

# Inhalt

<b>1</b>	<b>Die Programmiersprache Swift</b>	<b>1</b>
1.1	Die Geschichte von Swift	2
1.2	Swift-Updates	3
1.3	Voraussetzungen zur Nutzung von Swift	4
1.4	Installation von Swift	5
1.4.1	macOS	6
1.4.2	Linux	7
1.5	Xcode	10
1.5.1	Erstellen von Dateien und Projekten	11
1.5.2	Der Aufbau von Xcode	14
1.5.3	Einstellungen	20
1.6	Playgrounds	20
1.6.1	Erstellen eines Playgrounds	21
1.6.2	Aufbau eines Playgrounds	23
1.6.3	Pages, Sources und Resources	27
1.6.4	Playground-Formatierungen	29
1.6.5	Swift Playgrounds-App für das iPad	38
1.7	Weitere Code-Editoren zur Arbeit mit Swift	40
1.7.1	Visual Studio Code	40
1.7.2	Syntra Small	42
1.7.3	IBM Swift Sandbox	43
1.8	Swift-Ressourcen und weiterführende Informationen	44
<b>2</b>	<b>Grundlagen der Programmierung</b>	<b>47</b>
2.1	Grundlegendes	47
2.1.1	Swift Standard Library	47
2.1.2	print	49
2.1.3	Befehle und Semikolons	49
2.1.4	Operatoren	50
2.2	Variablen und Konstanten	52
2.2.1	Erstellen von Variablen und Konstanten	52

2.2.2	Variablen und Konstanten in der Konsole ausgeben .....	53
2.2.3	Type Annotation und Type Inference .....	54
2.2.4	Gleichzeitiges Erstellen und Deklarieren mehrerer Variablen und Konstanten .....	55
2.2.5	Namensrichtlinien .....	56
2.3	Kommentare .....	57
<b>3</b>	<b>Schleifen und Abfragen .....</b>	<b>59</b>
3.1	Schleifen .....	59
3.1.1	for-in .....	59
3.1.2	while .....	61
3.1.3	repeat-while .....	62
3.2	Abfragen .....	63
3.2.1	if .....	63
3.2.2	switch .....	67
3.2.3	guard .....	71
3.3	Control Transfer Statements .....	73
3.3.1	Anstoßen eines neuen Schleifendurchlaufs mit continue .....	73
3.3.2	Verlassen der kompletten Schleife mit break .....	73
3.3.3	Weitere Control Transfer Statements .....	74
3.3.4	Labeled Statements .....	74
<b>4</b>	<b>Typen in Swift .....</b>	<b>77</b>
4.1	Integer .....	79
4.2	Fließkommazahlen .....	80
4.3	Bool .....	81
4.4	String .....	81
4.4.1	Erstellen eines Strings .....	81
4.4.2	Zusammenfügen von Strings .....	82
4.4.3	Character auslesen .....	83
4.4.4	Character mittels Index auslesen .....	84
4.4.5	Character entfernen und hinzufügen .....	86
4.4.6	Anzahl der Character zählen .....	87
4.4.7	Präfix und Suffix prüfen .....	87
4.4.8	String Interpolation .....	88
4.5	Array .....	88
4.5.1	Erstellen eines Arrays .....	89
4.5.2	Zusammenfügen von Arrays .....	90
4.5.3	Inhalte eines Arrays leeren .....	91
4.5.4	Prüfen, ob ein Array leer ist .....	91
4.5.5	Anzahl der Elemente eines Arrays zählen .....	92
4.5.6	Zugriff auf die Elemente eines Arrays .....	92
4.5.7	Neue Elemente zu einem Array hinzufügen .....	93
4.5.8	Bestehende Elemente aus einem Array entfernen .....	93

4.5.9	Bestehende Elemente eines Arrays ersetzen .....	94
4.5.10	Alle Elemente eines Arrays auslesen und durchlaufen .....	95
4.6	Set .....	96
4.6.1	Erstellen eines Sets .....	97
4.6.2	Inhalte eines bestehenden Sets leeren .....	98
4.6.3	Prüfen, ob ein Set leer ist .....	98
4.6.4	Anzahl der Elemente eines Sets zählen .....	98
4.6.5	Element zu einem Set hinzufügen .....	99
4.6.6	Element aus einem Set entfernen .....	99
4.6.7	Prüfen, ob ein bestimmtes Element in einem Set vorhanden ist .....	99
4.6.8	Alle Elemente eines Sets auslesen und durchlaufen .....	100
4.6.9	Sets miteinander vergleichen .....	100
4.6.10	Neue Sets aus bestehenden Sets erstellen .....	103
4.7	Dictionary .....	105
4.7.1	Erstellen eines Dictionary .....	105
4.7.2	Prüfen, ob ein Dictionary leer ist .....	106
4.7.3	Anzahl der Schlüssel-Wert-Paare eines Dictionary zählen .....	107
4.7.4	Wert zu einem Schlüssel eines Dictionary auslesen .....	107
4.7.5	Neues Schlüssel-Wert-Paar zu Dictionary hinzufügen .....	108
4.7.6	Bestehendes Schlüssel-Wert-Paar aus Dictionary entfernen .....	108
4.7.7	Bestehendes Schlüssel-Wert-Paar aus Dictionary verändern .....	109
4.7.8	Alle Schlüssel-Wert-Paare eines Dictionary auslesen und durchlaufen ..	109
4.8	Tuple .....	110
4.8.1	Zugriff auf die einzelnen Elemente eines Tuple .....	112
4.8.2	Tuple und switch .....	112
4.9	Optional .....	115
4.9.1	Deklaration eines Optionals .....	116
4.9.2	Zugriff auf den Wert eines Optionals .....	117
4.9.3	Optional Binding .....	119
4.9.4	Implicitly Unwrapped Optional .....	120
4.9.5	Optional Chaining .....	121
4.9.6	Optional Chaining über mehrere Eigenschaften und Funktionen .....	126
4.10	Any und AnyObject .....	130
4.11	Type Alias .....	130
4.12	Value Type versus Reference Type .....	131
4.12.1	Reference Types auf Gleichheit prüfen .....	133
<b>5</b>	<b>Funktionen .....</b>	<b>135</b>
5.1	Funktionen mit Parametern .....	136
5.1.1	Argument Labels und Parameter Names .....	137
5.1.2	Default Value für Parameter .....	140
5.1.3	Variadic Parameter .....	141
5.1.4	In-Out Parameter .....	142
5.2	Funktionen mit Rückgabewert .....	143

5.3	Function Types	145
5.3.1	Funktionen als Variablen und Konstanten	146
5.4	Verschachtelte Funktionen	148
5.5	Closures	148
5.5.1	Closures als Parameter von Funktionen	150
5.5.2	Trailing Closures	153
5.5.3	Autoclosures	154
<b>6</b>	<b>Enumerations, Structures und Classes</b>	<b>157</b>
6.1	Enumerations	157
6.1.1	Enumerations und switch	160
6.1.2	Associated Values	161
6.1.3	Raw Values	163
6.2	Structures	166
6.2.1	Erstellen von Structures und Instanzen	166
6.2.2	Eigenschaften und Funktionen	167
6.3	Classes	172
6.3.1	Erstellen von Klassen und Instanzen	173
6.3.2	Eigenschaften und Funktionen	173
6.4	Enumeration vs. Structure vs. Class	175
6.4.1	Gemeinsamkeiten und Unterschiede	176
6.4.2	Wann nimmt man was?	177
6.5	self	178
<b>7</b>	<b>Eigenschaften und Funktionen von Typen</b>	<b>181</b>
7.1	Properties	181
7.1.1	Stored Property	182
7.1.2	Lazy Stored Property	184
7.1.3	Computed Property	188
7.1.4	Read-Only Computed Property	190
7.1.5	Property Observer	192
7.1.6	Type Property	195
7.2	Globale und lokale Variablen	197
7.3	Methoden	200
7.3.1	Instance Methods	201
7.3.2	Type Methods	203
7.4	Subscripts	204
<b>8</b>	<b>Initialisierung</b>	<b>209</b>
8.1	Aufgabe der Initialisierung	210
8.2	Erstellen eigener Initializer	211
8.3	Initializer Delegation	216
8.3.1	Initializer Delegation bei Value Types	217
8.3.2	Initializer Delegation bei Reference Types	218

8.4	Failable Initializer .....	220
8.5	Required Initializer .....	223
8.6	Deinitialisierung .....	224
<b>9</b>	<b>Vererbung .....</b>	<b>227</b>
9.1	Überschreiben von Eigenschaften und Funktionen einer Klasse .....	230
9.2	Überschreiben von Eigenschaften und Funktionen einer Klasse verhindern .....	233
9.3	Zugriff auf die Superklasse .....	233
9.4	Initialisierung und Vererbung .....	234
9.4.1	Zwei-Phasen-Initialisierung .....	235
9.4.2	Überschreiben von Initializern .....	241
9.4.3	Vererbung von Initializern .....	244
9.4.4	Required Initializer .....	244
<b>10</b>	<b>Speicherverwaltung mit ARC .....</b>	<b>247</b>
10.1	Strong Reference Cycles .....	250
10.1.1	Weak References .....	252
10.1.2	Unowned References .....	255
10.1.3	Weak Reference vs. Unowned Reference .....	257
<b>11</b>	<b>Weiterführende Sprachmerkmale von Swift .....</b>	<b>259</b>
11.1	Nested Types .....	259
11.2	Extensions .....	261
11.2.1	Computed Properties .....	261
11.2.2	Methoden .....	262
11.2.3	Initializer .....	263
11.2.4	Subscripts .....	265
11.2.5	Nested Types .....	265
11.3	Protokolle .....	266
11.3.1	Deklaration von Eigenschaften und Funktionen .....	268
11.3.2	Der Typ eines Protokolls .....	277
11.3.3	Protokolle und Extensions .....	279
11.3.4	Vererbung in Protokollen .....	284
11.3.5	Class-only-Protokolle .....	285
11.3.6	Optionale Eigenschaften und Funktionen .....	286
11.3.7	Protocol Composition .....	289
11.3.8	Delegation .....	290
<b>12</b>	<b>Type Checking und Type Casting .....</b>	<b>295</b>
12.1	Type Checking mit is .....	298
12.2	Type Casting mit as .....	299
<b>13</b>	<b>Error Handling .....</b>	<b>303</b>
13.1	Deklaration und Feuern eines Fehlers .....	303

13.2 Reaktion auf einen Fehler .....	307
13.2.1 Mögliche Fehler mittels do-catch auswerten .....	308
13.2.2 Mögliche Fehler in Optionals umwandeln .....	311
13.2.3 Mögliche Fehler weitergeben .....	311
13.2.4 Mögliche Fehler ignorieren .....	313
<b>14 Generics .....</b>	<b>315</b>
14.1 Generic Functions .....	316
14.2 Generic Types .....	319
14.3 Type Constraints .....	322
14.4 Associated Types .....	322
<b>15 Dateien und Interfaces .....</b>	<b>327</b>
15.1 Modules und Source Files .....	327
15.2 Access Control .....	328
15.2.1 Access Level .....	328
15.2.2 Explizite und implizite Zuweisung eines Access Levels .....	332
15.2.3 Besonderheiten .....	333
<b>16 Cocoa, Objective-C und C .....</b>	<b>339</b>
16.1 Interoperability .....	340
16.1.1 Swift und Cocoa .....	340
16.1.2 Swift und Objective-C .....	361
16.1.3 Swift und C .....	374
16.2 Mix and Match .....	377
16.2.1 Mix and Match innerhalb eines App-Targets .....	378
16.2.2 Mix and Match innerhalb eines Framework-Targets .....	380
16.3 Migration .....	381
<b>17 Objektorientierte vs. protokollorientierte Programmierung .....</b>	<b>383</b>
17.1 Objektorientierte Programmierung .....	384
17.1.1 Praxis .....	385
17.1.2 Vor- und Nachteile .....	388
17.2 Protokollorientierte Programmierung .....	389
17.2.1 Praxis .....	389
17.2.2 Vor- und Nachteile .....	392
17.3 Fazit .....	393
<b>18 Weitere Sprachmerkmale und Profi-Wissen .....</b>	<b>395</b>
18.1 Zahlenliterale .....	395
18.2 Fortgeschrittene Operatoren .....	396
18.2.1 Ternary Conditional Operator .....	396
18.2.2 Nil-Coalescing Operator .....	398

18.2.3	Unary Minus- und Unary Plus-Operatoren .....	399
18.2.4	Bitweise Operatoren .....	399
18.2.5	Operator Methods .....	403
18.2.6	Eigene Operatoren erstellen .....	409
18.3	Option Sets .....	411
18.4	Closures .....	416
18.4.1	Escaping Closures .....	416
18.4.2	Closure Capture List .....	419
18.5	Recursive Enumerations .....	422
18.6	Optionals im Detail .....	424
18.7	Generic Where Clause .....	426
18.8	Dynamic Method Lookup .....	426
18.9	Weitere Objective-C-Makros .....	427
18.9.1	NS_SWIFT_NAME .....	428
18.9.2	NS_SWIFT_UNAVAILABLE .....	428
<b>Index</b>	.....	<b>429</b>