Einblick in die Programmiersprache Python

Michael J.M. Wagner

Fastlane

Revision 235

Inhaltsverzeichnis

Fastlane
Karriere

Grundlagen

Python ist ein Interpreter-Sprache mit vielen Vorteilen:

- Eine einfache, eindeutige Syntax
- Klare Strukturen: Die Anordnung der Programmzeilen ergibt gleichzeitig die logische Struktur des Programms.
- Unabhängigkeit vom Betriebssystem

Fastlane Karriere

Entwicklungsgeschichte:

- 1990: Erste Anfänge durch Guido van Rossum (Amsterdam)
- 1994: Python 1.0. Anfänglich mit Konzepten der funktionalen Programmierung
- 2000: Python 2.0. Garbage Collection, Unicode-Zeichensatz
- 2008: Python 3.0. Entfernen von Redundanzen bei Befehlssätzen und veralteten Konstrukten
  ABER: Verlust der Abwärtskompatibilität. Daher wird die Version 2.7 weiter unterstützt.

Fastlane Karriere

Da die graphischen Oberflächen der meisten Linux-Distributionen Python verwenden, ist der Python-Interpreter dort stets installiert. In diesem Kurs wird die Entwicklungumgebung *idle* verwendet.
Die offizielle Python-Dokumentation findet sich unter `http://docs.python.org`.

Beispiel:

- Installierte Python-Versionen
- Python-Interpreter

Fastlane Karriere

Wie in Skriptsprachen üblich, ist das „Hauptprogramm" dadurch gekennzeichnet, dass die Befehlszeilen direkt in der Datei, ohne weitere Angaben einer Funktion o.ä. stehen. Ein „Hallo Welt"-Programm besteht also nur aus der Zeile:

```
print ("Hallo␣Welt")
```

Fastlane
Karriere

Zur Ausführung von Python-Skripten gibt es folgende Möglichkeiten:

- Ausführung in der Entwicklungsumgebung: *Ausführen als* → *Python Skript*
- Ausführung mit dem Aufruf des Interpreters:

```
python3 HalloWelt.py
```

- Unter POSIX-Systemen (Linux, ...) besteht als weitere Möglichkeit den Interpreter im Skript selbst anzugeben. Die erste Zeile des Skripts lautet dann:

```
#!/usr/bin/python3
```

Fastlane Karriere

Python Sprachelemente

- Kommtare

```
# Mein erstes Programm
print("Hallo Welt")    # Eine Ausgabe
"""Kommentar in
mehreren Zeilen"""
```

- Variablen
- Verzweigungen
- Schleifen
- Funktionen
- Pakete und Module

Fastlane Karriere

Programmierbeispiel

Im Laufe dieser Veranstaltung soll eine kleine Anwendung entstehen. Für einer Bücherei werden für die Verwaltung die Medien in einer Datenverwaltung gespeichert.

- Medien können Bücher, CDs, ... sein
- Zur Datenablage dient eine csv-Datei
- Ein Datensatz besteht aus
  - Eindeutiger Signatur
  - Autor
  - Titel
  - Typ (`B` oder `C`)
  - Seitenzahl (für Bücher)
  - Spieldauer (für CDs)

Fastlane Karriere

Zeichenketten

- In Python gibt es mächtige Werkzeuge zur Bearbeitung von Zeichenketten.
- Zur Definition von Zeichenketten können verschiedene Hochkommata verwendet werden.
- Operatoren: `+`, `*`, `in`
- Indizierung mit positiven und negativen Indizes

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Element	R	o	b	i	n	s	o	n		C	r	u	s	o	e
Negativer Index	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1

- Bildung von Slices (Teilzeichenketten)

Fastlane Karriere

Es gibt in Python viele Funktionen zur Bearbeitung von Zeichenketten. Die folgenden seien herausgehoben:

- `find`: Teilzeichenkette suchen
- `replace(old, new[, count])`: Liefert eine Kopie der Zeichenkette, bei der alle Zeichenketten `old` durch `new` ersetzt sind. Wenn das optionale Argument `count` gesetzt ist, werden nur die ersten `count` Stellen ersetzt.
- `startswith(prefix)`: Liefert `True`, wenn die Zeichenkette mit `prefix` beginnt, sonst `False`.
- `split(sep=None, maxsplit=-1)`: Liefert eine Liste von Wörtern aus der Zeichenkette mit `sep` als Trenner. Wenn `maxsplit` gegeben ist, wird die Zeichenkette in maximal `maxsplit` Bestendteile zerlegt.
- `rstrip, lstrip, strip`: Entfernt Leerzeichen

Listen

- Listen sind Folgen von Objekten, die von verschiedenen Typen sein können.
- Da Listen geordnet sind, kann mittels Index auf die einzelnen Elemente zugegriffen werden.
- Listen können mit + konkatiniert und mit * vervielfältigt werden.
- Im Gegensatz zu Zeichenketten sind Listen veränderbar.
- Nützliche Listenoperationen

Python kennt auch eine nicht veränderbare Variante der Liste, das Tupel. werden in dieser Form definiert:

```
t1 = 13, "Hallo", 15.0
t2 = ( 13, "Hallo", 15.0 )
```

Tupel können auch über die Parameterübergabe erzeugt werden:

```
def printer(*tup):
    print(tup)

printer("Hallo", "Welt")  # Schreibt ("Hallo","
    Welt")
```

Verzeichnisse und Mengen

Verzeichnisse:

- Schlüssel-Wert-Paare werden als Verzeichnisse, Wörterbücher oder *dictionaries* bezeichnet.
- Operationen auf Verzeichnisse
- Views und Iteratoren
- Verzeichnisse können auch über die Parameterübergabe erzeugt werden:

```
def printer(**verz):
    print(verz)

printer(a="Hallo", b="Welt")
# Schreibt {a:"Hallo",b:"Welt"}
```

Fastlane Karriere

Mengen:

> *Mengen (englisch: sets) unterscheiden sich von Listen und Tupeln dadurch, dass jedes Element nur einmal existiert. Außerdem sind Mengen ungeordnet, daher ist auch die Reihenfolge bei der Ausgabe eines gesamten Sets nicht festgelegt.*

Syntax: `s = {'a', 12, 4.5}`
Listen lassen sich mit der Funktion `set()` in Mengen umwandeln.
Duplikate werden dabei entfernt.

Aufgabe:
Mit den nächsten Übungen soll Schritt für Schritt eine Bücherverwaltung für eine Bücherei erstellt werden. Führen Sie folgende Schritte aus:

- Legen Sie ein neues Projekt *Buecherei* an.
- Legen Sie die Datei `Medienverwaltung.py` an.
- Schreiben Sie eine Funktion `addMedium`, die sechs Parameter nimmt:
  `addMedium(signatur, autor, titel, typ, seitenzahl, spieldauer)`
- `addMedium` soll die Daten an die Datei `medien.csv` kommasepariert anhängen.
- Rufen Sie die Prozedur aus dem Hauptprogramm (mit konstanten Werten) auf.

Schreiben Sie im Modul `Medienverwaltung.py` eine Funktion `isSignatureInFile(signatur)`, die

- die Datei `medien.csv` Zeile für Zeile ausliest,
- jede Zeile prüft, ob sie mit der übergebenen Signatur beginnt,
- als Ergebnis einen bool'schen Wert zurückgibt: `True`, falls die Signatur bereits in der Datei vorhanden ist, sonst `False`.

Ergänzen Sie die Medienverwaltung:

- In der Funktion `addMedium` um den Aufruf von `isSignatureInFile`. Falls die Signatur schon vorhanden ist, geben Sie einen entsprechenden Integer-Fehlercode an das Hauptprogramm zurück
- Werten Sie im Hauptprogramm die Fehlercodes aus.

Fastlane
Karriere

Motivation

- Strukturen sind eine sehr nützliche Sache, um Daten, die logisch zusammen gehören, zusammen zu verwalten.
- Wird eine Struktur im Speicher angelegt, wurde es in großen Programmwerken schnell unübersichtlich, wer diese Struktur für welchen Zweck gebraucht und wer Änderungen daran vornimmt.
- Unklar war oft, ab welchem Zeitpunkt welche Bestandteile einen gültigen Wert besitzen.
- Vor diesem Hintergrund kam die Idee auf, die Zugriffe auf Strukturen (lesend, wie schreibend) zu kontrollieren.
- Eine Datenstruktur mit den dazugeförigen Zugriffsfunktionen nennt sich *Klasse*.
- Alle Zugriffe erfolgen prozedural über öffentliche *Methoden*.

Fastlane Karriere

So weit die Theorie: Python ist da nicht so streng.

Klassen

In Python werden Objektinstanzen ausschließlich im Freispeicher angelegt und vom *garbage collector* wieder frei gegeben. Darüber hinaus kann eine Instanz durch das explizite Aufrufen von `del` freigegeben werden. Eine Variable im Programm stellt daher lediglich einen Verweis in den Freispeicher dar. Über eine Zuweisung wird nur der Verweis kopiert, nicht die Instanz selbst. Um die Instanz selbst zu kopieren stellt Python die Funktion `copy.deepcopy(<obj>)` zur Verfügung.

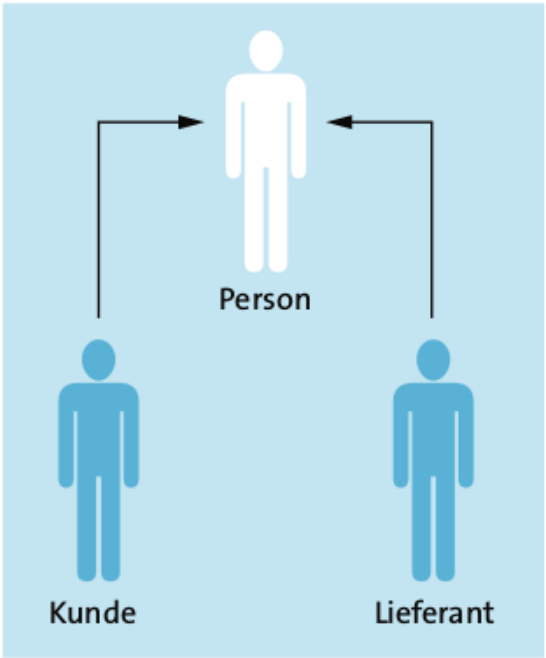
Fastlane
Karriere

Bestandteile einer Klasse:

- Klassendefinition
- Methoden / *getter*: In Python ist (leider) der direkte Zugriff auf Eigenschaften üblich.
- Konstruktor
- Kein Überladen von Funktionen/Methoden nicht überladen werden. Es gibt also stets genau einen Konstruktor.
- Weitere `__<NAME>__`-Methoden:
  - `__str__`: Die Umwandlung in eine Zeichenkette
  - `__repr__`: Soll eine für die Objetinstanz eindeutige Zeichenkette liefern. `__repr__` wird bei der Ausgabe der in Listen enthaltenen Objekten verwendet.
  - `__eq__`: Definition, wann zwei Instanzen gleich sind, definiert den ==-Operator
- `self` ist erster Parameter jeder Methode.

Vererbung

Vererbung ist ein nützliches Merkmal der Objektorientierung um gemeinsame Eigenschaften aus Klassen herauszufaktorieren.



Fastlane
Karriere

Aufgabe:
Ergänzen Sie die Bibliotheksanwendung:

- Ergänzen Sie das Modul `MediumModul.py`. Leiten Sie von der Klasse `MediumBase` die Klassen `Buch` und `CD` ab.
- Verteilen Sie die Attribute sinnvoll auf die Klassen.
- Ergänzen Sie jede Klasse um eine Methode `__str__`, die den Datensatz als Zeichenkette in dem Format zurückgibt, wie er in der Eingabedatei erwartet wird.
- Testen Sie den Code durch direkte Verwendung im Hauptprogramm.
- Ergänzen Sie die Medienverwaltung um eine Funktion `addMediumHira(mediumBase)`,
  die die neue `__str__`-Methode zum Schreiben in die Date
  verwendet. Zur Duplikatsprüfung kann die bestehende

Fastlane Karriere

`isSignatureInFile` verwendet werden.

- Verwenden Sie die Methode im Hauptprogramm, indem Sie einmal ein Buch, einmal eine CD übergeben.
- Sollen, wie in kompilierten Sprachen üblich, Methodennamen überschrieben werden, kann dies in Python dadurch erreicht werden, dass die übergebenen Parameter mit `isinstance` auf ihre Zugehörigkeit zu einer Vererbungshierarchie überprüft werden. Ergänzen Sie die Funktion `addMedium` in der Weise, dass
  - nur der erste Parameter Pflicht ist,
  - wenn sich der erste Parameter von `MediumBase` ableitet, in die Funktion `addMediumHira` verzweigt wird.

Fastlane
Karriere

Python Standard-Bibliothek

Die Implementierung der verschiedenen Typen ist in Python in der Standard-Bibliothek organisiert. Selbst für die eingebauten Datentypen definiert der Sprachkern lediglich die Schnittstelle. Die Standard-Bibliothek ist sehr umfangreich.
Auf Linux-Systemen wird die Bibliothek üblicherweise in mehrere Pakete zerlegt, die ggf. nachinstalliert werden müssen.
Im Folgenden werden nun einige Elemente der Bibliothek näher vorgestellt.

Fastlane
Karriere

Datum und Zeit

- Aktueller Zeitpunkt mit `time` und `localtime`
- Formatieren der Zeitpunkte mit `strftime`
- Zeitpunkt erzeugen mit `mktime`
- Mit Zeitangaben rechnen

Fastlane
Karriere

Container „double-ended queue”

- Erzeugung
- Kopieren
- Iterieren
- Operatoren `+`, `*`
- Erweitern der Kette
- Einfügen/Entfernen von Elementen
- Rotieren der Kette

Brüche

- Erzeugen von Brüchen
- Umwandlung in Dezimalbruch

Fastlane
Karriere

Reguläre Ausdrücke

- Suchen von Teiltexten
- Ersetzen von Teiltexten

Aufgabe:
Ergänzen Sie die Bibliotheksanwendung. Um die Datei nicht jedes mal auf's Neue lesen zu müssen, sollen die Daten in einem Verzeichnis abgelegt werden. Dazu bekommt die Medienverwaltung jetzt interne Daten, wird vom Funktionsmodul zur Klasse.

- Legen Sie die Klasse `MedienverwaltungClass` mit einem internen Verzeichnis an.
- Schreiben Sie eine Methode `load()`, die die Datei einliest und das Verzeichnis füllt. Als Schlüssel soll dabei die Signatur dienen, als Wert eine Buch- oder CD-Instanz.
  Nehmen Sie die Implementierung von `isSignatureInFile(signatur)` als Vorlage.
- Der Konstruktor der Klasse soll den Namen der Mediendatei als Paramter übernehmen und die `load()`-Methode aufrufen.

- Schreiben Sie eine Methode `checkDuplicate(signatur)`, die prüft, ob eine bestimmte Signatur im Verzeichnis vorhanden ist.
- Schreiben Sie eine Methode `addMedium(medium)`, die `checkDuplicate` aufruft und das Medium an die Datei anhängt, danach das Verzeichnis mit `load` aktualisiert.
- Verwenden Sie den neuen Code im Hauptprogramm.

Ergänzen Sie die Klasse `MedienverwaltungClass`:

- um die Eigenschaft `sortedMedia`, eine *double ended queue*
- Im Konstruktor soll Medium für Medium nach der Signatur sortiert eingegefügt werden.
- um die Methode `printMediaSorted(reverse=False)`, die die Medien sortiert ausgibt.
  Falls der `reverse`-Parameter auf `True` steht, sollen die Medien in umgekehrter Reihenfolge ausgegeben werden.

Webprogrammierung

Pythonprogramme können auf verschiedene Weisen in einen Webserver eingebunden werden:

- CGI-Skript: Wie bei Perl, langsam, überall verfügbar
- Python-Handler `mod_python`: Wie bei PHP, schnell, nur bei Apache
- WSGI: Ein eigener Python-Application-Server, *state of the art*

Fastlane
Karriere

Wird der Apache2-Webserver verwendet, so muss im Verzeichnis `/etc/apache2/sites-available` eine Datei `myapp.conf` mit folgendem Inhalt angelegt werden:

```
Alias /myapp "/path/to/my/app"
<Directory /path/to/my/app>
  Require all granted
  AddHandler mod_python .py
  PythonHandler mod_python.publisher
  PythonDebug On
</Directory>
```

Die Einstellungen müssen mit `sudo a2ensite myapp` aktiviert werden. Danach wird der Apache neu gestartet.

Fastlane Karriere

Ein Python-Modul, das eine HTTP-Anfrage beantwortet, hat folgenden Aufbau:

```
def myfunc(req, att1, att2):
        # work with request data
        return "<HTML>";
```

Der *python handler* `mod_python.publisher` hat gut verwendbare Dispatch-Funktionen. Es ermöglicht es mit der URL das Python-Modul zu wählen, darin Funktion und Attributwerte anzugeben. Die URL `http://my.server/myapp.py/myfunc?att1=val1&att2=val2` ruft die Funktion `myfunc` im Modul `myapp` mit den angegebenen Attributwerten.

Fastlane
Karriere

Python in Ansible

Every business is a digital business. Technology is your innovation engine, and delivering your applications faster helps you win. Historically, that required a lot of manual effort and complicated coordination. But today, there is Ansible - the simple, yet powerful IT automation engine that thousands of companies are using to drive complexity out of their environments and accelerate DevOps initiatives.

- Werkzeug zur Automatisierung
- Unterstützt IT-Administratoren

Fastlane
Karriere

Ansible is a radically simple IT automation engine that automates cloud provisioning, configuration management, application deployment, intra-service orchestration, and many other IT needs.

Ansible works by connecting to your nodes and pushing out small programs, called "Ansible modules" to them. These programs are written to be resource models of the desired state of the system. Ansible then executes these modules (over SSH by default), and removes them when finished.

- Einfachheit
- Ansible-Module (=Python-Module) werden auf Zielrecher kopiert, ausgeführt, gelöscht
- Kommunikation über `ssh`
- Python und `ssh` sind auf jedem Linuxrechner vorhanden und können auf Windows nachinstaliert werden.

So funktioniert Ansible:

- In `/etc/ansible/hosts` werden die verwalteten Knoten definiert.
- Einzelkommandos werden mit dem Befehl `ansible` initiiert.
- Szenarien lassen sich in einem *playbook* zusammenfassen.
- Viele Standardmodule (`ping`, `shell`, `setup`)
- Benutzerdefinierte Module werden in Python geschrieben.

Fastlane
Karriere

Aufgabe:
Legen Sie eine statische Webseite mit folgender Funktionalität an:

- Eine Eingabemaske für das Anlegen eines neuen Mediums
- Das *Submit* ruft ein `BuchereiGUI.py`, das die Daten aus dem Request übernimmt und das `addMedium` der `Medienverwaltung` aufruft.

Modifizieren Sie `AnsibleModule.py` derart, dass zwei Parameter (`command` und `medium`) übergeben werden können.

- Für `command=add` wird über die im Modul enthaltene Funktion `addMedium` der übergebene Datensatz hinzufügt.
- Für `command=get` wird die Liste der Medien zurückgegeben.

Fastlane
Karriere

Quellen

Theis, Thomas	Einstieg in Python. Ideal für Programmiereinsteiger, 5. Auflage, 2017
C++/Vererbung	Jürgen Wolf, Grundkurs C++
Ansible	https://www.ansible.com/ (21.3.2018)
Andible-Doku	http://docs.ansible.com/ (21.3.2018)
Python	https://de.wikipedia.org/wiki/Python_(Programmiersprache) (25.7.2017)
Python-Doku	https://docs.python.org/ (1.8.2017)
Apache	https://www.howtoforge.de/anleitung/python-in-apache2-mit-mod_python-debian-etch-einbetten/ (11.2.2018)
Web-Handler	http://modpython.org/live/current/doc-html/handlers.html

Fastlane
Karriere