Flutter技术分享

UI相关

> Flutter的最大优势就是画UI，Flutter中大部分绘制单元是Widget，各种复杂的UI都是通过Widget组合一起的

typing_learn .zip
11.61MB

简单例子

一个简单的文本

```
Text("你好")
```

加一个背景红色

```
Container(
                decoration: const BoxDecoration(
                  color: Colors.red,
                ),
                child: const Text("你好"),
              )
```

增加一个10px的内边距

```
Padding(
                  padding: const EdgeInsets.all(10.0),
                  child: Container(
                    decoration: const BoxDecoration(
                      color: Colors.red,
                    ),
                    child: const Text("你好"),
                  ),
                  )
```

进一步封装

> 其实画UI就是拼积木，Flutter提供了各种各样的Widget，各种Widget可以 互相组合实现不同的功能，一般这些是最基础的Widget，我们如果需要实现具体的界面UI就需要我们自己封装，比如 个人信息的显示 或者头像的显示等，封装界面的时候在Flutter中也非常统一，就两个Widget，区别就是看看是否有状态,StateFulWidget、StateLessWidget，如下所示

StateLessWidget

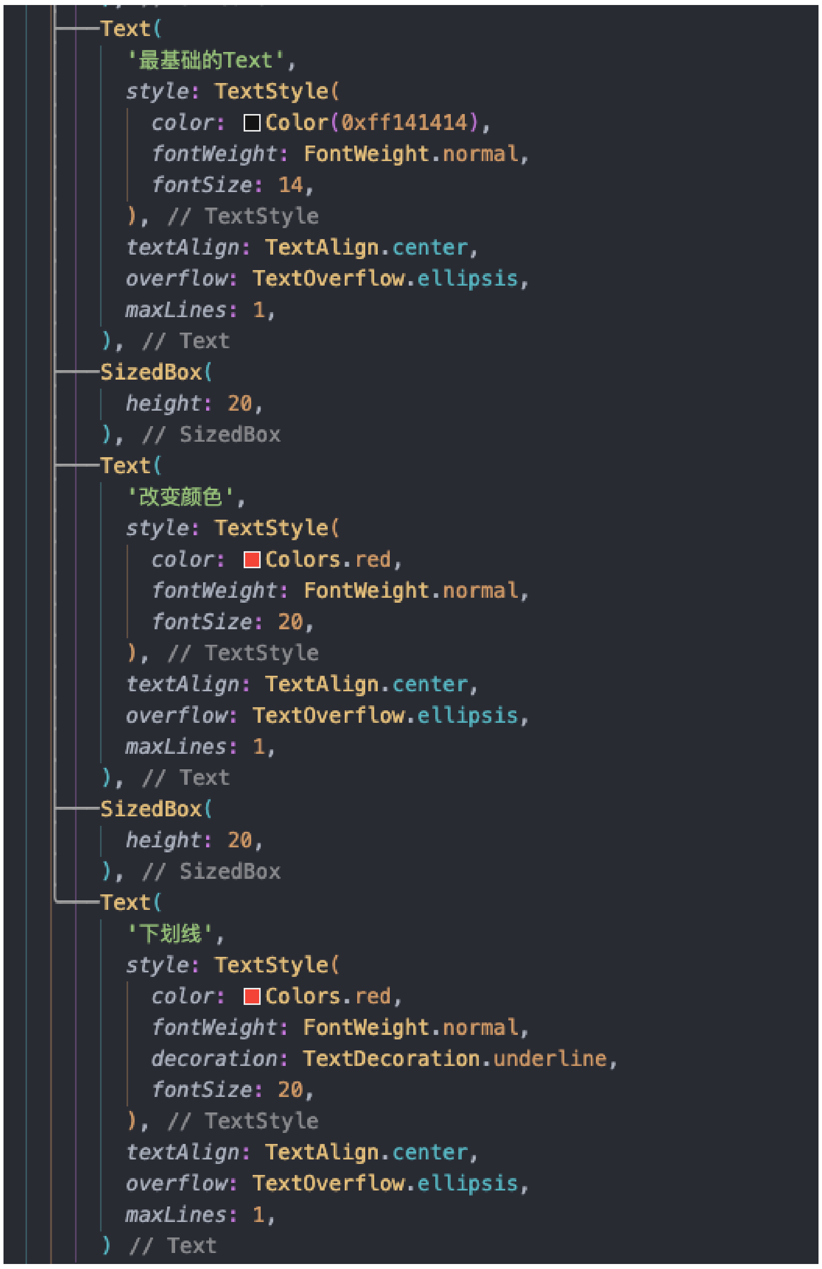
```dart
class PersonInfoView extends StatelessWidget {
  const PersonInfoView({super.key, required this.userName});

  final String userName;

  @override
  Widget build(BuildContext context) {
    return Container(
      decoration: const BoxDecoration(
        color: Colors.white,
      ),
      child: Padding(
        padding: const EdgeInsets.all(8.0),
        child: Text("你好,我是$userName"),
      ),
    );
  }
}
```

StateFulWidget

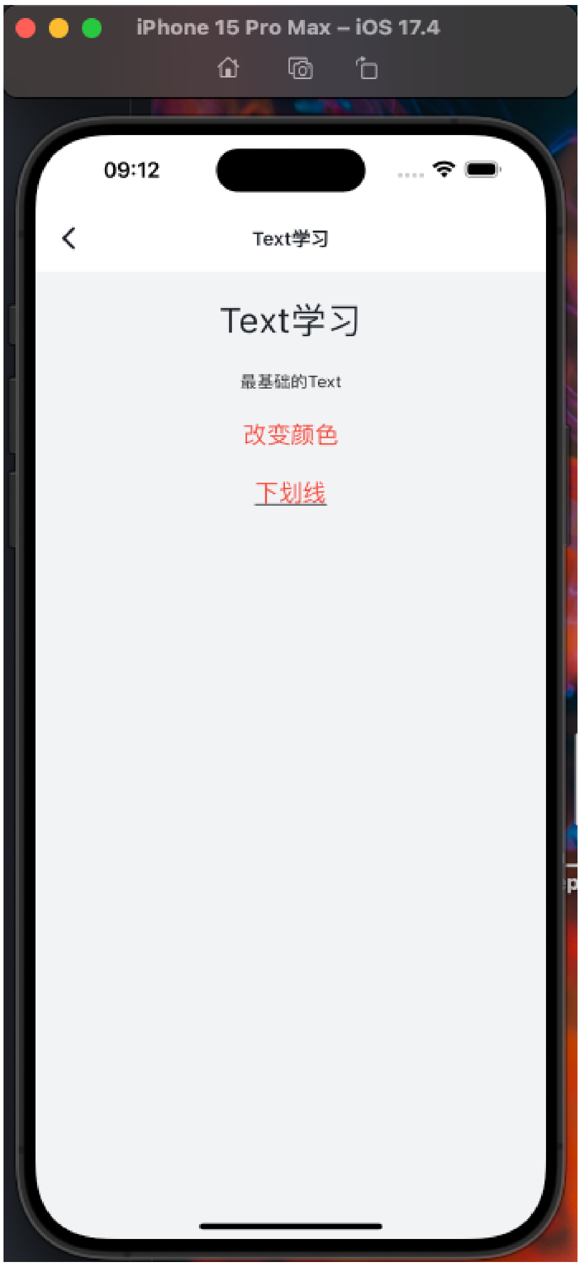
```dart
class PersonInfoView extends StatefulWidget {
  const PersonInfoView({super.key, required this.userName});

  final String userName;
  @override
  PersonInfoViewState createState() => PersonInfoViewState();
}

class PersonInfoViewState extends State<PersonInfoView> {
  @override
  Widget build(BuildContext context) {
    return Container(
      decoration: const BoxDecoration(
        color: Colors.white,
      ),
      child: Padding(
        padding: const EdgeInsets.all(8.0),
        child: Text("你好,我是${widget.userName}"),
      ),
    );
  }

  @override
  void initState() {
    super.initState();
  }

  @override
  void dispose() {
    super.dispose();
  }
}
```

一些基本Widget

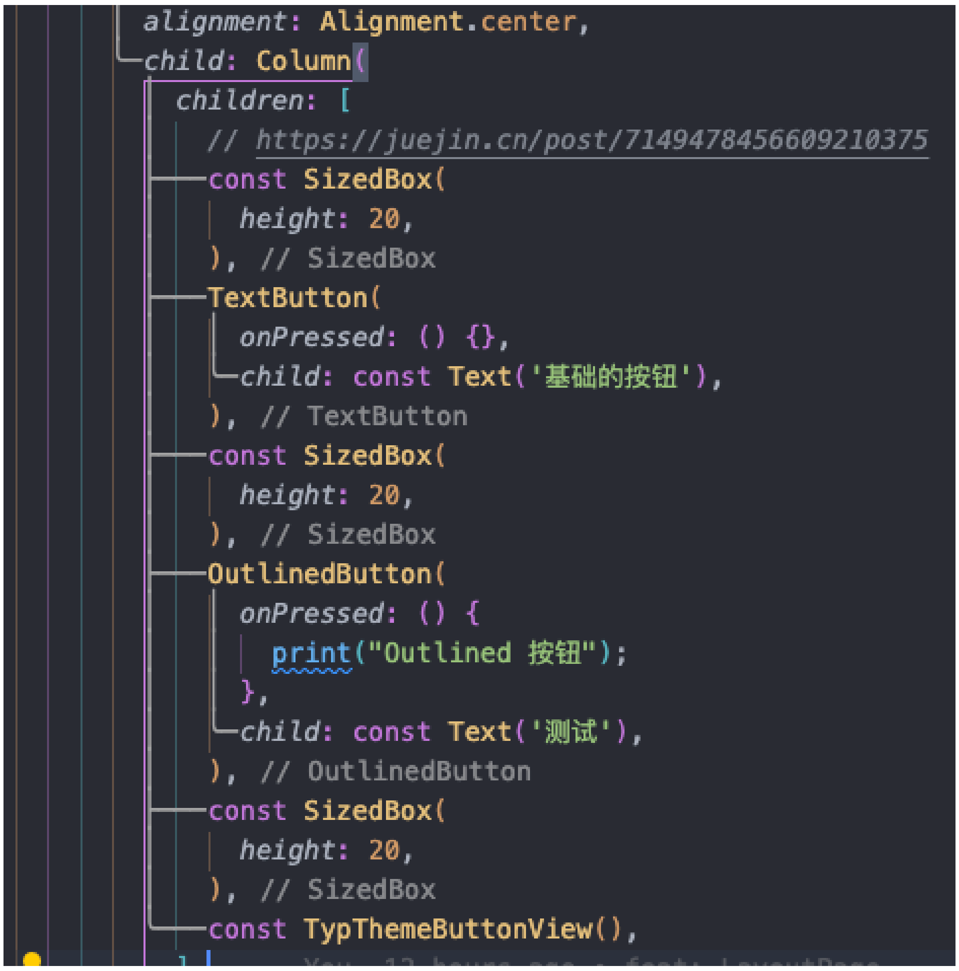
1. Text



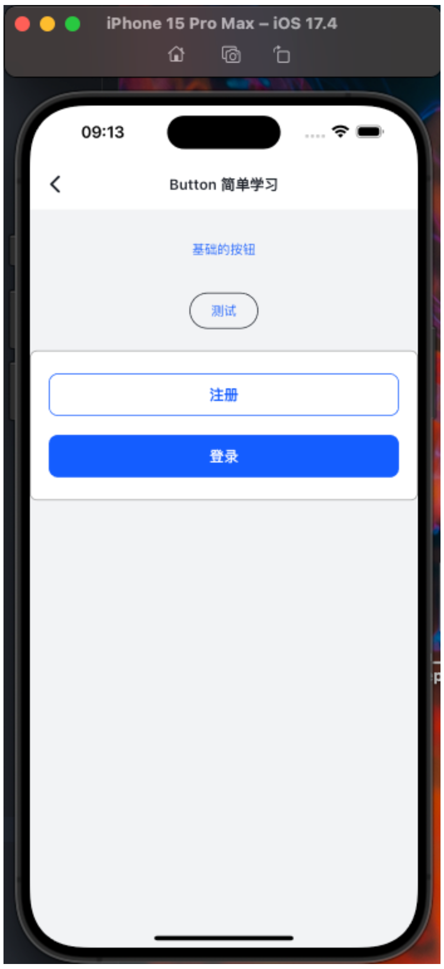
```
Text(
  '最基础的Text',
  style: TextStyle(
    color: Color(0xff141414),
    fontWeight: FontWeight.normal,
    fontSize: 14,
  ), // TextStyle
  textAlign: TextAlign.center,
  overflow: TextOverflow.ellipsis,
  maxLines: 1,
), // Text
SizedBox(
  height: 20,
), // SizedBox
Text(
  '改变颜色',
  style: TextStyle(
    color: Colors.red,
    fontWeight: FontWeight.normal,
    fontSize: 20,
  ), // TextStyle
  textAlign: TextAlign.center,
  overflow: TextOverflow.ellipsis,
  maxLines: 1,
), // Text
SizedBox(
  height: 20,
), // SizedBox
Text(
  '下划线',
  style: TextStyle(
    color: Colors.red,
    fontWeight: FontWeight.normal,
    decoration: TextDecoration.underline,
    fontSize: 20,
  ), // TextStyle
  textAlign: TextAlign.center,
  overflow: TextOverflow.ellipsis,
  maxLines: 1,
) // Text
```



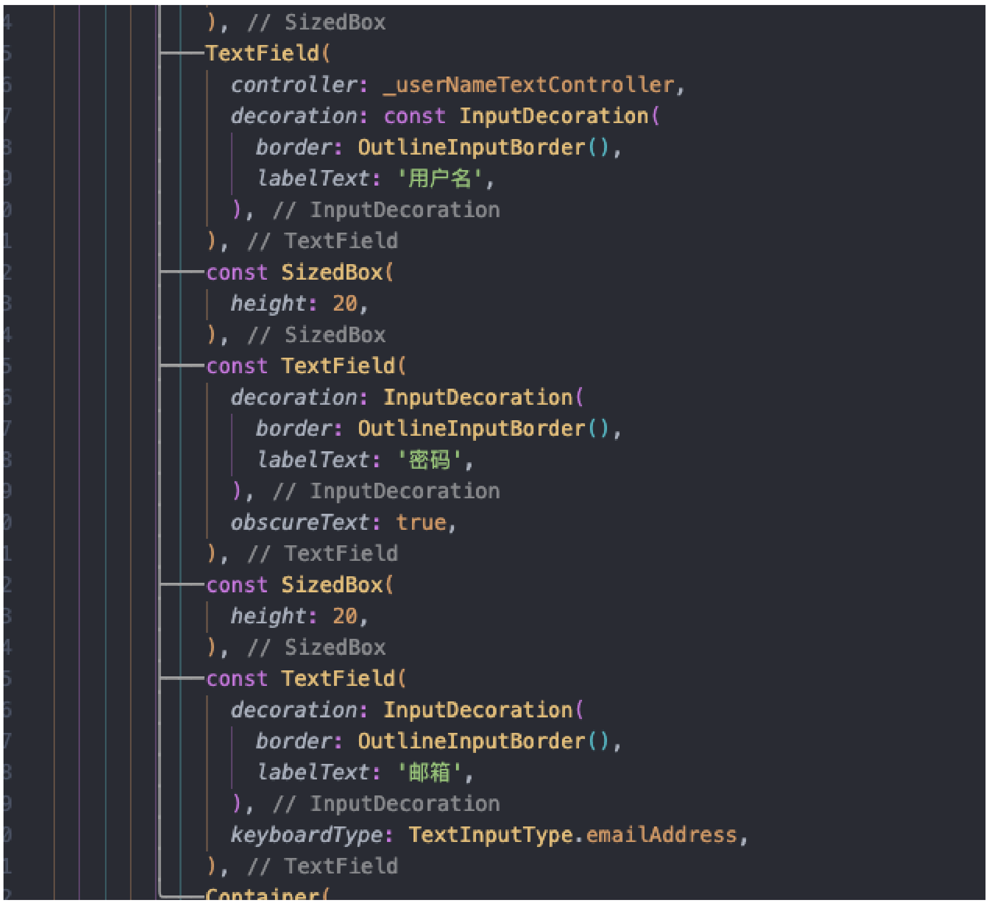
2. Button



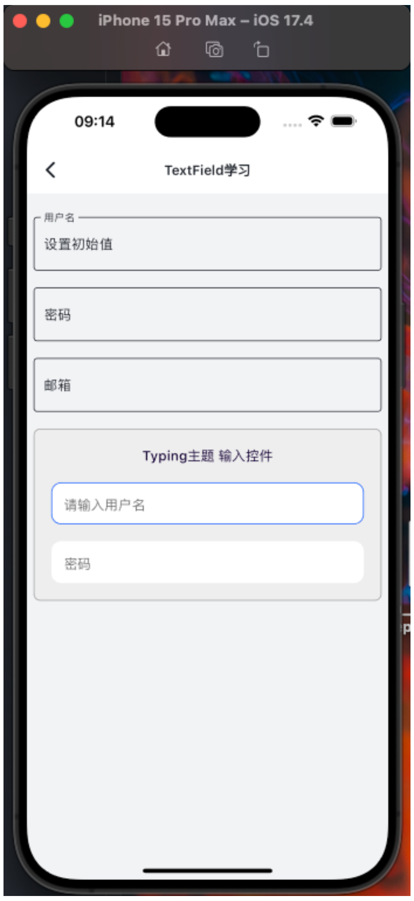
```
    alignment: Alignment.center,
    child: Column(
      children: [
        // https://juejin.cn/post/7149478456609210375
        const SizedBox(
          height: 20,
        ), // SizedBox
        TextButton(
          onPressed: () {},
          child: const Text('基础的按钮'),
        ), // TextButton
        const SizedBox(
          height: 20,
        ), // SizedBox
        OutlinedButton(
          onPressed: () {
            print("Outlined 按钮");
          },
          child: const Text('测试'),
        ), // OutlinedButton
        const SizedBox(
          height: 20,
        ), // SizedBox
        const TypThemeButtonView(),
```



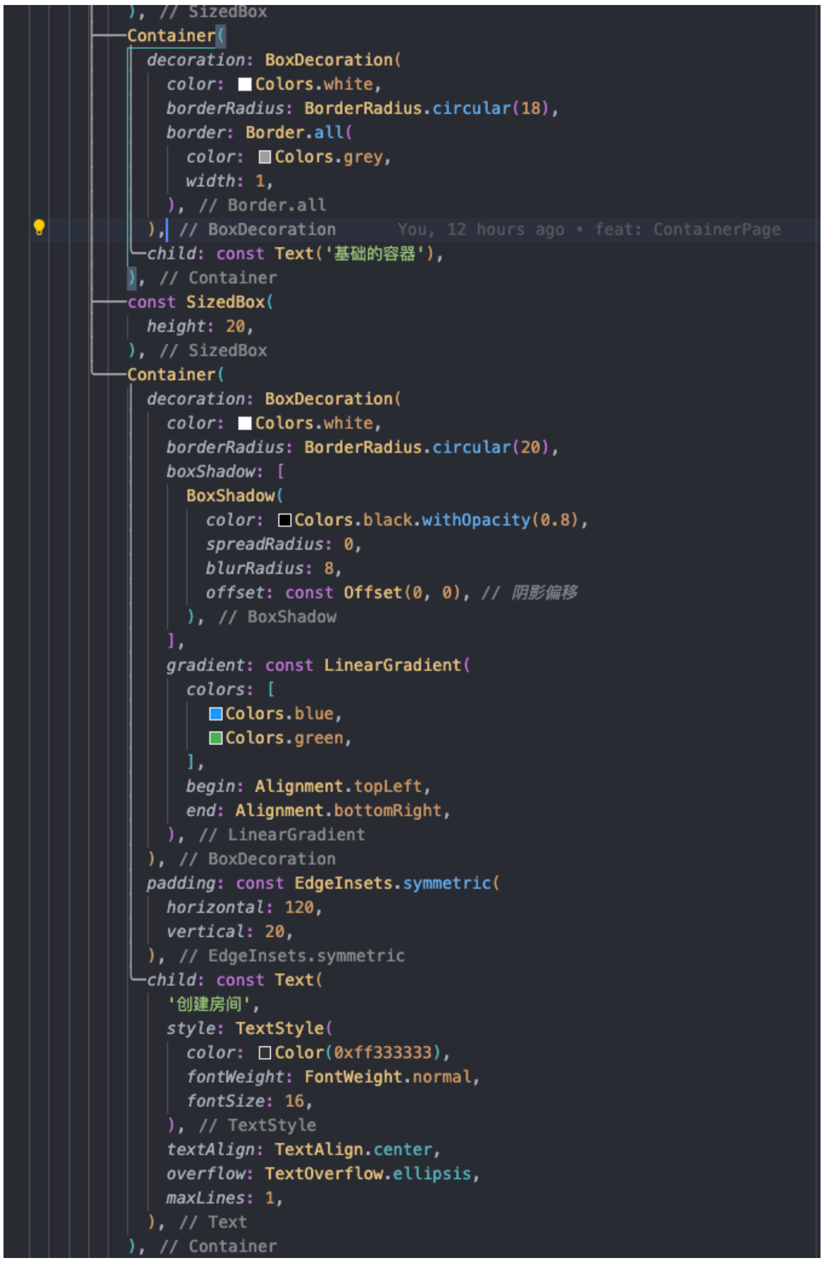
3. TextField



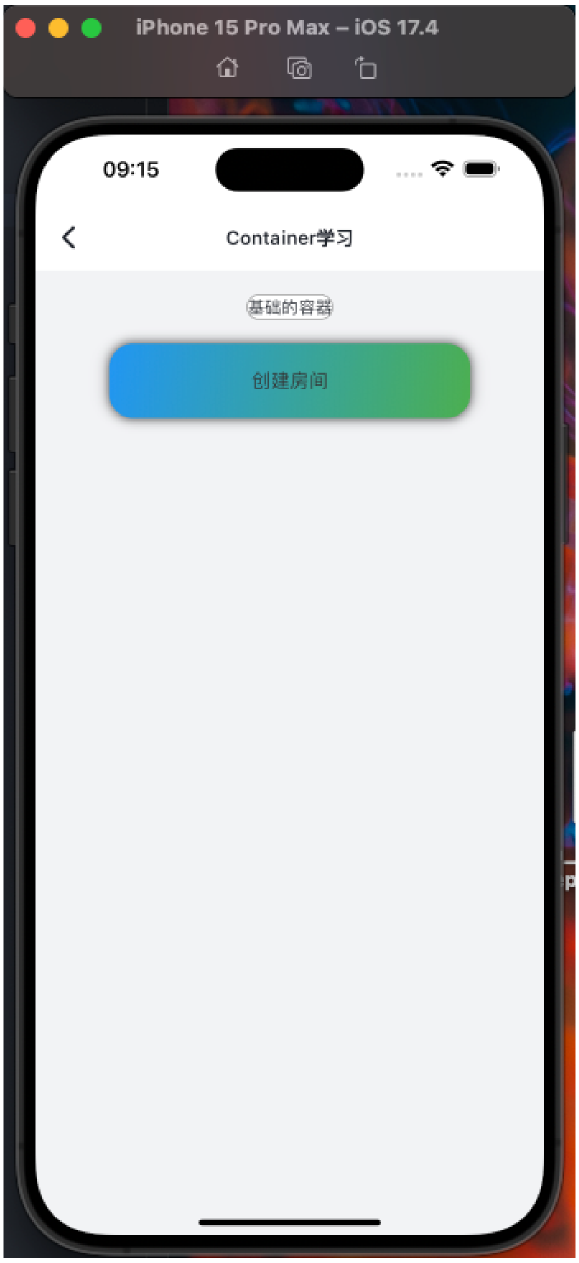
```
        ), // SizedBox
        TextField(
          controller: _userNameTextController,
          decoration: const InputDecoration(
            border: OutlineInputBorder(),
            labelText: '用户名',
          ), // InputDecoration
        ), // TextField
        const SizedBox(
          height: 20,
        ), // SizedBox
        const TextField(
          decoration: InputDecoration(
            border: OutlineInputBorder(),
            labelText: '密码',
          ), // InputDecoration
          obscureText: true,
        ), // TextField
        const SizedBox(
          height: 20,
        ), // SizedBox
        const TextField(
          decoration: InputDecoration(
            border: OutlineInputBorder(),
            labelText: '邮箱',
          ), // InputDecoration
          keyboardType: TextInputType.emailAddress,
        ), // TextField
        Container(
```



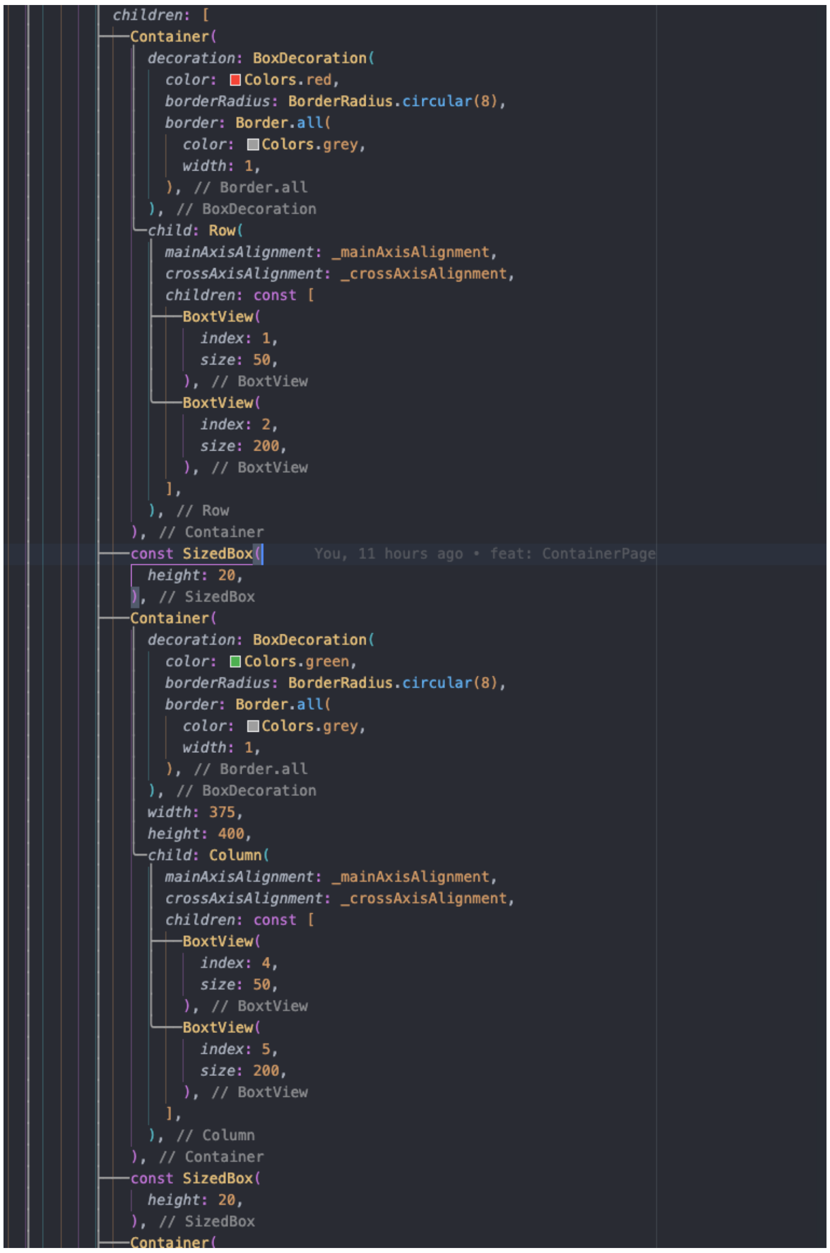
4. Container



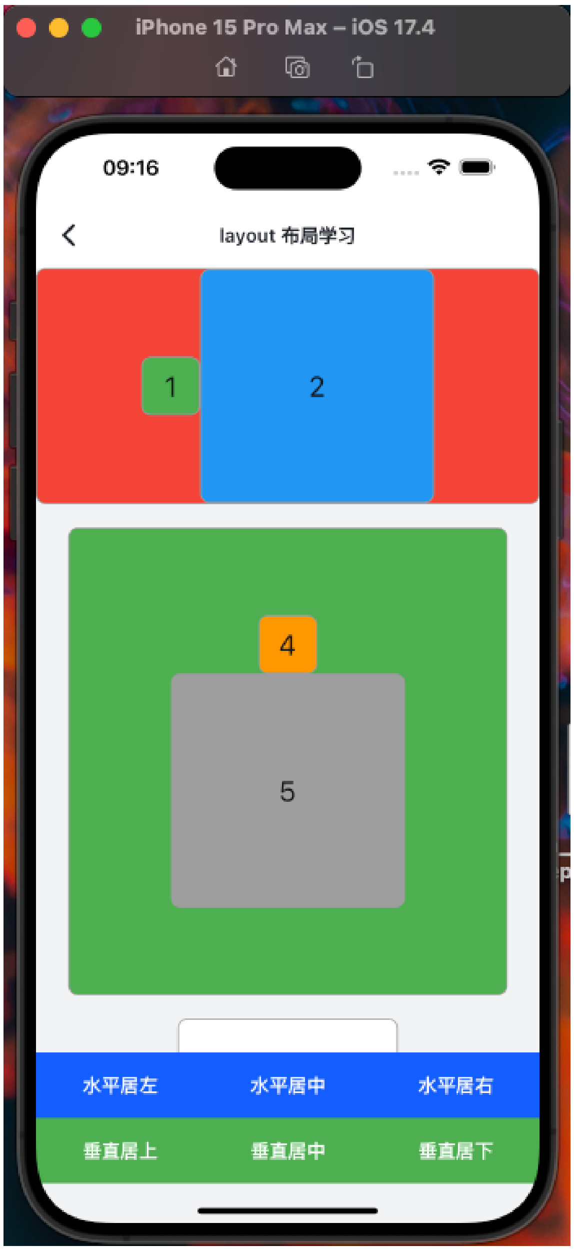
```
), // SizedBox
Container(
  decoration: BoxDecoration(
    color: Colors.white,
    borderRadius: BorderRadius.circular(18),
    border: Border.all(
      color: Colors.grey,
      width: 1,
    ), // Border.all
  ), // BoxDecoration    You, 12 hours ago • feat: ContainerPage
  child: const Text('基础的容器'),
), // Container
const SizedBox(
  height: 20,
), // SizedBox
Container(
  decoration: BoxDecoration(
    color: Colors.white,
    borderRadius: BorderRadius.circular(20),
    boxShadow: [
      BoxShadow(
        color: Colors.black.withOpacity(0.8),
        spreadRadius: 0,
        blurRadius: 8,
        offset: const Offset(0, 0), // 阴影偏移
      ), // BoxShadow
    ],
    gradient: const LinearGradient(
      colors: [
        Colors.blue,
        Colors.green,
      ],
      begin: Alignment.topLeft,
      end: Alignment.bottomRight,
    ), // LinearGradient
  ), // BoxDecoration
  padding: const EdgeInsets.symmetric(
    horizontal: 120,
    vertical: 20,
  ), // EdgeInsets.symmetric
  child: const Text(
    '创建房间',
    style: TextStyle(
      color: Color(0xff333333),
      fontWeight: FontWeight.normal,
      fontSize: 16,
    ), // TextStyle
    textAlign: TextAlign.center,
    overflow: TextOverflow.ellipsis,
    maxLines: 1,
  ), // Text
), // Container
```



5. Layout



```
children: [
  Container(
    decoration: BoxDecoration(
      color: Colors.red,
      borderRadius: BorderRadius.circular(8),
      border: Border.all(
        color: Colors.grey,
        width: 1,
      ), // Border.all
    ), // BoxDecoration
    child: Row(
      mainAxisAlignment: _mainAxisAlignment,
      crossAxisAlignment: _crossAxisAlignment,
      children: const [
        BoxtView(
          index: 1,
          size: 50,
        ), // BoxtView
        BoxtView(
          index: 2,
          size: 200,
        ), // BoxtView
      ],
    ), // Row
  ), // Container
  const SizedBox(        You, 11 hours ago • feat: ContainerPage
    height: 20,
  ), // SizedBox
  Container(
    decoration: BoxDecoration(
      color: Colors.green,
      borderRadius: BorderRadius.circular(8),
      border: Border.all(
        color: Colors.grey,
        width: 1,
      ), // Border.all
    ), // BoxDecoration
    width: 375,
    height: 400,
    child: Column(
      mainAxisAlignment: _mainAxisAlignment,
      crossAxisAlignment: _crossAxisAlignment,
      children: const [
        BoxtView(
          index: 4,
          size: 50,
        ), // BoxtView
        BoxtView(
          index: 5,
          size: 200,
        ), // BoxtView
      ],
    ), // Column
  ), // Container
  const SizedBox(
    height: 20,
  ), // SizedBox
  Container(
```



状态管理

> Flutter 是declarative(声明式的)，类似RxSwift,Vue等前端响应式的,不需要主动去找到某一个控件然后去修改它，操作的更多的是数据，操作数据即可

Riverpod学习

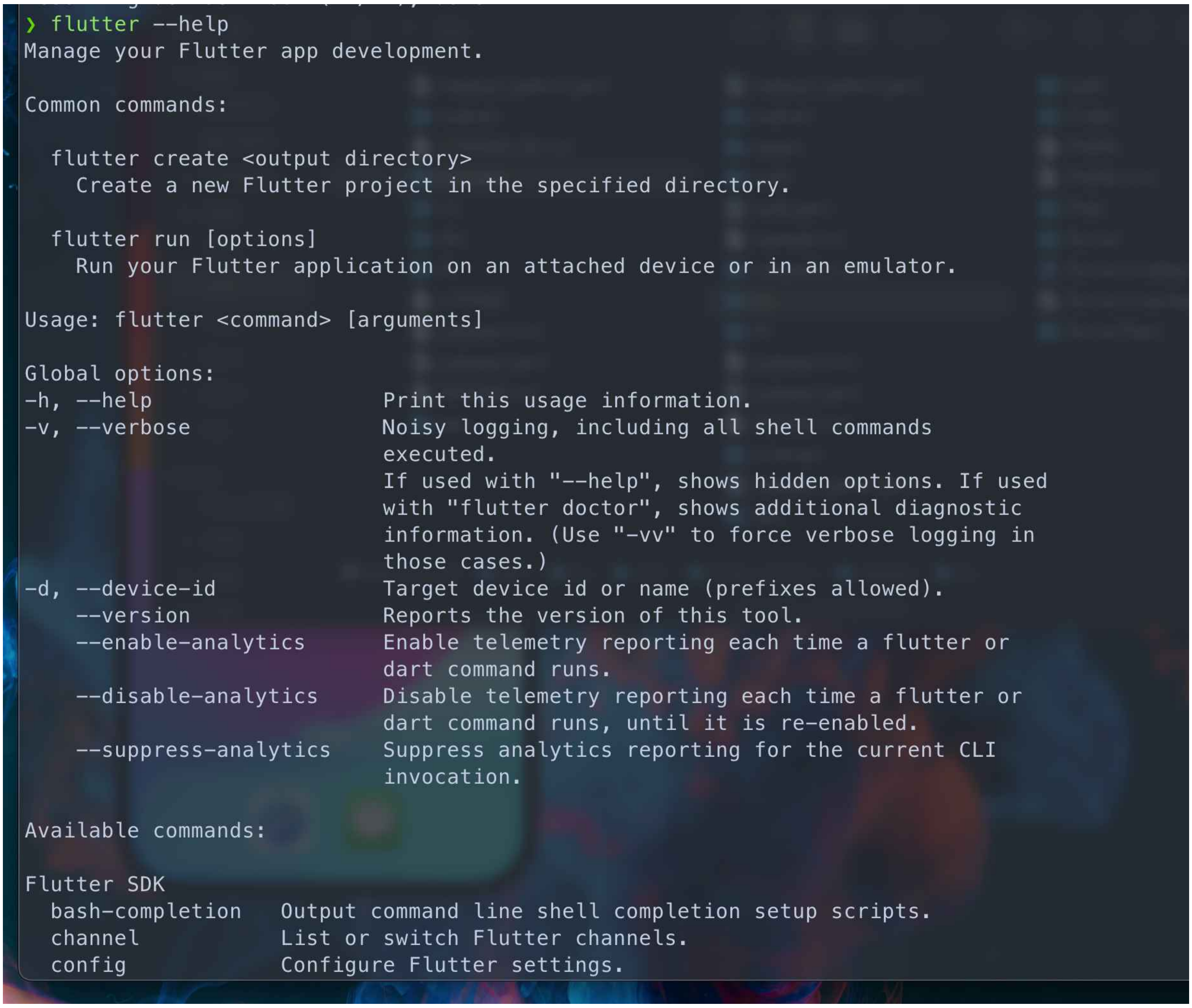
脚本相关

> "flutter -h"就可以了解具体的参数，项目创建、model生成、项目打包、环境依赖检查等都 需要依赖到脚本

1. Flutter相关脚本

帮助脚本

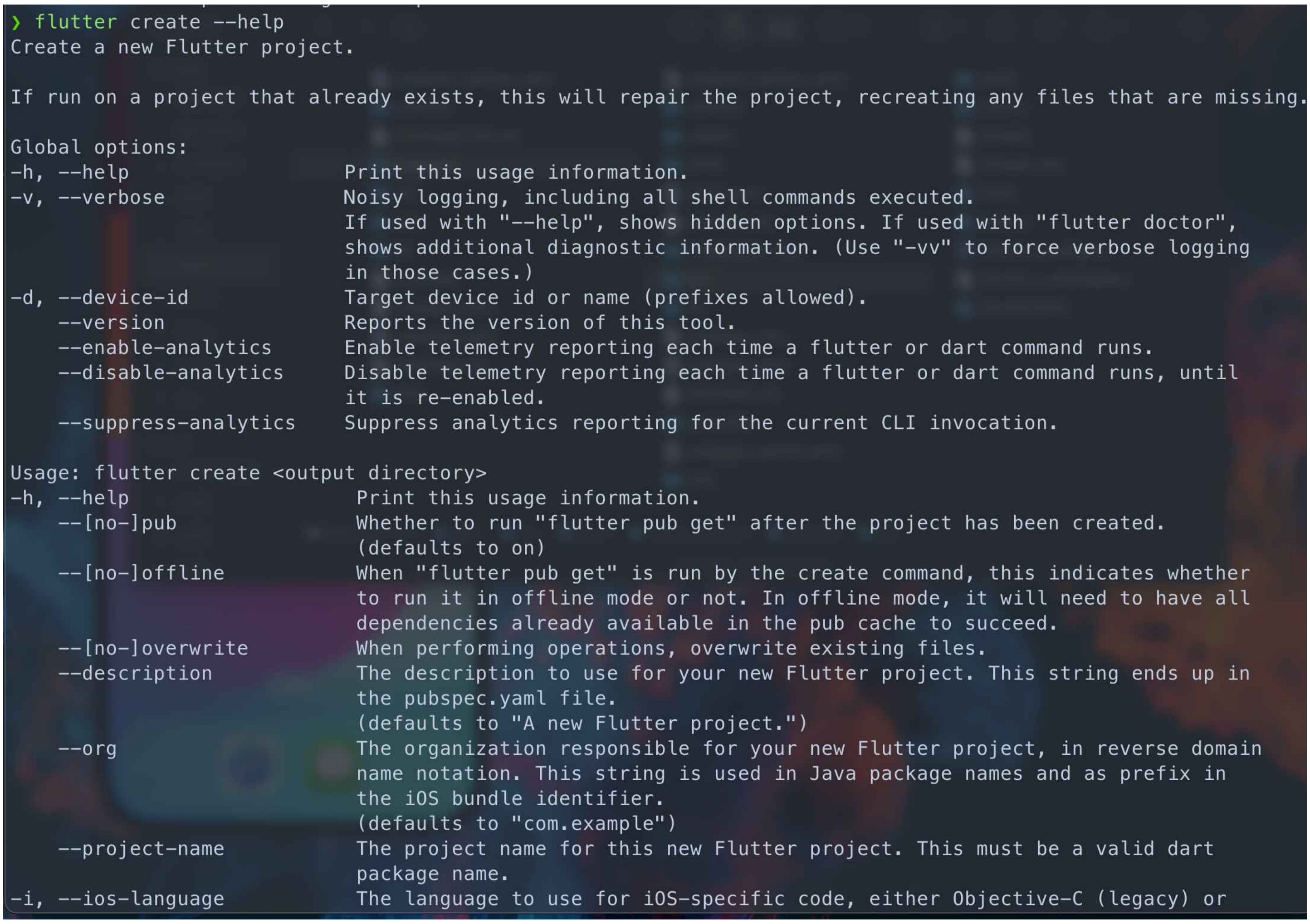
```
flutter --help
```



```
❯ flutter --help
Manage your Flutter app development.

Common commands:

  flutter create <output directory>
    Create a new Flutter project in the specified directory.

  flutter run [options]
    Run your Flutter application on an attached device or in an emulator.

Usage: flutter <command> [arguments]

Global options:
-h, --help                  Print this usage information.
-v, --verbose               Noisy logging, including all shell commands
                            executed.
                            If used with "--help", shows hidden options. If used
                            with "flutter doctor", shows additional diagnostic
                            information. (Use "-vv" to force verbose logging in
                            those cases.)
-d, --device-id             Target device id or name (prefixes allowed).
    --version               Reports the version of this tool.
    --enable-analytics      Enable telemetry reporting each time a flutter or
                            dart command runs.
    --disable-analytics     Disable telemetry reporting each time a flutter or
                            dart command runs, until it is re-enabled.
    --suppress-analytics    Suppress analytics reporting for the current CLI
                            invocation.

Available commands:

Flutter SDK
  bash-completion   Output command line shell completion setup scripts.
  channel           List or switch Flutter channels.
  config            Configure Flutter settings.
```

比如要查看create命令，可以执行

```
flutter create --help
```



```
❯ flutter create --help
Create a new Flutter project.

If run on a project that already exists, this will repair the project, recreating any files that are missing.

Global options:
-h, --help                  Print this usage information.
-v, --verbose               Noisy logging, including all shell commands executed.
                            If used with "--help", shows hidden options. If used with "flutter doctor",
                            shows additional diagnostic information. (Use "-vv" to force verbose logging
                            in those cases.)
-d, --device-id             Target device id or name (prefixes allowed).
    --version               Reports the version of this tool.
    --enable-analytics      Enable telemetry reporting each time a flutter or dart command runs.
    --disable-analytics     Disable telemetry reporting each time a flutter or dart command runs, until
                            it is re-enabled.
    --suppress-analytics    Suppress analytics reporting for the current CLI invocation.

Usage: flutter create <output directory>
-h, --help                   Print this usage information.
    --[no-]pub               Whether to run "flutter pub get" after the project has been created.
                             (defaults to on)
    --[no-]offline           When "flutter pub get" is run by the create command, this indicates whether
                             to run it in offline mode or not. In offline mode, it will need to have all
                             dependencies already available in the pub cache to succeed.
    --[no-]overwrite         When performing operations, overwrite existing files.
    --description            The description to use for your new Flutter project. This string ends up in
                             the pubspec.yaml file.
                             (defaults to "A new Flutter project.")
    --org                    The organization responsible for your new Flutter project, in reverse domain
                             name notation. This string is used in Java package names and as prefix in
                             the iOS bundle identifier.
                             (defaults to "com.example")
    --project-name           The project name for this new Flutter project. This must be a valid dart
                             package name.
-i, --ios-language           The language to use for iOS-specific code, either Objective-C (legacy) or
```

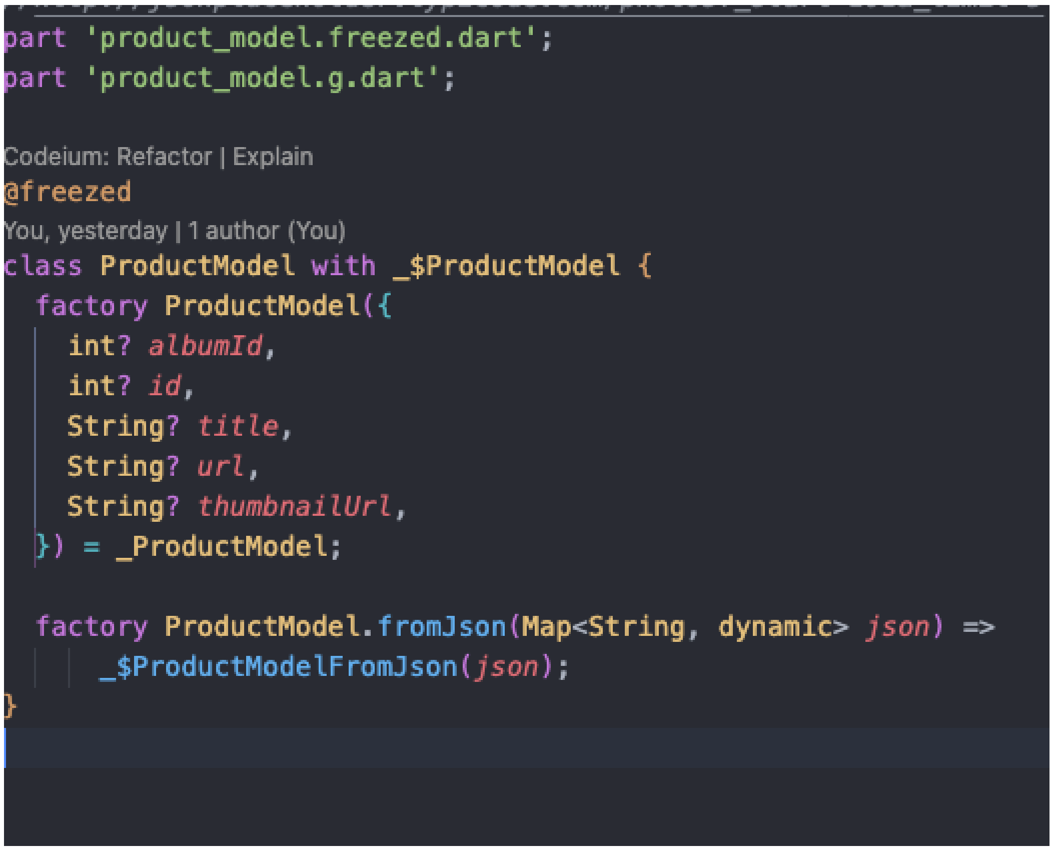
2. build_runner

> Dart语言不支反射，导致很多技巧是不能用的，比如iOS中常用的model **MJExtension** 的解析，一般model的生成都需要依赖build_runner在编译前动态生成代码,一般都是.g.dart文件，当然也可以自定义，比较常用的三方插件有: freezed *json_serializable 等都是借助build_runner 编译前生成对应的解析文件，我们只需要定义对应的属性和类名即可定义一个dart中的model*

查看帮助

```
flutter pub run build_runner -h
```

类似的model如下:



```dart
part 'product_model.freezed.dart';
part 'product_model.g.dart';

Codeium: Refactor | Explain
@freezed
You, yesterday | 1 author (You)
class ProductModel with _$ProductModel {
  factory ProductModel({
    int? albumId,
    int? id,
    String? title,
    String? url,
    String? thumbnailUrl,
  }) = _ProductModel;

  factory ProductModel.fromJson(Map<String, dynamic> json) =>
      _$ProductModelFromJson(json);
}
```

不止是model，很多其他，比如资源管理文件以及多语言文件等很多都是依赖build_runner 生成，只要依赖build_runner 生成的就需要在终端执行命令即可生成:

```
flutter pub run build_runner build --delete-conflicting-outputs
```

--delete-conflicting-outputs表示重复的文件就删除现有的

或者是

```
dart run build_runner build
```

或者watch，只要有修改就会自动生成

```
flutter pub run build_runner watch --delete-conflicting-outputs
```

推荐资料

- https://chat.openai.com/
- https://docs.flutter.dev/get-started/

- *https://docs.flutter.dev/reference/widgets*
- Page 1 | Top packages
- Search results for is:flutter-favorite
- Featured Flutter Tutorials | Code With Andrea
- 张风捷特烈 的个人主页 - 文章 - 掘金
- Flutter工具推荐

Vscode 插件推荐

- https://marketplace.visualstudio.com/items?itemName=Codeium.codeium
- https://marketplace.visualstudio.com/items?itemName=FushiArt.copywidget
- https://marketplace.visualstudio.com/items?itemName=hzgood.dart-data-class-generator
- https://marketplace.visualstudio.com/items?itemName=robert-brunhage.flutter-riverpod-snippets