

Inhaltsverzeichnis

Vorwort	13
 I Einstieg	 21
 1 Einführung	 23
1.1 Python im Überblick	23
1.2 Los geht's – Installation	27
1.2.1 Python-Download	28
1.2.2 Installation von Python	28
1.2.3 Nacharbeiten nach der Installation	29
1.2.4 Python-Installation prüfen	31
1.2.5 Python-Programm als Skript ausführen	32
1.3 Entwicklungsumgebungen	32
1.3.1 Installation von PyCharm	33
1.3.2 PyCharm starten	35
1.3.3 Erstes Projekt in PyCharm	37
1.3.4 Erstes Modul in PyCharm	38
 2 Schnelleinstieg	 43
2.1 Hallo Welt (Hello World)	43
2.2 Variablen und Datentypen	44
2.2.1 Definition von Variablen	45
2.2.2 Variablen und Typen	46
2.2.3 Ausgaben mit print()	48
2.2.4 Bezeichner (Variablennamen)	50
2.3 Operatoren im Überblick	52
2.3.1 Arithmetische Operatoren	53

2.3.2	Zuweisungsoperatoren	56
2.3.3	Vergleichsoperatoren	58
2.3.4	Logische Operatoren	59
2.4	Fallunterscheidungen	61
2.5	Funktionen	65
2.5.1	Eigene Funktionen definieren	66
2.5.2	Nützliche Beispiele aus Python	69
2.6	Fehlerbehandlung und Exceptions	70
2.7	Kommentare	71
2.8	Module	72
2.8.1	Imports – Einbinden anderer Funktionalitäten	73
2.8.2	Zusammenfassung und Ergänzendes	74
2.9	Built-in-Datentypen	75
2.9.1	Listen (list)	76
2.9.2	Tupel (tuple)	77
2.9.3	Mengen (set)	79
2.9.4	Dictionaries (dict)	79
2.10	Schleifen	81
2.10.1	Besonderheit: Ranges	81
2.10.2	Indexbasierte for-in-Schleife	82
2.10.3	Wertebasierte for-in-Schleife	84
2.10.4	Die for-in-enumerate-Schleife mit Index und Wert	85
2.10.5	Die while-Schleife	87
2.11	Weiterführende Informationen	88
3	Strings	89
3.1	Schnelleinstieg	89
3.1.1	Gebräuchliche Stringaktionen	90
3.1.2	Suchen und Ersetzen	98
3.1.3	Informationen extrahieren und formatieren ..	101
3.1.4	Praxisrelevante Funktionen im Kurzüberblick	104
3.2	Nächste Schritte	105
3.2.1	Zeichenverarbeitung	105
3.2.2	Strings und Listen	106
3.2.3	Mehrzeilige Strings	108

4	Klassen und Objektorientierung	111
4.1	Schnelleinstieg	111
4.1.1	Grundlagen zu Klassen und Objekten	112
4.1.2	Eigenschaften (Attribute)	117
4.1.3	Verhalten (Methoden)	118
4.1.4	Typprüfung mit <code>isinstance()</code>	124
4.1.5	Objekte vergleichen – die Rolle von <code>__eq__()</code>	125
4.2	Nächste Schritte	128
4.2.1	Klassen ausführbar machen	128
4.2.2	Imports und Packages	131
4.2.3	Übergang zum Einsatz einer IDE	132
4.2.4	Verstecken von Informationen	136
4.2.5	Packages: Auswirkungen auf unsere Applikation	141
4.3	Vererbung	146
5	Collections	149
5.1	Schnelleinstieg	149
5.1.1	Die Klasse <code>list</code>	149
5.1.2	Die Klasse <code>set</code>	157
5.1.3	Die Klasse <code>dict</code>	161
5.2	Nächste Schritte	167
5.2.1	Comprehensions	167
5.2.2	Slicing – Zugriff auf Teilbereiche von Listen	169
5.2.3	Sortierung – <code>sort()</code> / <code>sorted()</code>	172
5.2.4	Tauschen von Elementen – <code>swap()</code>	176
5.2.5	Reihenfolge umkehren – <code>reverse()</code> und <code>reversed()</code>	179
5.2.6	Mehrdimensionale Listen	180
6	Ergänzendes Wissen	185
6.1	Benutzereingaben <code>input()</code>	185
6.2	Zufallswerte und das Modul <code>random</code>	187
6.3	Besonderheiten von Parametern	189
6.3.1	Parameter mit Position bzw. Name	189
6.3.2	Parameter mit Defaultwert	191
6.3.3	Var Args – variable Anzahl an Argumenten	191

6.4 Ternary-Operator	196
6.5 Aufzählungen mit Enum	197
6.6 Fallunterscheidungen mit match	200
6.7 break, continue und else in Schleifen	204
6.7.1 Funktionsweise von break und continue ...	204
6.7.2 Wie macht man es besser?	208
6.7.3 Besonderheit: else in Schleifen	211
6.8 Ausdrücke mit eval() auswerten	213
6.9 Rekursion	214
6.9.1 Einführendes Beispiel	214
6.9.2 Weiterführendes Beispiel: Fibonacci-Zahlen ..	215
6.9.3 Praxisbeispiel: Flächen füllen	217

II Aufstieg **219**

7 Collections Advanced	221
7.1 Sequenzielle Datentypen	221
7.2 Iteratoren	223
7.3 Generatoren	228
7.4 Datencontainer mit namedtuple	231
7.5 Einstieg in Lambdas	236
7.5.1 Syntax von Lambdas	236
7.5.2 Lambdas im Einsatz mit filter() und map() ..	237
7.5.3 Lambdas im Einsatz mit sort()	240
8 Verarbeitung von Dateien	243
8.1 Schnelleinstieg	243
8.1.1 Anlegen von Dateien und Verzeichnissen ...	244
8.1.2 Aktuelles Verzeichnis wechseln	245
8.1.3 Aktuelles Verzeichnis und absoluten Pfad ermitteln	246
8.1.4 Inhalt eines Verzeichnisses auflisten	246
8.1.5 Pfad ist Datei oder Verzeichnis?	247
8.1.6 Auf Existenz prüfen	247

8.1.7	Informationen in Dateien schreiben und daraus lesen	248
8.1.8	Einfluss der Verarbeitungsmodi	252
8.1.9	Diverse Informationen ermitteln	254
8.1.10	Kopieren	256
8.1.11	Umbenennen	257
8.1.12	Löschen	258
8.2	JSON-Verarbeitung	259
8.2.1	JSON in eine Datei schreiben	260
8.2.2	Lesen von JSON aus einer Datei	261
8.2.3	Pretty Printing	262
9	Fehlerbehandlung mit Exceptions	263
9.1	Schnelleinstieg	263
9.1.1	Fehlerbehandlung	265
9.1.2	Exceptions selbst auslösen – raise	271
9.1.3	Eigene Exception-Typen definieren	273
9.1.4	Propagation von Exceptions	275
9.2	Fehlerbehandlung in der Praxis	278
9.2.1	Elegante Prüfungen mit assert	279
9.3	Automatic Resource Management (with)	282
10	Datumsverarbeitung	283
10.1	Schnelleinstieg	283
10.1.1	Zeitpunkte und die Klasse datetime	283
10.1.2	Datumswerte und die Klasse date	285
10.1.3	Zeit und die Klasse time	289
10.1.4	Zeitdifferenzen und die Klasse timedelta	291
10.1.5	Berechnungen mit Datumswerten	292
10.1.6	Formatierung und Parsing	294

III Anhang	297
A Schlüsselwörter im Überblick	299
B Schnelleinstieg Python-REPL	303
Literaturverzeichnis	307
Index	309