

2. AUFLAGE

---

# Einführung in Python

*Mark Lutz & David Ascher*

***Deutsche Überarbeitung und Aktualisierung:***

*Dinu C. Gherman*

***Fachliche Unterstützung:***

*Christian Tismer*

**O'REILLY®**

Beijing • Cambridge • Farnham • Köln • Paris • Sebastopol • Taipei • Tokyo

<b>Vorwort</b> .....	<b>XV</b>
----------------------	-----------

---

## Teil I    Einleitung

<b>1   Fragen und Antworten zu Python</b> .....	<b>3</b>
Warum Python? .....	3
Ist Python eine Skriptsprache? .....	6
Wo ist der Haken? .....	7
Wer benutzt heute Python? .....	8
Was kann ich mit Python machen? .....	8
Welche technischen Stärken hat Python? .....	12
Python verglichen mit anderen Sprachen .....	15
<b>2   Programmausführung durch Python</b> .....	<b>17</b>
Einführung in den Python-Interpreter .....	17
Programmausführung .....	19
Varianten des Ausführungsmodells .....	22
<b>3   Wie man Python-Programme ausführt</b> .....	<b>29</b>
Interaktives Codieren .....	29
System-Kommandozeilen und Dateien .....	33
Windows-Datei-Icons anklicken .....	37
Module importieren und neu laden .....	41
Die Entwicklungsumgebung IDLE .....	45
Weitere IDEs .....	49

Eingebettete Aufrufe .....	50
Eingefrorene ausführbare Binärdateien .....	51
Startmöglichkeiten im Texteditor .....	52
Weitere Startoptionen .....	52
Die Qual der Wahl .....	52
Übungen zu Teil I .....	52

---

## Teil II Typen und Operationen

<b>4 Zahlen .....</b>	<b>57</b>
Python-Programmstruktur .....	57
Warum eingebaute Typen benutzen? .....	57
Zahlen .....	59
Python-Ausdrucksoperatoren .....	62
Zahlen in Aktion .....	65
Intermezzo: Dynamische Typisierung .....	73
<b>5 Strings .....</b>	<b>80</b>
String-Literale .....	81
Strings in Aktion .....	88
String-Formatierung .....	94
String-Methoden .....	97
Generische Typenkategorien .....	102
<b>6 Listen und Dictionaries .....</b>	<b>104</b>
Listen .....	104
Listen in Aktion .....	106
Dictionaries .....	111
Dictionaries in Aktion .....	112
<b>7 Tupel, Dateien und alles andere .....</b>	<b>120</b>
Tupel .....	120
Mengen .....	123
Dateien .....	124
Typenkategorien, die Zweite .....	126
Allgemeinheit .....	126
Verweise und Kopien .....	128
Boolesche Wahrheitswerte .....	130

Vergleiche und Gleichheit .....	131
Typenhierarchien in Python .....	133
Weitere Typen in Python .....	135
Fallen bei eingebauten Typen .....	135
Übungen zu Teil II .....	137

---

## Teil III Anweisungen und Syntax

<b>8 Zuweisungen, Ausdrücke und Print .....</b>	<b>143</b>
Zuweisungen .....	144
Ausdrücke .....	151
Print-Anweisungen .....	152
<b>9 if-Tests .....</b>	<b>157</b>
if-Anweisungen .....	157
Python-Syntaxregeln .....	160
Wahrheitstests .....	164
<b>10 while und for-Schleifen .....</b>	<b>167</b>
while-Schleifen .....	167
break, continue, pass und das Schleifen-else .....	169
for-Schleifen .....	172
Schleifenvarianten .....	178
<b>11 Python-Code dokumentieren .....</b>	<b>184</b>
Das Python-Dokumentationsintermezzo .....	184
Häufige Fallen beim Codieren .....	196
Übungsaufgaben zu Teil III .....	197

---

## Teil IV Funktionen

<b>12 Funktionen .....</b>	<b>203</b>
Weshalb Funktionen? .....	203
Codierung von Funktionen .....	204
Ein erstes Beispiel: Definitionen und Aufrufe .....	207
Ein zweites Beispiel: Schnittmenge von Sequenzen .....	209

<b>13 Gültigkeitsbereiche und Argumente</b>	<b>212</b>
Gültigkeitsregeln	212
Die global-Anweisung	218
Gültigkeitsbereiche und verschachtelte Funktionen	219
Argumentübergabe	222
Spezielle Arten des Argumentabgleichs	225
<b>14 Funktionen für Fortgeschrittene</b>	<b>235</b>
Anonyme Funktionen: lambda	235
Funktionen auf Argumente anwenden	239
Funktionen auf Sequenzen anwenden	242
Werkzeuge bei der funktionalen Programmierung	243
Partielle Anwendung von Funktionen	244
List Comprehensions	245
Dekoratoren	249
Generatoren und Iteratoren	251
Konzepte beim Entwurf von Funktionen	255
Fallen bei Funktionen	258
Übungsaufgaben zu Teil IV	262

---

## Teil V Module

<b>15 Module: Ein Überblick</b>	<b>267</b>
Warum Module?	268
Architektur von Python-Programmen	268
Funktionsweise eines Imports	271
<b>16 Grundlagen der Modulerzeugung</b>	<b>278</b>
Erzeugung von Modulen	278
Benutzung von Modulen	279
Modulnamensräume	283
Nachladen von Modulen	288
<b>17 Modul-Pakete</b>	<b>292</b>
Grundlagen	292
Beispiele für Paket-Importe	295
Wozu Paket-Importe?	297
Eine Geschichte von drei Systemen	298

<b>18 Weiterführende Modul-Themen</b>	<b>301</b>
Datenkapselung in Modulen	301
Zukünftige Sprachmerkmale aktivieren	302
Gemischte Verwendungsmodi: <code>__name__</code> und <code>__main__</code>	303
Modulsuchpfad ändern	304
Die Import-Erweiterung »as«	304
Konzepte des Modulentwurfs	305
Fallstricke bei Modulen	308
Übungsaufgaben zu Teil V	315

---

## Teil VI Klassen und OOP

<b>19 Objektorientierung in Python</b>	<b>319</b>
Warum Klassen?	319
OOP aus 10.000 Meter Höhe betrachtet	321
<b>20 Grundlagen von Klassen</b>	<b>330</b>
Klassen erzeugen mehrere Instanz-Objekte	330
Klassen werden durch Vererbung spezialisiert	334
Klassen können Python-Operatoren abfangen	337
<b>21 Details der Klassenkodierung</b>	<b>341</b>
Die class-Anweisung	341
Methoden	344
Vererbung	347
Operatorenüberladung	351
Namensräume: Die ganze Geschichte	362
<b>22 Design mit Klassen</b>	<b>369</b>
Python und OOP	369
Klassen als Datensätze	371
OOP und Vererbung: »ist-ein«	372
OOP und Komposition: »hat-ein«	374
OOP und Delegation	378
Mehrfach-Vererbung	380
Klassen sind Objekte: Generische Objektfabriken	383
Methoden sind Objekte: gebunden oder ungebunden	385
Wiederholung: Dokumentations-Strings	387
Klassen und Module im Vergleich	388

<b>23 Weiterführende Themen zu Klassen</b>	<b>390</b>
Erweiterung eingebauter Typen	390
Pseudo-private Klassenattribute	394
Klassen »neuen Stils« in Python 2.2	396
Fallstricke bei Klassen	404
Übungsaufgaben zu Teil VI	407

---

## Teil VII Ausnahmen und Werkzeuge

<b>24 Grundlagen zu Ausnahmen</b>	<b>415</b>
Warum Ausnahmen?	415
Ausnahmebehandlung: Die Kurzfassung	417
Die try/except/else-Variante	421
Die try/finally-Variante	427
Die vereinheitlichte try/except/finally-Anweisung	428
Die raise-Anweisung	429
Die assert-Anweisung	431
<b>25 Ausnahme-Objekte</b>	<b>433</b>
Stringbasierte Ausnahmen	433
Klassenbasierte Ausnahmen	434
Allgemeine Formen der raise-Anweisung	442
<b>26 Entwurf mit Ausnahmen</b>	<b>444</b>
Verschachtelte Ausnahme-Handler	444
Code-Idiome zu Ausnahmen	448
Tipps zum Entwurf mit Ausnahmen	451
Fallstricke bei Ausnahmen	455
Zusammenfassung zum Sprachkern	456
Übungsaufgaben zu Teil VII	461

---

## Teil VIII Die Hülle

<b>27 Typische Aufgaben in Python</b>	<b>465</b>
Konvertierung, Zahlen und Vergleiche	470
Manipulation von Strings	474

Modifikation von Datenstrukturen .....	479
Bearbeiten von Dateien und Verzeichnissen .....	486
Internet-bezogene Module .....	501
Ausführung von Programmen .....	505
Fehlersuche, Testen, Zeitmessung, Optimierung .....	508
Übungsaufgaben .....	511
<b>28 Frameworks .....</b>	<b>513</b>
Ein automatisches Beschwerdesystem .....	514
Verbindung mit COM: Billige Öffentlichkeitsarbeit .....	520
Ein auf Tkinter basierender GUI-Editor zum Bearbeiten von Formulardaten ..	526
Jython: Die glückliche Ehe zwischen Python und Java .....	533
Übungsaufgaben .....	541
<b>29 Python-Ressourcen .....</b>	<b>542</b>
Verschiedene Ebenen der Gemeinschaft .....	542
Der Entwicklungsprozess .....	547
Dienstleistungen und Produkte .....	547
Der juristische Rahmen: Die Python Software Foundation .....	547
Software .....	548
Beliebte Software von Drittanbietern .....	549
Webanwendungs-Frameworks .....	559
Werkzeuge für Python-Entwickler .....	561

---

## Teil IX Anhang

<b>A Installation und Konfiguration .....</b>	<b>565</b>
<b>B Lösungen zu den Übungsaufgaben .....</b>	<b>572</b>
<b>Index .....</b>	<b>609</b>