

Unsere bisherigen Programme waren ausschließlich Skripte in einer Datei. Bei komplexeren Programmen stößt die Lesbarkeit dabei schnell an seine Grenzen, denn niemand möchte eine Datei mit 10 000 Zeilen Code lesen.

Außerdem kann es sein, dass man externe Codebibliotheken verwenden möchte – zum Beispiel für kompliziertere mathematische Operationen, Zufall usw. Hier kommen die sogenannten **Module** ins Spiel.

💡 Module und Imports

Ein **Modul** ist eine Codebibliothek, die durch einen **Import** im Programm verwendet werden kann. Selbst erstellte Module dienen dazu, verschiedene Programmteile voneinander zu trennen. Ein Modul erstellt man durch eine neue Datei, die den Modulnamen als Dateinamen hat (z. B. `mathe.py`). Innerhalb dieser Datei können nun Variablen, Funktionen usw. angelegt werden:

```
# Datei mathe.py
pi = 3.14159265359
def quadriere(a):
    return a**2
```

In anderen Python-Dateien im selben Verzeichnis kann man dieses Modul nun mit dem Keyword `import` importieren:

```
# Datei kreisflaeche.py
import mathe
radius = float(input("Bitte geben Sie den Radius ein: "))
area = mathe.pi * mathe.quadriere(radius)
print("Der Flächeninhalt ist", area)
```

👤 Aufgabe 1: Listenoperationen

Schreiben Sie ein Modul `list-operations`, in dem Sie die folgenden Funktionen implementieren:

- `minimum(lst)`: Diese Funktion sucht das kleinste Element in der Liste `lst` und gibt es zurück.
- `maximum(lst)`: Gibt das größte Element der Liste `lst` zurück.
- `average(lst)`: Berechnet den Durchschnitt aller Werte in der Liste `lst` und gibt ihn zurück.

Schreiben Sie dazu ein kleines Testprogramm `listentest.py`, in dem das Modul importiert wird. Erstellen Sie eine Liste (z. B. durch Nutzereingabe) und testen Sie die drei Funktionen.



Aufgabe 2: Zufall und Mathematik

Schreiben Sie ein Programm, das einen Integer `n` vom Nutzer einliest. Der folgende Code soll dann `n` mal ausgeführt werden:

- Generieren Sie mithilfe der Funktion `random()` aus dem Modul `random` einen Winkel α zwischen 0 und 360° .
- Berechnen Sie mithilfe der Funktionen `sin(alpha)` und `cos(alpha)` aus dem Modul `math` den Sinus- und Cosinus des Winkels α . Speichern Sie die Werte in zwei Listen ab (eine für die Sinuswerte, eine für die Cosinuswerte).

Berechnen Sie anschließend den Mittelwert der Sinus- und Cosinuswerte. Können Sie das Ergebnis mathematisch begründen?



Bonusaufgabe 3: Approximation von Pi

Lesen Sie eine Zahl `n` vom Nutzer ein und führen Sie folgende Schritte `n` mal aus:

- Generieren Sie einen zufälligen Punkt (x, y) , so dass $x, y \in [-1; 1]$.
- Bestimmen Sie, ob der Punkt im Einheitskreis¹ liegt und zählen Sie eine entsprechende Variable hoch.

Anschließend können Sie mit der folgenden Formel Pi approximieren:

$$\pi \approx 4 \cdot \frac{\text{Punkte im Kreis}}{\text{Punkte insgesamt}}$$

Vergleichen Sie den berechneten Wert mit dem tatsächlichen Wert von Pi (in der Variablen `pi` aus dem Modul `math`) für verschiedene Werte von `n`.

¹Der Einheitskreis ist der Kreis um den Punkt $(0, 0)$ mit Radius 1.