
Inhalt

Teil 1	Grundlagen	1
1	Erste Schritte	3
Die Programmierumgebung einrichten		3
Python 2 und 3		4
Python-Codeausschnitte ausführen		4
Hello World!		5
Python auf verschiedenen Betriebssystemen		5
Python unter Linux		5
Python unter OS X		9
Python unter Windows		12
Fehlersuche		17
Python-Programme im Terminal ausführen		18
Unter Linux und OS X		18
Unter Windows		19
Zusammenfassung		20
2	Variablen und einfache Datentypen	21
Was bei der Ausführung von <code>hello_world.py</code> wirklich geschieht		21
Variablen		22
Variablen benennen und verwenden		23
Fehler bei Variablennamen vermeiden		24
Strings		25
Groß- und Kleinschreibung mithilfe von Methoden ändern		26
Strings verketteten		27
Weißraum hinzufügen		28
Weißraum entfernen		29
Syntaxfehler bei der Stringverarbeitung vermeiden		30
Die Print-Anweisung in Python 2		32

Zahlen	33
Integer	33
Fließkommazahlen	34
Typisierungsfehler mithilfe der Funktion <code>str()</code> vermeiden	34
Integer in Python 2	35
Kommentare	36
Wie werden Kommentare geschrieben?	37
Was für Kommentare sind sinnvoll?	37
The Zen of Python	38
Zusammenfassung	40
3 Eine Einführung in Listen	41
Was sind Listen?	41
Elemente in einer Liste ansprechen	42
Indizes beginnen bei 0, nicht bei 1	43
Einzelne Werte aus einer Liste verwenden	43
Elemente ändern, hinzufügen und entfernen	44
Elemente in einer Liste ändern	45
Elemente zu einer Liste hinzufügen	45
Elemente aus einer Liste entfernen	47
Listen ordnen	52
Listen mit <code>sort()</code> dauerhaft sortieren	52
Listen mit der Funktion <code>sorted()</code> vorübergehend sortieren	52
Listen in umgekehrter Reihenfolge ausgeben	53
Die Länge einer Liste ermitteln	54
Indexfehler vermeiden	55
Zusammenfassung	57
4 Mit Listen arbeiten	59
Eine komplette Liste durchlaufen	59
Die Schleife im Detail	60
Weitere Aufgaben in einer <code>for</code> -Schleife erledigen	61
Aktionen nach der <code>for</code> -Schleife	63
Einrückungsfehler vermeiden	64
Vergessene Einrückung der ersten Zeile in einer Schleife	64
Vergessene Einrückung nachfolgender Zeilen	65
Unnötige Einrückung	65
Unnötige Einrückung nach einer Schleife	66
Vergessener Doppelpunkt	67

Numerische Listen	68
Die Funktion range()	68
Numerische Listen mithilfe von range() aufstellen	69
Einfache Statistiken für numerische Listen	70
Listennotation	71
Teillisten	72
Einen Slice erstellen	72
Einen Slice in einer Schleife durchlaufen	74
Listen kopieren	75
Tupel	78
Ein Tupel definieren	78
Die Werte in einem Tupel durchlaufen	79
Tupel überschreiben	79
Code formatieren	80
Die Gestaltungsrichtlinien	80
Einrückung	81
Zeilenlänge	81
Leerzeilen	82
Zusammenfassung	83
5 if-Anweisungen	85
Ein einfaches Beispiel	85
Bedingungen	86
Prüfung auf Gleichheit	86
Groß- und Kleinschreibung bei der Prüfung auf Gleichheit	87
Prüfung auf Ungleichheit	88
Numerische Vergleiche	89
Prüfung auf mehrere Bedingungen	90
Prüfung auf Vorhandensein eines Werts in einer Liste	91
Prüfung auf Abwesenheit eines Werts in einer Liste	91
Boolesche Ausdrücke	92
if-Anweisungen	93
Einfache if-Anweisungen	93
if-else-Anweisungen	94
Die if-elif-else-Kette	95
Mehrere elif-Blöcke	97
Den else-Block weglassen	97
Mehrere Bedingungen prüfen	98

if-Anweisungen für Listen	101
Prüfung auf besondere Elemente	101
Prüfung auf nicht leere Liste	102
Mehrere Listen verwenden	103
if-Anweisungen gestalten	105
Zusammenfassung	106
6 Dictionaries	107
Ein einfaches Dictionary	108
Umgang mit Dictionaries	108
Zugriff auf die Werte in einem Dictionary	109
Schlüssel-Wert-Paare hinzufügen	110
Ein leeres Dictionary als Ausgangspunkt	110
Werte in einem Dictionary ändern	111
Schlüssel-Wert-Paare entfernen	113
Ein Dictionary aus ähnlichen Objekten	113
Dictionaries in einer Schleife durchlaufen	116
Alle Schlüssel-Wert-Paare durchlaufen	116
Alle Schlüssel in einem Dictionary durchlaufen	118
Die Schlüssel in einem Dictionary geordnet durchlaufen	120
Alle Werte in einem Dictionary durchlaufen	121
Verschachtelung	123
Dictionaries in einer Liste	123
Listen in einem Dictionary	126
Dictionaries in einem Dictionary	128
Zusammenfassung	130
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
Benutzereingaben in Python 2.7	136
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
Gleichwertige Funktionsaufrufe	156
Argumentfehler vermeiden	157
Rückgabewerte	158
Einen einfachen Wert zurückgeben	159
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
Funktionen formatieren	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
Vererbung in Python 2.7	195
Attribute und Methoden der Kindklasse definieren	195
Methoden der Elternklasse überschreiben	196
Instanzen als Attribute	197
Reale Objekte modellieren	199
Klassen importieren	201
Eine einzelne Klasse importieren	201
Mehrere Klassen in einem Modul speichern	203
Mehrere Klassen aus einem Modul importieren	204
Ein gesamtes Modul importieren	205
Alle Klassen eines Moduls importieren	205
Ein Modul in ein Modul importieren	206
Ihren eigenen Arbeitsablauf finden	207
Die Standardbibliothek von Python	207
Klassen formatieren	209
Zusammenfassung	210
10 Dateien und Ausnahmen	211
Aus Dateien lesen	212
Eine gesamte Datei lesen	212
Dateipfade	214
Zeilenweises Lesen	215
Eine Liste aus den Zeilen einer Datei erstellen	216
Dateiinhalte verarbeiten	217
Große Dateien: eine Million Stellen	218
Ist Ihr Geburtsdatum in Pi enthalten?	219
In Dateien schreiben	220
In eine leere Datei schreiben	221

Mehrere Zeilen schreiben	222
Text an eine Datei anhängen	222
Ausnahmen	223
Division durch null	224
try-except-Blöcke	224
Abstürze mithilfe von Ausnahmen verhindern	225
Der else-Block	226
Datei nicht gefunden	227
Text analysieren	228
Umgang mit mehreren Dateien	230
Fehler stillschweigend übergehen	231
Welche Fehler sollten Sie melden und welche nicht?	232
Daten speichern	234
json.dump() und json.load()	234
Benutzergenerierte Daten speichern und lesen	235
Refactoring	237
Zusammenfassung	240
11 Code testen	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 Zusicherungsmethoden	251
Eine Beispielklasse zum Testen	252
Die Klasse AnonymousSurvey testen	254
Die Methode setUp()	256
Zusammenfassung	258
Teil 2 Projekte	259
Alien Invasion – ein Python-Spiel	259
Datenvisualisierung	260
Webanwendungen	260

Projekt 1: Alien Invasion	261
12 Das eigene Kampfschiff	263
Das Projekt planen	264
Pygame installieren	265
Python-Pakete mit pip installieren	265
Pygame unter Linux installieren	267
Pygame unter OS X installieren	269
Pygame unter Windows installieren	269
Das Spielprojekt beginnen	270
Ein Pygame-Fenster anlegen und auf Benutzereingaben reagieren	270
Die Hintergrundfarbe festlegen	272
Eine Klasse für Einstellungen anlegen	272
Das Bild eines Raumschiffs hinzufügen	274
Die Klasse Ship	275
Das Schiff auf den Bildschirm zeichnen	277
Refactoring des Moduls game_functions	278
Die Funktion check_events()	278
Die Funktion update_screen()	279
Das Schiff bewegen	281
Auf Tastenbetätigungen reagieren	281
Kontinuierliche Bewegung	282
Bewegung nach rechts und links	284
Die Geschwindigkeit des Schiffes anpassen	285
Den Bewegungsbereich des Schiffes einschränken	287
Refactoring von check_events()	288
Zwischenstand	288
alien_invasion.py	289
settings.py	289
game_functions.py	289
ship.py	289
Geschosse	290
Einstellungen für Geschosse hinzufügen	290
Die Klasse Bullet	290
Geschosse in Gruppen speichern	292
Geschosse abfeuern	294
Alte Geschosse löschen	295

Die Anzahl der Geschosse begrenzen	296
Die Funktion <code>update_bullets()</code>	297
Die Funktion <code>fire_bullet()</code>	298
Zusammenfassung	299
13 Die Außerirdischen	301
Überblick über das Projekt	302
Das erste Invasionsschiff	303
Die Klasse Alien	304
Eine Instanz von Alien erstellen	304
Das Invasionsschiff auf den Bildschirm zeichnen	305
Die Invasionsflotte erstellen	306
Wie viele Invasionsschiffe passen in eine Reihe?	306
Reihen von Invasionsschiffen erstellen	307
Die Flotte erstellen	308
Refactoring von <code>create_fleet()</code>	310
Reihen hinzufügen	311
Die Flotte in Bewegung setzen	314
Die Invasoren nach rechts bewegen	315
Einstellungen für die Flugrichtung der Flotte	316
Auf Randberührungen prüfen	316
Sinken und Flugrichtung ändern	317
Invasoren abschießen	319
Kollisionen von Geschossen erkennen	319
Größere Geschosse zu Testzwecken	320
Die Flotte auffüllen	321
Die Geschosse beschleunigen	322
Refactoring von <code>update_bullets()</code>	323
Spielende	324
Kollisionen zwischen Invasoren und dem eigenen Schiff erkennen	324
Auf Kollisionen zwischen Invasoren und dem eigenen Schiff reagieren	325
Wenn Invasoren den unteren Bildschirmrand erreichen	328
Game over!	329
Welche Teile des Spiels müssen ausgeführt werden?	329
Zusammenfassung	330

14 Das Wertungssystem	331
Eine Play-Schaltfläche hinzufügen	331
Die Klasse Button	332
Die Schaltfläche auf den Bildschirm zeichnen	334
Das Spiel starten	335
Das Spiel zurücksetzen	336
Die Play-Schaltfläche deaktivieren	338
Den Mauszeiger ausblenden	338
Levels	339
Die Geschwindigkeitseinstellungen ändern	340
Die Geschwindigkeit zurücksetzen	341
Die Punktwertung	342
Den Punktestand anzeigen	343
Eine Anzeigetafel erstellen	344
Den Punktestand bei jedem Abschuss erhöhen	346
Alle Treffer berücksichtigen	348
Den Punktwert erhöhen	348
Den Punktestand runden	349
Highscore	351
Das Level anzeigen	354
Die Anzahl der verfügbaren Schiffe anzeigen	357
Zusammenfassung	362
Projekt 2: Datenvisualisierung	363
15 Daten generieren	365
Die Bibliothek matplotlib installieren	366
Unter Linux	366
Unter OS X	367
Unter Windows	367
matplotlib testen	368
Die matplotlib-Galerie	368
Einfache Liniendiagramme	368
Beschriftung und Linienstärke ändern	369
Das Diagramm korrigieren	370
Einzelne Punkte mit scatter() darstellen und gestalten	371
Eine Folge von Punkten mit scatter() ausgeben	373
Daten automatisch berechnen	374

Die Umrandungen der Punkte entfernen	375
Eigene Farben festlegen	375
Eine Colormap verwenden	376
Diagramme automatisch speichern	377
Zufallsbewegungen	377
Die Klasse RandomWalk()	378
Richtungen wählen	379
Den Zufallspfad als Diagramm ausgeben	380
Mehrere Zufallspfade erstellen	381
Den Pfad gestalten	382
Die Punkte färben	382
Start- und Endpunkte anzeigen	383
Die Achsen entfernen	384
Datenpunkte hinzufügen	384
Die Größe an den Bildschirm anpassen	386
Würfeln mit Pygal	387
Pygal installieren	387
Die Pygal-Galerie	388
Die Klasse Die	388
Würfeln	388
Die Ergebnisse analysieren	389
Ein Histogramm erstellen	390
Ergebnisse bei zwei Würfeln	391
Würfel unterschiedlicher Flächenzahl	393
Zusammenfassung	396
16 Daten herunterladen	397
Das Dateiformat CSV	398
CSV-Spaltenköpfe analysieren	398
Die Spaltenköpfe und ihre Position ausgeben	400
Daten entnehmen und lesen	400
Daten in einem Temperaturdiagramm darstellen	402
Das Modul datetime	403
Datumsangaben im Diagramm darstellen	404
Ein Diagramm für einen längeren Zeitraum	405
Eine zweite Datenreihe darstellen	406
Einen Diagrammbereich einfärben	408
Fehlerprüfung	409

Globale Daten im JSON-Format visualisieren	412
Daten zur Weltbevölkerung herunterladen	412
Relevante Daten entnehmen	412
Strings in numerische Werte umwandeln	414
Zweistellige Ländercodes ermitteln	415
Eine Weltkarte zeichnen	417
Numerische Daten auf einer Weltkarte darstellen	419
Eine vollständige Karte der Weltbevölkerung	420
Länder nach der Bevölkerungszahl gruppieren	422
Weltkarten in Pygal gestalten	424
Die Farbpalette aufhellen	425
Zusammenfassung	427
17 APIs	429
Web-APIs	429
Git und GitHub	430
Daten über einen API-Aufruf anfordern	430
Das Paket requests installieren	431
API-Antworten verarbeiten	432
Das Antwort-Dictionary verarbeiten	433
Ein Überblick über die höchstbewerteten Repositories	435
Grenzwerte für die API-Aufruftrate	436
Repositories mit Pygal visualisieren	437
Pygal-Diagramme verbessern	439
Eigene Tooltips hinzufügen	441
Die Beschreibungen als Tooltips ausgeben	442
Aktive Links zu dem Diagramm hinzufügen	444
Die API von Hacker News	444
Zusammenfassung	448
Projekt 3: Webanwendungen	449
18 Erste Schritte mit Django	451
Ein Projekt einrichten	452
Eine Spezifikation schreiben	452
Eine virtuelle Umgebung erstellen	452
Das Paket virtualenv installieren	453
Die virtuelle Umgebung aktivieren	454

Django installieren	454
Ein Projekt in Django erstellen	455
Die Datenbank erstellen	455
Das Projekt anzeigen	456
Eine App anlegen	458
Modelle definieren	458
Modelle aktivieren	460
Die Admin-Site von Django	461
Das Modell für die Einträge definieren	464
Das Modell Entry in die Datenbank aufnehmen	465
Das Modell Entry auf der Admin-Site registrieren	465
Die Django-Shell	466
Seiten erstellen: die Startseite von Learning Log	469
Eine URL zuordnen	469
Eine Ansicht schreiben	471
Eine Vorlage schreiben	472
Weitere Seiten erstellen	473
Vererbung bei Vorlagen	474
Die Seite Topics	476
Einzelne Fachgebietsseiten	479
Die Vorlage für eine Fachgebietsseite	481
Zusammenfassung	483
19 Benutzerkonten	485
Dateneingabe durch die Benutzer	486
Neue Fachgebiete hinzufügen	486
Neue Einträge hinzufügen	491
Einträge bearbeiten	495
Benutzerkonten einrichten	499
Die App users	499
Die Anmeldeseite	500
Abmelden	503
Die Registrierungsseite	505
Die Benutzer als Besitzer ihrer eigenen Daten	508
Den Zugriff mit @login_required beschränken	509
Daten mit Benutzern verknüpfen	511
Den Zugriff auf die Fachgebiete auf die zuständigen Benutzer einschränken	514

Die Fachgebiete eines Benutzers schützen	515
Die Seite edit_entry schützen	516
Neue Fachgebiete dem aktuellen Benutzer zuordnen	516
Zusammenfassung	518
20 Eine App gestalten und bereitstellen	519
Learning Log gestalten	520
Die App django-bootstrap3	520
Learning Log mit Bootstrap gestalten	521
Änderungen an base.html	522
Die Startseite mit einem Jumbotron gestalten	526
Das Anmeldeformular gestalten	527
Die Seite new_topic gestalten	528
Die Seite Topics gestalten	529
Einträge auf den Fachgebichtsseiten gestalten	530
Learning Log bereitstellen	532
Ein Heroku-Konto anlegen	532
Heroku Toolbelt installieren	533
Die erforderlichen Pakete installieren	533
Eine Paketliste mit der Datei requirements.txt erstellen	533
Die Python-Laufzeitversion angeben	535
Die Datei settings.py für Heroku anpassen	535
Ein Procfile zum Starten der Prozesse erstellen	536
Die Datei wsgi.py für Heroku anpassen	537
Ein Verzeichnis für statische Dateien erstellen	537
Den Server gunicorn lokal nutzen	537
Mit Git den Überblick über die Projektdateien bewahren	538
Die Datenbank auf Heroku einrichten	542
Die Heroku-Bereitstellung verbessern	543
Das Onlineprojekt schützen	544
Änderungen mit Commit bestätigen und übertragen	545
Eigene Fehlerseiten erstellen	547
Weiterentwicklung des Projekts	550
Die Einstellung SECRET_KEY	550
Projekte auf Heroku löschen	551
Zusammenfassung	552
Nachwort	553

Anhang **555****A Python installieren** **555**

Python unter Linux	555
Die installierte Version herausfinden	555
Python 3 unter Linux installieren	556
Python unter OS X	556
Die installierte Version herausfinden	556
Python 3 mit Homebrew installieren	557
Python unter Windows	558
Python 3 unter Windows installieren	558
Den Python-Interpreter finden	559
Python zur Pfadvariablen hinzufügen	559
Schlüsselwörter und integrierte Funktionen	560
Python-Schlüsselwörter	560
Integrierte Python-Funktionen	560

B Texteditoren **563**

Geany	564
Geany unter Linux installieren	564
Geany unter Windows installieren	565
Python-Programme in Geany ausführen	565
Geany-Einstellungen anpassen	566
Sublime Text	567
Sublime Text unter OS X installieren	567
Sublime Text unter Linux installieren	567
Sublime Text unter Windows installieren	567
Python-Programme in Sublime Text ausführen	567
Sublime Text konfigurieren	568
Die Einstellungen von Sublime Text anpassen	568
IDLE	569
IDLE unter Linux installieren	569
IDLE unter OS X installieren	569
IDLE unter Windows installieren	570
IDLE-Einstellungen anpassen	570
Emacs und vim	570

C Hilfe finden	571
Erste Schritte	571
Versuchen Sie es erneut	572
Legen Sie eine Pause ein	572
Nutzen Sie das Onlinematerial zu diesem Buch	573
Online nach Hilfe suchen	573
Stack Overflow	573
Die offizielle Python-Dokumentation	574
Offizielle Dokumentation der Bibliotheken	574
r/learnpython	574
Blogs	574
IRC (Internet Relay Chat)	575
Ein IRC-Konto anlegen	575
Hilfreiche Kanäle	575
IRC-Kultur	576
D Versionssteuerung mit Git	577
Git installieren	578
Git unter Linux installieren	578
Git unter OS X installieren	578
Git unter Windows installieren	578
Git konfigurieren	578
Ein Projekt anlegen	579
Dateien ignorieren	579
Ein Repository initialisieren	580
Den Projektstatus überprüfen	580
Dateien zum Repository hinzufügen	581
Einen Commit durchführen	581
Das Protokoll einsehen	582
Der zweite Commit	582
Änderungen zurücknehmen	583
Vorherige Commits auschecken	585
Das Repository löschen	587
Stichwortverzeichnis	589