

Hello Flutter

Widget

- 플러터는 모든 것이 위젯으로 이루어져 있다.

```
Run | Debug | Profile  
void main() {  
  runApp(App());  
}
```

- 플러터는 runApp을 통해 위젯을 불러와서 실행시키는 형태이다
 - runApp은 void
- 위젯을 만들기 위해서는 flutter SDK에 있는 3개의 core widget중에 하나를 상속받아야 한다.
- 가장 다루기 쉬운 StatelessWidget 을 상속받음

```
class App extends StatelessWidget {  
  
}
```

- 오류가 뜨는데 StatelessWidget는 build 메소드를 구현해야 함
- build 메소드는 flutter가 실행하게 되는데
- build가 return하는 것을 화면에 띄운다.

```
class App extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return  
  }  
}
```

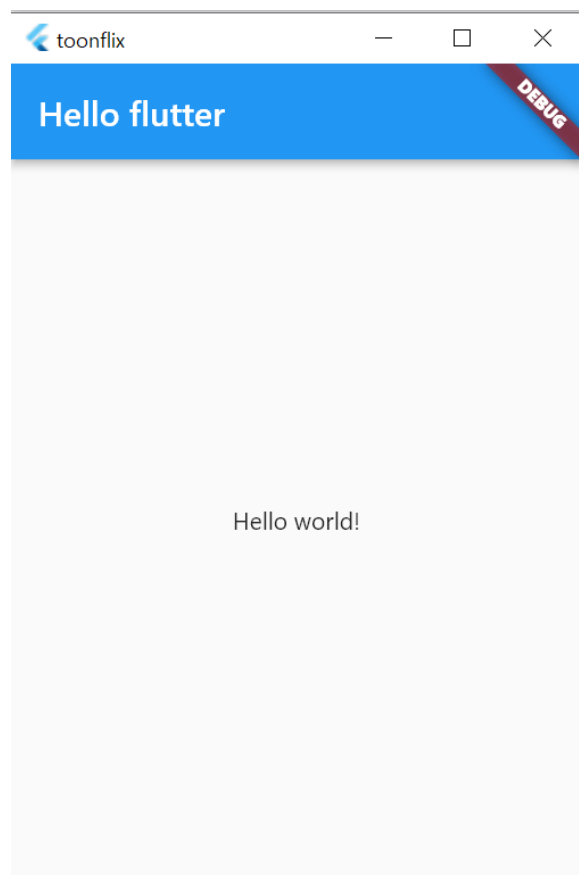
- App 이 root라고 생각하고
- build는 앱을 리턴하면 된다
- root는 두가지 디자인 옵션 중 하나를 선택해야 한다.
 - 두 가지 옵션
 - material (구글 디자인 시스템)
 - cupertino (애플 디자인 시스템)

vsc 자동정렬

- shift + alt + F
- 잘 활용하기위해 class 만들 때 마다 , 찍는 것을 잊지말자

```
void main() {
  runApp(App());
}

class App extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(
          title: Text('Hello flutter'),
        ), // AppBar
        body: Center(
          child: Text('Hello world!'),
        ), // Center
      ), // Scaffold
    ); // MaterialApp
  }
}
```



- 해당 코드의 실행 결과
- material 옵션을 선택하고 home을 리턴함
 - 사용자가 우리 앱의 home에 있을 때 보여질 화면
- home 안에는 scaffold를 사용하여 구조를 만들었다.

- scaffold의 옵션으로 appBar 와 body를 선택
- appBar 에는 AppBar 옵션
- body에는 Center 옵션을 선택
- 이런 식으로 플러터의 구조는 위젯을 선언해 나가는 식이다.