

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 14 / iOS 16 / Swift 5.7 対応

本書の出版時は、Xcode14.0 での動作確認を行っています。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ときがあります。

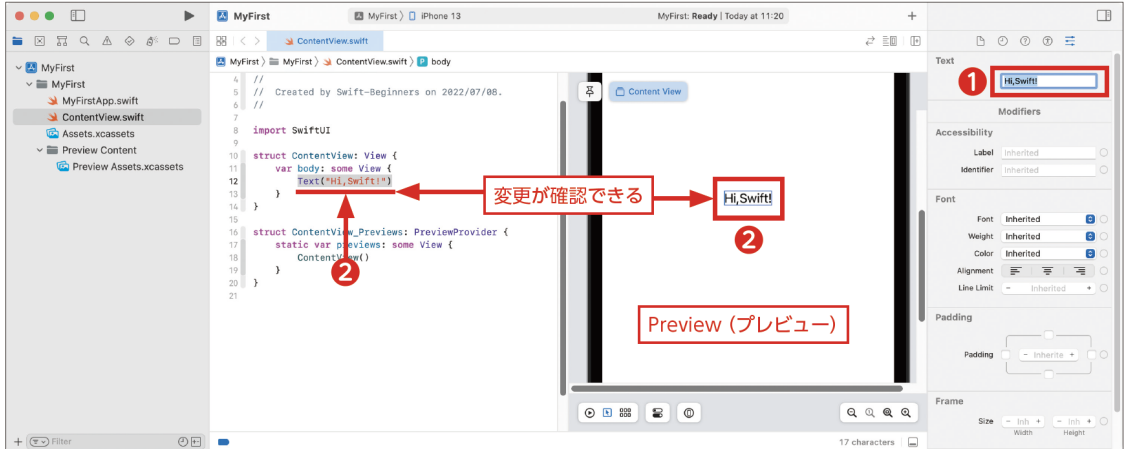
変更が必要な箇所を、このサポート情報にてご連絡いたします。

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。

■既知の不具合内容:

XcodeやSwiftにて不具合・バグとして認識されている箇所です。

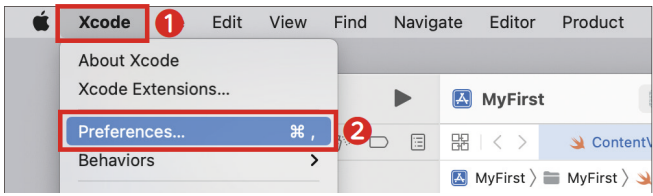
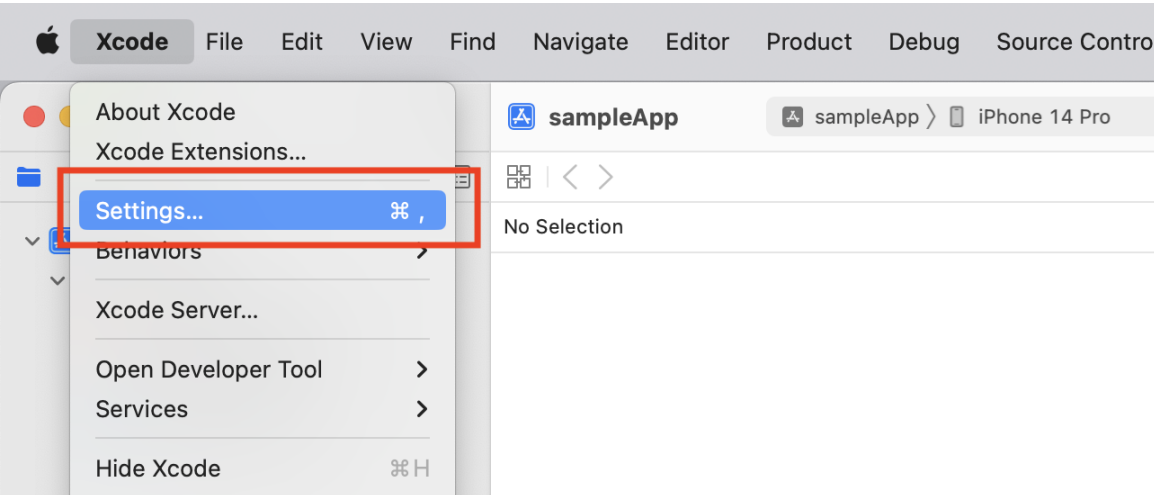
P.66: 図「[Text]の文字列を変更」にてテキストが変更されない

<p>本書での解説</p>	<p>▼ [Text] の文字列を変更</p>  <ol style="list-style-type: none"> 1 [Attributes inspector] の [Text] の文字を「Hi, Swift!」に変更しキーボードの [return] を実行してください。 2 コードの文字列が「Hello, World!」から「Hi, Swift!」に自動で変更され、Preview (プレビュー) にも反映されます。コードの文字を修正しても同様で、[Text] の文字と Preview の表示が「Hi, Swift!」に切り替わります。
<p>不具合の箇所</p>	<p>本来であれば、①の箇所でテキストを入力すると、プレビューやコードに反映される仕様ですが、現状は反映されません。テキストを修正する際には、②のコードを修正してください。</p>

■変更内容:

執筆時点では問題がありませんでしたが、XcodeやmacOSのバージョンアップやApple社の影響に伴い変更になった箇所です。ご自身のバージョンを確認した上で、次の変更箇所を参考にしてください。

本書全般:「macOS Ventura」にて[Preferences]が[Settings]へ表記がかわりました。

<p>macOS Monterey</p>	<h3>1 Xcode の環境を設定しよう</h3> <p>Xcode の [Preferences] (プリファレンス: 環境設定) を、自分好みに設定することでより使いやすくなります。</p> <p>環境設定以降は、再度設定するまでその設定が継続されるので、最初に Xcode の環境設定をしましょう。</p> <p>ここでは、行数番号の表示設定、フォント & カラーの設定、コード折りたたみ機能の設定、Minimap (ミニマップ) の設定方法を説明します。</p> <ol style="list-style-type: none">1 アプリケーションメニューから [Xcode] を選択します。2 [Preferences] をクリックします。 <p>[Preferences] (環境設定) 画面が表示されます。ここでは、Xcode をより使</p> <p>▼ プロジェクトテンプレートの選択</p> 
<p>macOS Ventura</p>	<p>macOS Venturaでは、[Settings]に変更されています。</p> 

P.343: Initializer for conditional binding must have Optional type, not 'Image' のエラーが表示される

<p>Xcode14.2まで</p>	<p>Xcode14.2まででは、P343のコードは正常に動作します。</p> <p>ContentViewに、以下の赤枠のコードを追加しましょう。</p> <div data-bbox="609 353 1209 878"><p>▼ ShareLink でシェア機能の作成</p><pre>62 // UIImagePickerController (写真撮影) を表示 63 UIImagePickerController(isShowSheet: \$isShowSheet, captureImage: \$captureImage) 64 } // 「カメラを起動する」ボタンのsheetここまで 65 66 // captureImageをアンラップする 67 if let captureImage, 68 // captureImageから共有する画像を生成する 69 let shareImage = Image(uiImage: captureImage) { 70 // 共有シート 71 ShareLink(item: shareImage, subject: nil, message: nil, 72 preview: SharePreview("Photo", image: shareImage)) { 73 // テキスト表示 74 Text("SNSに投稿する") 75 // 横幅いっぱい 76 .frame(maxWidth: .infinity) 77 // 高さ50ポイント指定 78 .frame(height: 50) 79 // 背景を青色に指定 80 .background(Color.blue) 81 // 文字色を白色に指定 82 .foregroundColor(Color.white) 83 // 上下左右に余白を追加 84 .padding() 85 } // ShareLinkここまで 86 } // アンラップここまで 87 } // VStackここまで 88 } // bodyここまで 89 } // ContentViewここまで</pre><p style="text-align: right;">追加</p></div> <p style="text-align: right;">343</p>
<p>Xcode14.3から</p>	<p>Xcode14.3にバージョンアップすると、if let の箇所で「Initializer for conditional binding must have Optional type, not 'Image' 」というエラーが表示され動作しません。エラーへの対応として、以下のように修正していただくと、エラーが解消され正常に動作します。</p> <div data-bbox="370 1131 1476 2004"><pre>64 } // 「カメラを起動する」ボタンのsheetここまで 65 66 // captureImageをアンラップする 67 if let captureImage { 68 // captureImageから共有する画像を生成する 69 let shareImage = Image(uiImage: captureImage) 70 // 共有シート 71 ShareLink(item: shareImage, subject: nil, message: nil, 72 preview: SharePreview("Photo", image: shareImage)) { 73 // テキスト表示 74 Text("SNSに投稿する") 75 // 横幅いっぱい 76 .frame(maxWidth: .infinity) 77 // 高さ50ポイント指定 78 .frame(height: 50) 79 // 背景を青色に指定 80 .background(Color.blue) 81 // 文字色を白色に指定 82 .foregroundColor(Color.white) 83 // 上下左右に余白を追加 84 .padding() 85 } // ShareLinkここまで 86 } // アンラップここまで 87 } // VStackここまで</pre><p style="text-align: right;">変更</p></div>

P.363: Initializer for conditional binding must have Optional type, not 'Image' のエラーが表示される

<p>Xcode14.2まで</p>	<p>Xcode14.2まででは、P363のコードは正常に動作します。</p> <p>「シェア」ボタンの作成です。 赤枠のコードを追加しましょう。 追加する場所に注意してください。 先程、追加した「エフェクト」ボタンの下に追加します。 「エフェクト」ボタンと同じく、追加するコードは、これまでのレッスンで解説済みです。ここでのコード解説は割愛します。</p> <p>▼ 「シェア」ボタンの作成</p> <pre>51 } // 「エフェクト」ボタンここまで 52 // 上下左右に余白を追加 53 .padding() 54 55 // showImageをアンラップする 56 if let showImage, 57 // captureImageから共有する画像を生成する 58 let shareImage = Image(uiImage: showImage) { 59 // 共有シート 60 ShareLink(item: shareImage, subject: nil, message: nil, 61 preview: SharePreview("Photo", image: shareImage)) { 62 63 // テキスト表示 64 Text("シェア") 65 // 横幅いっぱい 66 .frame(maxWidth: .infinity) 67 // 高さ50ポイント指定 68 .frame(height: 50) 69 // 背景を青色に指定 70 .background(Color.blue) 71 // 文字色を白色に指定 72 .foregroundColor(Color.white) 73 } // ShareLinkここまで 74 // 上下左右に余白を追加 75 .padding() 76 } // アンラップここまで 77 } // VStackここまで 78 }</pre> <p style="text-align: right;">追加</p>
<p>Xcode14.3から</p>	<p>Xcode14.3にバージョンアップすると、if let の箇所で「Initializer for conditional binding must have Optional type, not 'Image'」というエラーが表示され動作しません。 エラーへの対応として、以下のように修正していただくと、エラーが解消され正常に動作します。</p> <pre>52 // 上下左右に余白を追加 53 .padding() 54 55 // showImageをアンラップする 56 if let showImage { 57 // captureImageから共有する画像を生成する 58 let shareImage = Image(uiImage: showImage) 59 // 共有シート 60 ShareLink(item: shareImage, subject: nil, message: nil, 61 preview: SharePreview("Photo", image: shareImage)) { 62 63 // テキスト表示 64 Text("シェア") 65 // 横幅いっぱい 66 .frame(maxWidth: .infinity) 67 // 高さ50ポイント指定 68 .frame(height: 50) 69 // 背景を青色に指定 70 .background(Color.blue) 71 // 文字色を白色に指定</pre> <p style="text-align: right;">変更</p>

■正誤表:

本書において下記のとおり、誤りがございました。

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。

P.135: 表「Swiftの基本的なデータ型」にて説明に赤字に誤りがあります。

【誤】	▼ Swift の基本的なデータ型		
	種類	データ型	説明
	整数型	Int 型	整数を扱う場合は Int 型を利用。
	整数型	UInt 型	符号のない整数を扱う場合は UInt 型を利用。ただし、特別な理由のある場合を除いては、Int 型の利用を推奨。
	実数型	Float 型	整数に小数値を加えた数を実数と呼ぶ。または、浮動小数点とも呼ばれる。最大有効桁数 7 桁。
	実数型	Double 型	Double 型にも実数を格納できる。最大有効桁数 16 桁。 <u>Float 型と Double 型は実装を格納できますが</u> 、特別な理由のある場合を除いては、Double 型の利用を推奨。
	文字型	String 型	文字列を扱う場合は、String 型を利用。
	論理型	Bool 型	条件の真偽を表すデータを扱う場合は Bool 型を利用。true (真)、false (偽)
	Float 型とDouble 型は 実装 を格納できますが、		
【正】	Float 型とDouble 型は 実数 を格納できますが、		

P.380: 下から7行目の「captureImage!」の「!」は不要です。

【誤】	<p>選択画面では、captureImage は状態変数で宣言されています。エフェクト編集画面の引数 captureImage には「<u>captureImage!</u>」と状態変数に「\$」をつけずにセットしています。状態変数の参照ではなく、写真画像データそのものをエフェクト編集画面へ渡していることを理解してください。</p>
	<p>2 写真を取得後にエフェクト編集画面へ遷移できるようにしよう</p> <p>撮影画面（フォトライブラリー画面）とエフェクト編集画面とも「カメラを起動する」ボタンの sheet エフェクト編集画面の引数 captureImage1 には「captureImage!」と状態変数に「\$」をつけずにセットしています。</p>
【正】	エフェクト編集画面の引数 captureImage1 には「 captureImage 」と状態変数に「\$」をつけずにセットしています。