

Inhalt

Vorwort	XI
1 Über SwiftUI	1
1.1 Programmierung mit SwiftUI	3
1.2 Die Preview	5
1.3 Voraussetzungen	8
1.4 Integration	9
2 Grundlagen	11
2.1 Das View-Protokoll	11
2.1.1 Ablauf der View-Generierung	14
2.1.2 Structure vs. Klasse	14
2.2 Grundlagen der View-Erstellung	15
2.2.1 Text und Image	15
2.2.2 Views organisieren mittels Stacks	16
2.2.3 Views mittels Modifier anpassen	19
2.2.3.1 Funktionsweise von Modifiern	22
2.2.3.2 Auszug verfügbarer Modifier	26
2.2.4 Einsatz von Library und Preview	28
2.2.5 Layout-System	30
2.3 Status	31
2.3.1 Property	32
2.3.2 State	33
2.3.3 Binding	34
3 Views, Controls und Container	39
3.1 Text und Grafiken	39
3.1.1 Text	39
3.1.2 TextField	42
3.1.3 SecureField	45
3.1.4 TextEditor	46
3.1.5 Image	47

3.2	Buttons	53
3.2.1	Button	53
3.2.2	EditButton	57
3.2.3	PasteButton	57
3.2.4	Menu	58
3.2.5	Weitere Buttons	60
3.3	Value Selectors	61
3.3.1	Toggle	61
3.3.2	Picker	66
3.3.3	DatePicker	70
3.3.4	Slider	74
3.3.5	Stepper	78
3.4	Value Indicators	83
3.4.1	ProgressView	83
3.4.2	Label	85
3.4.3	Link	86
3.5	Stacks	87
3.5.1	HStack	88
3.5.2	VStack	92
3.5.3	ZStack	96
3.5.4	LazyHStack und LazyVStack	98
3.6	Grids	99
3.7	Listen und Scroll-Views	103
3.7.1	List	103
3.7.2	ForEach	112
3.7.3	ScrollView	125
3.8	Container-Views	128
3.8.1	Form	128
3.8.2	Group	130
3.8.3	GroupBox	134
3.8.4	Section	136
3.9	Weitere Views	139
3.9.1	Spacer	139
3.9.2	Divider	142
4	Navigation und Präsentation	143
4.1	NavigationView	143
4.1.1	Grundlagen	143
4.1.2	Festlegen einer Standardansicht für die Ziel-View	146
4.1.3	Ändern des NavigationView-Styles	148
4.1.4	Setzen eines NavigationView-Titels	150
4.1.5	Navigation-Bar ausblenden	152
4.1.6	Setzen von Navigation-Bar-Items	153

4.1.7	Alternatives Auslösen eines NavigationLink	155
4.1.8	Navigationsstrukturen unter watchOS	157
4.2	TabView	158
4.2.1	Grundlagen	159
4.2.2	Programmatisches Wechseln eines Tab-Bar-Items	162
4.3	HSplitView und VSplitView	164
4.4	Sheet	165
4.4.1	Sheet auf Basis eines Boolean	165
4.4.2	Sheet auf Basis eines Identifiable-Items	166
4.4.3	Reaktion auf Ausblenden eines Sheets	168
4.5	Alert	169
4.5.1	Erstellen eines Alert	169
4.5.2	Einblenden eines Alert auf Basis eines Boolean	171
4.5.3	Einblenden eines Alert auf Basis eines Identifiable-Items	172
4.6	ActionSheet	174
4.6.1	Erstellen eines ActionSheet	175
4.6.2	Einblenden eines ActionSheet auf Basis eines Boolean	177
4.6.3	Einblenden eines ActionSheet auf Basis eines Identifiable-Items ...	177
5	Status	179
5.1	Property	180
5.2	State	182
5.3	Binding	183
5.4	ObservedObject	186
5.4.1	Datenmodell vorbereiten	187
5.4.2	Datenmodell in SwiftUI-View einbinden	187
5.4.3	Auf Änderungen reagieren	190
5.5	StateObject	193
5.6	EnvironmentObject	194
5.7	Environment	200
5.8	Zusammenfassung: Welcher Status für welche Situation?	202
6	Integration	205
6.1	Hosting	205
6.1.1	NSHostingController und UIHostingController	206
6.1.2	WKHostingController	207
6.2	Representables	208
6.2.1	Erstellen einer Representable-View	210
6.2.2	Aktualisieren einer Representable-View	213
6.2.3	Weitergabe von Aktualisierungen an SwiftUI	215

7	Preview und Xcode	221
7.1	Funktionsweise der Preview	222
7.2	Arbeiten mit der Preview	224
7.3	Konfiguration der Preview	227
7.4	Mehrere Previews parallel einsetzen	228
7.5	Preview ausführen	231
7.6	Preview auf Device ausführen	233
7.7	Library	234
7.8	Kontext-Actions	239
	Nachwort	243
	Index	245