

Inhalt

TEIL I Einstieg in Python

1 Einleitung	17
1.1 Über dieses Buch	17
1.1.1 Warum haben wir dieses Buch geschrieben?	17
1.1.2 Was leistet dieses Buch, was nicht?	18
1.1.3 Wie ist dieses Buch aufgebaut?	19
1.1.4 Wer sollte dieses Buch wie lesen?	20
1.1.5 Neuerungen in der zweiten Auflage	21
1.1.6 Danksagung	21
2 Überblick über Python	23
2.1 Geschichte und Entstehung	23
2.2 Grundlegende Konzepte	24
2.3 Einsatzmöglichkeiten und Stärken	25
2.4 Aktuelle Einsatzgebiete	26
3 Die Arbeit mit Python	27
3.1 Die Verwendung von Python	27
3.1.1 Windows	29
3.1.2 Linux	29
3.1.3 Mac OS X	30
3.2 Tippen, kompilieren, testen	30
3.2.1 Shebang	32
3.2.2 Interne Abläufe	32
4 Der interaktive Modus	35
4.1 Ganze Zahlen	36
4.2 Gleitkommazahlen	37
4.3 Zeichenketten	37
4.4 Variablen	38
4.5 Logische Ausdrücke	39
4.6 Bildschirmausgaben	41

5 Grundlegendes zu Python-Programmen	43
5.1 Grundstruktur eines Python-Programms	43
5.2 Das erste Programm	45
5.3 Kommentare	47
5.4 Der Fehlerfall	48
6 Kontrollstrukturen	51
6.1 Fallunterscheidungen	51
6.1.1 if, elif, else	51
6.1.2 Conditional Expressions	55
6.2 Schleifen	56
6.2.1 While-Schleife	56
6.2.2 Vorzeitiger Abbruch einer Schleife	58
6.2.3 Vorzeitiger Abbruch eines Schleifendurchlaufs	59
6.2.4 For-Schleife	61
6.3 Die pass-Anweisung	65
7 Das Laufzeitmodell	67
7.1 Die Struktur von Instanzen	69
7.2 Referenzen und Instanzen freigeben	74
7.3 Mutable vs. immutable Datentypen	75
8 Basisdatentypen	79
8.1 Operatoren	79
8.2 Das Nichts – NoneType	81
8.3 Numerische Datentypen	82
8.3.1 Ganzzahlen – int	85
8.3.2 Gleitkommazahlen – float	91
8.3.3 Boolesche Werte – bool	93
8.3.4 Komplexe Zahlen – complex	98
8.4 Methoden und Parameter	100
8.5 Sequentielle Datentypen	103
8.5.1 Listen – list	111
8.5.2 Unveränderliche Listen – tuple	122
8.5.3 Strings – str, bytes	124
8.6 Mappings	151
8.6.1 Dictionary – dict	151

8.7 Mengen	160
8.7.1 Mengen – set	167
8.7.2 Unveränderliche Mengen – frozenset	168

9 Dateien 171

9.1 Datenströme	171
9.2 Daten aus einer Datei auslesen	172
9.3 Daten in eine Datei schreiben	176
9.4 Verwendung des Dateiobjekts	177

10 Funktionen 181

10.1 Schreiben einer Funktion	183
10.2 Funktionsparameter	187
10.2.1 Optionale Parameter	187
10.2.2 Schlüsselwortparameter	188
10.2.3 Beliebige Anzahl von Parametern	189
10.2.4 Seiteneffekte	192
10.3 Lokale Funktionen	195
10.4 Anonyme Funktionen	196
10.5 Namensräume	196
10.5.1 Zugriff auf globale Variablen – global	197
10.5.2 Zugriff auf übergeordnete Namensräume – nonlocal	198
10.6 Rekursion	200
10.7 Vordefinierte Funktionen	201

TEIL II Fortgeschrittene Programmietechniken

11 Modularisierung 223

11.1 Einbinden externer Programmbibliotheken	224
11.2 Eigene Module	226
11.2.1 Modulinterne Referenzen	227
11.3 Pakete	228
11.3.1 Absolute und relative Import-Anweisungen	230
11.3.2 Importieren aller Module eines Pakets	231
11.3.3 Die Built-in Function <code>_import_</code>	232

12 Objektorientierung	235
12.1 Klassen	240
12.1.1 Definieren von Methoden	242
12.1.2 Konstruktor, Destruktor und die Erzeugung von Attributen	243
12.1.3 Private Member	246
12.1.4 Versteckte Setter und Getter	250
12.1.5 Statische Member	252
12.2 Vererbung	255
12.2.1 Mehrfachvererbung	258
12.3 Magic Members	262
12.3.1 Allgemeine Magic Members	263
12.3.2 Datentypen emulieren	269
12.4 Objektphilosophie	277
13 Weitere Spracheigenschaften	279
13.1 Exception Handling	279
13.1.1 Eingebaute Exceptions	280
13.1.2 Werfen einer Exception	282
13.1.3 Abfangen einer Exception	283
13.1.4 Eigene Exceptions	287
13.1.5 Erneutes Werfen einer Exception	289
13.1.6 Exception Chaining	291
13.2 Comprehensions	292
13.2.1 List Comprehensions	293
13.2.2 Dict Comprehensions	295
13.2.3 Set Comprehensions	296
13.3 Docstrings	296
13.4 Generatoren	298
13.5 Iteratoren	301
13.6 Interpreter im Interpreter	310
13.7 Geplante Sprachelemente	312
13.8 Die with-Anweisung	313
13.9 Function Annotations	316
13.10 Function Decorator	318
13.11 assert	321
13.12 Weitere Aspekte der Syntax	322
13.12.1 Umbrechen langer Zeilen	322
13.12.2 Zusammenfügen mehrerer Zeilen	323

TEIL III Die Standardbibliothek

14 Mathematik	327
14.1 Mathematische Funktionen – math, cmath	327
14.1.1 Mathematische Konstanten	328
14.1.2 Zahlentheoretische Funktionen	328
14.1.3 Exponential- und Logarithmusfunktionen	330
14.1.4 Trigonometrische Funktionen	331
14.1.5 Winkelfunktionen	333
14.1.6 Hyperbolische Funktionen	333
14.1.7 Funktionen aus cmath	334
14.2 Zufallszahlengenerator – random	334
14.3 Präzise Dezimalzahlen – decimal	339
14.3.1 Verwendung des Datentyps	340
14.3.2 Nichtnumerische Werte	343
14.3.3 Das Context-Objekt	343
14.4 Spezielle Generatoren – itertools	345
15 Strings	351
15.1 Reguläre Ausdrücke – re	351
15.1.1 Syntax regulärer Ausdrücke	352
15.1.2 Verwendung des Moduls	362
15.1.3 Ein einfaches Beispielprogramm – Searching	372
15.1.4 Ein komplexeres Beispielprogramm – Matching	373
15.2 Lokalisierung von Programmen – gettext	376
15.2.1 Beispiel für die Verwendung von gettext	377
15.3 Hash-Funktionen – hashlib	380
15.3.1 Verwendung des Moduls	382
15.3.2 Beispiel	383
16 Datum und Zeit	385
16.1 Elementare Zeitfunktionen – time	385
16.2 Komfortable Datumsfunktionen – datetime	392
16.2.1 datetime.date	393
16.2.2 datetime.time	397
16.2.3 datetime.datetime	399

17 Schnittstelle zum Betriebssystem 405

17.1	Funktionen des Betriebssystems – os	405
17.1.1	Zugriff auf den eigenen Prozess und andere Prozesse	406
17.1.2	Zugriff auf das Dateisystem	407
17.2	Umgang mit Pfaden – os.path	413
17.3	Zugriff auf die Laufzeitumgebung – sys	418
17.3.1	Konstanten	418
17.3.2	Exceptions	421
17.3.3	Hooks	422
17.3.4	Sonstige Funktionen	423
17.4	Informationen über das System – platform	425
17.4.1	Funktionen	425
17.5	Kommandozeilenparameter – optparse	425
17.5.1	Taschenrechner – ein einfaches Beispiel	426
17.5.2	Weitere Verwendungsmöglichkeiten	428
17.6	Kopieren von Instanzen – copy	430
17.7	Zugriff auf das Dateisystem – shutil	433
17.8	Das Programmende – atexit	435

18 Parallel Programmierung 437

18.1	Prozesse, Multitasking und Threads	437
18.2	Die Thread-Unterstützung in Python	440
18.3	Das Modul thread	441
18.3.1	Datenaustausch zwischen Threads – locking	443
18.4	Das Modul threading	448
18.4.1	Locking im threading-Modul	450
18.4.2	Worker-Threads und Queues	454
18.4.3	Ereignisse definieren – threading.Event	457
18.4.4	Eine Funktion zeitlich versetzt ausführen – threading.Timer	457

19 Datenspeicherung 459

19.1	Komprimierte Dateien lesen und schreiben – gzip	459
19.2	XML	461
19.2.1	DOM – Document Object Model	463
19.2.2	SAX – Simple API for XML	474
19.2.3	ElementTree	479

19.3	Datenbanken	483
19.3.1	Pythons eingebaute Datenbank – sqlite3	487
19.4	Serialisierung von Instanzen – pickle	502
19.5	Das Tabellenformat CSV – csv	506
19.6	Temporäre Dateien – tempfile	511

20 Netzwerkkommunikation 515

20.1	Socket API	517
20.1.1	Client-Server-Systeme	518
20.1.2	UDP	520
20.1.3	TCP	522
20.1.4	Blockierende und nicht-blockierende Sockets	525
20.1.5	Verwendung des Moduls	526
20.1.6	Netzwerk-Byte-Order	531
20.1.7	Multiplexende Server – select	532
20.1.8	socketserver	536
20.2	URLs	540
20.2.1	Zugriff auf Ressourcen im Internet – urllib.request	540
20.2.2	Verarbeiten einer URL – urllib.parse	544
20.3	FTP – ftplib	549
20.4	E-Mail	557
20.4.1	SMTP – smtplib	557
20.4.2	POP3 – poplib	561
20.4.3	IMAP4 – imaplib	566
20.4.4	Erstellen komplexer E-Mails – email	572
20.5	Telnet – telnetlib	577
20.6	XML-RPC	580
20.6.1	Der Server	581
20.6.2	Der Client	585
20.6.3	Multicall	587
20.6.4	Einschränkungen	588

21 Debugging 591

21.1	Der Debugger	591
21.2	Inspizieren von Instanzen – inspect	594
21.2.1	Datentypen, Attribute und Methoden	595
21.2.2	Quellcode	597
21.2.3	Klassen und Funktionen	598
21.3	Formatierte Ausgabe von Instanzen – pprint	602

21.4	Logdateien – logging	605
21.4.1	Das Meldungsformat anpassen	607
21.4.2	Logging Handler	609
21.5	Automatisiertes Testen	611
21.5.1	Testfälle in Docstrings – doctest	611
21.5.2	Unit Tests – unittest	615
21.6	Traceback-Objekte – traceback	619
21.7	Analyse des Laufzeitverhaltens	622
21.7.1	Laufzeitmessung – timeit	623
21.7.2	Profiling – cProfile	626
21.7.3	Tracing – trace	629

TEIL IV Weiterführende Themen

22 Distribution von Python-Projekten 635

22.1	Erstellen von Distributionen – distutils	635
22.1.1	Schreiben des Moduls	636
22.1.2	Das Installationsscript	638
22.1.3	Erstellen einer Quellcodedistribution	642
22.1.4	Erstellen einer Binärdistribution	643
22.2	Distributionen installieren	644

23 Optimierung 645

23.1	Die Optimize-Option	646
23.2	Mutable vs. immutable	646
23.3	Funktionsaufrufe	647
23.4	Schleifen	648
23.5	C	648
23.6	Lookup	649
23.7	Exceptions	649
23.8	Keyword Arguments	650

24 Grafische Benutzeroberflächen 651

24.1	Toolkits	651
24.2	Einführung in tkinter	654
24.2.1	Ein einfaches Beispiel	654
24.2.2	Steuerelementvariablen	656
24.2.3	Der Packer	658
24.2.4	Ausrichtung	660

24.2.5 Padding	660
24.2.6 Übersicht	661
24.2.7 Events	663
24.2.8 Die Steuerelemente	669
24.2.9 Die Klasse Tk	705
24.2.10 Weitere Module	707

25 Anbindung an andere Programmiersprachen 719

25.1 Dynamisch ladbare Bibliotheken – ctypes	720
25.1.1 Ein einfaches Beispiel	720
25.1.2 Die eigene Bibliothek	721
25.1.3 Schnittstellenbeschreibung	725
25.1.4 Verwendung des Moduls	726
25.2 Schreiben von Extensions	728
25.2.1 Ein einfaches Beispiel	728
25.2.2 Exceptions	732
25.2.3 Erzeugen der Extension	733
25.2.4 Reference Counting	734
25.3 Python als eingebettete Scriptsprache	736
25.3.1 Ein einfaches Beispiel	736
25.3.2 Ein komplexeres Beispiel	738
25.3.3 Python-API-Referenz	741

26 Insiderwissen 747

26.1 URLs im Standardbrowser öffnen – webbrowser	747
26.2 Funktionsschnittstellen vereinfachen – functools	748
26.3 Versteckte Passworteingaben – getpass	750
26.4 Kommandozeilen-Interpreter – cmd	750

27 Von Python 2.6 nach Python 3.0 755

27.1 Die wichtigsten Unterschiede	755
27.1.1 Ein-/Ausgabe	756
27.1.2 Iteratoren	757
27.1.3 Strings	757
27.1.4 Ganze Zahlen	758
27.1.5 Exception Handling	759
27.1.6 Standardbibliothek	759
27.1.7 Neue Sprachelemente in Python 3.0	760
27.2 Automatische Konvertierung	761

A Anhänger	765
A.1 Entwicklungsumgebungen	765
A.1.1 Eclipse	766
A.1.2 Eric4	767
A.1.3 Komodo IDE	768
A.1.4 Wing IDE	769
A.2 Reservierte Wörter	770
A.3 Operatorrangfolge	770
A.4 Built-in Exceptions	771
A.5 Built-in Functions	775
Index	779