

# Formation Mobile Hybride Day 3

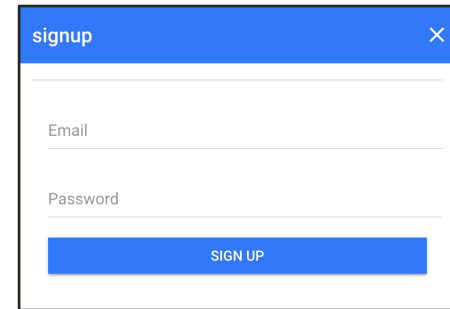
**Cordova, Ionic**

# Agenda

1. Correction travaux pratiques
2. Provider http « places »
3. Google maps component
4. NavParams navigation
5. Sass customisation

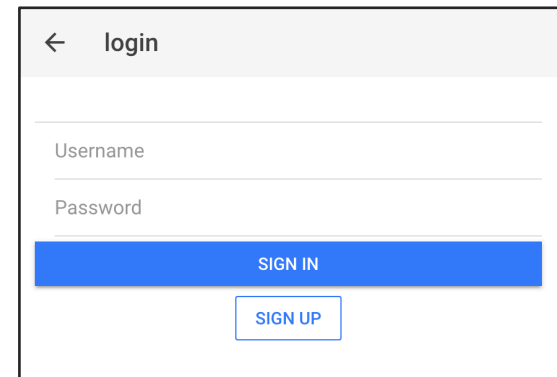
# Correction

## Sign up



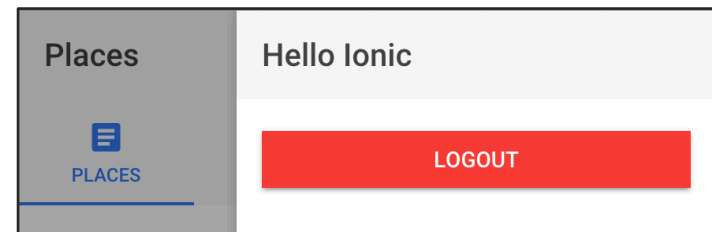
A mobile app mockup for a sign-up screen. It features a blue header bar with the text 'signup' and a close icon. Below the header are two input fields labeled 'Email' and 'Password'. At the bottom is a blue button labeled 'SIGN UP'.

## Sign In



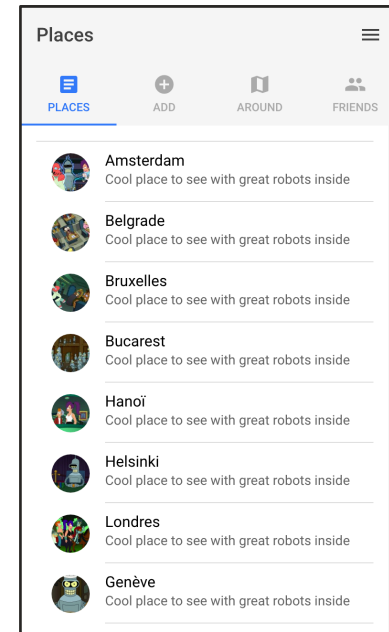
A mobile app mockup for a sign-in screen. It features a light gray header bar with a back arrow and the text 'login'. Below the header are two input fields labeled 'Username' and 'Password'. At the bottom are two buttons: a blue 'SIGN IN' button and a white 'SIGN UP' button with a blue border.

## Affichage email + Logout



A mobile app mockup for a user profile screen. It features a gray sidebar on the left with the text 'Places' and a blue icon. The main content area has a light gray header bar with the text 'Hello Ionic'. Below the header is a red button labeled 'LOGOUT'.

# Provider http « places »

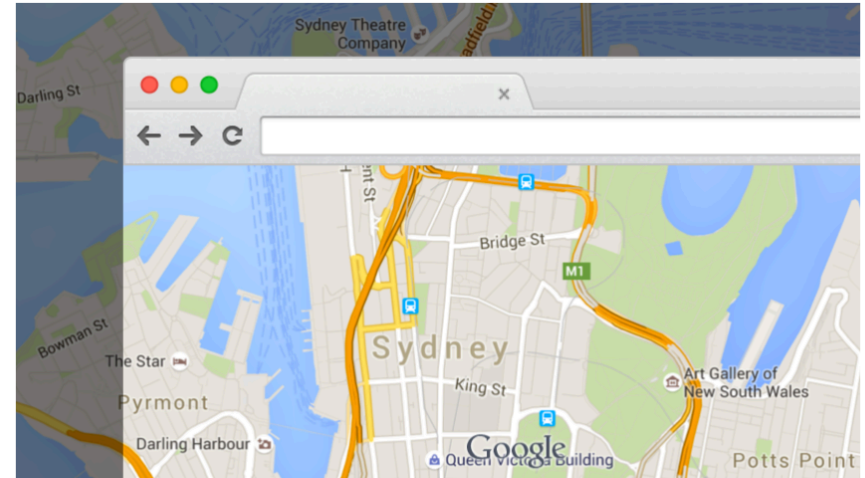


```
getPlaces(){  
  return this.API_PATH + "/user/places"  
}
```

```
return new Promise(resolve => {  
  this.authHttp.get(this.endpoints.getPlaces())  
    .map(res => res.json())  
    .subscribe(data => {  
      this.data = data;  
      resolve(this.data);  
    });  
});
```

# Google maps component

# Google Maps Javascript API



Cartes

---

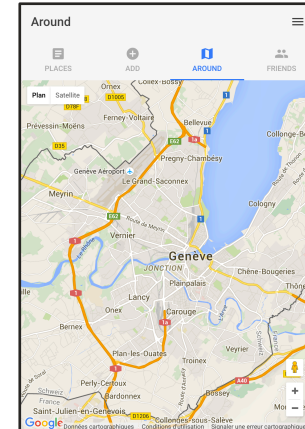
## Google Maps Javascript API:

```
<script  
src="https://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY"  
></script>
```

Key: AlzaSyB16sGmlekuGlvYOfNoW9T44377IU2d2Es

## \$ ionic g component map

```
this.map = new google.maps.Map(document.getElementById("map_canvas"), {
  center: new google.maps.LatLng(lat, long),
  zoom: zoom,
  mapTypeId: google.maps.MapTypeId.ROADMAP
});
```



### Around page:

```
export class AroundPage {

  @ViewChild(Maps)
  private map: Maps;

  constructor() {
  }

  ionViewDidEnter(){
    this.map.init(46.2043907, 6.1431576999999961)
  }
}
```

**npm install --save @types/google-maps**



# Customisation avec Sass

## SASS c'est quoi ?

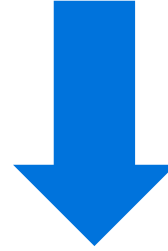
**SASS** est un système qui étend les possibilités de **CSS**. Il permet d'être plus productif et ainsi d'avoir un **CSS** mieux organisé et plus facile à maintenir.

---

- Né dans la communauté **Ruby On Rail**
- Il existe une **Gem Ruby**

# SASS c'est quoi ?

Écrire son script **Sass**



Le préprocesseur interprète le  
script **Sass**



Les fichiers **css** sont produits  
par le préprocesseur

## 2 syntaxes

### SCSS

---

Syntaxe classique qui utilise **brackets** et **semicolon**, tout ce qui est valide **css** est valide **scss**.  
C'est la syntaxe la plus utilisée.

### SASS

---

À la place d'utiliser les **brackets** et **semicolon**, ce dernier utilise l'indentation pour séparer les blocs de codes.  
Il utilise l'extension **.sass**

# Les Features



## 6 features principales

1. Variables
2. Nesting
3. Mixins
4. Partial
5. @extend
6. Function

# Les variables

---

Les variables vous permettent d'assigner une valeur afin de la réutiliser et de vous en souvenir plus facilement.

Fonctionne avec tout type : *Boolean, color, text, string ....*

## Fichier Sass:

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

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

## Fichier CSS compilé:

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

# Nesting

---

Si vous avez passé du temps à écrire du **CSS** dans un projet, vous avez probablement écrit beaucoup de lignes comme celle-ci:

```
nav.tabbed-navigation { ... }  
nav.tabbed-navigation a { ... }  
nav.tabbed-navigation a span { ... }
```

Cela crée beaucoup de lignes répétées dans votre feuille de style qui sont difficiles à lire, avec une forte dépendance à l'égard des sélecteurs CSS descendants

Si le nom de la classe de l'élément `< nav >` change , vous devrez réécrire les trois sélecteurs CSS dans l'exemple. Mais avec Sass ...

# Nesting

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

---

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



# Mixins

---

Les **mixins** vous permettent de définir des styles qui pourront être réutilisés partout dans votre **CSS**.

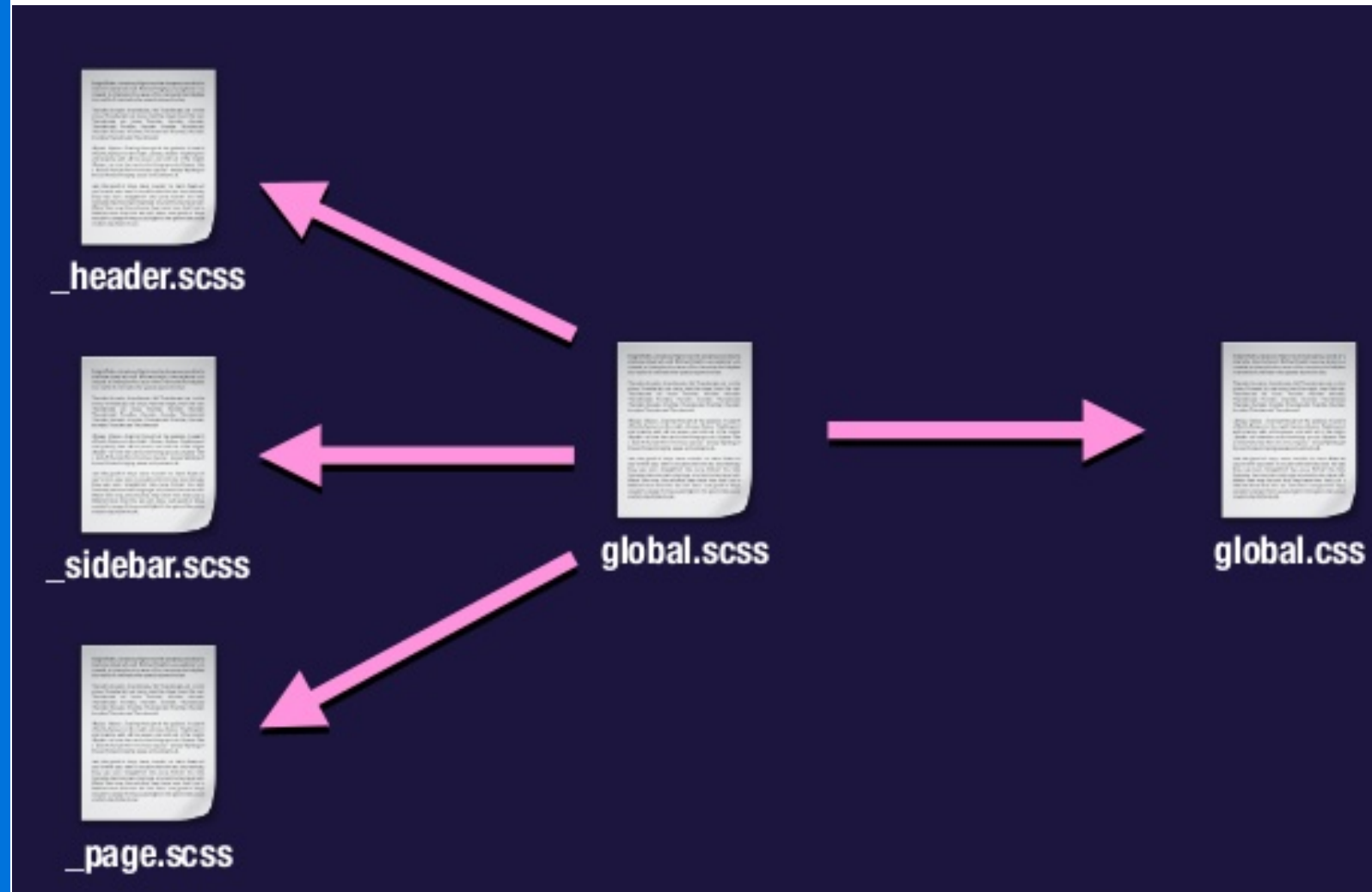
Vous pouvez même passer des valeurs pour rendre votre mixin plus souple .

```
@mixin border-radius($radius) {  
    -webkit-border-radius: $radius;  
    -moz-border-radius: $radius;  
    -ms-border-radius: $radius;  
    border-radius: $radius;  
}  
  
.box { @include border-radius(10px); }
```

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

# Partials

**Sass** permet de séparer les feuilles de style afin de rendre le code plus modulable et ainsi mieux l'organiser.



# @extend

La directive @extend vous permet de partager un ensemble de propriétés CSS d'un sélecteur à l'autre.

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

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

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

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

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

---

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

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

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

## Natives

```
$red: rgb(255, 0, 0);
```

---

## Personnalisés

```
@function addition($a, $b) {  
  @return $a + $b;  
}  
  
@function shade($color, $percent) {  
  @return mix(#000, $color, $percent);  
}  
  
body {  
  background: shade(grey, 50%);  
}
```

# Sass et Ionic



```
// App Shared Color Variables
// -----
// It's highly recommended to change the default colors
// to match your app's branding. Ionic uses a Sass map of
// colors so you can add, rename and remove colors as needed.
// The "primary" color is the only required color in the map.
// Both iOS and MD colors can be further customized if colors
// are different per mode.
```

```
$colors: (
  primary: #387ef5,
  secondary: #32db64,
  danger: #f53d3d,
  light: #f4f4f4,
  dark: #222,
  favorite: #69BB7B,
);
```

Gestion par template.

# Customisable avec Sass

- Customisation du composant Toggle



```
@include transition-timing-function(ease-in-out);
@include transition-duration($toggle-transition-duration);
@include transition-property((background-color, border));

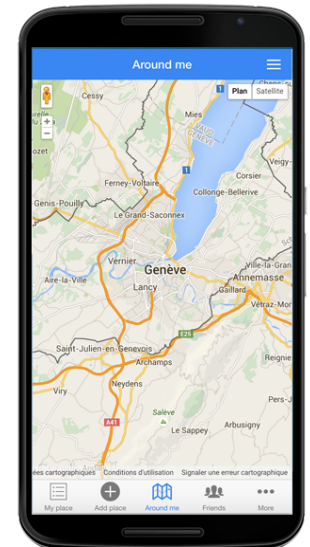
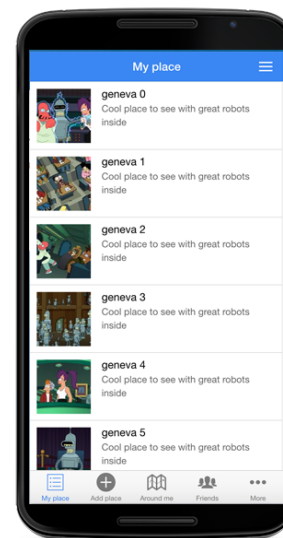
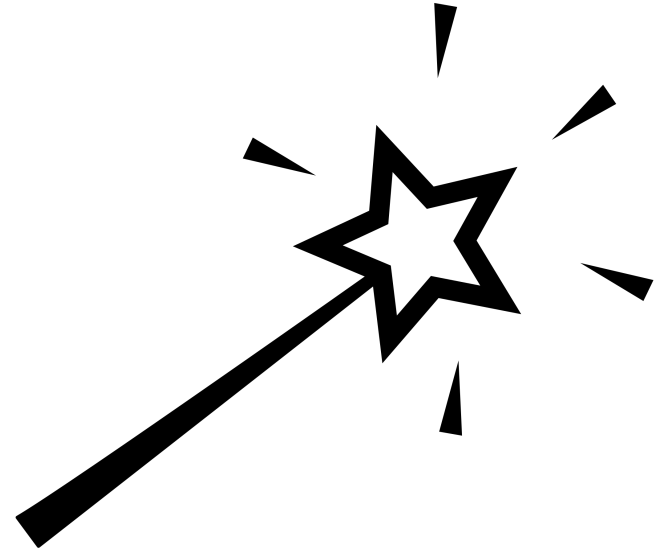
display: inline-block;
box-sizing: border-box;
width: $toggle-width;
height: $toggle-height;
border: solid $toggle-border-width $toggle-off-border-color;
border-radius: $toggle-border-radius;
background-color: $toggle-off-bg-color;
content: ' ';
cursor: pointer;
pointer-events: none;
```

## Deux options :

- Surcharge CSS (attention aux performances)
- *Fork* du fichier Sass

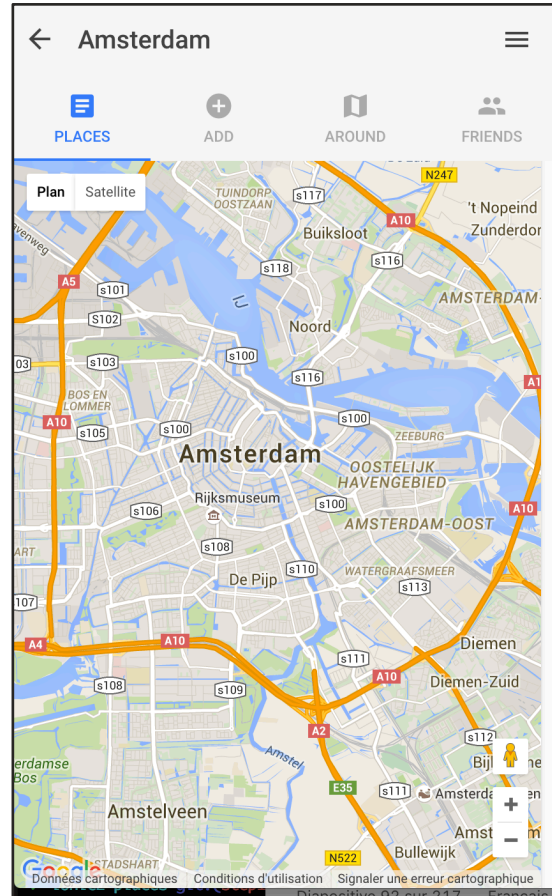
# Objectif

## Customiser l'application avec Sass





## Au clique sur une place, afficher la map au bon endroit.



```
export class PlacePage {  
  
  @ViewChild(Maps)  
  private map: Maps;  
  
  selectedPlace:any;  
  
  constructor(private nav: NavController, private params: NavParams, private routes: Routes) {  
    const selectedId = params.data.id;  
    this.selectedPlace = places.getPlaceById(selectedId)  
  }  
  
  ionViewDidEnter(){  
    const pos = this.selectedPlace.position  
    this.map.init(pos.lat, pos.long)  
  }  
  
  onClickBack(){  
    this.nav.setRoot(this.routes.getPage(this.routes.TABS))  
  }  
}
```

```
selectItem(id){  
  this.nav.insert(0,this.routes.getPage(this.routes.PLACE),{id:id})  
}
```

```
{  
  "id":0,  
  "name":"Amsterdam",  
  "description":"Cool place to see with great robots inside",  
  "image":"http://lorempicsum.com/futurama/100/100/1",  
  "position":{  
    "lat":"52.3702157",  
    "long":"4.895167899999933"  
  }  
}
```