a href="#">LESS Hat</a>	Mixins library
<a href="#">lessley</a>	A jasmine like testing suite written in pure less
<a href="#">Lessmore</a>	Mixins library. Cross-browser support for the CSS3 features and more
<a href="#">LESS-bidi</a>	Set of mixins for creating bi-directional styling
<a href="#">LESS-Mix</a>	Mixins library
<a href="#">media-query-to-type</a>	Media Queries to Media Types with Less
<a href="#">More-Colors.less</a>	Variables for easier color manipulation while you design in the browser
<a href="#">more-less</a>	Mixin lib supporting Less 1.7
<a href="#">More.less</a>	Mixins, animations, shapes and more
<a href="#">more-or-less</a>	for-loops and other functions + css3 mixins
<a href="#">normalize.less</a>	Modularized famous normalize.css using less
<a href="#">Oban</a>	Collection of mixins
<a href="#">Preboot</a>	Collection of variables and mixins. The precursor to Bootstrap

# ZADANIE – INSTALACJA ŚRODOWISKA

1. Zainstaluj Visual Studio Code (VSC) -  
<https://code.visualstudio.com>
2. W VSC zainstaluj rozszerzenia:
  - Easy Less -  
<https://marketplace.visualstudio.com/items?itemName=mrcrowl.easy-less>
  - Beautify css/sass/scss/less -  
<https://marketplace.visualstudio.com/items?itemName=michelemiluso.code-beautifier>
  - LESS IntelliSense -  
<https://marketplace.visualstudio.com/items?itemName=wangkechun.html-less-class-completion>  
<https://marketplace.visualstudio.com/items?itemName=mrmlnc.vscode-less>
  - Live HTML Previewer -  
<https://marketplace.visualstudio.com/items?itemName=hdg.live-html-previewer>

# BASIC STRUCTURE

- Superset of CSS
- Valid CSS should be valid LESS

```
.home {  
  background: black;  
}  
@media (max-width: 600px) {  
  .home {  
    background: white;  
  }  
}
```

LESS

```
.home {  
  background: black;  
}  
@media (max-width: 600px) {  
  .home {  
    background: white;  
  }  
}
```

CSS

# COMMENTS

- **CSS-style comments** - included in output by default
- **C-style single-line comments** - ignored in output by default

```
/* Ignored */  
// Ignored  
.menu {  
  color: red;  
}  
|
```

LESS

```
/* Ignored */  
.menu {  
  color: red;  
}
```

CSS

# VARIABLES

- Start with @ symbol
- Can hold different types of units, LESS is smart about units: px, em, hex/RGB colors
- **@var: value;** syntax

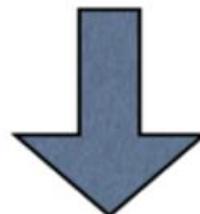
```
@ie8: true;  
@primarycolor: red;  
@secondarycolor: #448;  
@width: 2px;  
@height: 50%;  
@imageBase: '/images/';  
|
```

LESS

# VARIABLES

- Interpolation
  - For emphasis
  - Makes sense
  - Interpolation
  - Must use @

```
@baseUrl: "http://www.webucator.com/";  
@imageUri: "images/";  
background-image: url('@{baseUrl}@{imageUri}bg.png');
```



```
background-image: url('http://www.webucator.com/images/bg.png');
```

```
@var1: red;  
@var2: 'red';  
@var3: ~'red';  
.bg {  
  background: '@{var1}.png';  
  background: '@{var2}.png';  
  // duplicate removed:  
  background: '@{var3}.png';  
}  
// yikes!:  
.big@{var1} .big@{var2} .big@{var3} {  
  background: red;  
}
```

```
.bg i  
  background: '#ff0000.png';  
  background: 'red.png';  
}  
.big#ff0000 .big'red' .bigred {  
  background: red;  
}
```

LESS

CSS

# VARIABLES

- **Interpolation:**

- Property

```
@my-property: color;  
.myclass {  
    background-@{my-property}: #81F7D8;  
}
```

- Url

```
@images: "http://mkuciapski.com";  
.myclass {  
    background: url("@{images}/less/images/less_variables/intermediolan.jpg");  
    width: 800px;  
    height: 500px;  
}
```

- Import

```
@path : "http://www.tutorialspoint.com/less";  
@import "@{path}/external1.less";  
.myclass{  
    color : #A52A2A;  
}
```

# VARIABLES

- **Interrelation** - variable holding another variable.

```
.myclass{  
  @col: #ca428b;  
  @color: "col";  
  background-color: @@color;  
}
```

```
myclass {  
  background-color: #ca428b;  
}
```

# VARIABLES

- **Lazy Loading** - variables can be used even though they are not declared yet.

```
p {  
  font-size: @a;  
  color: #ca428b;  
}  
@a: @b;  
@b: 25px;
```

```
p {  
  font-size: 25px;  
  color: #ca428b;  
}
```

# VARIABLES

- **Scope:**
  - Variables scoped logically within braces
  - Variables from mixins available in scope where used

```
@primaryColor: blue;  
.home {  
  @primaryColor: red;  
  background: @primaryColor;  
}  
.home {  
  background: @primaryColor;  
}
```

LESS

```
.home {  
  background: #ff0000;  
}  
.home {  
  background: #0000ff;  
}
```

CSS

# VARIABLES

- Scope

```
@var: @a;  
@a: 15px;  
  
.myclass {  
  font-size: @var;  
  @a:20px;  
  color: green;  
}
```

```
.myclass {  
  font-size: 20px;  
  color: green;  
}
```

# VARIABLES

- **Escaping:**
  - If you need to output CSS that is not valid CSS syntax
  - Proprietary syntax not recognized by LESS
  - If not used, LESS compiler will throw an error
  - Simple prefix with **~ symbol** and put in quotes (string)

```
filter: ~"progid:DXImageTransform.Microsoft.gradient(startColorstr='#fdfdfd', endColorstr=
```

## ZADANIE – ZMIENNE PODSTAWY

Dla dokumentu “index.html”:

1. Zdefiniuj poprzez **zmienną** przechowującą:
  - formatowanie tekstu (czcionka, rozmiar, kolor, ...)
  - paddingi
  - marginesy

Kolorystyka ma być zdefiniowania z tonacji niebieskiej.

2. Zastosuj zmienne dla stylowania dokumentu.
3. Zmodyfikuj kolorystykę dokumentu, aby była w tonacji zielonej.

## ZADANIE – ZMIENNE PODSTAWY (OPCJONALNE)

Dla dokumentu “cv.html”:

1. Zdefiniuj **zmienne** przechowujące:
  - formatowanie tekstu (czcionka, rozmiar, kolor, ...)
  - paddingi
  - marginesy
  - obramowania
  - tła (np. sekcji)

Kolorystyka ma być zdefiniowania z tonacji niebieskiej.

2. Zastosuj zmienne dla stylowania dokumentu.
3. Zmodyfikuj kolorystykę dokumentu, aby była w tonacji zielonej.

# NESTING

- LESS allows nested selectors
- Compiled logically to combine selectors in resulting CSS
- **@media** handled appropriately

```
.home {  
  background: black;  
  .menu {  
    color: red;  
  }  
  @media (max-width: 200px) {  
    .menu {  
      color: yellow;  
    }  
  }  
}
```

LESS

```
.home {  
  background: black;  
}  
.home .menu {  
  color: red;  
}  
@media (max-width: 200px) {  
  .home .menu {  
    color: yellow;  
  }  
}
```

CSS

## ZADANIE – NESTING PODSTAWY

Dla dokumentu “index.html”:

1. Zmodyfikuj przypisanie stylu **stosując zagnieżdżenia** zgodnie z dobrymi praktykami.

# ZADANIE – NESTING PODSTAWY (OPCJONALNE)

Dla dokumentu “cv.html”:

1. Zmodyfikuj przypisanie stylu **stosując zagnieżdżenia** zgodnie z dobrymi praktykami.

# NESTING

- **Parent selector** - ampersand used to concatenate to parent selector. Will concatenate just about anything.

```
.menu {  
    color: red;  
    &.active {  
        font-weight: bold;  
    }  
    &:hover {  
        color: yellow;  
    }  
    ul { // Not necessary!!  
        li {  
            color: blue;  
        }  
    }  
}
```

LESS

```
.menu {  
    color: red;  
}  
.menu.active {  
    font-weight: bold;  
}  
.menu:hover {  
    color: yellow;  
}  
.menu ul li {  
    color: blue;  
}
```

CSS

## ZADANIE – PARENT SELECTOR

Dla dokumentu “index.html”:

1. Za pomocą **parent selector** zdefinuj styl dla listy, aby element, na który nastąpi najechanie był podświetlony.

## ZADANIE – PARENT SELECTOR (OPCJONALNE)

Dla dokumentu “cv.html” za pomocą **parent selector**:

1. Zdefinuj styl, aby najechanie na poszczególne sekcje cv delikatnie je podświetlało.
2. Zdefiniuj proste menu u góry dokumentu pozwalające na przechodzenie do poszczególnych sekcji, ze stylami najechania na link oraz bycia aktywnym.

# MIXINS

- Mixins will copy style information into context

```
.button {  
  background: gray;  
  border-radius: 5px;  
}  
.blue-button {  
  .button; // mixin!!  
  color: blue;  
}  
.red-button {  
  .button;  
  color: red;  
}
```

LESS

```
.button {  
  background: gray;  
  border-radius: 5px;  
}  
.blue-button {  
  background: gray;  
  border-radius: 5px;  
  color: blue;  
}  
.red-button {  
  background: gray;  
  border-radius: 5px;  
  color: red;  
}
```

CSS

# MIXINS

- **Not Outputting the Mixin** - create a mixin and it can be made to disappear in the output by simply placing the parentheses after it.

```
.a(){  
  padding-left: 100px;  
}  
.myclass{  
  background : #64d9c0;  
  .a;  
}
```

```
.myclass {  
  background: #64d9c0;  
  padding-left: 100px;  
}
```

# MIXINS

- **Selectors in Mixins** - The mixins can contain not just properties but they can also contain selectors too.

```
.mixin() {  
  &:hover {  
    background: #FFC0CB;  
  }  
}  
a {  
  .mixin();  
}
```

```
a:hover {  
  background: #FFC0CB;  
}
```

## ZADANIE – MIXIN

Dla dokumentu “index.html” napisz mixin:

1. O nazwie **zaokrSekcja**, **definujący**: padding, obramowanie oraz radius. Napisz styl dla nagłówka (header), dla którego przygotowania zastosuj mixin.
2. Który definiuje border-radius **dla różnych przeglądarek**. Zastosuj go w mixinie o nazwie zaokrSekcja.
3. O nazwie **listaBezFormatowania**, **definujący** listę bez wypunktowania, paddingów, marginesów, itp.. Napisz styl dla list (ul), dla którego przygotowania zastosuj mixin.

## ZADANIE – MIXIN (OPCJONALNE)

Dla dokumentu “cv.html”:

1. Napisz **mixin(y)** przygotowujący(e) **poziome menu**, które zawsze jest u góry strony. Napisz klasę(y) korzystające z mixina dodatkowo stylujące menu (kolor tła, najechanie, czcionki, ...).
2. Mixiny z zadania dotyczącego dokumentu “index.html” wstaw do **osobnego pliku**, który **zimportuj**. Następnie wykorzystaj mixin o nazwie **zaokrSekcja**.

# MIXINS

- **Parameters and variables:**
  - Mixins support parameters
  - Default values
  - Variables can be scoped within mixin definition

```
.button(@color:black,@radius:5px) {  
  background: gray;  
  border-radius: @radius;  
  color: @color;  
}  
.blue-button {  
  .button(blue, 10px);  
}  
.red-button {  
  .button(red);  
}
```

LESS

```
.blue-button {  
  background: gray;  
  border-radius: 10px;  
  color: #0000ff;  
}  
.red-button {  
  background: gray;  
  border-radius: 5px;  
  color: #ff0000;  
}
```

CSS

# MIXINS

- Named parameters

```
.mixin(@color: black; @fontSize: 10px) {  
  color: @color;  
  font-size: @fontSize;  
}  
.class1 {  
  .mixin(@fontSize: 20px; @color: #F5A9D0);  
}  
.class2 {  
  .mixin(#F79F81; @fontSize: 20px);  
}
```

```
.class1 {  
  color: #F5A9D0;  
  font-size: 20px;  
}  
.class2 {  
  color: #F79F81;  
  font-size: 20px;  
}
```

## ZADANIE – MIXIN PARAMETERS

Dla dokumentu “index.html”:

1. Zmodyfikuj mixin o nazwie zaokrSekcja, aby **wartości pobierane były jako parametry** - część z nich niech posiada wartości domyślne. Zmodyfikuj styl dla nagłówka (header), dla którego zastosowano mixin, aby wywołanie mixina zawierało parametry.

## ZADANIE – MIXIN PARAMETERS (OPCJONALNE)

Dla dokumentu “cv.html”:

1. Zmodyfikuj mixin przygotowujący poziome menu, aby **wartości pobierane były jako parametry**. Zmodyfikuj klasę(y) korzystające z mixina, aby stosowały one parametry.

# MIXINS

- **@arguments Variable:**

- When mixin was called, the @arguments includes all the passed arguments.
- The @arguments variable is useful when you don't want to work with individual parameters.

```
.box-shadow(@x: 0; @y: 0; @height: 3px; @width: 3px) {  
    -webkit-box-shadow: @arguments;  
    -moz-box-shadow: @arguments;  
    box-shadow: @arguments;  
}  
.myclass {  
    .box-shadow(2px; 2px);  
}
```

```
.myclass {  
    -webkit-box-shadow: 2px 2px 3px 3px;  
    -moz-box-shadow: 2px 2px 3px 3px;  
    box-shadow: 2px 2px 3px 3px;  
}
```

# MIXINS

- **Advanced Arguments and @rest Variable:**
  - Mixin takes variable number of arguments by using ....
  - You can assign arguments to the variable by placing ... after the variable name.

```
.mixin(...) {          // it matches arguments from 0-n
.mixin() {            // it matches exactly 0 arguments
.mixin(@x: 1) {      // it matches arguments from 0-1
.mixin(@x: 1; ...) { // it matches arguments from 0-n
.mixin(@x; ...) {
```

```
.mixin(@x; @rest...) {
    // after the variable @a, the @rest is bound to arguments
    // @arguments is bound to all arguments
}
```

# MIXINS

- **Return values** - Mixins are similar to functions and the variables that are defined in a mixin will behave as its return values.

```
.padding(@x, @y) {  
  @padding: ((@x + @y) / 2);  
}  
  
.myclass{  
  .padding(80px, 120px); // call to the mixin  
  padding-left: @padding; // returns the value  
}
```

```
.myclass {  
  padding-left: 100px;  
}
```

# MIXINS

- **Pattern matching:**

- Mixins matched based on criteria
- @\_ for any value, will always be included
- Match on parameter count

```
.mixin (dark; @color) {  
  color: darken(@color, 10%);  
}  
.mixin (light; @color) {  
  color: lighten(@color, 10%);  
}  
.mixin (@_; @color) {  
  display: block;  
}  
.class {  
  .mixin(light; #888);  
}
```

LESS

```
.class {  
  color: #a2a2a2;  
  display: block;  
}
```

CSS

# MIXINS

- **!important keyword** - When it is placed after mixin call, it marks all inherited properties as *!important*.

```
.mixin(){  
  color: #900;  
  background: #F7BE81;  
}  
.para1{  
  .mixin();  
}  
.para2{  
  .mixin() !important;  
}
```

```
.para1 {  
  color: #900;  
  background: #F7BE81;  
}  
.para2 {  
  color: #900 !important;  
  background: #F7BE81 !important;  
}
```

# MIXINS

- **Mixin inside mixin** - whenever a mixin is defined inside another mixin, it can be used as return value too.

```
.outerMixin(@value) {  
  margin: @value /2;  
  .nestedMixin() {  
    font-size: @value;  
  }  
}  
  
.myclass {  
  .outerMixin(30px);  
  .nestedMixin();  
}
```

```
.myclass {  
  margin: 15px;  
  font-size: 30px;  
}
```

# MIXINS

- **Passing Rulesets:**

- Detached ruleset contains an ordinary ruleset such as properties, nested rulesets, variables declaration, mixins etc.
- It is stored in a variable and included into other structure where all its properties of the ruleset will get copied to that structure.

```
@detached-ruleset: ~
{
  @variable: 2px;
  .mixin() {
    font-family: "Comic Sans MS";
    background-color: #AA86EE;
    margin: @variable @variable * 2
  }
}

.cont {
  @detached-ruleset();
  .mixin();
}
```

# NAMESPACES AND ACCESSORS

- Using namespaces you can avoid conflict in name and encapsulate group of mixins from outside.

```
.class1 {  
  .class2 {  
    .val(@param) {  
      font-size: @param;  
      color:green;  
    }  
  }  
}  
  
.myclass {  
  .class1 > .class2 > .val(20px);  
}
```

```
.myclass {  
  font-size: 20px;  
  color: green;  
}
```

# ZADANIE – MIXIN NESTING

Dla dokumentu “index.html”:

1. Napisz mixin o nazwie **stylNaglowek** z parametrami: naglowek (h1-h6), klasa (np. “wazne”), kolor (np. “green”). Zgodnie z parametrami mixin ma przygotować klasę o formie **nagłówek.klasa** (np. h2.wazne).
2. Poprzez mixin przygotuj kilka klas oraz zastosuj je dla nagłówków.

## ZADANIE – MIXIN NESTING (OPCJONALNE)

Dla dokumentu “cv.html”:

1. Napisz mixin o nazwie **stylDiv** z parametrami: klasa (np. “zwrocUwage”), kolor (np. “green”), czcionka, rozmiar, padding, margines, float i ew. inne. Zgodnie z parametrami mixin ma **przygotować klasę** o formie ***div.klasa*** (np. div.wazne).
2. Poprzez mixin przygotuj kilka klas oraz zastosuj je dla sekcji.

# GUARDS

- Basic inclusion/exclusion rules for mixins
- Uses the **when** keyword after mixin declaration
- LESS contains five guard comparison operators: <, >, <=, >= and =.
- Also: *when (isnumber(@a)) and (@a > 0) { ... }*

```
.mixin (@a) when (lightness(@a) >= 50%) {  
  background-color: black;  
}  
.mixin (@a) when (lightness(@a) < 50%) {  
  background-color: white;  
}  
.mixin (@a) {  
  color: @a;  
}  
.class1 { .mixin(#ddd) }  
.class2 { .mixin(#555) }  
|
```

LESS

```
.class1 {  
  background-color: black;  
  color: #dddddd;  
}  
.class2 {  
  background-color: white;  
  color: #555555;  
}
```

CSS

# GUARDS

- **Logical Operators** - combine the guard conditions using *and* keyword and negate the conditions using *not* keyword.

```
.mixin (@a) when (@a > 50%) and (@a > 5px){  
font-size: 14px;  
}  
.mixin (@a) when not (@a < 50%) and not (@a < 5px){  
font-size: 20px;  
}  
.mixin (@a) {  
  color: @a;  
}  
  
.class1 { .mixin(#FF0000) }  
.class2 { .mixin(#555) }
```

```
.class1 {  
  font-size: 20px;  
  color: #FF0000;  
}  
.class2 {  
  font-size: 20px;  
  color: #555;  
}
```

# GUARDS

- **Type Checking Functions** - available functions: iscolor, isnumber, isstring, iskeyword, isurl, ispixel, ispercentage, isem, isunit.

```
.mixin (@a; @b: red) when (iscolor(@b)){
  color:blue;
}
.mixin (@a) {
  font-size: @a;
}
.myclass { .mixin(20px) }
```

```
.myclass {
  color: blue;
  font-size: 20px;
}
```

# GUARDS

- Conditional mixins.

```
.mixin (@a) when (@a > 22px){  
color:blue;  
}  
  
.mixin (@a) when (@a <= 20px){  
color:red;  
}  
  
.mixin (@a) {  
  font-size: @a;  
}  
  
.myclass { .mixin(20px) }
```

```
.myclass {  
  color: red;  
  font-size: 20px;  
}
```

## ZADANIE – GUARDS

Dla dokumentu “index.html”:

1. Napisz **mixin**, który jako parametry otrzymuje kolor tła oraz rozmiar czcionki i “zwraca” **odpowiedni kolor dla czcionki** oraz wprowadza dla niej ewentualne pogrubienie.
2. Poprzez mixin przygotuj kilka klas oraz zastosuj je dla nagłówków.

## ZADANIE – GUARDS (OPCJONALNE)

Dla dokumentu “cv.html”:

1. Napisz **mixin** który jako parametry otrzymuje kolor tła oraz rozmiar czcionki i “zwraca” **odpowiedni kolor dla czcionki** oraz wprowadza dla niej ewentualne pogrubienie.
2. Poprzez mixin przygotuj kilka klas oraz zastosuj je dla sekcji.

# LOOPS

- Loops statement allows us to execute a statement or group of statements multiple times.
- Various iterative/loop structures can be created when recursive mixin combined with **Guard Expressions** and **Pattern Matching**.

```
.cont(@count) when (@count > 0) {  
  .cont((@count - 1));  
  width: (25px * @count);  
}  
div {  
  .cont(7);  
}
```

```
div {  
  width: 25px;  
  width: 50px;  
  width: 75px;  
  width: 100px;  
  width: 125px;  
  width: 150px;  
  width: 175px;  
}
```

```
// import normalize for CSS resets
@import "normalize"; // same as @import "normalize.less";

// import mixins for all of my "global" variables and mixins
@import "mixins";

// base for mobile devices
@import "base";

//tablets
@media only screen and (min-width: 768px) {
@import "768";
}

//desktops
@media only screen and (min-width: 1030px) {
@import "1030";
}
```

## ZADANIE – PARTIALS, IMPORT

Dla dokumentu “index.html”:

1. Wszystkie zmienne wstaw do pliku **zmienne.less**.
2. Zimportuj plik zmienne.less.

## ZADANIE – IMPORT (OPCJONALNE)

Dla dokumentu “cv.html”:

1. Wszystkie zmienne wstaw do pliku **zmienne.less**.
2. **Zimportuj** plik zmienne.less.

# EXTEND

- Joins to existing style rules by adding selectors to parent definition
- Can't have parameters like mixins

```
.button {  
  background: gray;  
  border-radius: 5px;  
}  
.blue-button:extend(.button) {  
  color: blue;  
}  
.red-button:extend(.button) {  
  color: red;  
}
```

LESS

```
.button,  
.blue-button,  
.red-button {  
  background: gray;  
  border-radius: 5px;  
}  
.blue-button {  
  color: blue;  
}  
.red-button {  
  color: red;  
}
```

CSS

# EXTEND

- **Multiple-extend**

```
.style:extend(.container, .img)
{
  background: #BF70A5;
}
.container {
  font-style: italic;
}
.img{
  font-size: 30px;
}
```

```
.style {
  background: #BF70A5;
}
.container,
.style {
  font-style: italic;
}
.img,
.style {
  font-size: 30px;
}
```

# EXTEND

- **Extend Inside Ruleset** - parent selector:
  - Extend matches all outputs of specific params
  - Extend multiple comma separated
  - This is very useful for pseudo classes, such as :focus and :hover

```
div pre {  
  color: red;  
}  
  
pre:hover,  
.c1 {  
  &:extend(div pre);  
}  
  
.bucket {  
  tr { // nested ruleset with target sel  
    color: blue;  
  }  
}  
.c2:extend(.bucket tr, div pre) {}
```

```
div pre,  
pre:hover,  
.c1,  
.c2 {  
  color: red;  
}  
.bucket tr,  
.c2 {  
  color: blue;  
}
```

LESS

CSS

# EXTEND

- Extending Nested Selectors:

```
.style {  
  h3{  
    color: #BF70A5;  
    font-size: 30px;  
  }  
}  
.img:extend(.style h3){}
```

```
.style h3,  
.img {  
  color: #BF70A5;  
  font-size: 30px;  
}
```

# EXTEND

- **Exact Matching:**

```
.style h3,  
h3.style{  
    color: #BF70A5;  
    font-style: italic;  
}  
.img:extend(.style h3){}
```

```
.style h3,  
h3.style,  
.img {  
    color: #BF70A5;  
    font-style: italic;  
}
```

# EXTEND

- **Expressions:**

- **nth**

```
:nth-child(n+2) {  
    color: #BF70A5;  
    font-style: italic;  
}  
  
.mk:extend(:nth-child(1n+2)) {  
    background-color: #ffd800;  
}
```

- **filter**

```
[title=tutorialspoint] {  
    font-style: italic;  
}  
[title='tutorialspoint'] {  
    font-style: italic;  
}  
[title="tutorialspoint"] {  
    font-style: italic;  
}  
.style:extend([title=tutorialspoint]) {}  
.container:extend([title='tutorialspoint']) {}  
.img:extend([title="tutorialspoint"]) {}
```

```
:nth-child(n+2) {  
    color: #BF70A5;  
    font-style: italic;  
}  
.mk {  
    background-color: #ffd800;  
}
```

```
[title=tutorialspoint],  
.style,  
.container,  
.img {  
    font-style: italic;  
}  
[title='tutorialspoint'],  
.style,  
.container,  
.img {  
    font-style: italic;  
}  
[title="tutorialspoint"],  
.style,  
.container,  
.img {  
    font-style: italic;  
}
```

# EXTEND

- **All:**
  - Identified at last in extend argument
  - LESS matches that selector as part of another selector.
  - The selector part which is matched will be replaced with extend, making a new selector.

```
.style.nav,  
.nav h3 {  
  color: orange;  
}  
.nav {  
  &:hover {  
    color: green;  
  }  
}  
.container:extend(.nav all) {}
```

```
.style.nav,  
.nav h3,  
.style.container,  
.container h3 {  
  color: orange;  
}  
.nav:hover,  
.container:hover {  
  color: green;  
}
```

# EXTEND

- **Selector Interpolation:**

- The @{variable} symbol is used as part of variable name, id and class name.
- Extend does not have ability to match selector with variables.
- Extend can be connected to interpolated selector.

```
.style {  
  color: #BF70A5;  
  font-style: italic;  
}  
@{variable}:extend(.style) {}  
@variable: .selector;
```

```
.style,  
.selector {  
  color: #BF70A5;  
  font-style: italic;  
}
```

# EXTEND

- **Scoping:**

- The extend matches the selector only that are present inside the same media declaration.
- Extend present in media declaration does not matches the selector that are present inside the nested declaration.

```
@media screen {  
  .style {  
    width:500px;  
  }  
  @media (min-width: 1023px) {  
    .style {  
      width:500px;  
    }  
  }  
}  
.cont:extend(.style) {}
```

```
@media screen {  
  .style,  
  .cont {  
    width: 500px;  
  }  
}  
@media screen and (min-width: 1023px) {  
  .style,  
  .cont {  
    width: 500px;  
  }  
}
```

# EXTEND

- **Duplication Detection** – LESS does not detect the duplication of selectors.

```
.cont-main,  
.style {  
  font-family: "Comic Sans MS";  
  font-size: 30px;  
}  
.cont-main{  
  font-size: 30px;  
}  
.cont:extend(.cont-main, .style) {}
```

```
.cont-main,  
.style,  
.cont,  
.cont {  
  font-family: "Comic Sans MS";  
  font-size: 30px;  
}  
.cont-main,  
.cont {  
  font-size: 30px;  
}
```

## ZADANIE – EXTEND/INHERITANCE

Dla dokumentu “index.html”:

1. Przygotuj **klasę** o nazwie sekcja z wartościami obramowania, czcionki, kolorów, padding, margines, itp.
2. Zdefiniuj style dla headera oraz footera **rozszerzając klasę** sekcja.

## ZADANIE – EXTEND/INHERITANCE (OPCJONALNE)

Dla dokumentu “cv.html”:

1. Przygotuj **klasę** o nazwie sekcja z wartościami obramowania, czcionki, kolorów, padding, margines, itp.
2. Zdefiniuj style dla sekcji na podstawie (rozszerzenie) klasy sekcja, uwzględniając również najechanie na nie (hover).

# OPERATIONS

- Units remain intact, can mix some types
- Can use hex color values in operations

```
@base: 5%;  
@filler: (@base * 2);  
@other: (@base + @filler);  
@base-color: #111;  
  
.mc {  
color: (#888 / 4);  
background-color: (@base-color + #100);  
height: (100% / 2 + @filler);  
}
```

LESS

```
.mc {  
color: #222222;  
background-color: #221111;  
height: 60%;  
}
```

CSS

# ZADANIE – OPERATORS

Dla dokumentu “index.html”:

1. Zdefiniuj, aby <div> z tekstem “Sample paragraph of text” oraz <div> z listą były obok siebie i miały **procentowy rozmiar**, gdzie:
  - Pierwszy z nich ma mieć **800px z 1024px**.
  - Drugi z nich ma mieć **pozostałą część**.

## ZADANIE – OPERATORS (OPCJONALNE)

Dla dokumentu “cv.html”:

1. Zdefiniuj, aby dane w sekcjach Education oraz Courses & Diploma miały formę

1995-1998 Bachelor degree. Accounting. University of Northcap.

Gdzie daty zajmują **200px** z **1024px**, a opis **pozostałą część**. Wartości szerokości mają być **wyrażone procentowo**.

# FUNCTIONS

- Lots of built-in functions, like:
  - Color: lighten, darken, desaturate, ...
  - Math: ceil, floor, round, ...

```
.mc {  
background-color: lighten(#111, 10%);  
height: round(100% / 2.1);  
}
```

LESS

```
.mc {  
background-color: #2b2b2b;  
height: 48%;  
}
```

CSS

# FUNCTIONS

- **Colour:**
  - // return a color which is 10% \*lighter\* than @color  
**lighten(@color, 10%);**
  - // return a color which is 10% \*darker\* than @color  
**darken(@color, 10%);**
  - // return a color 10% \*more\* saturated than @color  
**saturate(@color, 10%);**
  - // return a color 10% \*less\* saturated than @color  
**desaturate(@color, 10%);**
  - // return a color 10% \*less\* transparent than @color  
**fadein(@color, 10%);**
  - // return a color 10% \*more\* transparent than @color  
**fadeout(@color, 10%);**
  - // return @color with 50% transparency  
**fade(@color, 50%);**

# FUNCTIONS

- **Colour:**
  - // return a color with a 10 degree larger in hue than @color  
**spin(@color, 10);**
  - // return a color with a 10 degree smaller hue than @color  
**spin(@color, -10);**
  - // return a mix of @color1 and @color2  
**mix(@color1, @color2);**

## Examples

```
@base: #f04615;  
@width: 0.5;  
  
.class {  
  width: percentage(@width); // returns `50%`  
  color: saturate(@base, 5%);  
  background-color: spin(lighten(@base, 25%), 8);  
}
```

# FUNCTIONS

- **Math:**
  - **round(1.67);** // returns '2'
  - **ceil(2.4);** // returns '3'
  - **floor(2.6);** // returns '2'
  - **percentage(0.5);** // returns '50%

## ZADANIE – FUNKCJE

Dla dokumentu “index.html”:

1. Zdeklaruj **zmienną** dla przechowywania podstawowej wartości paddingu jako 3px.
2. Napisz **funkcję**, która zwraca wartość paddingu jako wielokrotność wartości podstawowej parametru.
3. Wykorzystaj funkcję dla przypisania paddingu poszczególnych sekcji. Zmień wartość zmiennej przechowywania podstawowej wartości paddingu na 10px.

## ZADANIE – FUNKCJE (OPCJONALNE)

Dla dokumentu “cv.html”:

1. Zdeklaruj **zmienną** dla przechowywania podstawowej wartości szerokości sekcji jako 100%.
2. Napisz **funkcję**, która zwraca wartość szerokości, jako wartość podstawową pomniejszoną o wartość parametru.
3. Wykorzystaj funkcję dla przypisania szerokości poszczególnych sekcji. Zmień wartość zmiennej przechowywania podstawowej szerokości na 90%.

# RETHINK YOUR HTML

- Using mixins and :extend functionality, move the choice of styling to the rules, not the HTML

The screenshot illustrates the LESS preprocessor ecosystem. At the top, the LESS logo is displayed. Below it, the LESS interface shows two panels: one for LESS code and one for CSS output. The LESS panel contains:

```
// Define current styles
.blue-button() {
  display: inline-block;
  padding: 6px 14px;
  border-radius: 5px;
  border: 1px solid #ccc;
  background: #eee;
  color: blue;
  &:hover { background: #fff; }
}
// Apply them where appropriate
.menu > a { .blue-button; }
```

The CSS panel shows the generated CSS:

```
.menu > a {
  display: inline-block;
  padding: 6px 14px;
  border-radius: 5px;
  border: 1px solid #ccc;
  background: #eee;
  color: blue;
}
.menu > a:hover {
  background: #fff;
```

Below the LESS and CSS panels, the HTML panel displays the generated HTML:

```
<div class="menu">
  <a href="#">Item 1</a>
  <a href="#">Item 2</a>
  <a href="#">Item 3</a>
  <a href="#">Item 4</a>
</div>
```

On the right side of the interface, four buttons labeled "Item 1", "Item 2", "Item 3", and "Item 4" are shown, each styled with the LESS-defined blue-button class.