# Module mit Python

Mehr Modularität in unseren Projekten

#### Module und Modularität

```
pi = 3.14159

def area(radius):
   return pi * (radius**2)

def circumference(radius):
   return 2 * pi * radius
```

```
1 # circle.py
2
3 pi = 3.14159
4
5 def area(radius):
6   return pi * (radius**2)
7
8 def circumference(radius):
9   return 2 * pi * radius
```

```
import circle
print(circle.area(radius=2))
```

```
1 # prog.py
2
3 import circle
4 print(circle.area(radius=2))
```

#### Module und Modularität

- Ermöglicht die Wiederverwendbarkeit von Code
- Modulname ist der Name der Python-Datei, die den Code enthält

Aussage	Bedeutung
import circle	Alle Funktionen können mit circle.function() aufgerufen werden, z. B. circle.area(radius=2)
from circle import area	Nur die Funktion area wird aus dem Modul circle importiert. Die importierte Funktion kann dann direkt verwendet werden, z. B. area(2)

Wichtig: Bei Gesamtimport bleibt der Prefix "circle"

Erstellen Sie einen Python Skript basicfunc.py

Fügen Sie nun eine Funktion average\_of\_4(a,b,c,d) hinzu, die 4 Parameter aufnimmt und den Durchschnitt der 4 Werte zurückgibt.

Stellen Sie sicher, dass average\_of\_4(a,b,c,d) korrekt funktioniert, indem Sie es testen.

Beispiel: average\_of\_4(1, 2, 3, 4) sollte die Zahl 2,5 zurückgeben, da 1 + 2 + 3 + 4 = 10 und 10/4 = 2,5.

```
1 # basicfunc.py
2
3 def average_of_4(a, b, c, d):
4    return (a + b + c + d) / 4
```

Schreiben Sie eine Funktion compare(x,y) im Modul **basicfunc** (**basicfunc.py**), die zwei Parameter x und y aufnimmt und die folgenden Werte zurückgibt:

1, wenn x > y

0, wenn x == y

-1, wenn x < y

Stellen Sie sicher, dass compare(x,y) korrekt funktioniert, indem Sie es testen.

Beispiel: compare(4, 3) sollte 1 zurückgeben.

```
1 # basicfunc.py
2
3 def compare(x, y):
4    if x < y:
5        return -1
6    elif x > y:
7        return 1
8    else:
9        return 0
```

Schreiben Sie eine Funktion fahrenheit2celsius(f) im Modul Modul basicfunc (basicfunc.py), die als Parameter den Temperaturwert f (in Fahrenheit) aufnimmt und die umgerechnete Temperatur in Celsius zurückgibt.

Für die Umrechnung von Fahrenheit-Temperaturen in Celsius-Temperaturen ist folgende Formel zu verwenden:

(f - 32) \* 5/9

Stellen Sie sicher, dass fahrenheit2celsius(f) korrekt funktioniert, indem Sie es testen.

Beispiel: fahrenheit2celsius(32) sollte 0.0 zurückgeben.

```
1 # basicfunc.py
2
3 def fahrenheit2celsius(temp_fahrenheit):
4    return (temp_fahrenheit - 32) * 5 / 9
```

Erstellen Sie einen neues Skript **reader.py**, das einige Hilfsfunktionen zum Lesen von Benutzereingaben bereitstellt.

Bitte implementieren Sie die folgenden Funktionen:

- read\_int(): Fordert den Benutzer auf, eine Zahl einzugeben und gibt die eingegebene Zahl als int zurück.
- read\_float(): Fordert den Benutzer auf, eine Zahl einzugeben, und gibt die eingegebene Zahl als Float zurück.

Fügen Sie einen Docstring zu beiden Funktionen hinzu Stellen Sie sicher, dass die Funktionen korrekt funktionieren, indem Sie sie testen

```
1 # reader.py
2
3 def read_int():
4    """Reads an integer from the input and returns the entered number as int""
5    return int(input("Please enter a number (integer-only): "))
6
7
8 def read_float():
9    """Reads a number from the input and returns the entered number as float"""
10    return float(input("Please enter a number: "))
```

Erstellen Sie einen neues Skript calculator.py mit gleichnamiger Funktion def calculator, das keine Parameter aufnimmt, sondern den Benutzer auffordert, zwei ganze Zahlen einzugeben, und die Summe beider Zahlen zurückgibt.

Verwenden Sie eine geeignete Funktion aus dem read\_input-Modul, um die Integer-Zahlen einzulesen.

```
# calculator.py
   from reader import read_float
   def calculator():
       number_1 = read_float()
       number_2 = read_float()
 8
10
       return number_1 + number_2
11
12
   calculator()
```

Versuchen Sie nun den calculator.py zu erweitern und weitere Hilfsfunktionen aus basicfunc.py zu importieren.

Aussage	Bedeutung
import circle	Alle Funktionen können mit circle.function() aufgerufen werden, z.B. circle.area(radius=2)
from circle import area	Nur die Funktion area wird aus dem Modul circle importiert. Die importierte Funktion kann dann direkt verwendet werden, z. B. area(2)

# Ende

Das war alles für dieses Kapitel