

## Mobile Flutter

### 1. Flutter គិតខ្លួន

- Native: iOS = Swift / Android = Kotlin
- Cross-platform: ផើយករ៉ាងទីយោ ព័ត៌មាន 2 របៀប

### 2. Flutter Concept ទំនួរការ

- Everything is Widget

```
Text('Hello Flutter');
```

- Widget Tree

```
Scaffold(  
    appBar: AppBar(title: Text('Home')),  
    body: Center(  
        child: Text('Hello'),  
    ),  
,);
```

- Build flow (Widget → Build → Frame)

Widget → build() → UI Frame

- Hot reload vs Hot restart

Hot reload: UI ផ្សេងៗ / state ម៉ោង

Hot restart: reset នូវអេណុល

### 3. Widget ផ្តល់ព័ត៌មាន

- Text
- Icon
- Image
- Scaffold
- AppBar

```
Scaffold(  
    appBar: AppBar(title: Text('My App')),  
    body: Column(  
        children: [  
            Text('Hello'),  
            Icon(Icons.star),  
            Image.network('https://picsum.photos/200'),  
,  
,);
```

## Dart พื้นฐาน + Layout ที่ใช้จริง



- อ่าน Dart ออก
- เข้าใจโครงสร้าง layout

### Dart ที่จำเป็น

- var / final

```
var name = 'John';
final age = 20;
```

- int / double / String / bool

```
int a = 10;
double b = 10.5;
String c = 'Flutter';
bool isActive = true;
```

- List / Map

```
List<int> numbers = [1, 2, 3];
Map<String, dynamic> user = {
  'name': 'John',
  'age': 20,
};
```

- function

```
int sum(int a, int b) {
  return a + b;
}
```

- if / for

```
if (age > 18) {
  print('Adult');
}
```

```
for (var n in numbers) {
  print(n);
}
```

- class

```
class User {  
    final String name;  
    final int age;  
    User({required this.name, required this.age});  
}
```

## Flutter Layout

- Row / Column

```
Row(  
    children: [  
        Text('A'),  
        Text('B'),  
    ],  
);
```

- Container
- Padding
- SizedBox

```
Container(  
    padding: EdgeInsets.all(16),  
    color: Colors.blue,  
    child: Text('Box'),  
);
```

- Expanded

```
Row(  
    children: [  
        Expanded(child: Text('Left')),  
        Text('Right'),  
    ],  
);
```

## UI จริง + Interaction

### Widgets

- TextField

```
TextField(  
  decoration: InputDecoration(  
    labelText: 'Email',  
  ),  
);
```

- ElevatedButton

```
ElevatedButton(  
  onPressed: () {  
    print('Clicked');  
  },  
  child: Text('Login'),  
);
```

- InkWell
- GestureDetector

```
InkWell(  
  onTap: () {  
    print('Tapped');  
  },  
  child: Text('Click me'),  
);
```

- ListView

```
ListView.builder(  
  itemCount: 3,  
  itemBuilder: (_, i) {  
    return ListTile(  
      title: Text('Item $i'),  
    );  
  },  
);
```

## State Management ด้วย GetX



- เข้าใจ Controller
- รู้จัก Rx / Obx
- ล่ามาร์ค์อัพ GetX "ได"



สอนแบบ “ไม่ทุกษี”

### 1. ปัญหาที่ไม่มี State

- setState วนวาย
- ส่งค่าข้าม widget ยาก

### 2. GetX Concept

- Controller = logic
- View = UI
- Rx = state

#### Controller

```
class CounterController extends GetxController {  
    final count = 0.obs;  
  
    void increment() {  
        count.value++;  
    }  
}
```

#### View + Obx

```
final c = Get.put(CounterController());  
  
Obx(() => Text('${c.count.value}'));
```

#### Button

```
ElevatedButton(  
    onPressed: c.increment,  
    child: Text('Add'),  
);
```

## โครงสร้างโปรเจกต์จริง + API

```
lib/
└── modules/
    └── user/
        ├── controller/
        ├── view/
        └── model/
```

View → การแสดงผล

Controller → คุม state + call API

Model → แปลง JSON

### การเข้มต่อ API ด้วย Dio (Basic)

Model (แปลง JSON → Object) แปลงในเว็บไซต์ json to dart

```
class UserModel {
    final int id;
    final String name;
    final String email;

    UserModel({
        required this.id,
        required this.name,
        required this.email,
    });

    factory UserModel.fromJson(Map<String, dynamic> json) {
        return UserModel(
            id: json['id'],
            name: json['name'],
            email: json['email'],
        );
    }
}
```

### Controller (เรียก API)

```
class UserController extends GetxController {
    final users = <UserModel>[].obs;
    final isLoading = false.obs;

    final dio = Dio();

    @override
    void onInit() {
```

```

loadUsers();
super.initState();
}

Future<void> loadUsers() async {
try {
isLoading.value = true;

final response = await dio.get(
'https://jsonplaceholder.typicode.com/users',
);

final data = response.data as List;

users.value = data
.map((e) => UserModel.fromJson(e))
.toList();
} catch (e) {
Get.snackbar('Error', 'โหลดข้อมูลไม่สำเร็จ');
} finally {
isLoading.value = false;
}
}
}

```

### View (แสดงผล)

```

class UserView extends StatelessWidget {
UserView({super.key});

final controller = Get.put(UserController());

@Override
Widget build(BuildContext context) {
return Scaffold(
appBar: AppBar(title: const Text('Users')),
body: Obx(() {
if (controller.isLoading.value) {
return const Center(
child: CircularProgressIndicator(),
);
}

return ListView.builder(
itemCount: controller.users.length,
itemBuilder: (_, index) {
final user = controller.users[index];

```

```
return ListTile(  
    title: Text(user.name),  
    subtitle: Text(user.email),  
);  
},  
);  
}  
};
```