

2.5	制御構文	39
	分岐	39
	if文	39
	if-case文	39
	switch文	40
	式としてのswitch	41
	ループ	42
	for文	42
	while文	43
	breakとcontinue	43
2.6	パターン	44
	マッチング機能しか持たないパターン	44
	論理演算子、比較演算子	44
	一致判定	45
	マッチングと分解宣言の2つの機能を持つパターン	46
	List	46
	Map	46
	Record	46
	Object	47
	Tips for-in文での分解宣言	48
	パターンを補助する構文	49
	キャスト	49
	nullチェック	49
	nullアサーション	49
	ワイルドカード	50
2.7	例外処理	50
	例外の型 ErrorとException	51
	例外の捕捉	51
	例外の再スロー	52
	finally句	52
	アサーション	52
	Flutterの例外処理	53
2.8	コメント	54
2.9	null安全	55
	null許容演算子	55
	null認識演算子	55
	nullアサーション演算子	55

タイププロモーション	56
------------------	----

その他のほかの便利なnull関連演算子	56
---------------------------	----

2.10 ライブラリと可視性 57

2.11 関数 57

引数	57
----------	----

省略可能引数	58
--------------	----

名前付き引数	58
--------------	----

関数の省略記法	59
---------------	----

第一級関数と匿名関数	59
------------------	----

2.12 クラス 61

ゲッターとセッター	62
-----------------	----

いろいろなコンストラクタ	62
--------------------	----

constantコンストラクタ	63
-----------------------	----

名前付きコンストラクタ	64
-------------------	----

factoryコンストラクタ	64
----------------------	----

クラス継承	65
-------------	----

スーパークラスのコンストラクタ	66
-----------------------	----

暗黙のインタフェース	67
------------------	----

拡張メソッド	68
--------------	----

mixin クラスに機能を追加する	69
-------------------------	----

Enum	70
------------	----

Enumの宣言	70
---------------	----

Enumの利用	71
---------------	----

クラス修飾子	72
--------------	----

abstract	73
----------------	----

base	73
------------	----

interface	75
-----------------	----

abstractとinterfaceの組み合わせ	76
--------------------------------	----

final	76
-------------	----

mixin	77
-------------	----

sealed	77
--------------	----

2.13 非同期処理 78

Future型	78
---------------	----

エラーハンドリング	79
-----------------	----

Stream型	81
---------------	----

Streamの購読をキャンセル、一時停止する	81
------------------------------	----

Stream型を生成する関数	82
----------------------	----

Streamの終わり.....	83
エラーハンドリング.....	84
StreamController.....	85
ブロードキャスト.....	87
Streamを変更する.....	88
Zone 非同期処理のコンテキスト管理.....	88
アイソレート.....	90
Flutterアプリとアイソレート.....	90
2.14 まとめ.....	92

第3章

フレームワークの中心となる
Widgetの実装体験

StatelessWidget、StatefulWidget.....	93
3.1 DartPadでアプリ開発を体験しよう.....	94
3.2 状態を持たないWidget StatelessWidget.....	96
独自のStatelessWidgetを定義する.....	98
3.3 状態を持つWidget StatefulWidget.....	100
Widgetのタップ操作を検知する.....	101
StatefulWidgetを継承する.....	102
Widgetの状態を変化させる.....	104
3.4 まとめ.....	105

第4章

アプリの日本語化対応、
アセット管理、環境変数

アセット管理.....	107
4.1 パッケージやツールを導入する.....	108
パッケージの導入方法.....	109
パッケージバージョンの指定方法.....	110
パッケージバージョンの更新方法.....	111

	Tips セマンティックバージョンングについて	111
4.2	アプリを日本語に対応させる	113
	意図せず英語表示されるケース	114
	フレームワークが提供する表示文字列	114
	日付フォーマット	115
	アプリを日本にローカライズする	117
	フレームワークが提供する表示文字列を日本語化する	117
	日付フォーマットを日本語化する	119
	iOSアプリの対応言語を設定する	121
	メッセージをローカライズする	121
	コードジェネレータを設定する	121
	arbファイルを作成する	122
	ローカライズされたメッセージを表示する	123
	arbファイルの扱い方	125
	プレースホルダ	125
	単数形と複数形の対応	126
	複数の言語への対応	128
4.3	プロジェクトにアセットを追加する	129
	アプリに画像を追加する	129
	端末の解像度に応じて画像を切り替える	131
	flutter_gen 型安全にアセットを扱うパッケージ	133
	flutter_genを導入する	133
	SVG画像の利用	134
	その他のアセット	137
4.4	dart-define-from-file 環境変数を扱う	137
	環境変数をJSON形式で記述する	137
	環境変数をコードから参照する	138
4.5	まとめ	139

第5章

テーマとルーティング

5.1	テーマ アプリ全体のビジュアルを管理	142
	フレームワークによるテーマの自動計算機能	142
	ダークモード対応	146

	アプリ独自のテーマ管理.....	147
	Theme Extensionを利用したアプリのサンプル	150
	Tips Cupertino(クパチーノ)デザイン	152
5.2	ナビゲーションとルーティング 画面遷移を実現する3つの手法	152
	NavigatorウィジェットとRouteクラスによる画面遷移 Navigator 1.0.....	154
	画面間でのデータの受け渡し.....	158
	名前付きルートによる画面遷移 Navigator 1.0(非推奨)	161
	名前付きルートの制限事項	166
	Routerウィジェットによる画面遷移 Navigator 2.0	167
	go_routerによる画面スタックの書き換えを体験する.....	167
	GoRouteで入れ子構造を作る.....	170
	goとpushの違い	175
5.3	まとめ	180

第6章

実践ハンズオン① 画像編集アプリを開発..... 181

6.1	開発するアプリの概要	182
	スタート画面	182
	画像選択画面	183
	画像編集画面	184
	開発の土台づくり.....	185
	テーマと画面遷移の方針	185
6.2	プロジェクトを作成する	185
	開発の土台づくり.....	186
6.3	アプリ起動後のスタート画面を作成する	186
	レイアウトを作成する	188
6.4	テーマをアレンジする	190
6.5	アプリを日本語化する	190
	パッケージを導入する	190
	ローカライズの構成ファイルを作成する	191
	arbファイルを作成する	191
	ローカライズされたメッセージを適用する	192

	App Storeでの表示言語を設定する.....	194
6.6	画像選択画面を作成する	195
	メッセージを追加する.....	195
	レイアウトを作成する.....	196
	画像ライブラリから画像を取得する.....	198
	パッケージを導入する.....	198
	iOSネイティブの設定を行う.....	199
	画像を取得する処理を実装する.....	199
	画像取得処理をWidgetに組み込む.....	201
6.7	画像編集画面を作成する	203
	メッセージを追加する.....	203
	レイアウトを作成する.....	204
	画像編集画面への遷移を実装する.....	205
	アイコンを追加する flutter_genによるアセット管理.....	207
	画像を編集する処理を実装する.....	210
6.8	まとめ	212
	Tips WidgetとStateのライフサイクルについて.....	213

第 7 章

状態管理とRiverpod..... 215

7.1	Flutterアプリにおける状態管理	216
7.2	Riverpodとはどのようなパッケージか	217
	Tips 宣言的UIとしてのFlutter.....	217
	Riverpodの主要なクラス.....	218
	実装サンプル.....	218
7.3	Riverpodの関連パッケージ	221
	基本機能を提供するパッケージ.....	221
	Providerのコードを生成するパッケージ.....	222
	静的解析を行うパッケージ.....	223
	関連パッケージまとめ.....	223
7.4	Riverpodの使い方	224

Providerの種類	224
関数ベースのProvider	224
クラスベースのProvider	225
非同期処理を行うProvider	227
非同期なProviderとRaw型	229
Providerから値を取得する	230
Providerの値をフィルタする	231
Providerのライフサイクル	231
Providerにパラメータを渡す	232
7.5 まとめ	234

第8章

実践ハンズオン② ひらがな変換アプリを開発

8.1 開発するアプリの概要	236
入力状態	237
レスポンス待ち状態	237
変換完了状態	238
開発の土台づくり	239
テーマと画面遷移の方針	239
8.2 プロジェクトを作成する	239
8.3 アプリで使用するパッケージを導入する	240
riverpod_lintを設定する	241
8.4 入力状態のウィジェットを実装する	241
レイアウトを作成する	241
レイアウトを表示する	243
入力値のバリデーションを行う	244
8.5 入力文字を取得する	246
8.6 ひらがな化するWeb APIを呼び出す実装をする	248
リクエスト、レスポンスオブジェクトを定義する	249
アプリケーションIDを設定する	251
Web APIを呼び出す	251

8.7	アプリの状態を管理する	254
	状態を表現するクラスを作成する.....	254
8.8	状態に応じて表示を切り替える	256
	レスポンス待ち状態のウィジェットを実装する	256
	変換完了状態のウィジェットを実装する.....	257
	画面の切り替えを行う	258
	ひらがな変換処理の呼び出しを修正する.....	259
8.9	まとめ	262

第9章

フレームワークによる パフォーマンスの最適化

BuildContext、Key	263
------------------------	-----

9.1	BuildContextは何者なのか Element	264
	祖先の情報にアクセスできるBuildContext.....	264
	Elementがツリーを構成していく工程.....	265
	StatefulWidgetの状態を保持する役割.....	268
	Tips 宣言的UIとElementの再利用	273
9.2	Elementの再利用とパフォーマンス RenderObject	273
	RenderObjectは高コストな計算を行う	274
	RenderObjectは状態を持つ.....	274
	Elementの再利用はパフォーマンスに影響する.....	275
9.3	Keyは何に使うのか	275
	Elementが再利用される条件	276
	Elementが再利用される様子を見てみよう	276
	Keyを利用したElementの再利用	279
	Keyの種類	281
9.4	局所的にWidgetを更新するしくみ InheritedWidget	282
9.5	まとめ	284