

Python Crashkurs

Eine praktische, projektbasierte
Programmierereinführung

DAS INHALTS- VERZEICHNIS

» Hier geht's
direkt
zum Buch

Inhalt

Der Autor	xxiv
Der Fachgutachter	xxiv
Danksagung	xxv
Vorwort zur dritten Auflage	xxvii
Einleitung	xxxI
Zielgruppe	xxxii
Lernstoff	xxxii
Onlinematerial	xxxiv
Warum Python?	xxxiv
Teil 1 Grundlagen	1
1 Erste Schritte	3
Die Programmierumgebung einrichten	3
Python-Versionen	4
Python-Codeausschnitte ausführen	4
Der Editor VS Code	5
Python auf verschiedenen Betriebssystemen	5
Python unter Windows	5
Python unter macOS	7
Python unter Linux	9
Das Hello-World-Programm ausführen	11
Die Python-Erweiterung für VS Code installieren	11
Hello_world.py ausführen	11
Fehlersuche	12

Python-Programme im Terminal ausführen	14
Unter Windows	14
Unter Linux und macOS	15
Zusammenfassung	16
2 Variablen und einfache Datentypen	17
Was bei der Ausführung von hello_world.py wirklich geschieht	17
Variablen	18
Variablen benennen und verwenden	19
Fehler bei Variablennamen vermeiden	20
Variablen sind Etiketten	21
Strings	22
Groß- und Kleinschreibung mithilfe von Methoden ändern	23
Variablen in Strings verwenden	24
Weißraum hinzufügen	25
Weißraum entfernen	26
Entfernen von Präfixen	27
Syntaxfehler bei der Stringverarbeitung vermeiden	28
Zahlen	30
Integer	30
Fließkommazahlen	31
Integer und Fließkommazahlen	32
Unterstriche in Zahlen	32
Mehrfachzuweisung	33
Konstanten	33
Kommentare	34
Wie werden Kommentare geschrieben?	34
Was für Kommentare sind sinnvoll?	34
The Zen of Python	35
Zusammenfassung	37
3 Eine Einführung in Listen	39
Was sind Listen?	39
Elemente in einer Liste ansprechen	40
Indizes beginnen bei 0, nicht bei 1	41
Einzelne Werte aus einer Liste verwenden	41

Elemente verändern, hinzufügen und entfernen	42
Elemente in einer Liste ändern	43
Elemente zu einer Liste hinzufügen	43
Elemente aus einer Liste entfernen	45
Listen ordnen	50
Listen mit <code>sort()</code> dauerhaft sortieren	50
Listen mit der Funktion <code>sorted()</code> vorübergehend sortieren	50
Listen in umgekehrter Reihenfolge ausgeben	51
Die Länge einer Liste ermitteln	52
Indexfehler vermeiden	53
Zusammenfassung	55
4 Mit Listen arbeiten	57
Eine komplette Liste durchlaufen	57
Die Schleife im Detail	58
Weitere Aufgaben in einer <code>for</code> -Schleife erledigen	59
Aktionen nach der <code>for</code> -Schleife	61
Einrückungsfehler vermeiden	62
Vergessene Einrückung der ersten Zeile in einer Schleife	62
Vergessene Einrückung nachfolgender Zeilen	63
Unnötige Einrückung	63
Unnötige Einrückung nach einer Schleife	64
Vergessener Doppelpunkt	65
Numerische Listen	66
Die Funktion <code>range()</code>	66
Numerische Listen mithilfe von <code>range()</code> aufstellen	67
Einfache Statistiken für numerische Listen	69
Listennotation	69
Teillisten	71
Einen Slice erstellen	71
Einen Slice in einer Schleife durchlaufen	73
Listen kopieren	74
Tupel	77
Ein Tupel definieren	77
Die Werte in einem Tupel durchlaufen	78
Tupel überschreiben	78

Code formatieren	79
Die Gestaltungsrichtlinien	80
Einrückung	80
Zeilenlänge	80
Leerzeilen	81
Zusammenfassung	82
5 if-Anweisungen	83
Ein einfaches Beispiel	83
Bedingungen	84
Prüfung auf Gleichheit	84
Groß- und Kleinschreibung bei der Prüfung auf Gleichheit	85
Prüfung auf Ungleichheit	86
Numerische Vergleiche	87
Prüfung auf mehrere Bedingungen	87
Prüfung auf Vorhandensein eines Werts in einer Liste	89
Prüfung auf Abwesenheit eines Werts in einer Liste	89
Boolesche Ausdrücke	90
if-Anweisungen	91
Einfache if-Anweisungen	91
if-else-Anweisungen	92
Die if-elif-else-Kette	93
Mehrere elif-Blöcke	95
Den else-Block weglassen	95
Mehrere Bedingungen prüfen	96
if-Anweisungen für Listen	99
Prüfung auf besondere Elemente	99
Prüfung auf nicht leere Liste	100
Mehrere Listen verwenden	101
if-Anweisungen gestalten	103
Zusammenfassung	104
6 Dictionaries	105
Ein einfaches Dictionary	106
Umgang mit Dictionaries	106
Zugriff auf die Werte in einem Dictionary	107
Schlüssel-Wert-Paare hinzufügen	108
Ein leeres Dictionary als Ausgangspunkt	108

Werte in einem Dictionary ändern	109
Schlüssel-Wert-Paare entfernen	111
Ein Dictionary aus ähnlichen Objekten	111
Mit get() auf Werte zugreifen	113
Dictionaries in einer Schleife durchlaufen	114
Alle Schlüssel-Wert-Paare durchlaufen	115
Alle Schlüssel in einem Dictionary durchlaufen	117
Die Schlüssel in einem Dictionary geordnet durchlaufen	119
Alle Werte in einem Dictionary durchlaufen	119
Verschachtelung	122
Dictionaries in einer Liste	122
Listen in einem Dictionary	125
Dictionaries in einem Dictionary	127
7 Benutzereingaben und while-Schleifen	131
Die Funktion input()	132
Klar verständliche Eingabeaufforderungen schreiben	132
Verwendung von int() für numerische Eingaben	133
Der Modulo-Operator	135
while-Schleifen	136
while-Schleifen in Aktion	136
Programmbeendigung durch den Benutzer	137
Flags	139
Eine Schleife mit break verlassen	140
Die Anweisung continue	141
Endlosschleifen vermeiden	142
while-Schleifen für Listen und Dictionaries	143
Elemente von einer Liste in eine andere verschieben	144
Alle Vorkommen eines Wertes aus einer Liste entfernen	145
Ein Dictionary mit Benutzereingaben füllen	145
Zusammenfassung	147
8 Funktionen	149
Funktionen definieren	150
Informationen an eine Funktion übergeben	150
Argumente und Parameter	151
Argumente übergeben	152
Positionsabhängige Argumente	152
Schlüsselwortargumente	154

Standardwerte	155
Verschiedene Formen für Funktionsaufrufe	156
Argumentfehler vermeiden	157
Rückgabewerte	158
Einen einfachen Wert zurückgeben	158
Optionale Argumente	159
Ein Dictionary zurückgeben	161
Funktionen in einer while-Schleife	162
Eine Liste übergeben	164
Eine Liste mithilfe einer Funktion ändern	165
Die Änderung einer Liste in einer Funktion verhindern	168
Beliebig viele Argumente übergeben	169
Positionsabhängige Argumente und Argumente beliebiger Anzahl kombinieren	170
Beliebig viele Schlüsselwortargumente übergeben	171
Funktionen in Modulen speichern	173
Ein komplettes Modul importieren	173
Einzelne Funktionen importieren	174
Eine Funktion mit »as« umbenennen	175
Ein Modul mit »as« umbenennen	176
Alle Funktionen eines Moduls importieren	176
Gestaltung von Funktionen	177
Zusammenfassung	178
9 Klassen	181
Eine Klasse erstellen und verwenden	182
Die Klasse Dog erstellen	182
Eine Instanz einer Klasse anlegen	184
Mit Klassen und Instanzen arbeiten	187
Die Klasse Car	187
Einen Standardwert für ein Attribut festlegen	188
Attributwerte bearbeiten	189
Vererbung	193
Die Methode <code>__init__()</code> für eine Kindklasse	193
Attribute und Methoden der Kindklasse definieren	195
Methoden der Elternklasse überschreiben	196
Instanzen als Attribute	197
Reale Objekte modellieren	199

Klassen importieren	200
Eine einzelne Klasse importieren	201
Mehrere Klassen in einem Modul speichern	202
Mehrere Klassen aus einem Modul importieren	204
Ein gesamtes Modul importieren	204
Alle Klassen eines Moduls importieren	205
Ein Modul in ein Modul importieren	205
Aliase verwenden	206
Ihren eigenen Arbeitsablauf finden	207
Die Standardbibliothek von Python	208
Gestaltung von Klassen	209
Zusammenfassung	210
10 Dateien und Ausnahmen	211
Aus Dateien lesen	212
Den Inhalt einer Datei lesen	212
Relative und absolute Dateipfade	214
Lesen der Zeilen einer Datei	215
Dateiinhalte verarbeiten	216
Große Dateien: eine Million Stellen	217
Ist Ihr Geburtsdatum in Pi enthalten?	218
In Dateien schreiben	220
Eine einzelne Zeile schreiben	220
Mehrere Zeilen schreiben	220
Ausnahmen	222
Division durch null	222
try-except-Blöcke	223
Abstürze mithilfe von Ausnahmen verhindern	224
Der else-Block	225
Datei nicht gefunden	226
Text analysieren	227
Umgang mit mehreren Dateien	228
Fehler stillschweigend übergehen	230
Welche Fehler sollten Sie melden und welche nicht?	231
Daten speichern	232
json.dump() und json.load()	233
Benutzergenerierte Daten speichern und lesen	234
Refactoring	236
Zusammenfassung	239

11 Code testen	241
Pytest mit pip installieren	242
pip aktualisieren	242
Pytest installieren	243
Funktionen testen	244
Unit-Tests und Testfälle	245
Ein bestandener Test	245
Ein nicht bestandener Test	247
Was tun bei einem nicht bestandenen Test?	248
Neue Tests hinzufügen	249
Klassen testen	251
Verschiedene Annahmen	251
Eine Beispielklasse zum Testen	252
Die Klasse AnonymousSurvey testen	254
Verwenden von Fixtures	256
Zusammenfassung	258
Teil 2 Projekte	261
Alien Invasion – ein Python-Spiel	261
Datenvisualisierung	262
Webanwendungen	262
Projekt 1: Alien Invasion	263
12 Das eigene Kampfschiff	265
Das Projekt planen	266
Pygame installieren	267
Erste Schritte für das Spielprojekt	267
Ein Pygame-Fenster anlegen und auf Benutzereingaben reagieren	267
Steuerung der Framerate	269
Die Hintergrundfarbe festlegen	270
Eine Klasse für Einstellungen anlegen	271
Das Bild eines Raumschiffs hinzufügen	272
Die Klasse Ship	273
Das Schiff auf den Bildschirm zeichnen	275

Refactoring: Die Methoden <code>_check_events()</code> und <code>_update_screen()</code>	277
Die Methode <code>_check_events()</code>	277
Die Methode <code>_update_screen()</code>	278
Das Schiff bewegen	279
Auf Tastenbetätigungen reagieren	279
Kontinuierliche Bewegung	280
Bewegung nach rechts und links	282
Die Geschwindigkeit des Schiffes anpassen	283
Den Bewegungsbereich des Schiffes einschränken	285
Refactoring von <code>_check_events()</code>	286
Beenden mit Q	286
Das Spiel im Vollbildmodus ausführen	287
Zwischenstand	288
<code>alien_invasion.py</code>	288
<code>settings.py</code>	288
<code>ship.py</code>	289
Geschosse	289
Einstellungen für Geschosse hinzufügen	289
Die Klasse <code>Bullet</code>	290
Geschosse in Gruppen speichern	291
Geschosse abfeuern	292
Alte Geschosse löschen	294
Die Anzahl der Geschosse begrenzen	295
Die Methode <code>_update_bullets()</code>	296
Zusammenfassung	297
13 Die Außerirdischen	299
Überblick über das Projekt	300
Das erste Invasionsschiff	300
Die Klasse <code>Alien</code>	301
Eine Instanz von <code>Alien</code> erstellen	302
Die Invasionsflotte erstellen	304
Reihen von Invasionsschiffen erstellen	304
Refactoring von <code>_create_fleet()</code>	306
Reihen hinzufügen	306

Die Flotte in Bewegung setzen	309
Die Invasoren nach rechts bewegen	309
Einstellungen für die Flugrichtung der Flotte	311
Auf Randberührungen prüfen	311
Sinken und Flugrichtung ändern	312
Invasoren abschießen	313
Kollisionen von Geschossen erkennen	314
Größere Geschosse zu Testzwecken	315
Die Flotte auffüllen	316
Die Geschosse beschleunigen	317
Refactoring von <code>_update_bullets()</code>	317
Spielende	318
Kollisionen zwischen Invasoren und dem eigenen Schiff erkennen . . .	318
Auf Kollisionen zwischen Invasoren und dem eigenen Schiff reagieren	319
Wenn Invasoren den unteren Bildschirmrand erreichen	322
Game over!	323
Welche Teile des Spiels müssen ausgeführt werden?	324
Zusammenfassung	325
14 Das Wertungssystem	327
Eine Play-Schaltfläche hinzufügen	327
Die Klasse <code>Button</code>	328
Die Schaltfläche auf den Bildschirm zeichnen	330
Das Spiel starten	331
Das Spiel zurücksetzen	332
Die Play-Schaltfläche deaktivieren	333
Den Mauszeiger ausblenden	333
Levels	334
Die Geschwindigkeitseinstellungen ändern	335
Die Geschwindigkeit zurücksetzen	336
Die Punktwertung	337
Den Punktestand anzeigen	338
Eine Anzeigetafel erstellen	339
Den Punktestand bei jedem Abschuss erhöhen	341
Den Punktestand zurücksetzen	341

Alle Treffer berücksichtigen	342
Den Punktwert erhöhen	343
Den Punktestand runden	344
Highscore	345
Das Level anzeigen	347
Die Anzahl der verfügbaren Schiffe anzeigen	350
Zusammenfassung	353

Projekt 2: Datenvisualisierung 355

15 Daten generieren 357

Matplotlib installieren	358
Einfache Liniendiagramme	359
Beschriftung und Linienstärke ändern	360
Das Diagramm korrigieren	361
Vordefinierte Formatierungen verwenden	363
Einzelne Punkte mit scatter() darstellen und gestalten	364
Eine Folge von Punkten mit scatter() ausgeben	365
Daten automatisch berechnen	366
Eigene Beschriftungen anlegen	368
Eigene Farben festlegen	368
Eine Colormap verwenden	368
Diagramme automatisch speichern	370
Zufallsbewegungen	370
Die Klasse RandomWalk	371
Richtungen wählen	371
Den Zufallspfad als Diagramm ausgeben	373
Mehrere Zufallspfade erstellen	374
Den Pfad gestalten	375
Würfeln mit Plotly	380
Plotly installieren	381
Die Klasse <i>Die</i>	381
Würfeln	382
Die Ergebnisse analysieren	382
Ein Histogramm erstellen	383
Das Diagramm anpassen	385
Ergebnisse bei zwei Würfeln	386

Weitere Anpassungen	387
Würfel unterschiedlicher Flächenzahl	388
Diagramme speichern	389
Zusammenfassung	390
16 Daten herunterladen	391
Das Dateiformat CSV	392
CSV-Spaltenköpfe analysieren	392
Die Spaltenköpfe und ihre Position ausgeben	394
Daten entnehmen und lesen	394
Daten in einem Temperaturdiagramm darstellen	395
Das Modul datetime	396
Datumsangaben im Diagramm darstellen	398
Ein Diagramm für einen längeren Zeitraum	399
Eine zweite Datenreihe darstellen	400
Einen Diagrammbereich einfärben	401
Fehlerprüfung	402
Daten selbst herunterladen	406
Globale Daten im GeoJSON-Format visualisieren	407
Erdbebendaten herunterladen	408
JSON-Daten untersuchen	408
Eine Liste aller Erdbeben aufstellen	411
Die Stärken entnehmen	411
Ortsdaten entnehmen	412
Eine Weltkarte zeichnen	413
Darstellung der Stärke	414
Die Farben der Markierungen anpassen	415
Weitere Farbpaletten	417
Maustext hinzufügen	417
Zusammenfassung	419
17 APIs	421
APIs	421
Git und GitHub	422
Daten mithilfe eines API-Aufrufs anfordern	422
Das Paket requests installieren	423
API-Antworten verarbeiten	424
Das Antwort-Dictionary verarbeiten	425

Ein Überblick über die höchstbewerteten Repositories	428
Grenzwerte für die API-Aufruftrate	429
Angaben zu Repositories mit Plotly visualisieren	430
Das Diagramm gestalten	431
Eigenen Maustext hinzufügen	432
Links zu dem Diagramm hinzufügen	434
Marker-Farben anpassen	435
Mehr über Plotly und die GitHub-API	436
Die API von Hacker News	436
Zusammenfassung	440
Projekt 3: Webanwendungen	443
18 Erste Schritte mit Django	445
Ein Projekt einrichten	446
Eine Spezifikation schreiben	446
Eine virtuelle Umgebung erstellen	447
Die virtuelle Umgebung aktivieren	447
Django installieren	448
Ein Projekt in Django erstellen	449
Die Datenbank erstellen	449
Das Projekt anzeigen	450
Eine App anlegen	452
Modelle definieren	453
Modelle aktivieren	454
Die Admin-Site von Django	455
Das Modell für die Einträge definieren	459
Das Modell Entry in die Datenbank aufnehmen	460
Das Modell Entry auf der Admin-Site registrieren	460
Die Django-Shell	461
Seiten erstellen: die Startseite von Learning Log	464
Eine URL zuordnen	465
Eine Ansicht schreiben	467
Eine Vorlage schreiben	467
Weitere Seiten erstellen	469
Vererbung bei Vorlagen	469
Die Seite Topics	472
Einzelne Fachgebietsseiten	475
Zusammenfassung	479

19 Benutzerkonten	481
Dateneingabe durch die Benutzer	482
Neue Fachgebiete hinzufügen	482
Neue Einträge hinzufügen	487
Einträge bearbeiten	491
Benutzerkonten einrichten	495
Die App accounts	495
Die Anmeldeseite	496
Abmelden	500
Die Registrierungsseite	501
Die Benutzer als Besitzer ihrer eigenen Daten	504
Den Zugriff mit @login_required beschränken	504
Daten mit Benutzern verknüpfen	506
Den Zugriff auf die Fachgebiete auf die zuständigen Benutzer einschränken	510
Die Fachgebiete eines Benutzers schützen	511
Die Seite edit_entry schützen	512
Neue Fachgebiete dem aktuellen Benutzer zuordnen	512
Zusammenfassung	513
20 Eine App gestalten und bereitstellen	515
Learning Log gestalten	516
Die App django-bootstrap5	516
Learning Log mit Bootstrap gestalten	517
Änderungen an base.html	518
Die Startseite mit einem Jumbotron gestalten	524
Das Anmeldeformular gestalten	525
Die Seite Topics gestalten	527
Einträge auf den Fachgebieten gestalten	528
Learning Log bereitstellen	530
Ein Platform.sh-Konto anlegen	530
Die Platform.sh-Befehlszeile installieren	530
Platformsshconfig installieren	531
Die Datei requirements.txt erstellen	531
Zusätzliche Voraussetzungen für die Bereitstellung	532
Konfigurationsdateien hinzufügen	532
Die Konfigurationsdatei services-yaml	535

settings-py für Platform.sh anpassen	536
Mit Git den Überblick über die Projektdateien bewahren	537
Die Platform.sh-Bereitstellung verbessern	542
Das Onlineprojekt schützen	544
Änderungen mit Commit bestätigen und übertragen	545
Eigene Fehlerseiten erstellen	546
Weiterentwicklung des Projekts	548
Projekte auf Platform.sh löschen	548
Zusammenfassung	550

Nachwort	551
---------------------------	------------

Anhang	553
-------------------------	------------

A Installation und Fehlerbehebung	553
--	------------

Python unter Windows	553
Py statt Python	554
Das Installationsprogramm erneut ausführen	554
Python unter macOS	555
Versehentliches Installieren der Apple-Version von Python	555
Python 2 und ältere Versionen von macOS	555
Python unter Linux	555
Die Standardinstallation vornehmen	556
Die letzte Python-Version installieren	556
Die verwendete Python-Version herausfinden	557
Schlüsselwörter und integrierte Funktionen	557
Python-Schlüsselwörter	557
Integrierte Python-Funktionen	558

B Texteditoren und IDEs	559
--	------------

Effizient mit VS Code arbeiten	561
VS Code konfigurieren	561
Tabulatoren und Leerzeichen verwenden	561
Das Farbschema ändern	562
Den Zeilenlängenmarker festlegen	562
Die Ausgabe vereinfachen	562
Kurzbefehle in VS Code	564

Codeblöcke einrücken und Einrückungen aufheben	564
Codeblöcke auskommentieren	564
Weitere Texteditoren und IDEs	565
IDLE	565
Geany	566
Sublime Text	566
Emacs und Vim	566
PyCharm	567
Jupyter Notebooks	567
C Hilfe finden	569
Erste Schritte	569
Versuchen Sie es erneut	570
Legen Sie eine Pause ein	570
Nutzen Sie das Onlinematerial zu diesem Buch	571
Online nach Hilfe suchen	571
Stack Overflow	571
Die offizielle Python-Dokumentation	572
Offizielle Dokumentation der Bibliotheken	572
r/learnpython	573
Blogs	573
Discord	573
Slack	573
D Versionssteuerung mit Git	575
Git installieren	576
Git konfigurieren	576
Ein Projekt anlegen	577
Dateien ignorieren	577
Ein Repository initialisieren	577
Den Projektstatus überprüfen	578
Dateien zum Repository hinzufügen	578
Einen Commit durchführen	579
Das Protokoll einsehen	580
Der zweite Commit	580
Änderungen zurücknehmen	581
Vorherige Commits auschecken	583
Das Repository löschen	584

E Versionssteuerung mit Git	587
Git installieren	587
Grundlegende Fehlersuche	589
Den Hinweisen nachgehen	589
Das Protokoll analysieren	590
Betriebssystemspezifische Fehlersuche	592
Von Windows bereitstellen	593
Weitere Entwicklungsansätze	596
Stichwortverzeichnis	599