

# Informatik I: Einführung in die Programmierung

Exkurs: Python-Programme schreiben, kommentieren und starten

Albert-Ludwigs-Universität Freiburg



UNI  
FREIBURG

Peter Thiemann

13. November 2019



# Programme

Programme

Programme  
schreiben

Programme  
starten

Programme  
entwickeln



Programme

Programme  
schreiben

Programme  
starten

Programme  
entwickeln

- Programme = konkretisierte Algorithmen?
- Ja, aber nicht immer!
- Folge von Anweisungen und Ausdrücken, die einen bestimmten Zweck erfüllen sollen.
  - Interaktion mit der Umwelt (Benutzer, Sensoren, Dateien)
  - Unter Umständen nicht terminierend (OS, Sensorknoten, ...)
  - Auf jeden Fall meistens länger als 4 Zeilen!



# Programme schreiben

Programme

Programme  
schreiben

Programme  
starten

Programme  
entwickeln



- Umbrechen, wenn Zeilen zu lang.
- Implizite Fortsetzung mit öffnenden Klammern und Einrückung (siehe PEP8):

## Lange Zeilen

```
foo = long_function_name(var_one, var_two,
                          var_three, var_four)

def long_function_name(
    var_one, var_two, var_three,
    var_four):
    print(var_one)
```

Programme

Programme  
schreiben

Programme  
starten

Programme  
entwickeln



- Kommentiere dein Programm!
- Programme werden öfter **gelesen** als geschrieben!
- Auch der Programmierer selbst vergisst. . .
- Nicht das Offensichtliche kommentieren, sondern Hintergrundinformationen geben:  
**Warum ist das Programm so geschrieben und nicht anders?**
- Möglichst in Englisch kommentieren.

Programme

Programme  
schreiben

Programme  
starten

Programme  
entwickeln



- Der Rest einer Zeile nach # ist Kommentar.
- Blockkommentare: Zeilen, die jeweils mit # beginnen und genauso wie die umgebenden Zeilen eingerückt sind beziehen sich auf die folgenden Zeilen.

## Block-Kommentare

```
def fib(n : int):  
    # runtime is exponential in the argument  
    if n == 0:
```

- Fließtext-Kommentare kommentieren einzelne Zeilen.

## Schlechte und gute Kommentare

```
x = x + 1 # Increment x  
  
y = y + 1 # Compensate for border
```

Programme

Programme  
schreiben

Programme  
starten

Programme  
entwickeln



- #-Kommentare sind nur für den Leser.
- **docstring**-Kommentare geben dem Benutzer Informationen.
- Ist der erste Ausdruck in einer Funktion oder einem Programm (Modul) ein String, so wird dieser der *docstring*, der beim Aufruf von `help` ausgegeben wird.
- Konvention: Benutze den mit drei "-Zeichen eingefassten String, der über mehrere Zeilen gehen kann.

```
docstring
```

```
def fib(n):  
    """Computes the n-th Fibonacci number.  
    The argument must be a positive integer.  
    """  
  
    ...
```

Programme

Programme  
schreiben

Programme  
starten

Programme  
entwickeln





- Programme werden in Dateien abgelegt.
- Als Dateiname wähle *Modulname* .py
- Die Dateierweiterung .py zeigt an, dass es sich um ein Python-Programm handelt.
- *Windows*: Wähle immer *Alle Dateien* beim Sichern damit nicht .txt angehängt wird.

Programme

Programme  
schreiben

Programme  
starten

Programme  
entwickeln



# Programme starten

Programme

Programme  
schreiben

**Programme  
starten**

Programme  
entwickeln

# 5 Wege ein Programm zu starten



- Starten mit explizitem Aufruf von Python3
- Starten als Skript
- Starten durch Klicken
- Starten durch Import
- Starten in einer IDE

```
Beispielprogramm: example.py
```

```
print("Hello world")
```

Programme

Programme  
schreiben

Programme  
starten

Programme  
entwickeln



## Shell

```
# python3 example.py  
Hello world
```

- Voraussetzungen:
  - Die Datei `example.py` liegt im aktuellen Ordner.
  - Die Pfad-Variable (PATH) wurde so gesetzt, dass der Python-Interpreter gefunden wird.
- Wird normalerweise bei der Installation geleistet.
- Kann „per Hand“ nachgetragen werden:
  - *Windows*: Systemsteuerung → System und Sicherheit → Erweiterte Systemeinstellungen → Erweitert → Umgebungsvariablen
  - *Unix*: Setzen der PATH-Variable im entsprechenden Login-Skript oder in der Shell-Konfigurationsdatei (z.B. `~/.bash_profile`)

Programme

Programme  
schreiben

Programme  
starten

Programme  
entwickeln



## Shell

```
# example.py  
Hello world
```

### ■ Voraussetzungen:

- Die Datei `example.py` liegt im aktuellen Ordner.
- *Windows*: `.py` wurde als Standard-Dateierweiterung für Python registriert.
- *Unix*: Die erste Zeile in der Datei `example.py` ist:  
`#!/usr/bin/env python3`  
und die Datei hat das `x`-Bit (ausführbare Datei) gesetzt.

Programme

Programme  
schreiben

Programme  
starten

Programme  
entwickeln



- Wenn `.py` als Standard-Dateierweiterung für Python registriert ist (geht eigentlich bei allen Plattformen mit Desktop-Oberfläche), kann die Datei durch Klicken (oder Doppelklicken) gestartet werden.
- Leider wird nur kurz das Shell-Fenster geöffnet, mit Ende des Programms verschwindet es wieder.
- *Abhilfe*: Am Ende der Datei die Anweisung `input()` in das Programm schreiben.
- *Allerdings*:
  - Bei Fehlern verschwindet das Fenster
  - Beim Aufruf können keine Parameter übergeben werden.
- Für fertig entwickelte Programme mit GUI geeignet.

Programme

Programme  
schreiben

Programme  
starten

Programme  
entwickeln



- Nach dem Start von Python im Ordner, in dem `example.py` liegt:

## Python-Interpreter

```
>>> import example  
Hello world
```

- *Beachte:* Angabe ohne die Dateierweiterung!
- Die Anweisungen in der Datei werden nur beim ersten Import ausgeführt.

## Python-Interpreter

```
>>> import example  
Hello world  
>>> import example  
>>>
```

Programme

Programme  
schreiben

Programme  
starten

Programme  
entwickeln



# Programme entwickeln

Programme

Programme  
schreiben

Programme  
starten

Programme  
entwickeln

IDE

IDLE



# IDE = Integrated development environment



Editor aufrufen, Programm in der Shell starten, wieder Editor starten, ...

IDEs sind einsetzbar für:

- Projektverwaltung
- Programm editieren
- Ausführen
- Testen und *Debuggen*
- Dokumentation erzeugen
- ...

Gibt es in den verschiedensten Komplexitäts- und Qualitätsabstufungen.

Programme

Programme  
schreiben

Programme  
starten

Programme  
entwickeln

IDE  
IDLE



Wohlmöglich benannt nach Eric Idle.

- Ist 100% in Python geschrieben und benutzt die *tkinter* GUI (graphical user interface).
- Läuft auf allen Plattformen.
- Multi-Fenster-Texteditor mit Syntaxkennzeichnung, multipler Zurücknahme, smarter Einrückung.
- Enthält ein Fenster mit Python-Shell.
- Rudimentäre Debug-Möglichkeiten.
- Beschreibung siehe: <http://docs.python.org/3/library/idle.html>.

Programme

Programme  
schreiben

Programme  
starten

Programme  
entwickeln

IDE  
IDLE



Programme

Programme  
schreiben

Programme  
starten

Programme  
entwickeln

IDE

**IDLE**



- Wollen wir richtige Programme schreiben, brauchen wir Werkzeuge (**Tools**).
- **Texteditor** (nicht Word!), möglichst mit integriertem **Syntaxchecker**.
- Werden Zeilen zu lang, müssen sie **umgebrochen** werden.
- **Kommentare** sind hilfreich, um das Programm zu verstehen.
- Block-, Fließtext und `doctsring`-Kommentare
- Python-Programme können auf viele verschiedene Arten **gestartet** werden.
- **IDLE** ist eine einfache Python **IDE** (Integrated Development Environment).

Programme

Programme  
schreiben

Programme  
starten

Programme  
entwickeln

IDE  
IDLE