



Übungen zu Scientific Computing mit Python
Sommersemester 2025

Übungsblatt 1

Ausgabe 14.4., Übungen KW 16+17, Abgabe bis 25.4.

Scientific Computing, Computer und Python Übungen

1. Themen zur Diskussion in der Übung: Scientific Computing, Software, Programmierung

- (a) Was ist Scientific Computing? Was ist Computational Physics?
- (b) Wofür werden Computer in der Wissenschaft genutzt?
- (c) Was unterscheidet gute und schlechte Software?
- (d) Wie kann man Programmiersprachen einteilen?
- (e) Was unterscheidet eine Skriptsprache von einer kompilierten Sprache?
- (f) Welche Vorteile/Nachteile hat Python für die wissenschaftliche Programmierung?

2. Aufgabe: Python Einführung

- (a) Erstelle ein Programm, das deinen Vornamen und Nachnamen ausgibt.
- (b) Schreibe ein Programm, das zwei Zahlen einliest und deren Summe ausgibt.
TIPP: Verwende `input()` und `float()`.
- (c) Schreibe ein Programm, das 3 äquidistante Werte zwischen 0 und 1 (inklusive) ausgibt. Verallgemeinere es dann auf N Werte und das Intervall $[a, b]$.
- (d) Schreibe ein Programm, das Temperaturen von Grad Celsius in Kelvin und Fahrenheit umrechnet.
TIPP: $\text{Celsius} = 5/9 * (\text{Fahrenheit} - 32)$, $\text{Celsius} = \text{Kelvin} - 273,15$, Nullpunkt: 0 K.
- (e) Schreibe ein Programm, das zu einem gegebenen Winkel im Bogenmass (Eingabe durch den Benutzer), den entsprechenden Winkel im Gradmass berechnet. Die Ausgabe soll mit Grad ($^\circ$), Bogenminuten ($'$) und Bogensekunden ($''$) ausgegeben werden.
TIPP: Die Zahl π kann vom Modul `math` mit `from math import pi` Befehl geladen werden: `from math import pi`
- (f) Schreibe eine Funktion, die überprüft, ob eine Zahl gerade oder ungerade ist.
- (g) Schreibe jeweils eine Funktion, die eine komplexe Zahl von der kartesischen ($a + ib$) in die Polarform ($|z|e^{i\varphi}$) bzw. von der Polarform in die kartesische Form umwandelt. Teste sie an einfachen Beispielen.