

サポート情報

たった2日でマスターするiPhoneアプリ開発集中講座
Xcode 12 Swift 5.3 対応

本書の出版時は、Xcode12.3 での動作確認を行っていますが、Xcode・Swift・iOS のバージョンアップに伴い、本書の記載内容やサンプルアプリに変更が必要な場合があります。

変更が必要な箇所を、このサポート情報にてご連絡いたします。

ご使用されている **Xcode** のバージョンをご確認の上で、このサポート情報を利用してください。

また、正誤も合わせて記載させていただきます。

■変更内容:

なし

■正誤表:

本書において下記のとおり、誤りがございました。

内容を訂正すると共に、読者の皆様にご迷惑をおかけしたことを、深くお詫び申し上げます。

恐れ入りますが、本正誤表をご確認の上、ご利用いただきますようお願い申し上げます。

赤字が修正箇所になります。

P43

【誤】	実際の利用方法は、 サンプルアプリ の開発を通して繰り返し操作して慣れていきますので、今は、こんな画面の区切りがあるという認識をしていただければ大丈夫です。
【正】	実際の利用方法は、 サンプルアプリ の開発を通して繰り返し操作して慣れていきますので、今は、こんな画面の区切りがあるという認識をしていただければ大丈夫です。

P185

【誤】	図「ギターを鳴らすコード」で、39行目、40行目のコードに誤りがあります。 self.guitarPlayer = try AVAudioPlayer(data: self.guitarData) self.guitarPlayer.play()
【正】	selfは省略できるため、不要です。 guitarPlayer = try AVAudioPlayer(data: guitarData) guitarPlayer.play()

P264

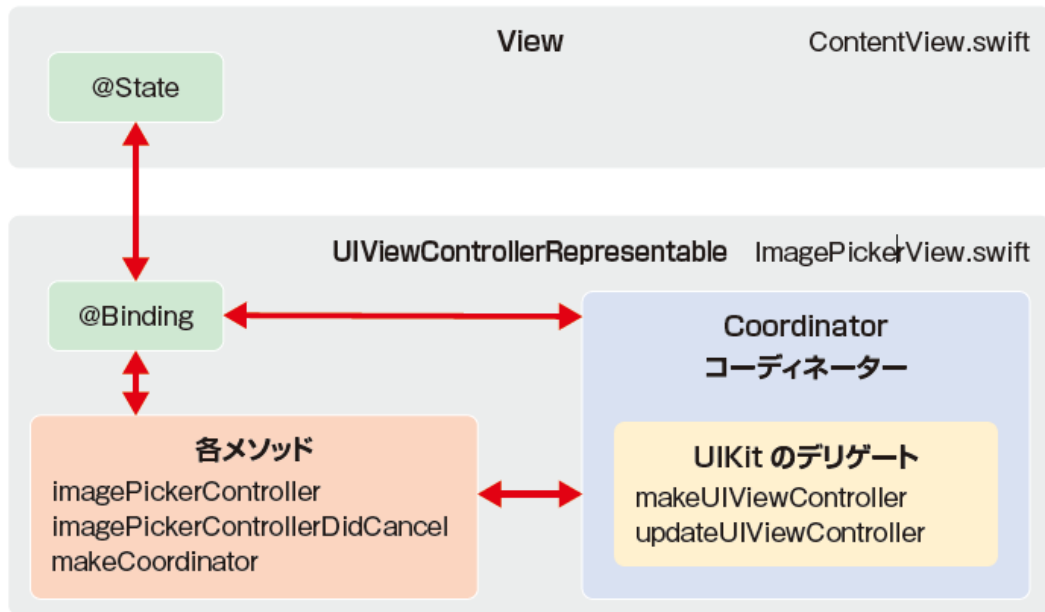
【誤】	図「画面遷移を実装」で、15行目のコメント「// VSaack ここまで」に誤りがあります。
【正】	「// VSaack ここまで」ではなく「// VStack ここまで」が正しい

P311

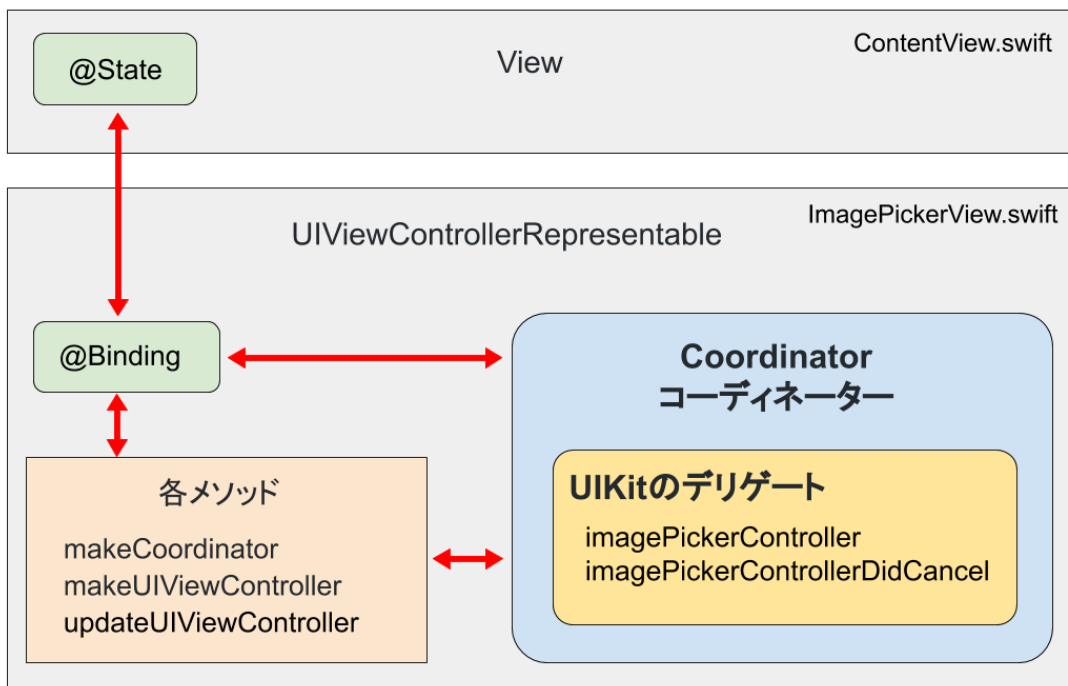
【誤】

図「Coordinator(コーディネーター)の役割」で、makeUIViewController、updateUIViewControllerの配置に誤りがあります。

▼ Coordinator (コーディネーター) の役割



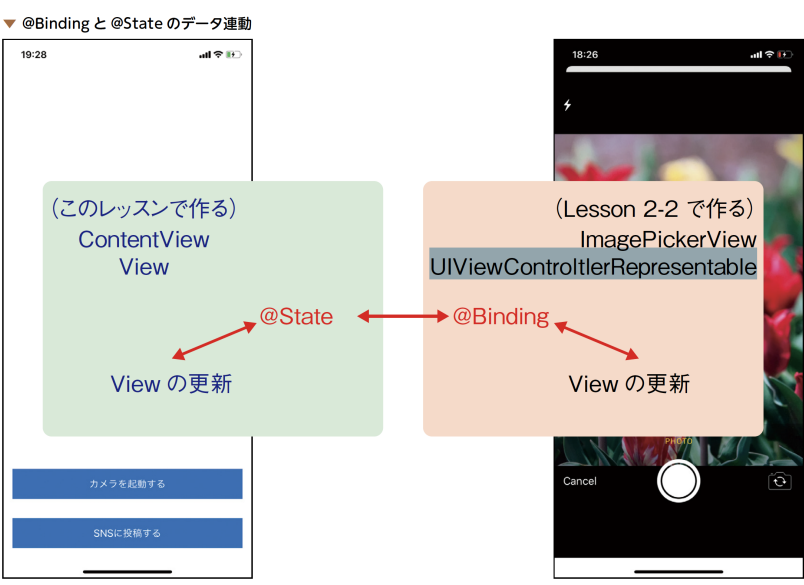
【正】



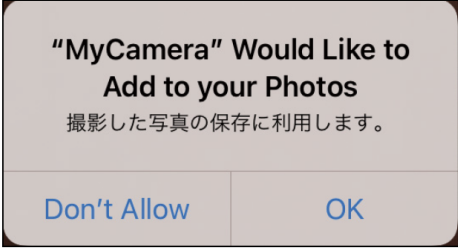
P311

【誤】	「let parent」の解説に誤りがあります。 <pre>// ImagePickerView型の変数を用意 let parent: ImagePickerView</pre> Coordinator クラスのインスタンス生成（最初に使用する）のときに、Coordinator を利用する「親」を指定します。Coordinator の中で「親」の値を直接に編集できるようにすることで利用しやすくします。ここでの「親」は「struct ImagePickerView {}」のことです。 そのため、「親」を格納する変数 parent には、構造体 ImagePickerView のデータ型を指定します。
【正】	「let parent」は変数ではなく、定数が正しいです。 <pre>// ImagePickerView型の変数を用意 let parent: ImagePickerView</pre> Coordinator クラスのインスタンス生成（最初に使用する）のときに、Coordinator を利用する「親」を指定します。Coordinator の中で「親」の値を直接に編集できるようにすることで利用しやすくします。ここでの「親」は「struct ImagePickerView {}」のことです。 そのため、「親」を格納する変数^{定数} parent には、構造体 ImagePickerView のデータ型を指定します。

P332

【誤】	図「@Binding と@Stateのデータ連動」で、UIViewControllerRepresentableの記述に誤りがあります。 
【正】	UIViewControllerRepresentableではなくUIViewControllerRepresentableが正しい

P336

【誤】	図「カメラ使用許可の確認」のタイトルに誤りがあります。 許可してもらう する場合には、ア 許可を得る必要があ するのかを利用者 の許可を得るた ペティリストで設 ▼ カメラ使用許可の確認 
【正】	「カメラ使用許可の確認」ではなく「フォトライブラリー使用許可の確認」が正しい

P363

【誤】	ActionSheetの「self.isPhotolibrary」での「self」は不要です。 actionSheet モディファイアは、状態変数 isShowAction が「true」になったときに実行します。 <pre> ActionSheet(title: Text("確認"), message: Text("選択してください"), buttons: [.default(Text("カメラ"), action: { // カメラを選択 self.isPhotolibrary = false }), .default(Text("フォトライブラリー"), action: { // フォトライブラリーを選択 self.isPhotolibrary = true }), // キャンセル .cancel(),]) // ActionSheet ここまで </pre> ActionSheet 構造体は、表示するタイトル、メッセージ、ボタンメニューを定義します。ここでは「カ
【正】	「self.isPhotolibrary = false」ではなく「isPhotolibrary = false」が正しい 「self.isPhotolibrary = true」ではなく「isPhotolibrary = true」が正しい

P370

【誤】	図「状態変数の追加」の20行目のコードに誤りがあります。 // 表示有無を管理する状態変数 @State var isShowActivity = false
【正】	@State var isShowActivity = false

P378

【誤】	表「Core Image でよく使用するクラス」のCImageクラスの解説に誤りがあります。 画像に関連するデータを管理するためのクラス。Core Image の各クラス(CIFilter、CIContext など)では、CImage クラスで生成されたCImage オブジェクトを使用
【正】	「CImage オブジェクト」ではなく「CImage オブジェクト」が正しい

P401

【誤】	Pointのコードコメント解説に誤りがあります。 a += 1 // a = a + 1 1を加算 a -= 1 // a = a - 1 1を減算 a *= 1 // a = a * 1 1を乗算 a /= 1 // a = a / 1 1を除算
【正】	a /= 1 // a = a / 1 1で除算

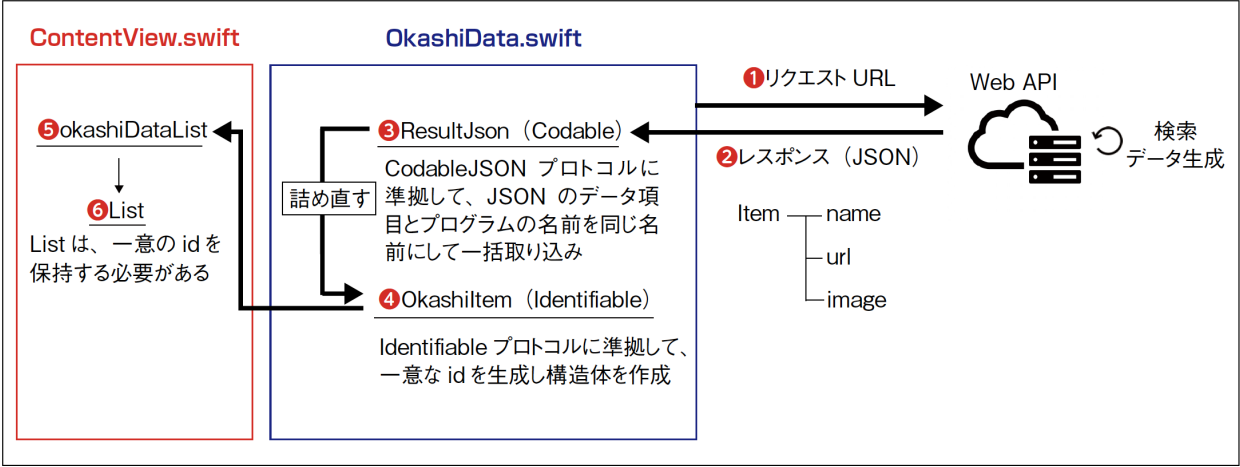
P437

【誤】	本文の「データ取得が完了したタイミンで、セッションを終了します。」に誤りがあります。 <pre>// セッションを終了 session.finishTasksAndInvalidate()</pre> データ取得が完了したタイミンで、セッションを終了します。 <pre>// JSONDecoderのインスタンス取得 let decoder = JSONDecoder()</pre>
【正】	データ取得が完了したタイミングで、セッションを終了します。

P437

【誤】	「変数json」の解説に誤りがあります。 decoder.decode で、取得した JSON データ（data!）をパースして構造体 ResultJson のデータ構造に合わせて、変数 json に格納します。 この 1 行で取得したデータが、変数 json に格納されます。この仕組みは前述の Codable を活用しています。
【正】	「変数json」ではなく「定数json」が正しい。 decoder.decode で、取得した JSON データ（data!）をパースして構造体 ResultJson のデータ構造に合わせて、定数 json に格納します。 この 1 行で取得したデータが、定数 json に格納されます。この仕組みは前述の Codable を活用しています。

P443

【誤】	図「データ変換ロジックのイメージ」で、「CodableJSONプロトコル」の記述に誤りがあります。 ▼ データ変換ロジックのイメージ  The diagram illustrates the data transformation logic. On the left, a box for ContentView.swift contains 5 okashiDataList and 6 List. A note states: 'List は、一意の id を保持する必要がある' (List must maintain a unique id). An arrow points from okashiDataList to List. In the center, a box for OkashiData.swift contains 3 ResultJson (Codable) and 4 OkashiItem (Identifiable). A note for ResultJson says: 'CodableJSON プロトコルに準拠して、JSON のデータ項目とプログラムの名前を同じ名前にして一括取り込み' (Compliant with CodableJSON protocol, batch import with same names as JSON data items and program names). A note for OkashiItem says: 'Identifiable プロトコルに準拠して、一意な id を生成し構造体を作成' (Compliant with Identifiable protocol, generate unique id and create struct). An arrow labeled '詰め直す' (repackage) points from OkashiItem to List. On the right, a Web API cloud icon is shown. An arrow labeled 1 リクエスト URL points from OkashiData.swift to the Web API. An arrow labeled 2 レスポンス (JSON) points from the Web API back to OkashiData.swift. A circular arrow around the Web API is labeled '検索 データ生成' (search data generation). A tree diagram shows an Item with name, url, and image properties.
【正】	「CodableJSONプロトコル」ではなく「CodableJSONプロトコル」が正しい

以上