

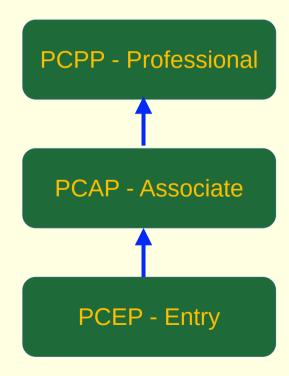
Tagesplan



- Einführung
- Kennenlernen
- Programmiersprachen
- Installation von Python3 auf Rechnern
- Erste praktische Schritte
- [evtl. Syntax]



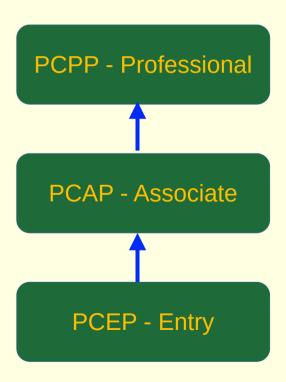
- PCAP: Python Certified Associate Programmer
- Teil eines Stufensystems





- PCAP: Python Certified Associate Programmer
- Teil eines Stufensystems

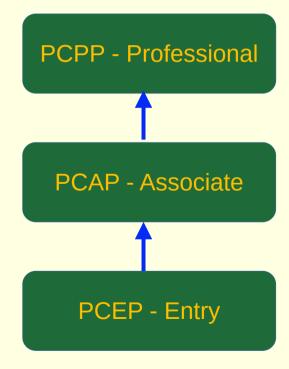
- mittlere Stufe
- brauchen erste





Stoffpläne:

Dateien/Material





- 4 Wochen für zwei Stufen
- machbar, aber herausfordernd

Vorstellungsrunde



Erfahrung mit

- Rechnern
- Programmierung
- Python

Ziele für den Kurs

- Python-Wissen
- PCAP-Zertifikat
- PCPP?

Vorwissen



- gute Sprachkenntnisse in Deutsch in Wort und Schrift
- Englischkenntnisse von Vorteil
- Gute PC Kenntnisse und hohe IT-Affinität

Vorwissen



- Dateisystem (Ordner / Dateien)
- Texteditoren
- Kommandozeile
- Programmierung?

Stoffplan



PCAP:

- Module und Pakete 12%
- Ausnahmen 14%
- Strings 18%
- Objektorientierung 34%
- Vermischtes 22%u.a. Eingabe/Ausgabe

PCAP™ - Certified Associate in Python Programmierung	Anzahl UE
Einführung und Überblick Python	45
Erste Schritte mit Python	9
Python Syntax	9
Datentypen	9
Kontrollstrukturen	9
Funktionen Parameter und Attribute	9
Funktionen und Module	54
Kombinierte Datentypen	9
Funktionen und Funktionsparameter	5
Eingebaute Funktionen	4
Seiteneffekte und Namensräume	9
Modularisierung	9
Objekt-orientierte Programmierung (OOP)	18
Datenbanken, SQL und NoSQL	36
MySQL-Workbench und NoSQL	18
Datenbankabfragen mit SQL	9
Relationale Datenbanken und Einblick Big Data / IoT	9
Grafische Benutzeroberflächen	18
Toolkits Tkinter und PyOT	5
Widgets	4
Koordinatensystem im Canvas	5
Zeichnen-Funktionen	4
Exceptionhandling	9
Projektarbeit	18
Prüfungsvorbereitung mit Vorklausur	22
Zertifizierungsprüfung	5
Gesamt UE	189



- Kameras vorzugsweise an
- Jede/Jeder sagt jeden Tag einmal etwas
- Projektarbeit zweigeteilt: imperativ
 objektorientiert
 schrittweise
- Klausur wahrscheinlich 3. November



- 7 Stunden gemeinsam statt 9
- 2 Stunden selbständig:
 - Programmieraufgaben
 - Lesen zur Vorbereitung (begleitendes Buch)

8.00 - 8.30	Gleitender Einstieg
8.30 - 9.15	Kenntnisvermittlung
9.15 - 10.00	Kenntnisvermittlung
10.00 - 10.15	Pause
10.15 - 11.00	Kenntnisvermittlung
11. ⁰⁰ - 11. ⁴⁵	Kenntnisvermittlung
11. ⁴⁵ - 12. ³⁰	Kenntnisvermittlung
12. ³⁰ - 13. ¹⁵	Mittagspause
13. ¹⁵ - 14. ⁰⁰	Praxis/Vertiefung
14 . ⁰⁰ - 14 . ⁴⁵	Kenntnisvermittlung
14. ⁴⁵ - 15. ⁰⁰	Pause
15. ⁰⁰ - 15. ⁴⁵	Praxis/Vertiefung
15. ⁴⁵ - 16. ⁰⁰	Gleitender Ausstieg





8.45 - 9.15	Gleitender Einstieg
9.15 - 10.00	Kenntnisvermittlung
10. ⁰⁰ - 10. ⁴⁵	Kenntnisvermittlung
10. ⁴⁵ - 11. ⁰⁰	Pause
11. ⁰⁰ - 11. ⁴⁵	Kenntnisvermittlung
11. ⁴⁵ - 12. ³⁰	Kenntnisvermittlung
12. ³⁰ - 13. ¹⁵	Mittagspause
13. ¹⁵ - 14. ⁰⁰	Kenntnisvermittlung / Praxis/Vertiefung
14. ⁰⁰ - 14. ⁴⁵	Praxis/Vertiefung
14. ⁴⁵ - 15. ⁰⁰	Gleitender Ausstieg



Teilnahme über Kalender

• Ein- und Ausstieg: Zeit für Fragen



Was sind und wofür brauchen wir

Programmiersprachen?



Maschinencode für eine Addition

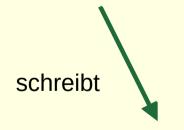
- 1 mov eax, DWORD PTR [rbp-8]
- 2 mov edx, DWORD PTR [rbp-4]
- з add eax, edx
- 4 mov DWORD PTR [rbp-12], eax

$$z.B.5 + 11$$

Python: 1 a+b







1 a+b

- 1 mov eax, DWORD PTR [rbp-8]
- 2 mov edx, DWORD PTR [rbp-4]
- з add eax, edx
- 4 mov DWORD PTR [rbp-12], eax

Interpreter / Compiler übersetzt



- Verschiedene Paradigmen
 imperativ / funktional / objektorientiert
- Verschiedene Prioritäten

Effizienz / Sicherheit / Flexibilität / Portabilität



- Entwickelt 1990er
- *Guido van Rossum*am Centrum voor Wiskunde en Informatica (CWI)
- Name von Monty Python
- Ein Ziel: leicht zu lernen



- Wir betrachten
- → Imperatives und objektorientiertes Programmieren
- Datentypen
- → Kontrollstrukturen
- **→** ...

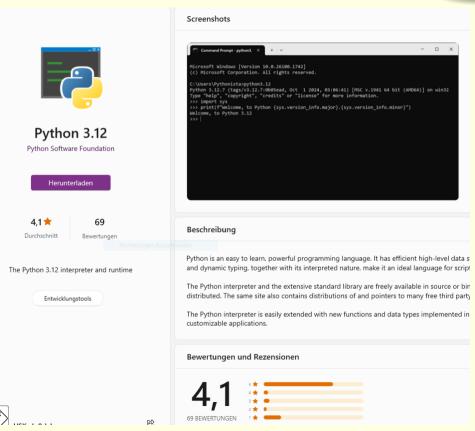
Installation von Python

Eingabeaufforderung:

>python

(nicht Powershell)





Installation von Python

- Evtl. mehrere Versionen 3.14 neueste
- Benutzbar über Kommandozeile

```
peter@Rechenkiste:~$ python3
```

Python 3.12.2 (main, Apr 16 2024, 09:36:35) [GCC 11.4.0] on linux

Type "help", "copyright", "credits" or "license" for more information.

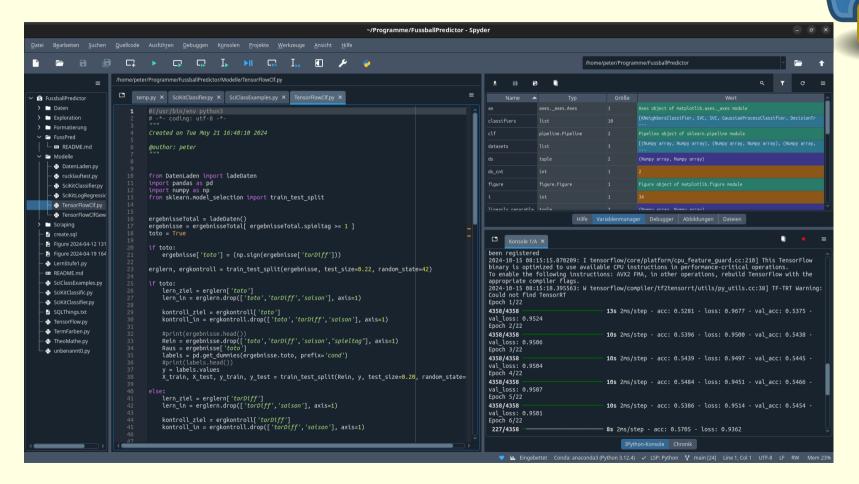
>>>



Zwei Benutzungsszenarien:

Direkt	Programmdateien
>>> 5 + 7 12	>python Dateiname.py
schnelleinfachfür das Lernen / Ausprobieren	 Eingabe verschwindet nicht Änderungen möglich für größere Projekte
 nicht für längere Eingaben Korrektur schwer verschwindet 	• mehr Aufwand

Spyder IDE



Spyder



IDE: Integrated Development Environment

Kombination aus

- Editor
- Debugger
- Laufzeitumgebung

Spyder



Herunterladen der Spyder IDE:

https://github.com/spyder-ide/spyder/releases/latest/download/Spyder-Windows-x86_64.exe

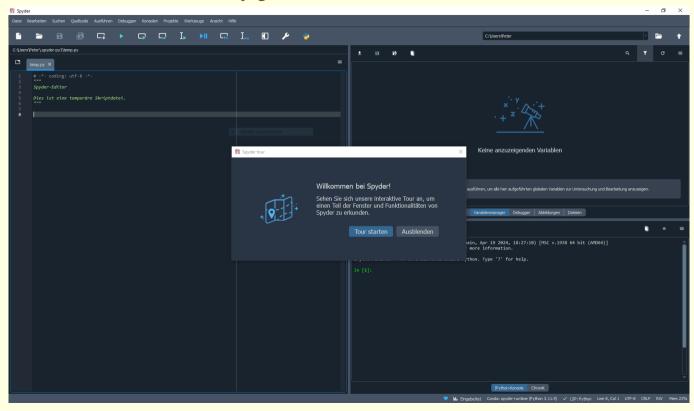
oder

https://www.spyder-ide.org/



Spyder

Installation der Spyder IDE:



Benutzung des direkten Modus



```
peter@Rechenkiste:~$ python3
Python 3.12.2 (main, Apr 16 2024, 09:36:35) [GCC 11.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> 2+3
>>> Hallo?
  File "<stdin>", line 1
    Hallo?
         Λ
SyntaxError: invalid syntax
>>> "Hallo?"
'Hallo?'
```



Benutzung der Spyder IDE o.ä.

- 1) Eingabe des Programms in Editor
- 2) Aufruf des Interpreters mit Programmname
- 3) Ausgabe erscheint nach und nach



Die Funktion print()

Gibt eine Zeichenkette aus mit Zeilenumbruch

```
>>> print("Aus")
Aus
```

Auch mehrere Zeichenketten

```
>>> print("Aus", "Zeit")
Aus Zeit
```

Trennzeichen

```
>>> print("Aus", "Zeit", sep = "-")
Aus-Zeit
```



Die Funktion print()

Zeilenumbruch kann überschrieben werden

```
>>> print( "Aus" ); print( "Zeit" )
Aus
Zeit
>>> print("Aus", end=""); print("Zeit")
AusZeit
```



Die Funktion input()

- liest Wert von Tastatur ein
- Zuweisung an Variable

```
>>> eingabe = input()
getippt
>>> print( eingabe )
getippt
```



Die Funktion input()

auch mit Prompt-Text

```
>>> eingabe = input( "Ihre Eingabe bitte: " )
Ihre Eingabe bitte: getippt
>>> print( eingabe )
getippt
```



Die Funktion input()

- liest Zeichenketten ein
- Zahlen können daraus erzeugt werden

```
>>> eingabe = int( input( "Eine Zahl bitte: " ) )
Eine Zahl bitte: 45
>>> 2 * eingabe
90
```



print() und input()

- erlauben Interaktion mit Benutzer
- auf Kommandozeilenebene



print() und input()

• Schreibweise mit ():

Namen von Funktionen