

Live template

1ère étape :

Nous importons le paquet de matériel.

Considérez le matériel comme un bureau devant vous. C'est juste une surface plane sur laquelle vous pouvez déplacer des objets et travailler avec.

2ème étape :

On va créer une nouvelle application matérielle. C'est une application qui utilise un matériau.

```
void main() {  
  runApp(new MaterialApp(  
    home: new MyApp(),  
  )); // MaterialApp  
}
```

Ensuite, nous avons notre fonction principale qui appelle runApp. runApp est le principal point d'entrée run app. C'est juste une fonction qui prend cette classe. Dans ce cas, mon application le pousse.

3ème étape :

Il faut définir MyApp, notre application principale. On veut suivre l'état de notre application donc on étend la classe MyApp d'un widget avec état.

```
class MyApp extends StatefulWidget {  
  @override  
  _State createState() => new _State();  
}
```

notre application étend un widget avec état, ce qui signifie qu'il va se souvenir de son état

4ème étape :

On va créer une nouvelle instance de la classe d'état qui va réellement décrire l'état de notre application.

On fait appel au widget build. On aura besoin d'un contexte dans lequel cette méthode va être exécutée. C'est le contexte actuel et c'est ce dont nous avons besoin. Dans ce contexte que le code principale va être ajouté.

On retourne un scaffold (un échafaudage), c'est une structure sur laquelle vous allez construire votre application matérielle. Ça veut dire tout ce qui est barre d'application, boutons, champs de texte

C'est comme un bâtiment à plusieurs niveaux, le premier étant la barre d'application avec un titre.

Après on définit un conteneur pour créer notre corps. Et on essaye d'organiser le corps de notre application comment on le veut.

```

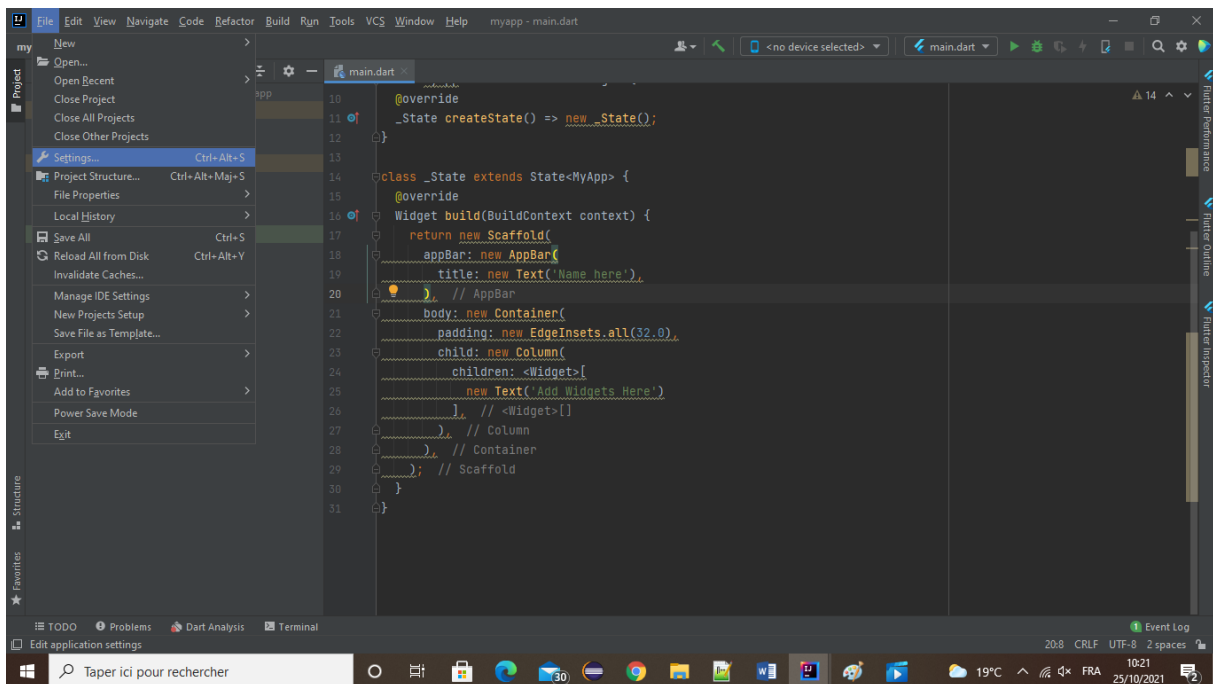
class _State extends State<MyApp> {
  @override
  Widget build(BuildContext context) {
    return new Scaffold(
      appBar: new AppBar(
        title: new Text('Name here'),
      ), // AppBar
      body: new Container(
        padding: new EdgeInsets.all(32.0),
        child: new Column(
          children: <Widget>[
            new Text('Add Widgets Here')
          ], // <Widget>[]
        ), // Column
      ), // Container
    ); // Scaffold
  }
}
  
```

Titre de votre application
 ils sont des widgets enfants
 Ajoutez le texte que vous voulez afficher

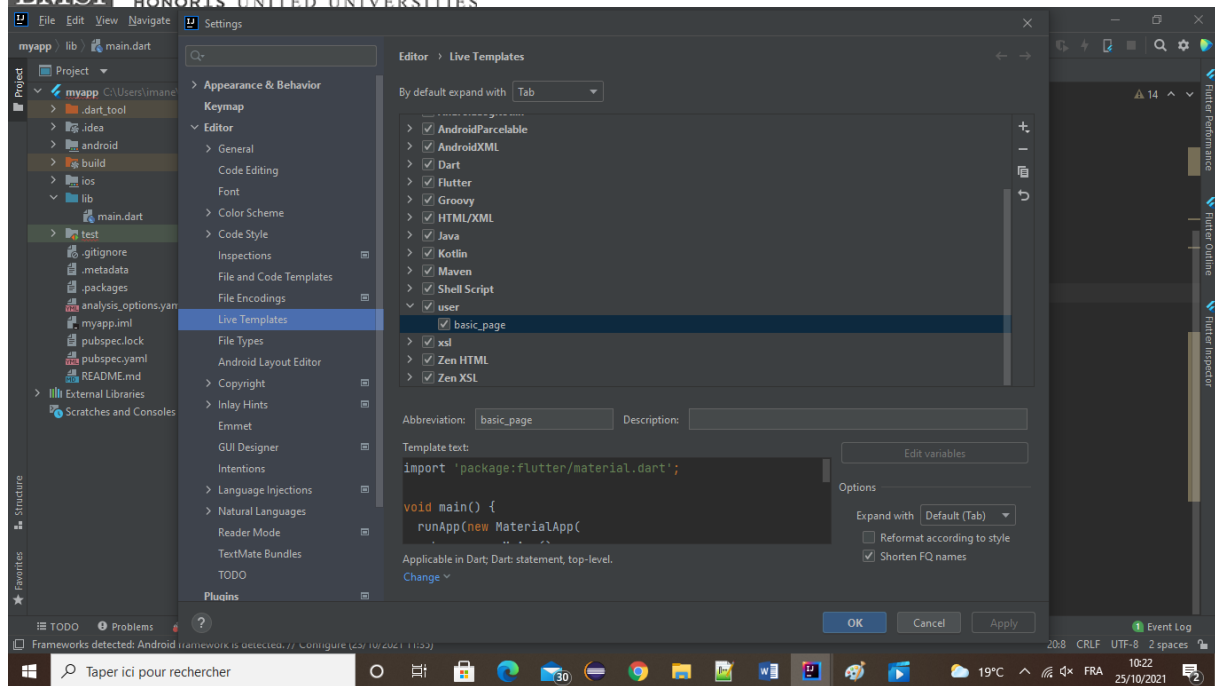
5ème étape :

On veut enregistrer ce code comme modèle d'application sur lequel on va se baser pour les prochains TP.

Copiez votre code. Cliquez sur file -> settings.



Choisissez live template, et cochez flutter.



Cliquez sur le petit + à droite. Collez votre code dans la partie template text, donnez un nom à votre template et cliquez sur Ok.

A chaque fois vous voulez utiliser ce modèle il suffit de taper le nom de votre template, et le code sera automatiquement généré.