

Auf einen Blick

Teil I Einstieg in Python

| | | |
|---|--|-----|
| 1 | Überblick über Python | 27 |
| 2 | Die Arbeit mit Python | 31 |
| 3 | Der interaktive Modus | 39 |
| 4 | Grundlegendes zu Python-Programmen | 51 |
| 5 | Kontrollstrukturen | 59 |
| 6 | Das Laufzeitmodell | 75 |
| 7 | Basisdatentypen | 87 |
| 8 | Dateien | 197 |
| 9 | Funktionen | 209 |

Teil II Fortgeschrittene Programmiertechniken

| | | |
|----|-----------------------------------|-----|
| 10 | Modularisierung | 257 |
| 11 | Objektorientierung | 269 |
| 12 | Weitere Spracheigenschaften | 329 |

Teil III Die Standardbibliothek

| | | |
|----|--|-----|
| 13 | Mathematik | 385 |
| 14 | Strings | 415 |
| 15 | Datum und Zeit | 451 |
| 16 | Schnittstelle zum Betriebssystem | 481 |
| 17 | Parallele Programmierung | 525 |
| 18 | Datenspeicherung | 549 |
| 19 | Netzwerkkommunikation | 607 |
| 20 | Debugging | 691 |

Teil IV Weiterführende Themen

| | | |
|----|---|-----|
| 21 | Distribution von Python-Projekten | 739 |
| 22 | Optimierung | 753 |
| 23 | Grafische Benutzeroberflächen | 759 |
| 24 | Wissenschaftliches Rechnen | 881 |
| 25 | Anbindung an andere Programmiersprachen | 903 |
| 26 | Insiderwissen | 933 |
| 27 | Von Python 2 nach Python 3 | 955 |

Inhalt

| | |
|------------------------|----|
| Über dieses Buch | 19 |
|------------------------|----|

Teil I Einstieg in Python

| | |
|--------------------------------------|-----------|
| 1 Überblick über Python | 27 |
|--------------------------------------|-----------|

| | |
|--|----|
| 1.1 Geschichte und Entstehung | 27 |
| 1.2 Grundlegende Konzepte | 28 |
| 1.3 Einsatzmöglichkeiten und Stärken | 29 |
| 1.4 Einsatzbeispiele | 30 |

| | |
|--------------------------------------|-----------|
| 2 Die Arbeit mit Python | 31 |
|--------------------------------------|-----------|

| | |
|---------------------------------------|----|
| 2.1 Die Verwendung von Python | 31 |
| 2.1.1 Windows | 33 |
| 2.1.2 Linux | 33 |
| 2.1.3 Mac OS X | 33 |
| 2.2 Tippen, kompilieren, testen | 34 |
| 2.2.1 Shebang | 35 |
| 2.2.2 Interne Abläufe | 36 |

| | |
|--------------------------------------|-----------|
| 3 Der interaktive Modus | 39 |
|--------------------------------------|-----------|

| | |
|-----------------------------------|----|
| 3.1 Ganze Zahlen | 40 |
| 3.2 Gleitkommazahlen | 41 |
| 3.3 Zeichenketten | 42 |
| 3.4 Listen | 43 |
| 3.5 Variablen | 43 |
| 3.6 Logische Ausdrücke | 45 |
| 3.7 Funktionen und Methoden | 47 |

| | | |
|------------|----------------------------------|-----------|
| 3.7.1 | Funktionen | 47 |
| 3.7.2 | Methoden | 48 |
| 3.8 | Bildschirm Ausgaben | 49 |

4 Grundlegendes zu Python-Programmen 51

| | | |
|------------|---|-----------|
| 4.1 | Grundstruktur eines Python-Programms | 51 |
| 4.2 | Das erste Programm | 53 |
| 4.3 | Kommentare | 55 |
| 4.4 | Der Fehlerfall | 56 |

5 Kontrollstrukturen 59

| | | |
|------------|---|-----------|
| 5.1 | Fallunterscheidungen | 59 |
| 5.1.1 | Die if-Anweisung | 60 |
| 5.1.2 | Conditional Expressions | 63 |
| 5.2 | Schleifen | 64 |
| 5.2.1 | Die while-Schleife | 65 |
| 5.2.2 | Vorzeitiger Abbruch einer Schleife | 65 |
| 5.2.3 | Erkennen eines Schleifenabbruchs | 66 |
| 5.2.4 | Vorzeitiger Abbruch eines Schleifendurchlaufs | 68 |
| 5.2.5 | Die for-Schleife | 70 |
| 5.2.6 | Die for-Schleife als Zählschleife | 71 |
| 5.3 | Die pass-Anweisung | 73 |

6 Das Laufzeitmodell 75

| | | |
|------------|---|-----------|
| 6.1 | Die Struktur von Instanzen | 77 |
| 6.1.1 | Datentyp | 77 |
| 6.1.2 | Wert | 78 |
| 6.1.3 | Identität | 80 |
| 6.2 | Referenzen und Instanzen freigeben | 81 |
| 6.3 | Mutable vs. immutable Datentypen | 82 |
| 6.3.1 | Mutable Datentypen und Seiteneffekte | 84 |

| | | |
|------------|--|------------|
| 7 | Basisdatentypen | 87 |
| 7.1 | Operatoren | 88 |
| 7.2 | Das Nichts – NoneType | 92 |
| 7.3 | Numerische Datentypen | 93 |
| 7.3.1 | Arithmetische Operatoren | 93 |
| 7.3.2 | Vergleichende Operatoren | 95 |
| 7.3.3 | Konvertierung zwischen numerischen Datentypen | 96 |
| 7.3.4 | Ganzzahlen – int | 96 |
| 7.3.5 | Gleitkommazahlen – float | 102 |
| 7.3.6 | Boolesche Werte – bool | 104 |
| 7.3.7 | Komplexe Zahlen – complex | 110 |
| 7.4 | Parameter von Funktionen und Methoden | 113 |
| 7.4.1 | Wiederholung | 113 |
| 7.4.2 | Parameter | 114 |
| 7.5 | Sequentielle Datentypen | 115 |
| 7.5.1 | Operationen auf Instanzen sequentieller Datentypen | 118 |
| 7.5.2 | Listen – »list« | 128 |
| 7.5.3 | Unveränderliche Listen – tuple | 139 |
| 7.5.4 | Strings – str, bytes, bytearray | 142 |
| 7.6 | Mappings | 173 |
| 7.6.1 | Dictionary – dict | 174 |
| 7.7 | Mengen | 185 |
| 7.7.1 | Die Datentypen set und frozenset | 185 |
| 7.7.2 | Veränderliche Mengen – set | 193 |
| 7.7.3 | Unveränderliche Mengen – frozenset | 195 |
| 8 | Dateien | 197 |
| 8.1 | Datenströme | 197 |
| 8.2 | Daten aus einer Datei auslesen | 198 |
| 8.3 | Daten in eine Datei schreiben | 202 |
| 8.4 | Verwendung des Dateiobjekts | 204 |

9 Funktionen 209

| | | |
|------------|--|-----|
| 9.1 | Schreiben einer Funktion | 211 |
| 9.2 | Funktionsparameter | 215 |
| 9.2.1 | Optionale Parameter | 215 |
| 9.2.2 | Schlüsselwortparameter | 216 |
| 9.2.3 | Beliebige Anzahl von Parametern | 217 |
| 9.2.4 | Reine Schlüsselwortparameter | 219 |
| 9.2.5 | Entpacken einer Parameterliste | 221 |
| 9.2.6 | Seiteneffekte | 222 |
| 9.3 | Namensräume | 225 |
| 9.3.1 | Zugriff auf globale Variablen – global | 225 |
| 9.3.2 | Zugriff auf den globalen Namensraum | 226 |
| 9.3.3 | Zugriff auf übergeordnete Namensräume – nonlocal | 227 |
| 9.4 | Lokale Funktionen | 229 |
| 9.5 | Anonyme Funktionen | 230 |
| 9.6 | Rekursion | 231 |
| 9.7 | Eingebaute Funktionen | 231 |

Teil II Fortgeschrittene Programmiertechniken

10 Modularisierung 257

| | | |
|-------------|--|-----|
| 10.1 | Einbinden globaler Module | 257 |
| 10.2 | Lokale Module | 260 |
| 10.2.1 | Namenskonflikte | 261 |
| 10.2.2 | Modulinterne Referenzen | 262 |
| 10.3 | Pakete | 262 |
| 10.3.1 | Absolute und relative Import-Anweisungen | 265 |
| 10.3.2 | Importieren aller Module eines Pakets | 266 |
| 10.4 | Built-in Functions | 267 |

11 Objektorientierung 269

| | | |
|-------------|--|-----|
| 11.1 | Klassen | 274 |
| 11.1.1 | Definieren von Methoden | 276 |
| 11.1.2 | Der Konstruktor und die Erzeugung von Attributen | 277 |
| 11.2 | Vererbung | 280 |
| 11.2.1 | Technische Grundlagen | 281 |
| 11.2.2 | Die Klasse GirokontoMitTagesumsatz | 283 |
| 11.2.3 | Beispiele möglicher Erweiterungen | 289 |
| 11.2.4 | Ausblick | 293 |
| 11.2.5 | Mehrfachvererbung | 294 |
| 11.3 | Setter und Getter und Property Attributes | 295 |
| 11.4 | Klassenattribute und Klassenmethoden sowie statische Methoden | 298 |
| 11.5 | Magic Methods und Magic Attributes | 302 |
| 11.5.1 | Allgemeine Magic Methods | 302 |
| 11.5.2 | Zugriff auf Attribute anpassen | 306 |
| 11.5.3 | Operatoren überladen | 309 |
| 11.5.4 | Datentypen emulieren | 317 |
| 11.6 | Built-in Functions für Objektorientierung | 322 |
| 11.6.1 | Funktionen für die Verwaltung der Attribute einer Instanz | 323 |
| 11.6.2 | Funktionen für Informationen über die Klassenhierarchie | 324 |
| 11.7 | Objektphilosophie | 326 |

12 Weitere Spracheigenschaften 329

| | | |
|-------------|---------------------------------------|-----|
| 12.1 | Exception Handling | 329 |
| 12.1.1 | Eingebaute Exceptions | 330 |
| 12.1.2 | Werfen einer Exception | 334 |
| 12.1.3 | Abfangen einer Exception | 335 |
| 12.1.4 | Eigene Exceptions | 340 |
| 12.1.5 | Erneutes Werfen einer Exception | 342 |
| 12.1.6 | Exception Chaining | 344 |
| 12.2 | Comprehensions | 346 |
| 12.2.1 | List Comprehensions | 346 |
| 12.2.2 | Dict Comprehensions | 348 |
| 12.2.3 | Set Comprehensions | 349 |

| | | |
|--------------|---|-----|
| 12.3 | Generatoren | 349 |
| 12.4 | Iteratoren | 353 |
| 12.5 | Docstrings | 363 |
| 12.6 | Interpreter im Interpreter | 365 |
| 12.7 | Die with-Anweisung | 368 |
| 12.8 | Function Annotations | 371 |
| 12.9 | Function Decorator | 373 |
| 12.10 | assert | 377 |
| 12.11 | Weitere Aspekte der Syntax | 378 |
| 12.11.1 | Umbrechen langer Zeilen | 378 |
| 12.11.2 | Zusammenfügen mehrerer Zeilen | 379 |
| 12.12 | Geplante Sprachelemente | 380 |

Teil III Die Standardbibliothek

| | | |
|-------------|---|-----|
| 13 | Mathematik | 385 |
| 13.1 | Mathematische Funktionen – math, cmath | 385 |
| 13.1.1 | Zahlentheoretische Funktionen | 389 |
| 13.1.2 | Exponential- und Logarithmusfunktionen | 391 |
| 13.1.3 | Trigonometrische Funktionen | 392 |
| 13.1.4 | Funktionen aus cmath | 394 |
| 13.2 | Zufallszahlengenerator – random | 394 |
| 13.2.1 | Steuerungsfunktionen | 396 |
| 13.2.2 | Funktionen für ganze Zahlen | 397 |
| 13.2.3 | Funktionen für Sequenzen | 398 |
| 13.2.4 | Alternative Generatoren | 399 |
| 13.3 | Präzise Dezimalzahlen – decimal | 399 |
| 13.3.1 | Verwendung des Datentyps | 400 |
| 13.3.2 | Nichtnumerische Werte | 403 |
| 13.3.3 | Das Context-Objekt | 404 |
| 13.4 | Spezielle Generatoren – itertools | 405 |

14 Strings 415

| | | |
|-------------|---|-----|
| 14.1 | Reguläre Ausdrücke – re | 415 |
| 14.1.1 | Syntax regulärer Ausdrücke | 416 |
| 14.1.2 | Verwendung des Moduls re | 427 |
| 14.1.3 | Ein einfaches Beispielprogramm – Searching | 438 |
| 14.1.4 | Ein komplexeres Beispielprogramm – Matching | 439 |
| 14.2 | Lokalisierung von Programmen – gettext | 442 |
| 14.2.1 | Beispiel für die Verwendung von gettext | 443 |
| 14.3 | Hash-Funktionen – hashlib | 446 |
| 14.3.1 | Verwendung des Moduls | 448 |
| 14.3.2 | Beispiel | 450 |

15 Datum und Zeit 451

| | | |
|-------------|--|-----|
| 15.1 | Elementare Zeitfunktionen – time | 451 |
| 15.2 | Komfortable Datumsfunktionen – datetime | 460 |
| 15.2.1 | Attribute und Datentypen (Übersicht) | 460 |
| 15.2.2 | date | 461 |
| 15.2.3 | datetime.time | 466 |
| 15.2.4 | datetime.datetime | 468 |
| 15.2.5 | datetime.timedelta | 477 |

16 Schnittstelle zum Betriebssystem 481

| | | |
|-------------|---|-----|
| 16.1 | Funktionen des Betriebssystems – os | 481 |
| 16.1.1 | Zugriff auf den eigenen Prozess und andere Prozesse | 482 |
| 16.1.2 | Zugriff auf das Dateisystem | 484 |
| 16.2 | Umgang mit Pfaden – os.path | 490 |
| 16.3 | Zugriff auf das Dateisystem – shutil | 495 |
| 16.3.1 | Verzeichnis- und Dateioperationen | 497 |
| 16.3.2 | Archivoperationen | 499 |
| 16.4 | Zugriff auf die Laufzeitumgebung – sys | 501 |
| 16.4.1 | Konstanten | 503 |
| 16.4.2 | Exceptions | 506 |

| | | |
|-------------|---|------------|
| 16.4.3 | Hooks | 507 |
| 16.4.4 | Sonstige Funktionen | 509 |
| 16.5 | Informationen über das System – platform | 511 |
| 16.6 | Kommandozeilenparameter – argparse | 512 |
| 16.6.1 | Taschenrechner – ein einfaches Beispiel | 513 |
| 16.6.2 | Komplexeres Beispiel | 518 |
| 16.7 | Kopieren von Instanzen – copy | 519 |
| 16.8 | Das Programmende – atexit | 523 |

17 Parallele Programmierung 525

| | | |
|-------------|--|------------|
| 17.1 | Prozesse, Multitasking und Threads | 525 |
| 17.2 | Die Thread-Unterstützung in Python | 528 |
| 17.3 | Das Modul <code>_thread</code> | 528 |
| 17.3.1 | Ein Beispiel für eine rechenintensive Funktion – die Approximation von Pi | 528 |
| 17.3.2 | Parallele Berechnung mithilfe von Threads | 529 |
| 17.3.3 | Datenaustausch zwischen Threads – locking | 531 |
| 17.4 | Das Modul <code>threading</code> | 536 |
| 17.4.1 | Locking im <code>threading</code> -Modul | 539 |
| 17.4.2 | Worker-Threads und Queues | 542 |
| 17.4.3 | Ereignisse definieren – <code>threading.Event</code> | 546 |
| 17.4.4 | Barrieren definieren – <code>threading.Barrier</code> | 546 |
| 17.4.5 | Eine Funktion zeitlich versetzt ausführen – <code>threading.Timer</code> | 546 |

18 Datenspeicherung 549

| | | |
|-------------|---|------------|
| 18.1 | Komprimierte Dateien lesen und schreiben – <code>gzip</code> | 549 |
| 18.2 | XML | 551 |
| 18.2.1 | DOM – Document Object Model | 553 |
| 18.2.2 | SAX – Simple API for XML | 565 |
| 18.2.3 | ElementTree | 570 |
| 18.3 | Datenbanken | 575 |
| 18.3.1 | Beispieldatenbank | 577 |
| 18.3.2 | Pythons eingebaute Datenbank – <code>sqlite3</code> | 579 |

| | | |
|-------------|--|------------|
| 18.4 | Serialisierung von Instanzen – pickle | 595 |
| 18.5 | Das Tabellenformat CSV – csv | 599 |
| 18.6 | Temporäre Dateien – tempfile | 604 |

19 Netzwerkkommunikation 607

| | | |
|-------------|---|------------|
| 19.1 | Socket API | 609 |
| 19.1.1 | Client-Server-Systeme | 610 |
| 19.1.2 | UDP | 612 |
| 19.1.3 | TCP | 614 |
| 19.1.4 | Blockierende und nicht-blockierende Sockets | 616 |
| 19.1.5 | Verwendung des Moduls | 618 |
| 19.1.6 | Die Socket-Klasse | 621 |
| 19.1.7 | Netzwerk-Byte-Order | 625 |
| 19.1.8 | Multiplexende Server – select | 627 |
| 19.1.9 | socketserver | 630 |
| 19.2 | URLs | 634 |
| 19.2.1 | Zugriff auf Ressourcen im Internet – urllib.request | 634 |
| 19.2.2 | Verarbeiten einer URL – urllib.parse | 640 |
| 19.3 | FTP – ftplib | 646 |
| 19.4 | E-Mail | 655 |
| 19.4.1 | SMTP – smtplib | 655 |
| 19.4.2 | POP3 – poplib | 660 |
| 19.4.3 | IMAP4 – imaplib | 665 |
| 19.4.4 | Erstellen komplexer E-Mails – email | 671 |
| 19.5 | Telnet – telnetlib | 676 |
| 19.6 | XML-RPC | 680 |
| 19.6.1 | Der Server | 680 |
| 19.6.2 | Der Client | 685 |
| 19.6.3 | Multicall | 687 |
| 19.6.4 | Einschränkungen | 688 |

20 Debugging 691

| | | |
|-------------|--|------------|
| 20.1 | Der Debugger | 691 |
| 20.2 | Inspizieren von Instanzen – inspect | 694 |

- 20.2.1 Datentypen, Attribute und Methoden 696
 - 20.2.2 Quellcode 697
 - 20.2.3 Klassen und Funktionen 699
- 20.3 **Formatierte Ausgabe von Instanzen – pprint** 703
- 20.4 **Logdateien – logging** 706
 - 20.4.1 Das Meldungsformat anpassen 709
 - 20.4.2 Logging Handler 711
- 20.5 **Automatisiertes Testen** 713
 - 20.5.1 Testfälle in Docstrings – doctest 713
 - 20.5.2 Unit Tests – unittest 718
- 20.6 **Traceback-Objekte – traceback** 722
- 20.7 **Analyse des Laufzeitverhaltens** 726
 - 20.7.1 Laufzeitmessung – timeit 727
 - 20.7.2 Profiling – cProfile 730
 - 20.7.3 Tracing – trace 734

Teil IV Weiterführende Themen

21 Distribution von Python-Projekten 739

- 21.1 **Erstellen von Distributionen – distutils** 739
 - 21.1.1 Schreiben des Moduls 740
 - 21.1.2 Das Installationsscript 742
 - 21.1.3 Erstellen einer Quellcodedistribution 746
 - 21.1.4 Erstellen einer Binärdistribution 748
 - 21.1.5 Distributionen installieren 749
- 21.2 **Erstellen von EXE-Dateien – cx_Freeze** 750

22 Optimierung 753

- 22.1 **Die Optimize-Option** 754
- 22.2 **Mutable vs. immutable** 754
- 22.3 **Schleifen** 755
- 22.4 **Funktionsaufrufe** 756

| | | |
|-------------|--------------------------------|-----|
| 22.5 | C | 756 |
| 22.6 | Lookup | 757 |
| 22.7 | Exceptions | 757 |
| 22.8 | Keyword Arguments | 758 |

23 Grafische Benutzeroberflächen 759

| | | |
|-------------|---|-----|
| 23.1 | Toolkits | 759 |
| 23.1.1 | Tkinter | 760 |
| 23.1.2 | PyGObject | 760 |
| 23.1.3 | PyQt | 760 |
| 23.1.4 | PySide | 761 |
| 23.1.5 | wxPython | 761 |
| 23.2 | Einführung in tkinter | 762 |
| 23.2.1 | Ein einfaches Beispiel | 762 |
| 23.2.2 | Steuerelementvariablen | 764 |
| 23.2.3 | Der Packer | 767 |
| 23.2.4 | Events | 771 |
| 23.2.5 | Die Steuerelemente | 779 |
| 23.2.6 | Die Klasse Tk | 817 |
| 23.2.7 | Weitere Module | 818 |
| 23.3 | Einführung in PyQt | 826 |
| 23.3.1 | Installation | 826 |
| 23.3.2 | Grundlegende Konzepte von Qt | 827 |
| 23.3.3 | Entwicklungsprozess | 829 |
| 23.4 | Signale und Slots | 836 |
| 23.5 | Wichtige Widgets | 839 |
| 23.5.1 | QCheckBox | 840 |
| 23.5.2 | QComboBox | 841 |
| 23.5.3 | QDateEdit, QTimeEdit, QDateTimeEdit | 842 |
| 23.5.4 | QDialog | 843 |
| 23.5.5 | QLineEdit | 844 |
| 23.5.6 | QListWidget, QListView | 844 |
| 23.5.7 | QProgressBar | 845 |
| 23.5.8 | QPushButton | 846 |
| 23.5.9 | QRadioButton | 846 |
| 23.5.10 | QSlider, QDial | 847 |

| | | |
|-------------|---|------------|
| 23.5.11 | QTextEdit | 848 |
| 23.5.12 | QWidget | 848 |
| 23.6 | Zeichenfunktionalität | 850 |
| 23.6.1 | Werkzeuge | 851 |
| 23.6.2 | Koordinatensystem | 853 |
| 23.6.3 | Einfache Formen | 853 |
| 23.6.4 | Grafiken | 856 |
| 23.6.5 | Text | 857 |
| 23.6.6 | Eye-Candy | 859 |
| 23.7 | Model-View-Architektur | 863 |
| 23.7.1 | Beispielprojekt: Ein Adressbuch | 864 |
| 23.7.2 | Auswählen von Einträgen | 874 |
| 23.7.3 | Bearbeiten von Einträgen | 875 |
| | | |
| 24 | Wissenschaftliches Rechnen | 881 |
| <hr/> | | |
| 24.1 | Installation | 882 |
| 24.2 | Das Modellprogramm | 883 |
| 24.2.1 | Der Import von numpy, scipy und matplotlib | 884 |
| 24.2.2 | Vektorisierung und der Datentyp numpy.ndarray | 885 |
| 24.2.3 | Visualisieren von Daten mit matplotlib.pyplot | 889 |
| 24.3 | Überblicke über die Module numpy und scipy | 891 |
| 24.3.1 | Überblick über den Datentyp numpy.ndarray | 891 |
| 24.3.2 | Überblick über scipy | 899 |
| | | |
| 25 | Anbindung an andere Programmiersprachen | 903 |
| <hr/> | | |
| 25.1 | Dynamisch ladbare Bibliotheken – ctypes | 904 |
| 25.1.1 | Ein einfaches Beispiel | 904 |
| 25.1.2 | Die eigene Bibliothek | 905 |
| 25.1.3 | Schnittstellenbeschreibung | 909 |
| 25.1.4 | Verwendung des Moduls | 911 |
| 25.2 | Schreiben von Extensions | 914 |
| 25.2.1 | Ein einfaches Beispiel | 914 |
| 25.2.2 | Exceptions | 918 |
| 25.2.3 | Erzeugen der Extension | 919 |
| 25.2.4 | Reference Counting | 921 |

| | | |
|-------------|--|------------|
| 25.3 | Python als eingebettete Scriptsprache | 922 |
| 25.3.1 | Ein einfaches Beispiel | 922 |
| 25.3.2 | Ein komplexeres Beispiel | 924 |
| 25.3.3 | Python-API-Referenz | 927 |

26 Insiderwissen 933

| | | |
|-------------|---|------------|
| 26.1 | URLs im Standardbrowser öffnen – webbrowser | 933 |
| 26.2 | Funktionsschnittstellen vereinfachen – functools | 934 |
| 26.2.1 | Funktionsschnittstellen vereinfachen | 934 |
| 26.2.2 | Caches | 935 |
| 26.2.3 | Ordnungsrelationen vervollständigen | 937 |
| 26.3 | Weitere Datentypen – collections | 938 |
| 26.3.1 | Zählen von Häufigkeiten | 938 |
| 26.3.2 | Dictionarys mit Standardwerten | 941 |
| 26.3.3 | Doppelt verkettete Listen | 942 |
| 26.3.4 | Benannte Tupel | 944 |
| 26.3.5 | Sortierte Dictionarys | 945 |
| 26.4 | Interpretieren von Binärdaten – struct | 946 |
| 26.5 | Versteckte Passworteingaben – getpass | 949 |
| 26.6 | Kommandozeilen-Interpreter – cmd | 949 |
| 26.7 | Dateiinterface für Strings – io.StringIO | 952 |

27 Von Python 2 nach Python 3 955

| | | |
|-------------|---|------------|
| 27.1 | Die wichtigsten Unterschiede | 958 |
| 27.1.1 | Ein-/Ausgabe | 958 |
| 27.1.2 | Iteratoren | 959 |
| 27.1.3 | Strings | 960 |
| 27.1.4 | Ganze Zahlen | 961 |
| 27.1.5 | Exception Handling | 962 |
| 27.1.6 | Standardbibliothek | 962 |
| 27.1.7 | Neue Sprachelemente in Python 3 | 963 |
| 27.2 | Automatische Konvertierung | 964 |

| | |
|--|-----|
| Anhang | 969 |
| <hr/> | |
| A.1 Reservierte Wörter | 969 |
| A.2 Built-in Functions | 969 |
| | |
| Index | 973 |