Méthode Intégration

OOCSS

Oriented Object CSS

Identifier les objets qui vont se répéter sur une page ou un site et d'y définir des classes réutilisables

```
<button class="btn btn-primary">bouton d'exemple</button>
```

```
.btn{
    width: 100%;
    padding: 6px 12px;
    border: 0;
    display: inline-block;
    position: relative;
    z-index: 1;
    overflow: hidden;
    &:before{...}
}

.btn-primary{
    background: $primary-color;
    color: white;
    &:before {...}
    &:hover{...}
}
```

BEM

```html
<header class="main-header">
    <div class="container">
        <div class="main-header__logo">
            <img src="img/logo.png" alt="sfi" class="main-header__logo__img"/>
        </div>
        <div class="main-header__title">
            <h1 class="main-header__title__text">Les bonnes pratiques Scss</h1>
        </div>
    </div>
</header>
```

```scss
.main-header{
    text-align: center;
    &__logo{
        &__img{
            width: 15%;
            display: block;
            margin: 20px auto;
        }
    }
    &__title{
        &__text{
            text-align: center;
            text-transform: uppercase;
            font-weight: 700;
            color: white;
        }
    }
}
```

Blocs : les blocs sont des entités indépendantes et autonomes constituant une page HTML

Éléments : les éléments constituent les différentes parties du bloc

Modificateurs : ces derniers vont permettre de constituer des variantes de blocs ou d'éléments

EXEMPLE

```scss
.main-header{
  background: lightblue;
  .logo{
    width: 50%;
    margin-left: 25%;
    .img{
      width: 100%;
    }
  }
}
```

VS

```scss
.main-header{
  background: lightblue;
  &__logo{
    width: 50%;
    margin-left: 25%;
    &__img{
      width: 100%;
    }
  }
}
```

EXEMPLE

```html
<section class="avis">
    <ul class="avis__list">
        <li class="avis__list__item">First</li>
        <li class="avis__list__item">Second</li>
    </ul>
</section>
```

```scss
//Le bloc avis
.avis{
    margin-top: 10px;
    //L'élement <ul>
    &__list{
        list-style: none;
        //L'élement <li>
        &__item{
            font-size: 1.2em;
        }
    }
}
```

VARIABLES

```scss
$primary-color : #F55837;
$img-path: "../img/";
$font-montserrat: 'Montserrat', sans-serif;

.main-title{
    background-image: url(#{$img-path}bg-title.png);
    color:  $primary-color;
    font-family: $font-montserrat;
}
```

Couleurs

Font

Chemin d’image

RESPONSIVE

Variables

Map

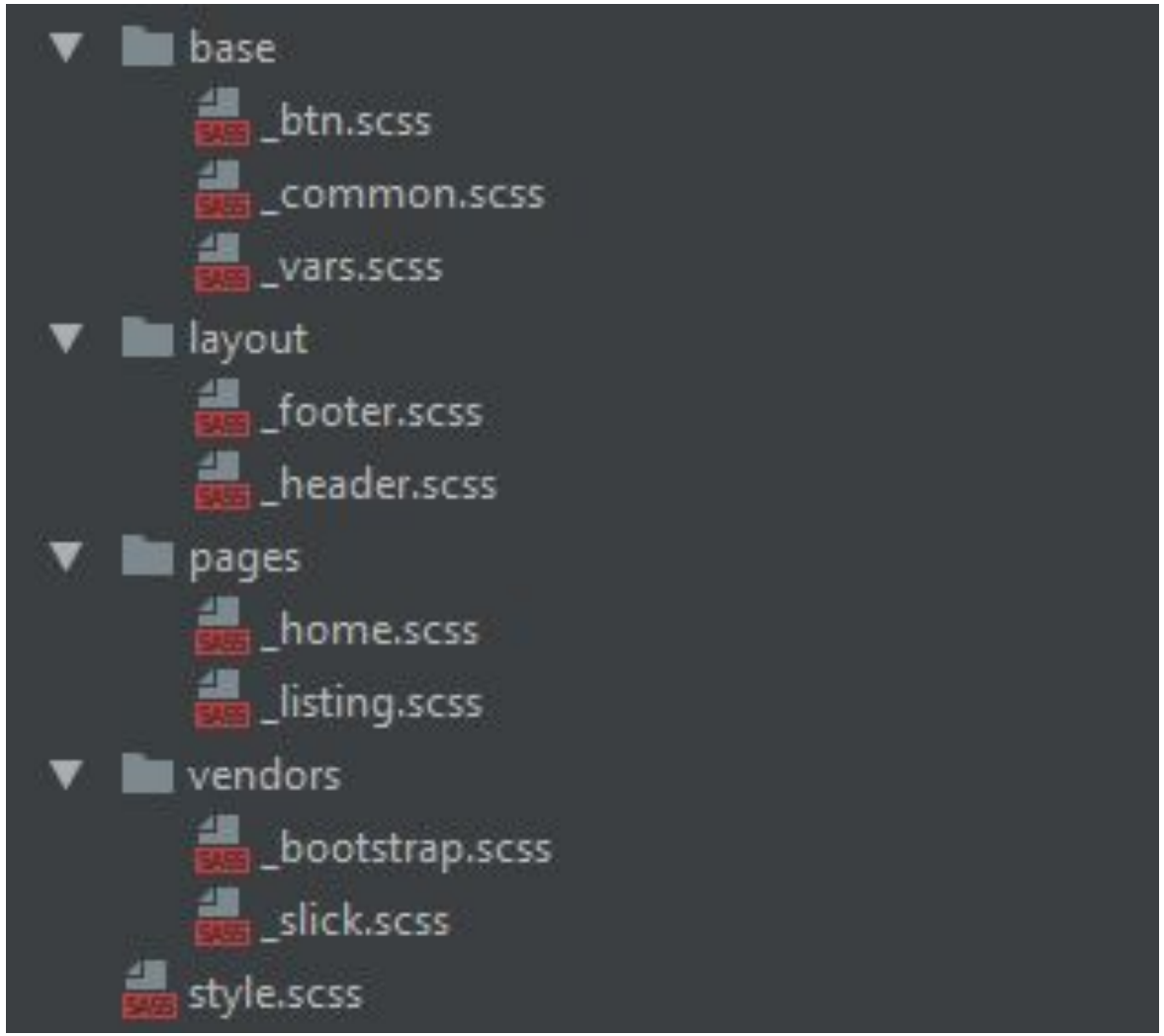
Mixin

```scss
$max-lg: 1199px;
$max-md: 991px;
$max-sm: 768px;
$max-xs: 575px;
```

```scss
$primary-color : #F55837;
$breakpoints: (
        'medium': (min-width: 768px),
        'large': (min-width: 992px),
        'huge': (min-width: 1200px)
) !global;

@mixin respond-to($breakpoint) {
    // Si la clé existe dans le map
    @if map-has-key($breakpoints, $breakpoint) {
        // On écrit les media-queries avec le paramêtre
        @media #{inspect(map-get($breakpoints, $breakpoint))} {
            @content;
        }
    }
    // Si la clé n'existe pas
    @else {
        @error 'No value found for `#{$breakpoint}`. '
        + 'Please make sure it is defined in `$breakpoints` map.';
    }
}
.main-header__logo{
    @include respond-to(medium){
        color:$primary-color;
    }
}
```

STRUCTURE


base
_btn.scss
_common.scss
_vars.scss
layout
_footer.scss
_header.scss
pages
_home.scss
_listing.scss
vendors
_bootstrap.scss
_slick.scss
style.scss

OUTILS

Best Practices	Example

Write multiple selectors on separate lines.	.btn,
	.btn-link {
	}

Include one space between the selector and the opening brace.	.selector { }
Use shorthand for hex values when possible.	`#fff` vs `#ffffff`
Write hex values in lowercase.	`#3d3d3d` vs `#3D3D3D`
Enclose URLs in single quotes. Generally, single quotes are preferred over double quotes, since they're easier to type.	`url ('/image.png')` vs `url ("/image.png")`
Don't use units for zero (0) values, except for angles (deg) and time (s or ms).	`margin-right: 0;` vs `margin-right: 0px;`

OUTILS