

모바일 스터디 5주차 과제 (닉네임:프로토타입)

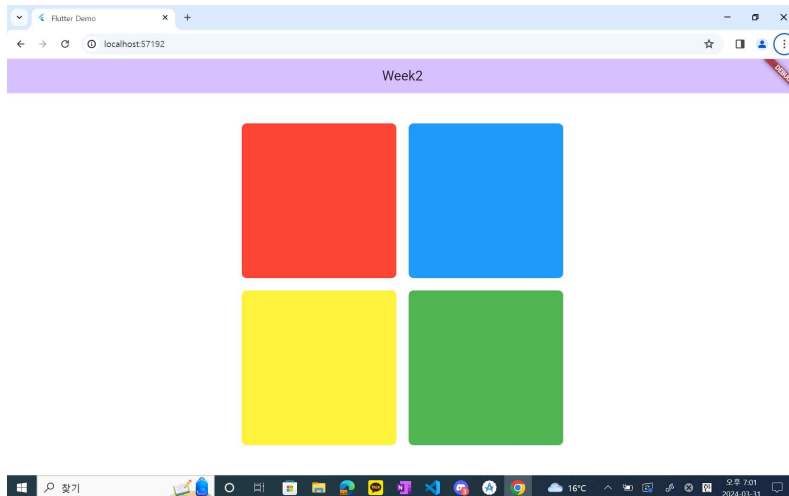
State란?

State의 뜻이 '상태'라는 것은 누구나 알고 있을 것이다. 그렇다면 flutter에서 이 '상태'라는 것은 어떤 상태를 의미하는 것일까? 예를 들어, 처음엔 숫자가 0이었는데 내가 버튼을 누를 때마다 숫자가 1씩 증가하는 카운터가 있다고 하자. 이 경우에 내가 버튼을 누름으로써 카운터의 상태, 즉 state가 바뀌었다고 할 수 있는 것이다. 이러한 state의 변화를 반영하는지 여부에 따라 위젯을 stateless widget과 stateful widget으로 분류할 수 있다. stateless widget은 state의 변화를 반영하지 않는 위젯이고, stateful widget은 state의 변화를 반영하는 위젯이다.

Stateless Widget이란?

Stateless Widget은 상태 변화를 필요로 하지 않는 위젯이다.

2~3주차 과제에서 사용했던 위젯은 모두 Stateless Widgets이었다. 2~3주차에서 했던 과제를 다시 상기해보자.



2주차는 간단하게 사각형 위젯을 구현하는 과제였다. 당연하지만 이 위젯들은 stateless widgets이다. 사각형이 구현되었으면 그것으로 끝일 뿐, 다른 상태 변화를 요구하지 않기 때문이다. 그렇다면 상태 변화를 요구한다는건 도대체 무엇일까? 다음에서 설명한다.

Stateful Widget이란?

Stateful Widget이 바로 상태 변화를 요구하는 위젯이다. 예를 들어 설명해보자.

Welcome, Guest!

Change Username

초기 화면이 위와 같은 앱이 있다. 이때, 버튼을 누르면 화면이 다음과 같이 바뀐다.

Welcome, John!

Change Username

처음에는 닉네임이 Guest였지만, 버튼을 누름으로써 닉네임이 John으로 바뀌어 화면에 출력된다. 이렇게 상태 변화를 요구하는 위젯이 Stateful Widget이다.

상태 바꾸기

Stateful Widget으로 상태를 바꾸려면 setState라는 함수를 이용하면 된다.

```
21 class _MyHomePageState extends State<MyHomePage> {
22   String _username = "Guest";
23
24   @override
25   Widget build(BuildContext context) {
26     return Scaffold(
27       appBar: AppBar(
28         title: Text('Flutter State Example'),
29       ), // AppBar
30       body: Center(
31         child: Column(
32           mainAxisAlignment: MainAxisAlignment.center,
33           children: <Widget>[
34             Text(
35               'Welcome, $_username!',
36               style: TextStyle(fontSize: 24),
37             ), // Text
38             SizedBox(height: 20),
39             ElevatedButton(
40               onPressed: () {
41                 // 사용자 이름 변경
42                 setState(() {
43                   _username = "John";
44                 });
45               },
46               child: Text('Change Username'),
47             ), // ElevatedButton
```

코드를 간단히 살펴보자.

21행에서 _username은 “Guest”로 초기화하였고, 35행에서 ‘Welcome, \$_username!’을 화면에 출력하도록 하고 있다. 따라서 초기 화면에서는 ‘Welcome, Guest’라는 문장이 화면에 출력된다.

한편, 39~47행을 보면 ‘Change Username’라는 텍스트가 적힌 버튼(Elevated Button) 위젯이 선언되어 있다. 이 버튼 위젯의 코드 내용을 보면 이 버튼을 눌렀을 때(onPressed), setState() 함수에 의해 _username이 “John”으로 바뀌도록 하고 있다.

아까 실행결과에서 버튼을 눌렀을 때, “Guest”가 “John”으로 바뀌었음을 확인했다. 즉, _username이 “John”으로 바뀌는 것에서 끝나는 것이 아니라, 35행의 \$_username 부분까지도 “John”으로 바뀌었고 이것이 화면에 최종 반영된 것이다. 만약에 Stateful Widget이 아니라 Stateless Widget이었으면 어떨까? 물론 _username은 똑같이 “John”으로 바뀐다. 하지만 화면에 출력되는 문장은 여전히 “Welcome, Guest!”이다. Stateless Widget은 상태 변화를 반영하지 않는 위젯이기 때문에 Text 위젯의 상태도 당연히 바뀌지 않는다.

상태관리 패턴

Provider : Flutter의 상태 관리 패턴 중 한 가지로, 앱의 상태를 관리하고 이에 관한 위젯을 제공한다. InheritedWidget를 기반으로 앱의 상태를 위젯 트리 전체에서 공유하며, 데이터 변경 시에 해당 데이터를 사용하는 위젯에 알려준다.

Riverpod : InheritedWidget과 비슷한 동작을 하는 새로운 매커니즘으로 재구현한 상태 관리 패키지이다. Riverpod의 공식 사이트에서는 Riverpod를 리액티브 캐싱, 데이터 바인딩 프레임워크라고 소개하는데, 리액티브 캐싱(Reactive Caching)은 데이터를 비동기적으로 계산할 때 캐싱하여 해당 데이터가 필요한 모든 곳에서 쉽게 접근할 수 있도록 하는 기술이고, 데이터 바인딩(Data Binding)은 UI와 데이터를 결합하는 기술로 데이터의 변경에 따라 UI를 자동으로 변경해주는 기술이다.

GetX : 매우 가볍고 강력한 패키지이다. 고성능 상태 관리, 지능형 종속성 주입, 라우트 관리 기능을 제공한다.