

Thomas Theis

Aktuell zu  
Python 3.2  
und  
Python 2.7

# Einstieg in Python

Ideal für Programmieranfänger geeignet

3. Auflage

- ▶ Schritt für Schritt eigene Programme entwickeln
- ▶ Mit vielen Beispielen und Übungsaufgaben
- ▶ Inkl. Benutzeroberflächen, objektorientierte Programmierung, Datenbank- und Internet-Anwendungen u.v.m.



Python-Versionen 3.2 und 2.7 sowie  
alle Code-Beispiele des Buchs

Galileo Computing 

# Inhalt

<b>1 Einführung</b> .....	13
1.1 Vorteile von Python .....	13
1.2 Verbreitung von Python .....	14
1.3 Aufbau des Buchs .....	14
1.4 Übungen .....	16
1.5 Installation von Python unter Windows .....	16
1.6 Installation von Python unter Linux .....	17
<b>2 Erste Schritte</b> .....	19
2.1 Python als Taschenrechner .....	19
2.1.1 Eingabe von Berechnungen .....	19
2.1.2 Addition, Subtraktion und Multiplikation .....	20
2.1.3 Division, Ganzzahldivision und Modulo .....	20
2.1.4 Rangfolge und Klammern .....	22
2.1.5 Variablen und Zuweisung .....	22
2.2 Erstes Programm .....	24
2.2.1 Hallo Welt .....	24
2.2.2 Eingabe eines Programms .....	25
2.3 Speichern und Ausführen .....	25
2.3.1 Speichern .....	25
2.3.2 Ausführen unter Windows .....	27
2.3.3 Ausführen unter Linux .....	29
2.3.4 Kommentare .....	31
2.3.5 Verkettung von Ausgaben .....	31
2.3.6 Lange Ausgaben .....	31
<b>3 Programmierkurs</b> .....	33
3.1 Ein Spiel programmieren .....	33
3.1.1 Das fertige Spiel .....	33
3.1.2 Der Weg zum fertigen Spiel .....	33
3.1.3 Mögliche Erweiterungen .....	34
3.2 Variablen und Operatoren .....	34
3.2.1 Berechnung und Zuweisung .....	34
3.2.2 Eingabe einer Zeichenkette .....	35
3.2.3 Eingabe einer Zahl .....	36

3.2.4	Spiel, Version mit Eingabe .....	37
3.2.5	Zufallszahlen .....	38
3.3	Verzweigungen .....	40
3.3.1	Vergleichsoperatoren .....	40
3.3.2	Einfache Verzweigung .....	40
3.3.3	Spiel, Version mit Bewertung der Eingabe .....	41
3.3.4	Mehrfache Verzweigung .....	43
3.3.5	Logische Operatoren .....	44
3.3.6	Mehrere Vergleichsoperatoren .....	47
3.3.7	Spiel, Version mit genauer Bewertung der Eingabe .....	47
3.3.8	Rangfolge der Operatoren .....	49
3.4	Schleifen .....	49
3.4.1	for-Schleife .....	50
3.4.2	Schleifenabbruch mit »break« .....	51
3.4.3	Geschachtelte Kontrollstrukturen .....	52
3.4.4	Spiel, Version mit for-Schleife und Abbruch .....	53
3.4.5	for-Schleife mit range() .....	54
3.4.6	Spiel, Version mit »range()« .....	58
3.4.7	while-Schleife .....	59
3.4.8	Spiel, Version mit while-Schleife und Zähler .....	60
3.5	Fehler und Ausnahmen .....	62
3.5.1	Basisprogramm .....	62
3.5.2	Fehler abfangen .....	63
3.5.3	Eingabe wiederholen .....	65
3.5.4	Exkurs: Schleifenfortsetzung mit »continue« .....	66
3.5.5	Spiel, Version mit Ausnahmebehandlung .....	67
3.6	Funktionen und Module .....	69
3.6.1	Einfache Funktionen .....	70
3.6.2	Funktionen mit einem Parameter .....	72
3.6.3	Funktionen mit mehreren Parametern .....	73
3.6.4	Funktionen mit Rückgabewert .....	74
3.6.5	Spiel, Version mit Funktionen .....	75
3.7	Das fertige Spiel .....	77
<b>4</b>	<b>Datentypen .....</b>	<b>81</b>
4.1	Zahlen .....	81
4.1.1	Ganze Zahlen .....	81
4.1.2	Zahlen mit Nachkommastellen .....	83
4.1.3	Operator ** .....	84
4.1.4	Rundung und Konvertierung .....	85

4.1.5	Modul »math«	86
4.1.6	Brüche	88
4.2	Zeichenketten	91
4.2.1	Eigenschaften	91
4.2.2	Operatoren	93
4.2.3	Operationen	94
4.2.4	Funktionen	96
4.2.5	Umwandlung von einer Zeichenkette in eine Zahl	100
4.2.6	Umwandlung von einer Zahl in eine Zeichenkette	102
4.2.7	Datentyp »bytes«	103
4.3	Listen	103
4.3.1	Eigenschaften	104
4.3.2	Operatoren	106
4.3.3	Funktionen und Operationen	107
4.4	Tupel	110
4.4.1	Eigenschaften	110
4.4.2	Operationen	111
4.4.3	Tupel entpacken	112
4.5	Dictionarys	115
4.5.1	Eigenschaften	115
4.5.2	Funktionen	117
4.5.3	Views	118
4.5.4	Vergleiche	120
4.6	Mengen, Sets	121
4.6.1	Eigenschaften	121
4.6.2	Funktionen	123
4.6.3	Operatoren	124
4.6.4	Frozenset	126
4.7	Wahrheitswerte und Nichts	127
4.7.1	Wahrheitswerte True und False	127
4.7.2	Nichts, None	131
4.8	Referenz, Identität und Kopie	133
4.8.1	Referenz und Identität	133
4.8.2	Ressourcen sparen	135
4.8.3	Objekte kopieren	136
<b>5</b>	<b>Weiterführende Programmierung</b>	<b>139</b>
5.1	Allgemeines	139
5.1.1	Kombinierte Zuweisungsoperatoren	139
5.1.2	Programmzeile in mehreren Zeilen	141

5.1.3	Eingabe mit Hilfestellung .....	142
5.1.4	Anweisung »pass« .....	144
5.1.5	Funktionen »eval()« und »exec()« .....	145
5.2	Ausgabe und Formatierung .....	147
5.2.1	Funktion »print()« .....	147
5.2.2	Formatierte Ausgabe .....	149
5.3	Conditional Expression .....	154
5.4	Iterierbare Objekte .....	155
5.4.1	Funktion »zip()« .....	155
5.4.2	Funktion »map()« .....	156
5.4.3	Funktion »filter()« .....	158
5.5	List Comprehension .....	159
5.6	Fehler und Ausnahmen .....	161
5.6.1	Allgemeines .....	161
5.6.2	Syntaxfehler .....	162
5.6.3	Laufzeitfehler .....	164
5.6.4	Logische Fehler und Debugging .....	164
5.6.5	Fehler erzeugen .....	168
5.6.6	Unterscheidung von Ausnahmen .....	170
5.7	Funktionen .....	172
5.7.1	Variable Anzahl von Parametern .....	172
5.7.2	Benannte Parameter .....	173
5.7.3	Voreinstellung von Parametern .....	174
5.7.4	Mehrere Rückgabewerte .....	175
5.7.5	Übergabe von Kopien und Referenzen .....	177
5.7.6	Lokal, global .....	179
5.7.7	Lambda-Funktion .....	181
5.8	Eingebaute Funktionen .....	181
5.8.1	Funktionen max(), min() und sum() .....	183
5.8.2	Funktionen »chr()« und »ord()« .....	184
5.8.3	Funktionen »reversed()« und »sorted()« .....	185
5.9	Eigene Module .....	186
5.9.1	Eigene Module erzeugen .....	186
5.9.2	Eigene Module verwenden .....	187
5.10	Parameter der Kommandozeile .....	188
5.10.1	Übergabe von Zeichenketten .....	188
5.10.2	Übergabe von Zahlen .....	189
5.10.3	Beliebige Anzahl von Parametern .....	190

<b>6</b>	<b>Objektorientierte Programmierung</b>	191
6.1	Was ist OOP?	191
6.2	Klassen, Objekte und eigene Methoden	192
6.3	Konstruktor und Destruktor	194
6.4	Besondere Methoden	196
6.5	Operatormethoden	197
6.6	Referenz, Identität und Kopie	199
6.7	Vererbung	201
6.8	Mehrfachvererbung	204
6.9	Spiel, objektorientierte Version	206
<b>7</b>	<b>Verschiedene Module</b>	211
7.1	Datum und Zeit	211
7.1.1	Spielen mit Zeitangabe	211
7.1.2	Aktuelle Zeit ausgeben	211
7.1.3	Zeitangabe erzeugen	215
7.1.4	Mit Zeitangaben rechnen	216
7.1.5	Programm anhalten	219
7.1.6	Spiel, Version mit Zeitmessung	220
7.1.7	Spiel, objektorientierte Version mit Zeitmessung	222
7.2	Modul »collections«	223
7.3	Multithreading	225
7.3.1	Wozu dient Multithreading?	226
7.3.2	Erzeugung eines Threads	226
7.3.3	Identifizierung eines Threads	227
7.3.4	Gemeinsame Objekte	229
7.3.5	Threads und Exceptions	230
7.4	Reguläre Ausdrücke	232
7.4.1	Suchen von Teiltextrn	232
7.4.2	Ersetzen von Teiltextrn	236
<b>8</b>	<b>Dateien</b>	239
8.1	Dateitypen	239
8.2	Öffnen und Schließen einer Datei	240
8.3	Sequentielle Dateien	241
8.3.1	Sequentielles Schreiben	241
8.3.2	Sequentielles Lesen	243

8.3.3	CSV-Datei schreiben .....	248
8.3.4	CSV-Datei lesen .....	250
8.4	Dateien mit festgelegter Struktur .....	253
8.4.1	Formatiertes Schreiben .....	253
8.4.2	Lesen an beliebiger Stelle .....	254
8.4.3	Schreiben an beliebiger Stelle .....	256
8.5	Serialisierung .....	257
8.5.1	Objekte in Datei schreiben .....	258
8.5.2	Objekte aus Datei lesen .....	260
8.6	Bearbeitung mehrerer Dateien .....	261
8.7	Informationen über Dateien .....	263
8.8	Dateien und Verzeichnisse verwalten .....	264
8.9	Spiel, Version mit Highscore-Datei .....	266
8.9.1	Eingabebeispiel .....	266
8.9.2	Aufbau des Programms .....	267
8.9.3	Code des Programms .....	267
8.10	Spiel, objektorientierte Version mit Highscore-Datei .....	272
<b>9</b>	<b>Internet .....</b>	<b>277</b>
9.1	Laden und Senden von Internetdaten .....	277
9.1.1	Lokaler Webserver unter Windows .....	278
9.1.2	Lokaler Webserver unter Linux .....	279
9.1.3	Daten lesen .....	279
9.1.4	Daten kopieren .....	282
9.1.5	Daten senden per GET .....	283
9.1.6	Daten senden per POST .....	287
9.2	Webserver-Programmierung .....	290
9.2.1	Erstes Programm .....	290
9.2.2	Beantworten einer Benutzereingabe .....	292
9.2.3	Formularelemente mit mehreren Werten .....	295
9.2.4	Typen von Formularelementen .....	298
9.3	Browser aufrufen .....	304
9.4	Spiel, Version für das Internet .....	304
9.4.1	Eingabebeispiel .....	305
9.4.2	Aufbau des Programms .....	307
9.4.3	Code des Programms .....	308

<b>10 Datenbanken</b> .....	315
10.1 Aufbau von Datenbanken .....	315
10.2 SQLite .....	316
10.2.1 Datenbank, Tabelle und Datensätze .....	316
10.2.2 Daten anzeigen .....	319
10.2.3 Daten auswählen, Operatoren .....	320
10.2.4 Operator »LIKE« .....	323
10.2.5 Sortierung der Ausgabe .....	324
10.2.6 Auswahl nach Eingabe .....	326
10.2.7 Datensätze ändern .....	327
10.2.8 Datensätze löschen .....	329
10.3 SQLite auf dem Webserver .....	331
10.4 Spiel, Version mit Highscore-Datenbank .....	333
10.5 Spiel, objektorientierte Version mit Highscore-Datenbank .....	337
<b>11 Benutzeroberflächen</b> .....	339
11.1 Einführung .....	339
11.1.1 Eine erste GUI-Anwendung .....	340
11.1.2 Ändern von Eigenschaften .....	342
11.2 Widget-Typen .....	343
11.2.1 Anzeigefeld, Label .....	344
11.2.2 Einzeilige Textbox, Entry .....	347
11.2.3 Versteckte Eingabe .....	349
11.2.4 Mehrzeilige Textbox, Text .....	351
11.2.5 Scrollende Textbox, ScrolledText .....	353
11.2.6 Listbox mit einfacher Auswahl .....	354
11.2.7 Listbox mit mehrfacher Auswahl .....	357
11.2.8 Scrollbar, scrollende Widgets .....	359
11.2.9 Radiobuttons zur Auswahl, Widget-Variablen .....	361
11.2.10 Radiobuttons zur Auswahl und Ausführung .....	363
11.2.11 Checkbuttons zur mehrfachen Auswahl .....	365
11.2.12 Schieberegler, Scale .....	367
11.2.13 Mausereignisse .....	370
11.2.14 Tastaturereignisse .....	373
11.3 Geometrische Anordnung von Widgets .....	375
11.3.1 Frame-Widget, Methode pack() .....	376
11.3.2 Ein einfacher Taschenrechner .....	378
11.3.3 Methode grid() .....	383
11.3.4 Methode place(), absolute Koordinaten .....	385



11.3.5	Methode place(), relative Koordinaten .....	387
11.3.6	Absolute Veränderung von Koordinaten .....	389
11.3.7	Relative Veränderung von Koordinaten .....	391
11.4	Menüs, Messageboxen und Dialogfelder .....	394
11.4.1	Menüleisten .....	395
11.4.2	Kontextmenüs .....	400
11.4.3	Messageboxen .....	404
11.4.4	Eigene Dialogfelder .....	409
11.4.5	Ausführung verhindern .....	411
11.5	Spiel, GUI-Version .....	412
<b>12</b>	<b>Neues in Python 3</b> .....	<b>419</b>
12.1	Neue und geänderte Eigenschaften .....	419
12.1.1	Auffällige Änderungen .....	419
12.1.2	Weitere Änderungen .....	420
12.2	Konvertierung von Python 2 zu Python 3 .....	421
<b>13</b>	<b>Lösungen</b> .....	<b>423</b>
	Index .....	431

# 1 Einführung

In diesem Kapitel stelle ich Ihnen Python kurz vor. Sie lernen die Vorteile von Python kennen und erfahren, wie Sie Python unter Windows und unter Linux installieren.

## 1.1 Vorteile von Python

Python ist eine sehr einfach zu erlernende Programmiersprache und für den Einstieg in die Welt der Programmierung ideal geeignet. Trotz ihrer Einfachheit bietet diese Sprache auch die Möglichkeit, komplexe Programme für vielfältige Anwendungsgebiete zu schreiben.

Leicht zu lernen

Python eignet sich besonders zur schnellen Entwicklung umfangreicher Anwendungen. Diese Technik ist unter dem Stichwort RAD (= *Rapid Application Development*) bekannt geworden. Python vereint zu diesem Zweck folgende Vorteile:

- ▶ Eine einfache, eindeutige Syntax: Python ist eine ideale Programmiersprache für Einsteiger. Sie beschränkt sich auf einfache, klare Anweisungen. In anderen Programmiersprachen werden vielfältige Lösungswege für das gleiche Problem angeboten, so dass der Entwickler weniger geleitet als verunsichert wird. Python beschränkt sich dagegen häufig auf einen einzigen möglichen Lösungsweg. Dieser prägt sich schnell ein und wird dem Entwickler vertraut.
- ▶ Klare Strukturen: Python zwingt den Entwickler, in einer gut lesbaren Struktur zu schreiben. Die Anordnung der Programmzeilen ergibt gleichzeitig die logische Struktur des Programms.
- ▶ Wiederverwendung von Code: Die Modularisierung, also die Zerlegung eines Problems in Teilprobleme und die anschließende Zusammenführung der Teillösungen zu einer Gesamtlösung, wird in Python sehr leicht gemacht. Die vorhandenen Teillösungen können unkompliziert für weitere Aufgabenstellungen genutzt werden, so dass Sie als Entwickler bald über einen umfangreichen Pool an Modulen verfügen.
- ▶ Objektbearbeitung: In Python werden alle Daten als Objekte gespeichert. Dies führt zu einer einheitlichen Datenbehandlung für Objekte

Einfach

Klar

Wiederverwendbar

Einheitliche Objekte

unterschiedlichen Typs. Andererseits erfolgt die physikalische Speicherung der Objekte von Python automatisch, also ohne Eingriff des Entwicklers. Der Entwickler muss sich nicht um die Reservierung und Freigabe geeigneter Speicherbereiche kümmern.

- Interpretiert ▶ Interpreter/Compiler: Python-Programme werden unmittelbar interpretiert. Sie müssen nicht, wie bei vielen anderen Programmiersprachen, erst kompiliert und gebunden werden. Dies ermöglicht einen häufigen, schnellen Wechsel zwischen Codierungs- und Testphase.
- Unabhängig ▶ Unabhängigkeit vom Betriebssystem: Sowohl Programme, die von der Kommandozeile aus bedient werden, als auch Programme mit grafischen Benutzeroberflächen können auf unterschiedlichen Betriebssystemen (Windows, Unix/Linux, Mac OS) ohne Neuentwicklung und Anpassung eingesetzt werden.

## 1.2 Verbreitung von Python

Aufgrund seiner vielen Vorzüge gehört Python zu den beliebtesten Programmiersprachen überhaupt. So wird es z. B. innerhalb des Projekts *100-Dollar-Laptop*, das der Schulausbildung von Kindern in aller Welt dient, für die Benutzeroberfläche verwendet. Aber auch in zahlreichen großen Unternehmen wird Python eingesetzt, hier ein paar Beispiele:

- ▶ Google: Python ist eine der drei offiziellen Programmiersprachen. (Guido van Rossum, der Entwickler von Python, arbeitet für Google.)
- ▶ YouTube: Wurde zum großen Teil mit Hilfe von Python entwickelt.
- ▶ NASA: Nutzt Python zur Softwareentwicklung im Zusammenhang mit den Space-Shuttle-Missionen.
- ▶ Industrial Light & Magic: Auch Hollywood setzt auf Python – die Produktionsfirma ILM (Star Wars, Indiana Jones, Fluch der Karibik) nutzt es z. B. bei der Entwicklung von Spezialeffekten.
- ▶ Honeywell: Python wird weltweit in vielen Firmen zur allgemeinen Hardware- und Softwareentwicklung eingesetzt.

## 1.3 Aufbau des Buchs

- Aktiv lernen Das vorliegende Buch soll Sie in die Programmiersprache Python einführen. Es wird besonderer Wert darauf gelegt, dass Sie selbst praktisch mit

Python arbeiten. Daher möchte ich Sie von Anfang an dazu auffordern, dem logischen Faden von Erklärungen und Beispielen zu folgen.

Python 3 ist die aktuelle Version der Programmiersprache und an vielen Stellen nicht abwärtskompatibel mit der alten Version Python 2. Allerdings ist Python 2 weiterhin sehr verbreitet und wird noch auf Jahre hinaus von den Python-Entwicklern unterstützt werden. Alle Erläuterungen und Programme dieses Buchs liegen für beide Versionen vor, damit stehen Ihnen beide Möglichkeiten offen. Auf die Unterschiede der beiden Versionen gehe ich besonders ein, so dass ein späterer Umstieg von der einen zur anderen Version leichtfällt.

Python-Versionen

Erste Zusammenhänge werden in Kapitel 2, »Erste Schritte«, anhand von einfachen Berechnungen vermittelt. Außerdem lernen Sie, ein Programm einzugeben, zu speichern und es unter den verschiedenen Umgebungen auszuführen.

Sie sollen die Sprache spielerisch kennenlernen. Daher wird Sie ein selbst programmiertes Spiel durch das Buch begleiten. Dieses Spiel wird mit dem »Programmierkurs« in Kapitel 3 eingeführt und im weiteren Verlauf des Buchs kontinuierlich erweitert und verbessert.

Spielerisch lernen

Nach der Vorstellung der verschiedenen Datentypen mit ihren jeweiligen Eigenschaften und Vorteilen in Kapitel 4, »Datentypen«, werden die Programmierkenntnisse in Kapitel 5, »Weiterführende Programmierung«, vertieft.

Kapitel 6, »Objektorientierte Programmierung«, widmet sich der objektorientierten Programmierung mit Python.

Objektorientiert

Einige nützliche Module zur Ergänzung der Programme werden in Kapitel 7, »Verschiedene Module«, vorgestellt.

In den Kapiteln 8, »Dateien«, und 10, »Datenbanken«, lernen Sie, Daten dauerhaft in Dateien oder Datenbanken zu speichern. Python wird zudem in der Internetprogrammierung eingesetzt. Die Zusammenhänge zwischen Python und dem Internet vermittelt Kapitel 9, »Internet«.

Sowohl Windows als auch Linux bieten komfortable grafische Benutzeroberflächen. Kapitel 11, »Benutzeroberflächen«, beschäftigt sich daher mit der Erzeugung von Benutzeroberflächen mit Hilfe des Moduls `tkinter`. Dieses stellt eine Schnittstelle zwischen dem grafischen Toolkit *Tk* und Python dar.

Grafische Oberflächen

**Python-Versionen** Python gibt es in den Versionen 2 und 3, die parallel nebeneinander bestehen. Sofern es keine Gründe gibt, die dagegensprechen, starten Sie am besten mit Python 3, das auch den Beispielen zugrunde liegt. Auf etwaige Abweichungen der Version 2 weise ich Sie in den Erläuterungen und Programmen in diesem Buch aber natürlich jederzeit hin. Kapitel 12 fasst die wesentlichen Unterschiede zwischen Python 2 und 3, die besonders für Einsteiger interessant sind, dann noch einmal übersichtlich zusammen.

Für die Hilfe bei der Erstellung dieses Buchs bedanke ich mich bei Petra Biedermann und dem ganzen Team von Galileo Press, ganz besonders bei Christine Siedle.

## 1.4 Übungen

In einigen Kapiteln finden Sie Übungsaufgaben, die Sie unmittelbar lösen sollten. Auf diese Weise können Sie Ihre Kenntnisse prüfen, bevor Sie zum nächsten Thema übergehen. Die Lösungen der Übungsaufgaben finden Sie in Kapitel 13 am Ende des Buchs. Dabei ist Folgendes zu beachten:

- Viele Lösungen** ▶ Es gibt für jedes Problem viele richtige Lösungen. Sie sollten sich also nicht davon beunruhigen lassen, dass Ihre Lösung eventuell nicht genauso aussieht wie die angegebene. Betrachten Sie die angegebene Lösung vielmehr als Anregung, was Sie anders und gegebenenfalls besser machen können.
- ▶ Bei der eigenen Lösung der Aufgaben wird sicherlich der eine oder andere Fehler auftreten – lassen Sie sich dadurch nicht entmutigen ...
- Aus Fehlern lernen** ▶ ... denn nur aus Fehlern kann man lernen. Auf die vorgeschlagene Art und Weise werden Sie Python wirklich erlernen – nicht allein durch das Lesen von Programmierregeln.

## 1.5 Installation von Python unter Windows

**Software auf CD** Python ist eine frei verfügbare Programmiersprache, die auf verschiedenen Betriebssystemplattformen eingesetzt werden kann. Auf der CD zu diesem Buch finden sich die Versionen 2.7 und 3.2 für Windows. Die jeweils neueste Version für Python 3 können Sie von der offiziellen Python-Website <http://www.python.org> aus dem Internet laden. Die Ver-

sion 2.7 wird die letzte Version für Python 2 bleiben und von den Python-Entwicklern noch auf Jahre hinaus unterstützt werden.

Rufen Sie zur Installation unter Windows die ausführbare Datei *python-3.2.msi* (bzw. *python-2.7.1.msi*) auf. Die Voreinstellungen des Installationsvorganges können Sie unverändert übernehmen. Dabei wird Python im Verzeichnis *C:\Python32* (bzw. *C:\Python27*) installiert. Anschließend verfügen Sie im Startmenü über einen Eintrag PYTHON 3.2 (bzw. PYTHON 2.7), siehe Abbildung 1.1.

Installation

Falls Sie sich mit beiden Versionen beschäftigen möchten: kein Problem. Sie können parallel installiert und benutzt werden.

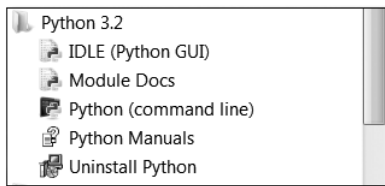


Abbildung 1.1 Startmenü Python

Der Eintrag *IDLE*, den Sie nach der Installation im Startmenü sehen, ist eine Entwicklungsumgebung, die selbst in Python geschrieben wurde und mit der Sie im Folgenden Ihre Programme schreiben werden.

IDLE

## 1.6 Installation von Python unter Linux

Stellvertretend für andere Linux-Distributionen wird in diesem Buch Ubuntu Linux 10.10 genutzt. Python 2 ist unter Ubuntu bereits installiert, weil Ubuntu selbst es für viele Funktionen benötigt. Es sollte auch nicht deinstalliert werden.

Ubuntu

Python 3 lässt sich leicht zusätzlich installieren. Sie finden es im UBUNTU SOFTWARE CENTER unter SOFTWARE INSTALLIEREN • ENTWICKLUNGSWERKZEUGE • PYTHON.

Dort finden Sie auch IDLE, sowohl für Python 2.7 als auch für Python 3.1. Bei der Installation von IDLE wird die zugehörige Python-Version (2.7 bzw. 3.1.2) direkt mit installiert.

IDLE

Es ist kein Problem, beide Versionen von Python parallel zu installieren und zu benutzen. Das Gleiche gilt für die beiden Versionen von IDLE.

# Index

- (minus)
  - Set* 125
  - Subtraktion* 20
- != (ungleich) 40
- # (Kommentar) 31
- \$\_GET 285
- \$\_POST 288
- % (Prozent)
  - Format* 152, 421
  - Modulo* 21, 88
- %= 141
- & 125
- ' 91
- \* (Stern)
  - Multiplikation* 20
  - Parameter* 172
  - Vervielfachung* 93, 106
- \*\* (Potenz) 84
- \*\*= 140
- \*= 140
- + (plus)
  - Addition* 20
  - Verkettung* 93, 106
- += 140
- / (Division) 20
- // (Ganzzahldivision) 21
- //= 141
- /= 141
- :(Doppelpunkt) 41, 60
- < (kleiner)
  - Format* 153
  - Python* 40
  - SQL* 322
- <= (kleiner gleich)
  - Python* 40
  - SQL* 322
- <> (ungleich) 40
  - Python* 420
  - SQL* 322
- = 140
- =
  - Python (Zuweisung)* 23, 34
  - SQL (gleich)* 322
- == (gleich) 40, 121, 133
- > (größer)
  - Format* 153
  - Python* 40
  - SQL* 322
- >= (größer gleich)
  - Python* 40
  - SQL* 322
- [ ], Dictionary 116
- ^ (hoch) 125
- \_\_add\_\_ 198
- \_\_cmp\_\_ 420
- \_\_del\_\_ 196
- \_\_eq\_\_ 198
- \_\_floordiv\_\_ 198
- \_\_ge\_\_ 198
- \_\_gt\_\_ 198
- \_\_init\_\_ 195
- \_\_le\_\_ 198
- \_\_lt\_\_ 198
- \_\_mod\_\_ 198
- \_\_mul\_\_ 198
- \_\_ne\_\_ 198
- \_\_pow\_\_ 198
- \_\_str\_\_ 197
- \_\_sub\_\_ 198
- \_\_truediv\_\_ 198
- \_\_thread
  - get\_ident()* 227
  - Modul* 226
  - start\_new\_thread()* 226
- { }
  - Dictionary* 116
  - Formatierung* 151
- | (oder) 125
- 0b 83
- 0o 83
- 0x 83
- 100-Dollar-Laptop 14
- 2to3 421

## A

---

- abs() 91
- add() 124
- add\_cascade() 399

add\_checkbutton() 399  
 add\_command() 398  
 add\_radiobutton() 399  
 add\_separator() 399  
 Aktionsabfrage (Datenbank) 318  
 Alt-Taste (Linux) 373  
 anchor 344  
 and 44  
 AND (SQL) 322  
 Antwort-Box (GUI) 404  
 Anweisung  
   *ausführen* 145  
   *leere* 144  
 Anzahl  
   *Elemente* 94  
   *Teiltex*te 97  
 Anzeigefeld (GUI) 344  
 Apache-Webserver 278  
 append() 110, 225  
 appendleft() 225  
 APT 279  
 argv  
   sys 188  
 Array 104  
 as 171  
 ASCII-Zeichen 103  
 askokcancel() 404  
 Ausdruck  
   *auswerten* 145  
   *bedingter* 154  
 Ausgabe  
   *formatieren* 149  
   *lange* 31  
 Ausnahme 62  
   Thread 230  
 Ausnahmebehandlung 64  
 Ausrichtung (GUI) 344  
 Auswahlabfrage (Datenbank) 319  
 Auswahlmenü  
   *einfaches (GUI)* 354  
   *mehrfaches (GUI)* 357  
 Auswahlmenü (HTML) 299

## B

---

b  
   *Format* 153  
   *Prefix* 103  
 Basisverzeichnis 278, 279

Bedingter Ausdruck 154  
 Bedingung 40, 60, 127  
   *verknüpfen* 44  
 Benannter Parameter 173, 420  
 Benutzeroberfläche 339  
 Betrag 91  
 bg 344  
 Bild einbinden (GUI) 344  
 Bild-Widget 346  
 bin() 82  
 Binärdaten 103  
 bind() 372, 374  
 Bogenmaß 87  
 bool 128  
 borderwidth 344  
 break 51  
 Breakpoint 168  
 Breite (GUI) 344  
 Browser 280  
   *aufrufen* 304  
 Bruch 88  
   *annähern* 90  
 Button 341  
 Button-Widget 341  
 Byte-Literal 103, 281  
 bytes 103

## C

---

Callback (GUI) 341  
 Carriage Return 103  
 cgi  
   *FieldStorage* 294  
   *Modul* 290, 294  
 cgi-bin 290, 293  
 CGI-Skript 290  
   *Standardverzeichnis* 293  
 cgitb  
   *enable()* 294  
   *Modul* 294  
 checkbox (HTML) 301  
 Checkbutton-Widget 365  
 checked (HTML) 300  
 chmod 30  
 chr() 184  
 class 192  
 clear() 124  
 close() 241  
   *connection* 319



cmd 28  
 collections  
   *deque()* 225  
   *Modul* 223  
 command 341, 363, 367  
 Comma-separated Values 248  
 commit()  
   *connection* 318  
 Compiler 14  
 Conditional Expression 154  
 configure() 343  
 connect() 318  
 connection  
   *close()* 319  
   *commit()* 318  
   *cursor()* 318  
   *sqlite3* 318  
 Content-type 291  
 continue 67  
 copy  
   *deepcopy()* 136, 199  
   *Modul* 136, 199  
 copy() 123  
 copyfile() 265  
 cos() 86  
 count() 110  
   *str* 97  
 cp1252 26, 291  
 CREATE TABLE (SQL) 318  
 CSV-Datei 248  
 cursor() 318

## D

---

d (Format) 153  
 Datei  
   *ausführbare* 30  
   *Eigenschaften* 263  
   *Ende erkennen* 246  
   *festgelegte Struktur* 253  
   *formatierte Ausgabe* 253  
   *Größe* 264  
   *kopieren* 265  
   *lesen* 243  
   *lesen und schreiben* 256  
   *Liste erstellen* 261  
   *löschen* 265  
   *öffnen* 240  
   *Position verändern* 254  
   Datei (Forts.)  
     *schließen* 241  
     *schreiben* 241  
     *speichern* 239  
     *Typ* 239  
     *umbenennen* 265  
     *verwalten* 264  
     *Zugriffzeitpunkt* 264  
   Dateiendung  
     *.csv* 248  
     *.py* 25  
   Dateizugriff  
     *binärer* 239  
     *sequentieller* 239  
     *wahlfreier* 239  
   Datenbank 315  
     *erzeugen* 316  
     *im Internet* 331  
     *Verbindung herstellen* 318  
   Datenbankmanagement-System 316  
   Datensatz 315  
     *ändern* 327  
     *anlegen* 316, 318  
     *Anzahl begrenzen* 335  
     *anzeigen* 319  
     *auswählen* 320  
     *Cursor* 318  
     *sortieren* 324  
   Datentyp  
     *bool* 128  
     *bytes* 103  
     *Datenbank* 315  
     *deque* 223  
     *ermitteln* 87  
     *float* 83  
     *FLOAT (SQL)* 316  
     *int* 81  
     *INTEGER (SQL)* 316  
     *long* 420  
     *NoneType* 131  
     *spezieller* 223  
     *str* 91  
     *TEXT (SQL)* 316  
   Datum und Zeit 211  
     *Differenz* 216  
     *erzeugen* 215  
     *formatieren* 211, 213  
     *Nullpunkt* 211  
   Debug Control 165

- Debug Off 168
- Debug On 165
- Debugger 164
- Decoding 26, 291
- deepcopy() 136, 199
- def 71
  - Klassenmethode* 192
- del 107, 117, 135
- DELETE (SQL) 329
- denominator 89
- deque 223
- deque() 225
- Deserialisierung 257
- destroy() 341
- Destruktor 194
- Dezimalformat 153
- Dezimalzahl 81
- Dezimalzeichen 20
  - ändern* 276
- Dialogfeld
  - eigenes* 409
  - modales* 412
- Dictionary 115
  - Element löschen* 117
  - Elementexistenz* 117
  - erzeugen* 116
  - Funktionen* 117
  - leeres* 128
  - vergleichen* 120
  - View* 118
- Differenzmenge 126
- discard() 124
- Double-ended Queue 223
- DoubleVar 362
- Duale Zahl 81
- Dualformat 153
- dump() 258
- Eingabe (Forts.)
  - versteckte (GUI)* 349
  - versteckte (HTML)* 299
  - wiederholen* 66
  - Zeichenkette* 35, 92
- Eingabeaufforderung 29
- Eingabefeld
  - einzeliges (GUI)* 347
  - HTML* 299
  - mehrzeiliges (GUI)* 351
  - scrollbares (GUI)* 353
- Eingebaute Funktion 181
- Einrücken 60
  - doppelt* 52
- Einzelschrittverfahren 164
- elif 44
- else 41, 44
- Elternelement (GUI) 376
- enable() 294
- Encoding 26, 291
- end 147
- Endlosschleife 128, 339
- endswith() 99
- Entry-Widget 347
- Entwicklungsumgebung 17
- Escape-Sequenz 103
- Eszett 26, 292
- EVA (Eingabe, Verarbeitung, Ausgabe) 37
- eval() 145, 383
- Excel 250, 272
- except 64, 169
- exec() 145
- execute() 318
- exit 28
- Exponentialformat 151
- extend() 225
- extendleft() 225

## E

---

- e
  - Eulersche Zahl* 86
  - Exponentialschreibweise* 84
  - Format* 151
- Eigenschaft 192
- Eindeutiger Index (Datenbank) 317, 329
- Eingabe
  - kommentierte* 142
  - nicht sinnvolle* 170

## F

---

- f (Format) 151
- Falsch 128
- False 128
- Fehler 62, 161
  - abfangen* 63, 65
  - erzeugen* 168
  - im Browser anzeigen* 295
  - Laufzeitfehler* 164
  - logischer* 164

Fehler (Forts.)  
*Nachricht (GUI)* 404  
*Syntaxfehler* 162  
*Unterscheidung* 170  
Feld (Datenbank) 315  
Fenster  
  *öffnen* 341  
  *schließen* 341  
fg 344  
FieldStorage  
  *cgi* 294  
  *value* 294  
filter() 158  
find() 98  
findall() 232  
Fließkommazahl 83  
float 83  
FLOAT (SQL) 316  
float() 36, 100  
font 344  
for 50  
form (HTML) 284  
format() 149, 254  
Fraction() 89  
fractions  
  *Fraction()* 89  
  *gcd()* 89  
  *Modul* 88  
Frame-Widget 376  
from 187  
Frozenset 126  
frozenset() 127  
Funktion 69  
  *Aufruf* 71  
  *beenden* 75  
  *benannter Parameter* 173  
  *Definition* 71  
  *eingebaute* 181  
  *mehrere Rückgabewerte* 175  
  *Mehrfachaufruf* 156  
  *Name* 72  
  *Parameter, ein* 72  
  *Parameter, mehrere* 73  
  *Parameterübergabe* 177  
  *Parametervoreinstellung* 174  
  *Rückgabewert* 74  
  *trigonometrische* 86  
  *überspringen* 167

Funktion (Forts.)  
  *variable Parameteranzahl* 172  
  *verlassen* 167

## G

---

Ganze Zahl 81  
Ganzzahldivision 21  
  *Rest* 21  
Ganzzahligkeit ermitteln 88  
gcd() 89  
Geometriemanager (GUI) 341, 375, 383,  
  385  
geometry() 403  
get() 348, 356, 363  
get\_ident() 227  
GET-Methode 284  
Gleich  
  *in Python* 40  
  *in SQL* 322  
glob() 262  
global 180  
Graphical User Interface 339  
grid() 383  
Groß- und Kleinschreibung 24  
Größter gemeinsamer Teiler 89  
GUI-Anwendung 339

## H

---

Haltepunkt 168  
has\_key() 421  
Header 291  
height 344  
hex() 82  
Hexadezimale Zahl 82  
Hexadezimalformat 153  
Hintergrundfarbe (GUI) 344  
Höhe (GUI) 344  
htdocs 278, 279  
HTML  
  *Datei kopieren* 282  
  *Datei lesen* 279  
  *Formular* 293  
  *Formulardaten* 294  
  *Formularelement* 295  
  *lernen* 280  
http 278, 279

**I**


---

Identität 133  
 IDLE 17, 19, 164  
     *Fehler* 180  
 if 41  
 image 344  
 import 39, 187  
 in  
     *Dictionary* 117  
     *Element-View* 120  
     *for* 50  
     *Key-View* 120  
     *Liste* 106  
     *Sequenz* 93  
     *Set* 122  
     *Werte-View* 119  
 Index einer Sequenz 94  
 index() 110  
 Info-Nachricht (GUI) 404  
 input (HTML) 284  
 input() 142  
 INSERT INTO (SQL) 318  
 insert() 110, 352, 356  
 Instanz 193  
 int 81  
 int() 36, 85, 100  
 INTEGER (SQL) 316  
 Internet 277  
     *Daten lesen* 279  
     *Daten senden* 283, 287  
     *Datenbank im* 331  
 Interpreter 14  
 IntVar 362  
 is 133, 201  
 items() 120  
 Iterable 155  
     *filtern* 158  
     *verbinden* 155  
 Iterierbares Objekt 155

**J**


---

Ja-Nein-Box (GUI) 404

**K**


---

Kalenderwoche 214  
 keys() 119

Klammersetzung 22  
 Klasse 191  
     *abgeleitete* 201  
     *Basisklasse* 201  
     *Definition* 192  
 Kommandozeile 27, 188  
 Kommentar 31  
 Konstruktor 194  
 Kontrollkästchen  
     *HTML* 299  
 Kontrollstruktur, geschachtelte 52  
 Konvertierung 421  
 Kopfrechnen 33

**L**


---

Label-Widget 344  
 Lambda 181  
 Lange Ausgabe 31  
 Laufzeitfehler 164  
 Layoutmanager (GUI) 375  
 len() 94  
 LIKE (SQL) 323  
 LIMIT 335  
 limit\_denominator() 90  
 Line Feed 103  
 Linksbündig 153  
 Linux 14, 17  
     *Terminal* 29, 188  
 List Comprehension 159  
 Listbox-Widget 354, 357  
 Liste 103  
     *Anzahl bestimmter Elemente* 110  
     *eingebettete* 105  
     *Element anfügen* 110  
     *Element einfügen* 110  
     *Element löschen* 107, 110  
     *filtern* 159  
     *leere* 128  
     *mehrdimensionale* 105  
     *Operation* 107  
     *Position ermitteln* 110  
     *sortieren* 110  
     *umdrehen* 110  
 load() 260  
 localhost 278, 279  
 localtime() 211  
 Logischer Fehler 164  
 Logischer Operator 44

Lokaler Namensraum 179  
 Lokaler Webserver 278, 279  
 long 420  
 Lösungen der Übungsaufgaben 16

## M

---

mainloop() 341  
 map() 156  
 math  
   *cos()* 86  
   *e* 86  
   Modul 86  
   *pi* 86  
   *sin()* 86  
   *tan()* 86  
 Mausereignis (GUI) 370  
 max() 183  
 Maximum 183  
 Menge 121  
   *Differenzmenge* 126  
   *Element hinzufügen* 124  
   *Element löschen* 124  
   *kopieren* 123  
   *leere* 128  
   *leeren* 124  
   *Schnittmenge* 126  
   *Teilmenge* 124  
   *Vereinigungsmenge* 126  
 Menu  
   *tkinter* 395  
 Menü  
   *Debug* 165  
   *File* 25  
   *Run* 27  
 Menü (GUI) 395  
   *abtrennen* 399  
 Menüleiste (GUI) 395  
 messagebox 404  
 Methode 192  
   *besondere* 196  
 min() 183  
 Minimum 183  
 mktime() 215  
 Modul  
   *\_thread* 226  
   *cgi* 290, 294  
   *cgitb* 294  
   *collections* 223

Modul (Forts.)  
   *copy* 136, 199  
   *eigenes* 186  
   *einbinden* 39, 187  
   *fractions* 88  
   *glob* 262  
   *math* 86  
   *os* 263, 264  
   *random* 39  
   *re* 232  
   *shutil* 264  
   *sqlite3* 316  
   *tkinter* 340  
   *tkinter.messagebox* 404  
   *tkinter.scrolledtext* 354  
   *urllib* 277  
   *urllib.parse* 277  
   *urllib.request* 277  
   *webbrowser* 304  
 Modularisierung 69, 186  
 Monatsname 213  
 multiple (HTML) 301  
 Multithreading 225

## N

---

Nachkommastellen 23  
   *Anzahl* 149  
 Nachrichtenbox (GUI) 404  
 Namensraum 179  
   *lokaler* 179  
 Nenner eines Bruchs 89  
 Nicht, logisches (Python) 45  
 NICHT, logisches (SQL) 322  
 Nichts 131  
 None 131  
 NoneType 131  
 not 44  
 NOT (SQL) 322  
 numerator 89

## O

---

o (Format) 153  
 Objekt 81, 192  
   *aus Datei lesen* 260  
   *ausgeben* 197  
   *erzeugen* 193  
   *Identität* 201

- Objekt (Forts.)
    - in Datei schreiben* 258
    - iterierbares* 155
    - kopieren* 136, 199
    - Referenz* 133, 199
    - Typ ermitteln* 87
    - vergleichen* 198
  - Objektorientierte Programmierung 191
  - oct() 82
  - Oder, logisches (Python) 45
  - ODER, logisches (SQL) 322
  - offvalue 367
  - OK-Abbrechen-Box (GUI) 404
  - Oktale Zahl 82
  - Oktalformat 153
  - Online-Bestellung 298
  - onvalue 367
  - OOP 191
  - open()
    - Datei* 240
    - webbrowser* 304
  - OpenOffice.org Calc 250, 272
  - Operator
    - für Sequenzen* 93
    - logischer* 44
    - mehrere Vergleichsoperatoren* 47
    - Rangfolge* 22, 49
    - Rechenoperator* 84, 198
    - Vergleichsoperator* 40, 198
    - Verkettung* 94
    - Vervielfachung* 94
  - Operatormethode 197
  - Optionsfeld (GUI) 361
  - or 44
  - OR (SQL) 322
  - ord() 184
  - ORDER BY (SQL) 324
  - os
    - Modul* 263, 264
    - remove()* 265
    - rename()* 265
    - stat()* 263
  - Out 167
  - Over 167
- P**
- 
- pack() 341
  - Paketverwaltungssystem 279
  - Parallele Verarbeitung 225
  - Parameter 72, 73
    - benannter* 173, 420
    - variable Anzahl* 172, 420
    - Voreinstellung* 174
  - partition() 99
  - pass 144
  - password (HTML) 301
  - Passworтеingabe
    - GUI* 349
    - HTML* 299
  - PhotoImage 346
  - PHP 278, 283
  - pi 86
  - pickle 257
    - dump()* 258
    - load()* 260
  - place() 385, 387, 389, 391
  - pop() 225
  - popleft() 225
  - Position Teiltext 98
  - POST-Methode 288
  - PRIMARY KEY (SQL) 318, 329
  - print() 25, 147
    - mehrere Ausgaben* 31
  - Programm
    - abbrechen* 167
    - Abbruch vermeiden* 63
    - anhalten* 219
    - ausführen in IDLE* 27
    - ausführen von Kommandozeile* 28
    - eingeben* 25
    - konvertieren* 421
    - Parameter* 188
  - Programmzeile, lange 141
  - Python
    - auf dem Webserver* 291
    - beide Versionen* 17
    - installieren* 17
    - Interpreter* 28, 291, 341
    - Shell* 19
    - starten* 19
    - Version 3* 419
  - Python 2 15, 24
  - python -V 28
  - python.exe 291
  - python.org 16
  - python-2.7.1.msi 17
  - python-3.2.msi 17

**Q**

---

Quit 167

**R**

---

RAD 13

radio (HTML) 300

Radiobutton

*GUI* 361*HTML* 299

Rahmen (GUI) 376

raise 168

Randart (GUI) 344

Randbreite (GUI) 344

randint() 40

random

*Modul* 39*randint()* 40*seed()* 39

range() 54

Rapid Application Development 13

raw\_input() 419

re

*findall()* 232*Modul* 232*sub()* 236

read() 243

readline() 243

readlines() 243

Rechenregel 20

Rechtsbündig 153

Referenz 133, 199

*erzeugen* 343*löschen* 135*übergeben* 177

Regulärer Ausdruck 232

relief 344

remove() 110

*os* 265

rename() 265

replace() 98

Reset-Button (HTML) 299

return 75

*mehrere Werte* 176

reverse() 110

reversed() 185

rfind() 98

rotate() 225

round() 85

rpartition() 99

Rückgabewert 74, 131

*mehrere* 175

Run Module 27, 165

Runden 85

RuntimeError 171

**S**

---

Scale-Widget 367

Schaltfläche (GUI) 341

Schieberegler (GUI) 367

Schleife 49

*Abbruch* 51*Durchlauf abbrechen* 67*for* 50*while* 59

Schlüssel-Wert-Kombination 115

Schnittmenge 126

Schrift (GUI) 344

Scrollbalken (GUI) 359

Scrollbar 360

Scrollbar-Widget 359

ScrolledText 354

ScrolledText() 354

ScrolledText-Widget 353

Scrollen (GUI) 353

seed() 39

seek() 254

select (HTML) 301

SELECT (SQL) 319

selected (HTML) 301

self 193

Selfhtml 280

Sendebutton (HTML) 299

sep 147

Separator 147

Sequenz 91

*Index* 94*Operatoren* 93*Teilbereich* 94*umdrehen* 185

Serialisierung 257, 266

Set 121

*unveränderliches* 126

set() 122

*GUI* 360, 363

Set-Literal 420

show 349  
 show() 404  
 showerror() 404  
 showinfo() 404  
 showwarning() 404  
 shutil  
   *copyfile()* 265  
   *Modul* 264  
 sin() 86  
 sleep() 219  
 Slice 94  
 sort() 110  
 sorted() 185  
 Sortieren 185  
 Spiel Kopfrechnen 33  
 split() 99  
 SQL 316  
 SQL-Befehl senden 318  
 SQLite 316  
 sqlite3  
   *connect()* 318  
   *connection* 318  
   *cursor* 318  
   *Modul* 316  
 start\_new\_thread() 226  
 startswith() 99  
 stat() 263  
 Step 166  
 str  
   *count()* 97  
   *Datentyp* 91  
   *endswith()* 99  
   *find()* 98  
   *partition()* 99  
   *replace()* 98  
   *rfind()* 98  
   *rpartition()* 99  
   *split()* 99  
   *startswith()* 99  
 str() 102  
 strftime() 213  
 String 91  
   *leerer* 128  
 StringVar 362  
 sub() 236  
 Suchen und Ersetzen 232  
 sum() 183  
 Summe 183

Syntaxfehler 162  
 sys 188

## T

---

Tabelle (Datenbank) 315  
   *anlegen* 316, 318  
 tan() 86  
 Taschenrechner 19  
 Tastaturreignis (GUI) 370  
 Taste F5 27, 165  
 Teilbereich einer Sequenz 94  
 Teilmenge 124  
 Teiltext ersetzen 98  
 TEXT (SQL) 316  
 text/html 291  
 Text-Area (HTML) 301  
 Text-Widget 351  
 Thread 225  
   *Ausnahme* 230  
   *globale Daten* 229  
   *Identifikation* 227  
 thread 230  
 time 211  
   *localtime()* 211  
   *mktime()* 215  
   *sleep()* 219  
   *strftime()* 213  
   *time()* 211  
 time() 211  
 Tk 339  
   *mainloop* 341  
   *tkinter* 341  
 Tkinter 342  
 tkinter  
   *Button* 341  
   *Menu* 395  
   *messagebox* 404  
   *Modul* 340  
   *PhotoImage* 346  
   *Scrollbar* 360  
   *Tk* 341  
 tkinter.messagebox  
   *askokcancel()* 404  
   *askretrycancel()* 404  
   *askyesno()* 404  
   *Modul* 404  
   *show()* 404  
   *showerror()* 404



tkinter.messagebox (Forts.)  
   *showinfo()* 404  
   *showwarning()* 404  
 tkinter.scrolledtext  
   *ScrolledText()* 354  
 tkMessageBox 409  
 Toplevel-Widget 409  
 True 128  
 try 64, 169  
 Tupel 110  
   *entpacken* 114  
   *leeres* 128  
   *mehrdimensionales* 112  
   *verpacken* 114  
 type() 87

## U

---

Ubuntu 17  
 Umlaut 26, 103, 292  
 Umrechnung  
   *in dual* 82  
   *in hexadezimal* 82  
   *in oktal* 82  
 Umwandlung  
   *in ganze Zahl* 85  
   *in Zahl* 36, 100  
   *in Zeichenkette* 102  
 Und, logisches (Python) 44  
 UND, logisches (SQL) 322  
 Ungleich  
   *Python* 40  
   *SQL* 322  
 Unicode 184  
   *Literal* 420  
 unique (SQL) 329  
 UPDATE (SQL) 327  
 update() 118  
 urlencode() 289  
 urllib 277  
 urllib.parse  
   *Modul* 277  
   *urlencode()* 289  
 urllib.request  
   *Modul* 277  
   *urlopen()* 281  
   *urlretrieve()* 283  
 urlopen() 281  
 urlretrieve() 283

## V

---

value, FieldStorage 294  
 ValueError 63  
 values() 119  
 Variable 22, 34  
   *kontrollieren* 166  
   *Name* 23  
 Verarbeitung, parallele 225  
 Vereinigungsmenge 126  
 Vererbung 201  
   *Mehrfachvererbung* 204  
 Vergleichsoperator 40  
 Versionsnummer 28  
 Verzeichnis verwalten 264  
 Verzweigung 40  
   *einfache* 40  
   *mehrfache* 43  
 View 118  
 Vordergrundfarbe (GUI) 344  
 Vorzeichen 49

## W

---

Wagenrücklauf 103  
 Wahr 128  
 Wahrheitswert 127  
 Wahrheitswerttabelle 45  
 Warnungsnachricht (GUI) 404  
 Webbrowser 280  
 webbrowser  
   *Modul* 304  
   *open()* 304  
 Webserver 278, 279  
   *lokaler* 278, 279  
   *Programmierung* 290  
 while 59  
 while(True) 128  
 Widget 339  
   *add\_cascade()* 399  
   *add\_checkbutton()* 399  
   *add\_command()* 398  
   *add\_radiobutton()* 399  
   *add\_separator()* 399  
   *anchor* 344  
   *anordnen* 375, 383, 385, 387  
   *Bewegung* 389, 391  
   *bg* 344  
   *bind()* 372, 374

Widget (Forts.)  
*borderwidth* 344  
*command* 341, 363, 367  
*configure()* 343  
*destroy()* 341  
*DoubleVar* 362  
*Eigenschaften ändern* 342  
*Ereignis* 372  
*fg* 344  
*font* 344  
*geometry()* 403  
*get()* 348, 356, 363  
*grid()* 383  
*Größe, Ort* 403  
*height* 344  
*image* 344  
*insert()* 352, 356  
*IntVar* 362  
*offvalue* 367  
*onvalue* 367  
*pack()* 341  
*place()* 385, 387, 389, 391  
*relief* 344  
*scrollbares* 359  
*set()* 360, 363  
*show* 349  
*StringVar* 362  
*Toplevel* 409  
*Variable* 361  
*width* 344  
*yview()* 360  
width 344  
Wiederholen-Abbrechen-Box (GUI) 404  
Wiederholung 49  
Windows 14  
Wochentagsname 213  
write() 241  
writelines() 241

## X

---

x (Format) 153  
XAMPP 278  
*Control Panel* 278

## Y

---

yview() 360

## Z

---

Zahl 81  
*als Bruch annähern* 90  
*ganze* 81  
*mit Nachkommastellen* 83  
Zahlensystem 81  
Zähler eines Bruchs 89  
Zeichenkette 91  
*beginnt mit* 99  
*Eingabe* 92  
*endet mit* 99  
*mehrzeilige* 92  
*partitionieren* 99  
*zerlegen* 99  
Zeilenende 147  
*Linux* 256  
Zeilenumbruch in Code 141  
Zeilenvorschub 103  
Zeitzone 214  
zip() 155  
Zufallszahl  
*Generator* 38  
*Generator initialisieren* 39  
*liefern* 40  
Zuweisung 23, 34  
*kombinierte Operatoren* 139  
*mehrfache* 112