

The background of the slide is a complex, abstract geometric pattern. It consists of a dense network of white lines connecting small white dots, creating a mesh-like structure. Overlaid on this network are various semi-transparent blue shapes, including triangles and polygons, which vary in opacity and color intensity, ranging from light blue to a darker, more saturated blue. The overall effect is a dynamic, digital-looking texture that suggests connectivity and data.

Grundlagen der Programmierung  
Prof. Dr. Pascal Reusch

# Prof. Dr. Pascal Reusch

Professur für Produktionswirtschaft und  
Industriebetriebslehre

Fachbereich Ingenieurwissenschaften und  
Mathematik – Campus Gütersloh

Ausbildung:

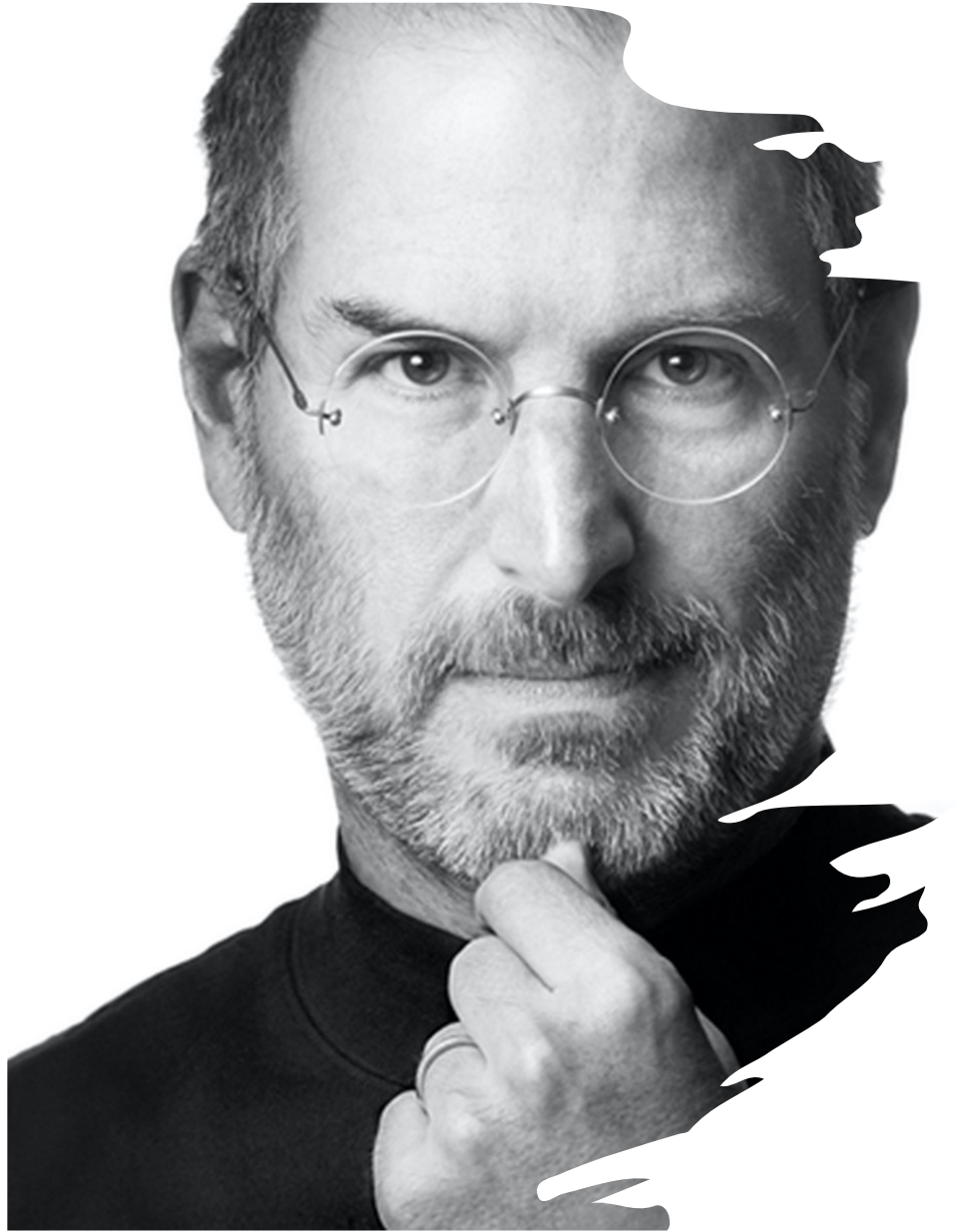
Diplom Wirtschaftsinformatik

Dr. rer. oec. (Produktionsplanung)

Forschungsaktivitäten:

Gründungsmitglied des CfADS (Center for  
Applied Data Science) der FH Bielefeld





# Learn how to think

“Everyone in this country should learn to program a computer, because it teaches you to think.” — Steve Jobs

# Problem Solving

“Almost all employers prioritize problem-solving skills first.

Problem-solving skills are almost unanimously the most important qualification that employers look for... more than programming languages proficiency, debugging, and system design.

Demonstrating computational thinking or the ability to break down large, complex problems is just as valuable (if not more so) than the baseline technical skills required for a job.”

— Hacker Rank (2018 Developer Skills Report)

	1 to 100 employees	101 to 1,000 employees	1,001+ employees	Average
Problem-solving	94.2%	94.7%	95.9%	94.9%
Programming language proficiency	55.2%	55.2%	59.0%	56.6%
Debugging	47.1%	45.0%	48.5%	47.1%
System design	37.0%	39.8%	44.1%	40.3%
Performance optimization	34.0%	35.0%	37.0%	36.0%
Code review capability	36.3%	36.3%	35.0%	35.8%
Testing	35.9%	34.4%	34.1%	34.8%
Frameworks proficiency	26.1%	23.3%	22.8%	24.2%
Database design	23.2%	20.5%	18.8%	20.9%
Codebase navigation	17.6%	14.2%	13.2%	15.1%

<https://research.hackerrank.com/developer-skills/2018>

## Which core competencies do employers look for?

	1 to 100 employees	101 to 1,000 employees	1,001+ employees	Average
Problem-solving	94.2%	94.7%	95.9%	94.9%
Programming language proficiency	55.2%	55.2%	59.0%	56.6%
Debugging	47.1%	45.0%	48.5%	47.1%
System design	37.0%	39.8%	44.1%	40.3%
Performance optimization	34.0%	35.0%	37.0%	36.0%
Code review capability	36.3%	36.3%	35.0%	35.8%
Testing	35.9%	34.4%	34.1%	34.8%
Frameworks proficiency	26.1%	23.3%	22.8%	24.2%
Database design	23.2%	20.5%	18.8%	20.9%
Codebase navigation	17.6%	14.2%	13.2%	15.1%

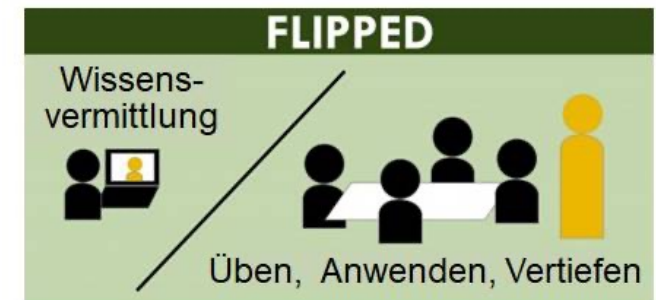


Flipped Classroom Konzept

# Flipped Classroom Konzept

## Umfang / Kapazität

Veranstaltungsart	in SWS	Kontaktzeit in Stunden	Selbststudium in Stunden
Vorlesung	2,00	0,00	56,00
Seminaristischer Unterricht	0,00	0,00	Null !!!
Übung	1,00	8,00	46,00
Seminar	0,00	0,00	0,00
Praktikum	1,00	16,00	0,00
Betreutes Selbststudium in SWS	1,00	16,00	0,00



Das Flipped Classroom Konzept ist eine Unterrichtsmethode, bei der der traditionelle Unterrichtsansatz umgedreht wird. In einem Flipped Classroom erhalten die Studierenden die Lerninhalte vor dem eigentlichen Unterricht und nutzen die Unterrichtszeit hauptsächlich für praktische Übungen, Diskussionen und den Austausch mit dem Dozenten und ihren Kommilitonen.

# Programmiersprache für den Kurs: Python

## Besonderheiten

- Dynamische Typisierung und automatisches Speichermanagement.
- Umfangreiche Standardbibliothek für verschiedene Funktionalitäten.
- Starke Community-Unterstützung und aktive Entwicklung.



---

## Grundlagen von Python

- Variablen und Datentypen (int, float, string, etc.).
- Arithmetische und logische Operatoren.
- Einfache Kontrollstrukturen (if-else, while, for).

```
i = 42  
print(type(i))
```

```
<class 'int'>
```

```
f = 17.0  
print(type(f))
```

```
<class 'float'>
```

```
s = "xxx"  
print(type(s))
```

```
<class 'str'>
```

```
sum = i + f  
print(sum)  
print(type(sum))
```

```
59.0
```

```
<class 'float'>
```

---

---

## Fortgeschrittene Datenstrukturen

- Listen, Tupel, Wörterbücher und Mengen.
- Manipulation und Iteration durch Datenstrukturen.
- List Comprehensions für prägnanten Code.

```
t = 5, 6, 7
print(type(t))
```

<class 'tuple'>

```
l = [8, 9, 10]
print(type(l))
l[0]
```

<class 'list'>

8

```
d = {"Maus": "mouse", "Katze": "cat"}
print(type(d))
print(d["Katze"])
print(d.get("Maus"))
```

<class 'dict'>

cat

mouse

```
[x for x in range(2, 10, 2)]
```

[2, 4, 6, 8]

---

---

## Funktionen und Module

- Erstellung von Funktionen und Übergabe von Argumenten.
- Importieren und Verwenden von Modulen zur Codeorganisation.
- Verständnis von Geltungsbereich und Namensräumen.

```
def fib(n):    # write Fibonacci series up to n
    a, b = 0, 1
    while a < n:
        print(a, end=' ')
        a, b = b, a+b
    print()
```

---

## Objektorientierte Programmierung in Python

- Einführung in die Konzepte von Klassen und Objekten.
- Vererbung und Polymorphismus in Python.
- Vorteile der objektorientierten Programmierung.

```
class Person:  
    def __init__(self, name, id):  
        self.name = name  
        self.age = id  
  
    def sayHello(self):  
        print("Hello! My name is " + self.name)  
  
p1 = Person("Dan", 1234)  
p1.sayHello()
```




---



## Generative KI als Tutor


PA Please explain to me how loops work in Python and give me a practical example.



In Python, loops are programming constructs that allow you to repeatedly execute a block of code based on a certain condition. There are two main types of loops in Python: `for` loops and `while` loops.   

#### 1. For Loops:


The `for` loop is used to iterate over a sequence (such as a list, tuple, string, etc.) and execute a block of code for each item in the sequence. The basic syntax of a `for` loop is:


```
python  Copy code  
  
for item in sequence:  
    # code to execute for each item
```

Here's a practical example that uses a `for` loop to iterate over a list of numbers and print each number:

```
python  Copy code  
  
numbers = [1, 2, 3, 4, 5]  
for num in numbers:  
    print(num)
```

#### 1. While Loops:

The `while` loop repeatedly executes a block of code as long as a certain condition remains true. The basic syntax of a `while` loop is:  Regenerate

Send a message 

---

## **Ziele der Lehrveranstaltung in Richtung KI**

- Entdecke die faszinierende Welt der generativen KI
- Erlebe den Spaß und die Kreativität, die generative KI in der Informatik bietet.
- Tauche ein in die Möglichkeiten, wie KI-Modelle neue Lösungen generieren können.
- Entdecke, wie generative KI-Modelle reale Probleme lösen können.
- Schärfe dein analytisches Denken und deine Fähigkeit zur kritischen Bewertung von KI-Systemen.
- Erkunde die Vor- und Nachteile des Einsatzes von KI in verschiedenen Kontexten.
- Entdecke die neuesten Trends und Technologien und passe dich schnell an.

A woman with long, wavy hair is shown from the side, wearing a blue top and a white jacket. She is sitting at a desk, typing on a silver laptop. The room is dimly lit, with a window in the background showing a bright, hazy light. The overall mood is professional and focused.

Notwendige Vorbereitungen

---

## Rahmenbedingungen Hardware

Jeder Studierende ist selber verantwortlich für sein voll funktionsfähiges Arbeitsgerät.

Empfohlen wird für die Veranstaltung die Nutzung eines **Rechners mit vollwertigem Betriebssystem** (damit dieser auch bei Präsenzveranstaltungen genutzt werden kann, vorzugsweise ein Laptop).

Die Nutzung von anderen digitalen Endgeräten wie z.B. Tablets mit eingeschränkten Betriebssystemen (iOS, Android) ist grundsätzlich auch möglich – aber mit Einschränkungen verbunden, auf die aufgrund der Vielfalt möglicher Plattformen/Apps in der Veranstaltung nicht eingegangen werden kann!



---

## Rahmenbedingungen Software

Die Studierenden sind EIGENVERANTWORTLICH bei der Einrichtung einer Entwicklungsumgebung.

Technische Probleme bei der Installation, Berechtigungen auf Firmenrechnern, WLAN-Zugang etc. können **nicht** im Rahmen der Lehrveranstaltungen gelöst werden.

Technische Probleme mit dem Hochschulnetz kann nur die IT-Servicedesk der Hochschule lösen – nicht der Dozent!

<https://www.fh-bielefeld.de/dvz/ueber-uns/it-servicedesk>

**Jeder Studierende bringt eine lauffähige Umgebung mit zu den Veranstaltungen und hat diese vorab selbstständig getestet.**



---

## Programmiersprache für den Kurs: Python

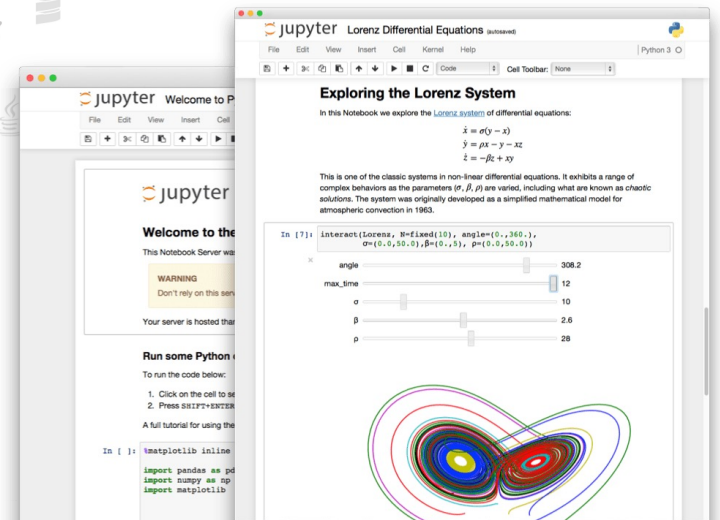
Python ist eine universelle, üblicherweise interpretierte höhere Programmiersprache. Sie hat den Anspruch, einen gut lesbaren, knappen Programmierstil zu fördern. Wegen ihrer klaren und übersichtlichen Syntax gilt Python als einfach zu erlernen.



# Standard Werkzeug für die Veranstaltung Jupyter Notebook



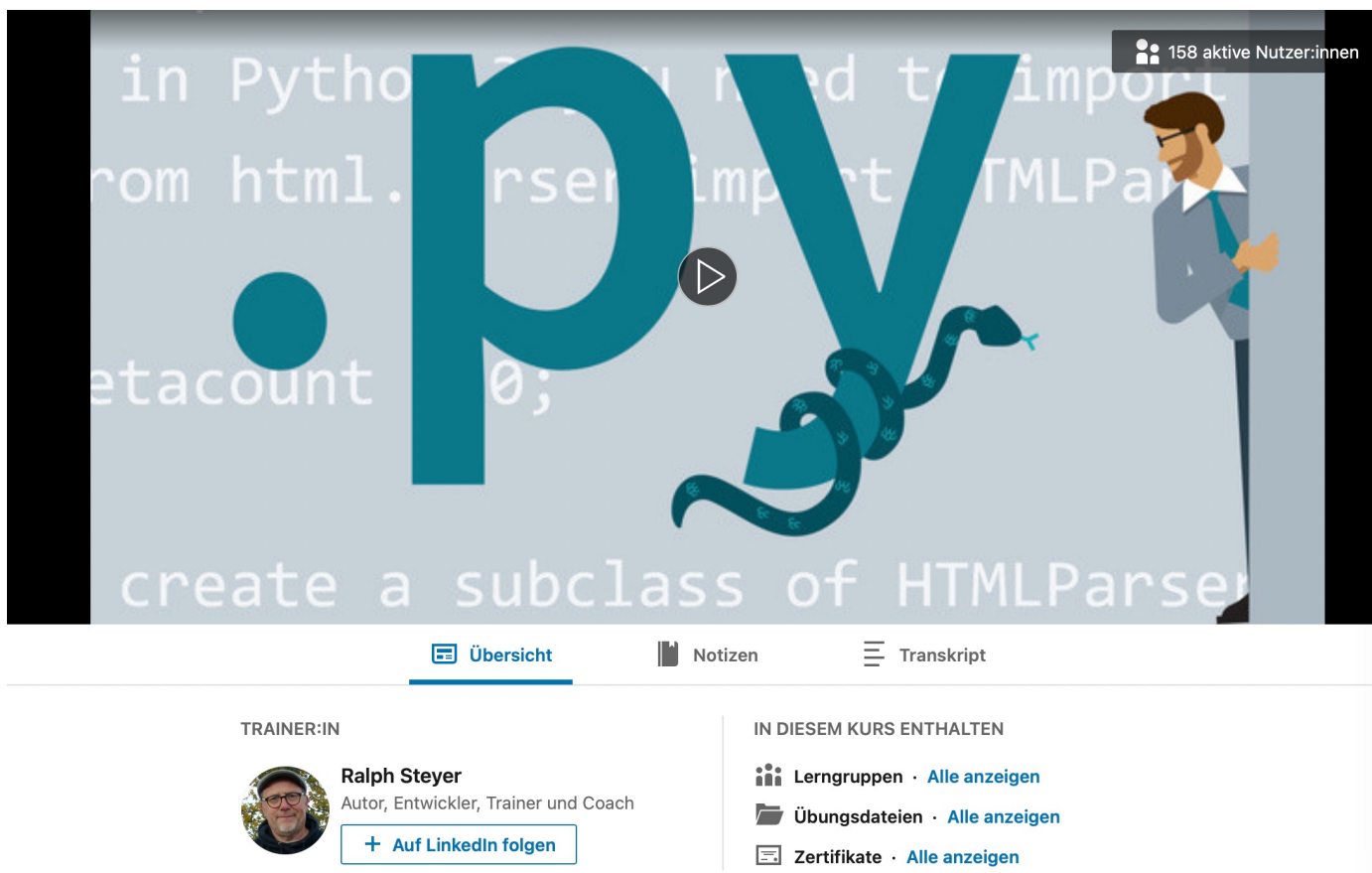
<http://jupyter.org>





Vorbereitendes Selbststudium

## Arbeiten Sie den Kurs auf LinkedIn Learning durch




in Python  
from html.  
etacount  
create a subclass of HTMLParser




158 aktive Nutzer:innen

Übersicht Notizen Transkript

TRAINER:IN

 **Ralph Steyer**  
Autor, Entwickler, Trainer und Coach  
[+ Auf LinkedIn folgen](#)

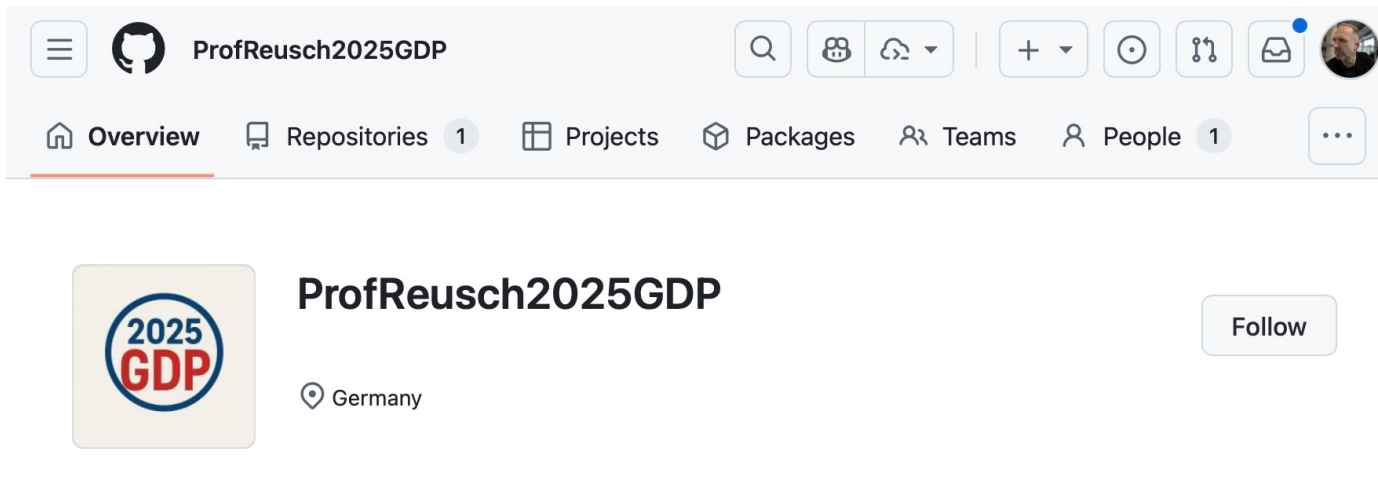
IN DIESEM KURS ENTHALTEN

-  Lerngruppen · [Alle anzeigen](#)
-  Übungsdateien · [Alle anzeigen](#)
-  Zertifikate · [Alle anzeigen](#)

<https://www.linkedin.com/learning/python-lernen/der-schnelle-und-leichte-einsteig-in-die-vielseite-programmiersprache-python>

---

**Richten Sie einen GitHub Account ein und übermitteln den Benutzernamen, damit sie zur GitHub Organisation für den Kurs hinzugefügt werden können.**



The screenshot shows the GitHub profile page for the user 'ProfReusch2025GDP'. The header includes the GitHub logo, the username 'ProfReusch2025GDP', and navigation icons for search, repositories, packages, teams, and people. Below the header, the navigation menu shows 'Overview' (selected), 'Repositories 1', 'Projects', 'Packages', 'Teams', and 'People 1'. The profile section features a profile picture with a '2025 GDP' logo, the username 'ProfReusch2025GDP', and a location tag 'Germany'. A 'Follow' button is visible on the right side of the profile.

A young woman with dark hair pulled back, wearing glasses on her head and large headphones around her neck, is seated at a light-colored wooden desk in a library. She is looking thoughtfully to her left, holding a yellow and blue pen over an open book. A smartphone is visible on the desk to her right. The background shows bookshelves filled with books and other library materials.

Ergänzende Unterlagen zum Selbststudium

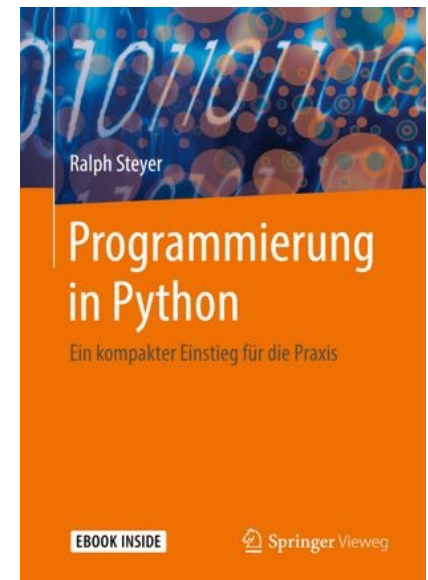
---

## Basis-Literatur zum Selbststudium

### Programmierung in Python

Ein kompakter Einstieg für die Praxis

von Ralph Steyer

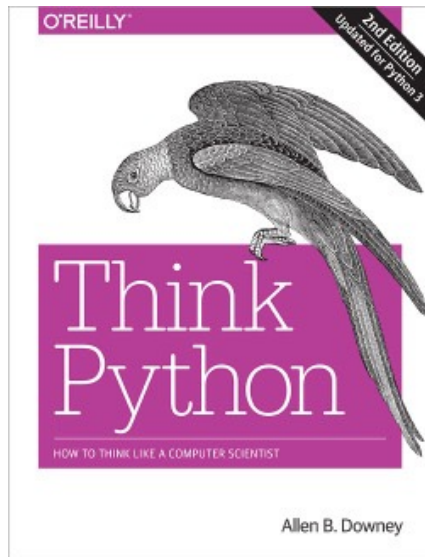


<https://link.springer.com/book/10.1007/978-3-658-20705-2>

## Basis-Literatur zum Selbststudium

*Think Python 2nd Edition* by Allen B. Downey

<https://greenteapress.com/wp/think-python-2e/>



Plus the corresponding GitHub page:

<https://github.com/AllenDowney/ThinkPython2>

Why GitHub? Team Enterprise Explore Marketplace Pricing

AllenDowney / ThinkPython2

<> Code Issues 11 Pull requests 15 Actions Projects Wiki Security Insights

master ThinkPython2 / code / Circle.py <> Jump to

AllenDowney Removing (object) from class defs

1 contributor

124 lines (89 sloc) | 2.42 KB

```
1 """This module contains a code example related to
2
3 Think Python, 2nd Edition
4 by Allen Downey
5 http://thinkpython2.com
```

# Videomaterial

Die riesige Python Community stellt Unmengen von Lernmöglichkeiten frei im Internet zur Verfügung. Nutzen Sie was immer Ihnen hilft!

Beispiele für gute Python YouTube Kanäle:

<https://www.youtube.com/user/schafer5>

The screenshot shows the YouTube channel page for Corey Schafer. At the top, there is a banner for 'COREYMS.com'. Below the banner, the channel name 'Corey Schafer' is displayed with 815,000 subscribers. Navigation tabs include 'ÜBERSICHT', 'VIDEOS', 'PLAYLISTS', 'COMMUNITY', 'KANÄLE', and 'KANALINFO'. Under 'Eigene Playlists', several video series are listed: 'Pandas Tutorials' (11 videos), 'Matplotlib Tutorials' (10 videos), 'Django Tutorials' (17 videos), 'Flask Tutorials' (15 videos), and 'Career Advice' (6 videos). A 'Python Tutorials' playlist is highlighted, showing a list of videos: 'PYTHON Installation and Setup' (15:29), 'PYTHON Strings' (21:12), 'PYTHON Integers and Floats' (11:55), 'PYTHON Lists, Tuples, and Sets' (29:08), and 'PYTHON Dictionaries' (9:59).

<https://www.youtube.com/playlist?list=PLi01XoE8jYohWFpPC17Z-wWhPOSuh8Er->

The screenshot shows a YouTube playlist titled 'Python Programming Tutorials (Computer Science)'. The playlist contains 35 videos with 2,954,184 views, last updated on 27.07.2021. The video description states: 'Our Python Tutorials will help you learn Python quickly and thoroughly. We start with "Hello World" and then move on to data structures (sets, lists, tuples and dictionaries). Next we'll cover classes, and give a variety of in-depth examples and applications.' The video list includes: 1. 'Lerne Python mit Socratica' (1:04), 2. 'Hello World in Python' (2:47), 3. 'Python Strings' (3:24), 4. 'Numbers in Python Version 2' (4:39), 5. 'Numbers in Python Version 3' (3:00), and 6. 'Arithmetic in Python V2' (6:02).

---

## **Aber der wichtigste Hinweis: Selber machen ist entscheidend!!!**

Sie können 100 Bücher lesen, wenn Sie das darin beschriebene nicht auch SELBER umsetzen, werden Sie nicht erfolgreich programmieren können.

Websites wie Udemy, Datacamp und Codecademy sind unglaubliche Ressourcen zum Erlernen neuer Programmiersprachen.

Hier ein paar ausgewählte Links für den sofortigen Einstieg:

<https://www.udemy.com/course/pythonforbeginnersintro/>

<https://www.datacamp.com/courses/intro-to-python-for-data-science>

<https://www.codecademy.com/learn/learn-python>

<https://www.w3schools.com/python/>

Wichtig ist aber vor allem das SELBER MACHEN.

Stellen Sie sich Coding Challenges im Internet.

Beispielsweise hier:

<https://www.codewars.com/>



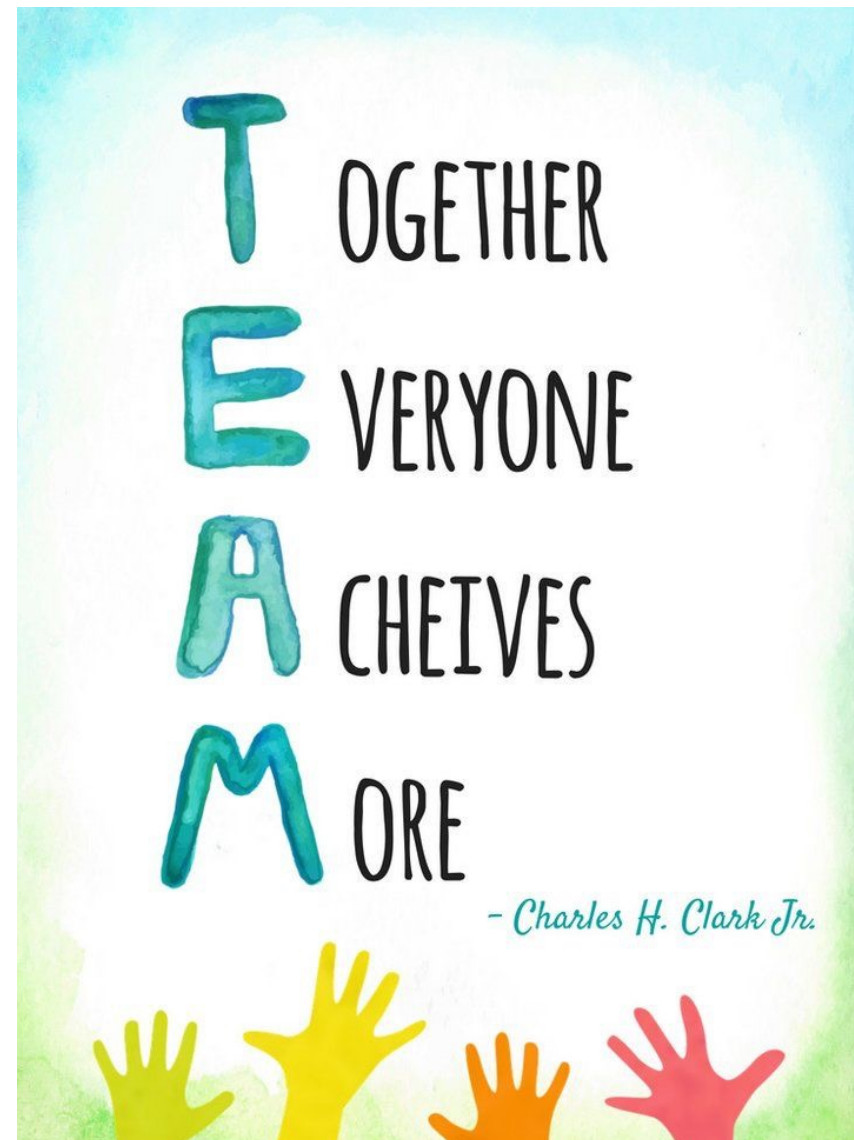
Schreib dir DEINEN Trainingsplan für die Veranstaltung!

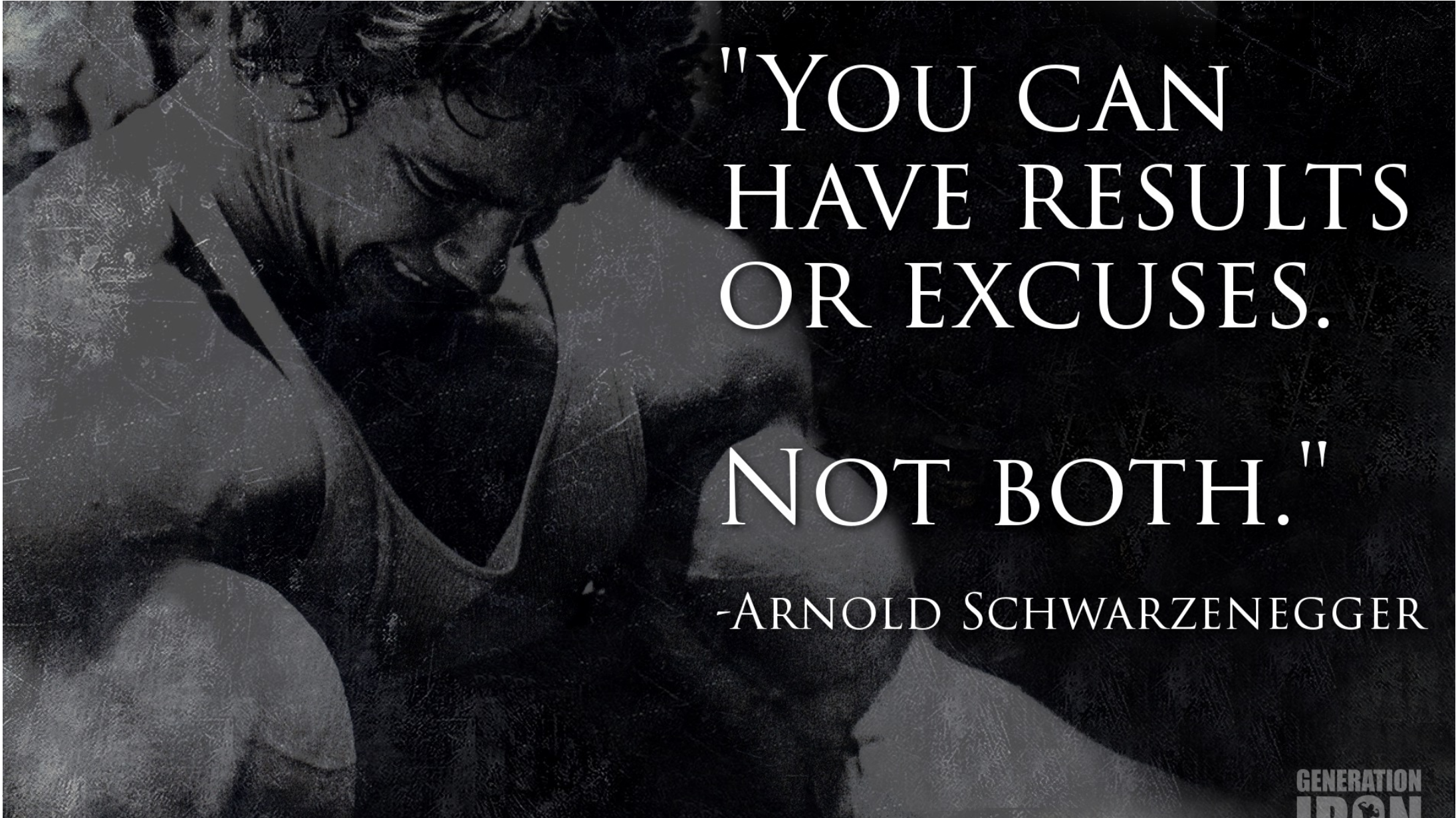
## Gemeinsam sind Sie stark!

Der größte Fehler, den Sie auf Ihrer Softwareentwicklungsreise machen können, ist, es alleine zu machen.

Formen sie ein Team und machen Sie gemeinsam Fehler und lernen, was im Laufe der Zeit funktioniert (indem wir viel versagen).

Auch sich die Zeit zu nehmen, den Code erfahrener Entwickler zu lesen, wird sich immer auszahlen.





"YOU CAN  
HAVE RESULTS  
OR EXCUSES.

NOT BOTH."

-ARNOLD SCHWARZENEGGER

GENERATION  
IRON