

Summary

- [缘起](#)
- [起步](#)
 - [移动开发技术简介](#)
 - [Flutter简介](#)
 - [搭建Flutter开发环境](#)
 - [Dart语言简介](#)
- [第一个Flutter应用](#)
 - [计数器示例](#)
 - [路由管理](#)
 - [包管理](#)
 - [资源管理](#)
 - [调试Flutter APP](#)
- [基础Widgets](#)
 - [Widget简介](#)
 - [文本、字体样式](#)
 - [按钮](#)
 - [图片和Icon](#)
 - [单选框和复选框](#)
 - [输入框和表单](#)
- [布局类Widgets](#)
 - [布局类Widgets简介](#)
 - [线性布局Row、Column](#)
 - [弹性布局Flex](#)
 - [流式布局Wrap、Flow](#)
 - [层叠布局Stack、Positioned](#)

- 容器类Widgets
 - Padding
 - 布局限制类容器ConstrainedBox、SizeBox
 - 装饰容器DecoratedBox
 - 变换Transform
 - Container容器
- 可滚动Widgets
 - 可滚动Widgets简介
 - SingleChildScrollView
 - ListView
 - GridView
 - CustomScrollView
 - 滚动监听及控制ScrollController
- 功能型Widgets
 - 导航返回拦截-WillPopScope
 - 数据共享-InheritedWidget
 - 主题-Theme
- 事件处理与通知
 - 原始指针事件处理
 - 手势识别
 - 全局事件总线
 - 通知Notification
- 动画
 - Flutter动画简介
 - 动画结构
 - 自定义路由过渡动画
 - Hero动画
 - 交错动画

- 自定义Widget
 - 自定义Widget方法简介
 - 通过组合现有Widget实现
 - 实例：TurnBox
 - CustomPaint与Canvas
 - 实例：圆形渐变进度条(自绘)
- 文件操作与网络请求
 - 文件操作
 - Http请求-HttpClient
 - Http请求-Dio package
 - WebSocket
 - 使用Socket API
 - Json转Model
- 包与插件
 - 开发package
 - 插件开发：平台通道简介
 - 插件开发：实现Android端API
 - 插件开发：实现IOS端API
 - 系统能力调用
- 国际化
 - 让App支持多语言
 - 实现Localizations
 - 使用Intl包
- Flutter 核心原理
 - Flutter UI系统
 - Widget、Element与BuildContext
 - RenderObject与RenderBox
 - Notification与事件冒泡
 - Scrollable原理

- `InheritedWidget`与数据传递
- `ParentDataWidget`
- 布局与绘制过程
- `Flutter`资源汇总