Python von Kopf bis Fuß

Grundlagen und Praxis der Python-Programmierung

» Hier geht‘s direkt zum Buch

DAS INHALTS-VERZEICHNIS

Der Inhalt (im Überblick)

Der Inhalt (jetzt ausführlich)

Intro

Ihr Gehirn und Python. *Sie* versuchen, etwas zu *lernen*, und Ihr *Hirn* tut sein Bestes, damit das Gelernte nicht *hängen bleibt*. Es denkt nämlich: »Wir sollten lieber ordentlich Platz für wichtigere Dinge lassen, z. B. für das Wissen darüber, welche Tiere einem gefährlich werden könnten, oder dass es eine ganz schlechte Idee ist, nackt Snowboard zu fahren.« Tja, wie schaffen wir es nun, Ihr Gehirn davon zu überzeugen, dass Ihr Leben davon abhängt, wie man in Python programmiert?

0 Warum Python?
Ähnlich und doch anders

Wie Sie vermutlich schon wissen, beginnt Python mit dem Zählen bei null. Python hat eine Menge mit anderen Programmiersprachen **gemeinsam**. Es gibt **Variablen**, **Schleifen**, **Bedingungen**, **Funktionen** und so weiter. In diesem Eröffnungskapitel nehmen wir Sie mit auf eine **kleine Besichtigungstour**, die Ihnen einen **Überblick** über die Grundlagen von Python vermitteln soll. Das heißt, wir zeigen Ihnen die Sprache, aber ohne zu sehr ins Detail zu gehen. Sie werden lernen, mit Jupyter Notebook (innerhalb von VS Code) eigenen Code zu **schreiben** und **auszuführen**. Dabei werden Sie staunen, wie viel Programmierfunktionalität direkt in Python **eingebaut** ist. Das werden Sie **nutzen**, um verschiedene Aufgaben zu erledigen. Außerdem erfahren Sie, dass Python viele Konzepte mit anderen Programmiersprachen gemeinsam hat, diese aber **etwas anders** umsetzt. Verstehen Sie uns hier nicht falsch: Wir meinen hier die **guten** Unterschiede, nicht die *schlechten*. Lesen Sie weiter, um mehr zu erfahren.

1 Eintauchen
Sprung ins kalte Wasser

Eine neue Sprache lernt man am besten, indem man Code schreibt.
Und wenn Sie Code schreiben wollen, brauchen Sie ein **echtes** Problem, das es zu lösen gilt. Wie der Zufall es will, gibt es in diesem Kapitel eins. Hier beginnen Sie Ihre Karriere in der Applikationsentwicklung mit Python, indem Sie zusammen mit unserem freundlichen **Schwimmcoach** einen Sprung ins kalte Wasser wagen. Sie beginnen mit Pythons **Strings** und wie Sie sie nach Herzenslust **manipulieren** können. Unterwegs erstellen Sie eine Python-basierte Lösung für das Problem des Coachs. Außerdem erfahren Sie mehr über Pythons eingebaute **Listen**-Datenstruktur. Sie lernen, wie **Variablen** funktionieren und was Pythons **Fehlermeldungen** bedeuten, ohne dass Sie hierfür gleich einen Tauchschein brauchen. Und das alles, während wir ein *echtes* Problem mit *echtem* Python-Code lösen. Also, nichts wie rein – und zwar kopfüber!

2 Listen aus Zahlen
Listendaten verarbeiten

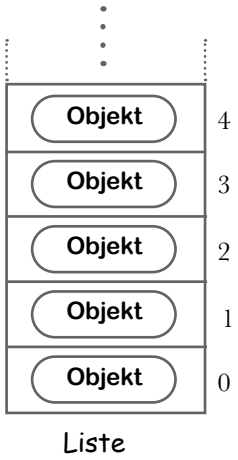
Je mehr Code Sie schreiben, desto besser werden Sie. Ganz einfach.

Auch in diesem Kapitel schreiben Sie Python-Code, um dem Coach zu helfen. Sie lernen, wie Sie Daten aus der **Datei** des Coachs **lesen** und die enthaltenen Zeilen in einer **Liste**, einer von Pythons eingebauten **Datenstrukturen**, speichern können. Neben der Erstellung von Listen aus Daten in einer Datei lernen Sie auch, Listen von Grund auf neu zu erstellen und bei Bedarf **dynamisch wachsen** zu lassen. Außerdem werden Sie Listen mit einer von Pythons beliebtesten Schleifenkonstrukten, der **for**-Schleife, verarbeiten. Sie werden Daten aus einem Datenformat in ein anderes **konvertieren,** und Sie werden einen neuen besten Freund (Ihren eigenen Python-**BFF**) kennenlernen. Nach Kaffee und Kuchen ist es jetzt Zeit, die Ärmel hochzukrempeln und sich wieder an die Arbeit zu machen.

3 Listen von Dateien
Funktionen, Module und Dateien

Ihr Code kann nicht ewig in einem Notebook leben. Er will frei sein.

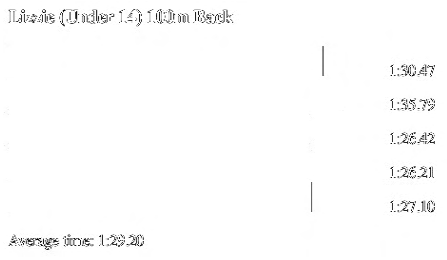
Und wenn es darum geht, Ihren Code zu befreien und mit anderen zu **teilen**, dann ist eine selbst erstellte **Funktion** der erste Schritt, auf den kurz darauf ein **Modul** folgt, mit dem Sie Ihren Code organisieren und weitergeben können. In diesem Kapitel werden Sie aus dem bisher geschriebenen Code direkt eine Funktion und auf dem Weg auch gleich ein **gemeinsam nutzbares** Modul erstellen. Ihr Modul wird sich sofort an die Arbeit machen, während Sie **for**-Schleifen, **if**-Anweisungen, Tests auf bestimmte **Bedingungen** sowie die Python-Standardbibliothek, **PSL** (*Python Standard Library*), verwenden, um die Schwimmdaten des Coachs zu verarbeiten. Außerdem werden Sie lernen, Ihre Funktionen zu **kommentieren** (was *immer* eine gute Idee ist). Es gibt viel zu tun, also an die Arbeit!



4 Formatierte String-Literale
Tabellen aus Daten

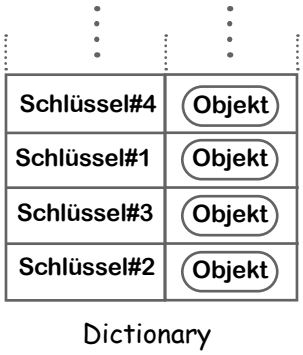
Manchmal ist die einfachste Lösung die beste.

In diesem Kapitel kommen wir endlich dazu, die Balkendiagramme für den Coach zu erstellen. Hierfür werden Sie nichts als **Strings** verwenden. Wie Sie schon wissen, besitzt Python bereits eine Menge **eingebauter Funktionalität**. Pythons **formatierte String-Literale**, auch *f-Strings* genannt, erweitern diese Möglichkeiten noch einmal auf ziemlich clevere Weise. Es klingt vielleicht seltsam, dass wir vorschlagen, Balkendiagramme mit Text zu erstellen, Sie werden aber sehr bald merken, dass es längst nicht so *absurd* ist, wie es scheint. Unterwegs werden Sie Python einsetzen, um **Dateien** zu erstellen und einen Webbrowser zu starten – und das alles mit nur wenigen Zeilen Code. Am Ende bekommt der Coach endlich, was er sich so lange gewünscht hat: die **automatische Erzeugung** von Diagrammen aus seinen Schwimmdaten. Also, nichts wie los!



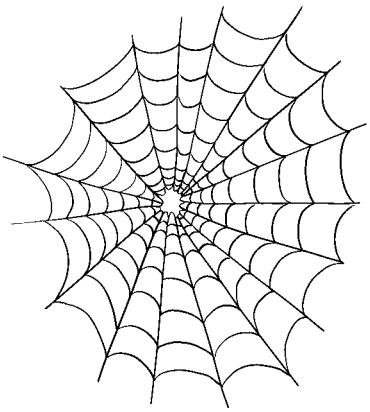
5 Daten organisieren
Die richtige Datenstruktur

Ihr Code muss seine Daten im Arbeitsspeicher ablegen können.
Und bei der Anordnung von Daten **im Arbeitsspeicher** entscheidet oft die Wahl der Datenstruktur darüber, ob es eine unordentliche Lösung gibt, die *irgendwie* funktioniert, oder eine **elegante** Lösung, die *gut* funktioniert. In diesem Kapitel lernen Sie eine weitere Python-Datenstruktur kennen, das **Dictionary**. Es wird oft mit der allgegenwärtigen Liste kombiniert, um **komplexe** Datenstrukturen zu schaffen. Der Coach muss auf einfache Weise die Daten eines beliebigen Schwimmers auswählen können. Mit einem Dictionary wird das zum Kinderspiel!



6 Eine Web-App erstellen
Webentwicklung

Fragen Sie zehn Programmierer, welches Web-Framework Sie nehmen sollen …
… und Sie bekommen wahrscheinlich elf Antworten! 😉 Bei der Webentwicklung mangelt es Python wirklich nicht an technischen *Wahlmöglichkeiten*. Jede von ihnen hat eine eigene loyale und unterstützende Entwicklergemeinschaft. In diesem Kapitel tauchen Sie ein in die Welt der **Webentwicklung**. Sie werden schnell eine Web-App für den Coach bauen, in der man die Balkendiagramme für beliebige Schwimmer betrachten kann. Unterwegs lernen Sie die Verwendung von HTML-**Templates** (Schablonen), Funktions**dekoratoren**, **GET-** und **POST**-HTTP-Methoden und vieles mehr. Es gilt keine Zeit zu verschwenden: Der Coach will endlich sein neues System vorführen. Also an die Arbeit!



7 Bereitstellung
Code überall ausführen

Code auf dem eigenen Rechner auszuführen, ist eine Sache …

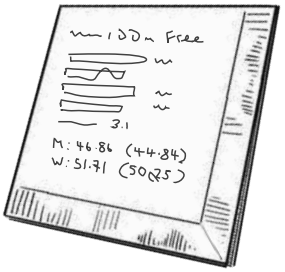
Aber eigentlich wollen Sie Ihren Code so **bereitstellen**, dass auch andere Nutzer ihn einfach weiterverwenden können. Je *geringer* der Aufwand, desto besser. In diesem Kapitel werden Sie die Web-App des Coachs fertigstellen, sie mit etwas **Stil** versehen und sie dann in der **Cloud bereitstellen**. Trotzdem wird die Komplexität nicht ins Unermessliche steigen. In einer früheren Auflage dieses Buchs haben wir damit angegeben, dass die Bereitstellung »etwa zehn Minuten« dauert, und das ist bereits ziemlich schnell … In diesem Kapitel werden Sie für die Bereitstellung nur noch zehn **Schritte** benötigen. Die Cloud wartet schon darauf, die Web-App des Coachs zu hosten. Zeit für die Bereitstellung!

8 Mit HTML arbeiten
Web-Scraping

In einer perfekten Welt wären alle benötigten Daten leicht zugänglich. Das ist aber nur selten der Fall. So werden Daten beispielsweise im Web veröffentlicht. In HTML eingebettete Daten müssen von Webbrowsern **gerendert** und von Menschen **gelesen** werden können. Was aber, wenn Sie diese Daten mit Code **verarbeiten** müssen? Geht das überhaupt? Glücklicherweise ist Python so etwas wie ein Champion, wenn es um das maschinelle Auslesen – das sogenannte ***Scraping*** – von Daten aus Webseiten geht, und in diesem Kapitel werden wir Ihnen zeigen, wie das funktioniert. Sie werden außerdem lernen, wie die ausgelesenen HTML-Seiten **geparst** werden, um nutzbare Daten zu **extrahieren**. Unterwegs werden Ihnen **Slices** (»Scheiben«) und **Soup** (»Suppe«) begegnen, aber keine Sorge, das hier ist immer noch *Python von Kopf bis Fuß* und nicht *Kochen von Kopf bis Fuß*.

9 Mit Daten arbeiten
Datenmanipulation

Manchmal sind die Daten nicht so angeordnet, wie sie gebraucht werden. Vielleicht haben Sie eine *verschachtelte Liste (List of Lists)*, brauchen aber eigentlich ein *verschachteltes Dictionary*. Oder vielleicht müssen Sie einen Wert in einer Datenstruktur mit einem Wert in einer anderen Datenstruktur verbinden, ohne dass sie richtig zusammenpassen. Das kann schnell ziemlich frustrierend werden. Aber keine Sorge: Die Macht von Python steht Ihnen auch hierbei zur Seite. In diesem Kapitel benutzen Sie Python, um Ihre Daten **durch die Mangel zu drehen**. Das Ziel ist es, die von der *Wikipedia*-Website ausgelesenen Daten vom Ende des vorherigen Kapitels in etwas wirklich *Nützliches* umzuwandeln. Sie werden lernen, wie Sie Pythons Dictionary als **Lookup**-Tabelle **nutzen** können. Hier geht es um Umwandlungen, Integrationen, Aktualisierungen, Bereitstellungen und mehr. Und ganz am Ende wird die Spezifikation, die der Coach auf der Papierserviette notiert hat, *Wahrheit*. Wetten, Sie können es kaum abwarten? Wir auch nicht … also los!



9½ Mit ~~Elefanten~~ Dataframes arbeiten

Tabellarische Daten

Manchmal wollen anscheinend alle Daten der Welt tabellarisch sein.
Tabellarische Daten gibt es *überall*. Die Schwimmweltrekorde aus dem vorherigen Kapitel liegen als **tabellarische** Daten vor. Wenn Sie alt genug sind, um sich an Telefonbücher zu erinnern, wissen Sie, dass auch diese Daten tabellarisch sind. Kontoauszüge, Rechnungen, sogar Tabellenkalkulationen sind – Sie haben es natürlich geahnt – tabellarische Daten. In diesem *kurzen* Kapitel lernen Sie einige Dinge über **pandas**, die beliebteste Python-Bibliothek zur Datenanalyse. Leider werden wir hier nur die Spitze des Eisbergs betrachten können. Trotzdem werden Sie genug lernen, um das nächste Mal, wenn Sie mit tabellarischen Daten arbeiten müssen, die am häufigsten verwendete pandas-Datenstruktur nutzen zu können, den **Dataframe**.

10 Datenbanken
Dinge ordnen

Irgendwann müssen Sie die Daten Ihrer Applikationen verwalten.
Und wenn Sie Ihre Daten besser **verwalten** müssen, reicht Python (für sich genommen) möglicherweise nicht aus. In solchen Fällen hilft der Griff zur **Datenbank**-Engine Ihrer Wahl. Um nicht den Überblick zu verlieren, beschränken wir uns an dieser Stelle auf Datenbank-Engines, die das gute alte **SQL** unterstützen. Sie werden hier nicht nur eine Datenbank **erstellen** und darin ein paar **Tabellen** anlegen, Sie werden auch Daten zur Datenbank **hinzufügen**, **daraus auswählen** und **löschen**. Für alle diese Aufgaben werden Sie SQL-Abfragen verwenden, die von Ihrem Python-Code koordiniert werden.

11 Listenabstraktionen
Datenbankintegrationen

Zeit für die Integration Ihrer Datenbanktabellen.

Mithilfe der Tabellen in der Datenbank kann Ihre Web-App die vom Coach benötigte **Flexibilität** erreichen. In diesem Kapitel erstellen wir ein Modul mit verschiedenen **Hilfsfunktionen** (Utilities), über die Ihre Web-App die Datenbank-Engine **nutzen** kann. Außerdem werden Sie eine echte Python-Superkraft kennenlernen, die Ihnen hilft, mit weniger Code mehr zu erreichen: **Listenabstraktionen**. Daneben werden Sie eine Menge Ihres schon vorhandenen Codes auf neue und interessante Weise **wiederverwenden**. Wir haben jede Menge **Integration** vor uns. Also los!


```
[sql for sql in dir(queries) if not sql.startswith("__")]
```

12 Bereitstellung in neuem Licht
Der letzte Schliff

Das Ende vom Anfang Ihrer Python-Reise ist fast erreicht.

In diesem letzten Kapitel passen Sie Ihre Web-App so an, dass sie nicht mehr SQLite, sondern **MariaDB** als Datenbank-Backend verwendet. Danach gibt es noch ein paar kleinere Änderungen, um die neueste Version Ihrer Web-App auf PythonAnywhere bereitzustellen. Dadurch unterstützt das System des Coachs eine **beliebige** Zahl von Schwimmern, die an einer **beliebigen** Zahl von Trainingseinheiten teilnehmen. In diesem Kapitel gibt es zu Python selbst nicht viel Neues. Die meiste Zeit werden Sie damit verbringen, den schon vorhandenen Code für die Zusammenarbeit mit MariaDB und PythonAnywhere anzupassen. Ihr Python-Code existiert nie **isoliert**, sondern **interagiert** mit seiner Umgebung und dem System, auf dem er läuft.


```
mysql> select count(*) from events;
+----------+
| count(*) |
+----------+
|       14 |
+----------+
1 row in set (0.00 sec)

mysql> select count(*) from swimmers;
+----------+
| count(*) |
+----------+
|       23 |
+----------+
1 row in set (0.00 sec)

mysql> select count(*) from times;
+----------+
| count(*) |
+----------+
|      467 |
+----------+
1 row in set (0.00 sec)
```

Anhang

Die zehn wichtigsten Dinge, die wir nicht behandelt haben

Wir sind fest davon überzeugt, dass man wissen muss, wann es reicht.
Besonders wenn Ihr Autor aus Irland stammt, einem Land, das dafür berühmt ist, Menschen mit der Gabe hervorzubringen, zu ignorieren, wann sie besser mit dem Reden aufhören sollten. 😉 Dabei genießen wir es, über unsere Lieblingsprogrammiersprache Python zu sprechen. In diesem Anhang zeigen wir Ihnen die zehn wichtigsten Themen, die wir Ihnen gerne »von Kopf bis Fuß« erzählt hätten. Leider hatten wir aber keine weiteren 400 Seiten zur Verfügung. Hier finden Sie Informationen zu Klassen, Exception-Handling, zum Testen eines Walrosses (Ernsthaft? Ein Walross? Ja, ein Walross), zu Switches, Dekoratoren, Kontextmanagern und Nebenläufigkeit sowie Tipps zu Typen, virtuellen Umgebungen und Programmierwerkzeugen. Wie gesagt: Es gibt immer etwas, über das man noch reden kann. Also, blättern Sie um – und haben Sie Freude an den nächsten zwölf Seiten!