

9.3 Beschreibung einer Funktion (DocString)

- mit dem DocString können Objekte, wie Funktionen, Klassen oder Module zu Beginn um eine Beschreibung erweitert werden
- diese Beschreibung kann dann abgerufen werden
- bei Funktionen mit `funktionsname.__doc__`

- Beispiel:

```
# Funktion mit variabler Parameteranzahl
def addieren(*summanden):
    """
    Die Funktion übernimmt eine beliebige Anzahl von Parametern.
    Die Funktion berechnet die Summe und gibt sie zurück.
    :param summanden:
    :return: summe
    """
    summe = 0
    for s in summanden:
        summe += s
    return summe

# Beschreibung der Funktion addieren ausgeben
print(addieren.__doc__)

# Aufruf mit zwei Werten
print(addieren(2, 4))

# Aufruf mit fünf Werten
print(addieren(4, 8, 12, -8))
```

```
Diese Funktion übernimmt eine beliebige Anzahl von Parametern.
Die Funktion berechnet die Summe und gibt sie zurück.
:param summanden:
:return: summe
```

6

16

9.4 Benannte Parameter und optionale Parameter einer Funktion

- die Reihenfolge der Parameter beim Aufruf muss nicht eingehalten werden, falls der Aufruf der Funktion mit benannten Parametern erfolgt
- optionale Parameter ermöglichen eine variable Parameteranzahl
- sie müssen einen Vorgabewert besitzen und am Ende der Parameterliste stehen

- Beispiel:

```
# Funktion mit benannten und optionalen Parametern
def gruss(name , grusswort="Hallo"):
    return f"{grusswort} {name}"

# == > "Hallo Herbert"
# Name ist angegeben - optionaler Parameter grusswort mit Vorgabewert
print(gruss("Herbert"))

# == > "Guten Tag Herbert"
# Name ist angegeben - optionaler Parameter grusswort ersetzt durch Wert
print(gruss(grusswort="Guten Tag", name="Herbert"))
```

```
Hallo Herbert
```

```
Guten Tag Herbert
```

9.5 Variable Anzahl von Parametern

- Funktionen lassen sich mit einer variablen Anzahl von Parametern definieren
- dabei wird bei der Definition der letzte Parameter mit * gekennzeichnet
- dieser Parameter enthält alle bis dahin noch nicht zugeordneten Parameter als Tupel

- Beispiel:

```
# Funktion mit variabler Parameteranzahl
def addieren(*summanden):
    summe = 0
    for s in summanden:
        summe += s
    return summe
```

```
# Aufruf mit zwei Werten
print(addieren(2, 4))
```

```
# Aufruf mit fünf Werten
print(addieren(4, 8, 12, -8))
```

6

16

9.6 Lokale und globale Namen

- Namen, die in Funktionen angelegt werden, werden als lokale Namen bezeichnet
- ihre Gültigkeit ist auf die Zeit der Abarbeitung der Funktion beschränkt und endet danach
- Namen außerhalb der Funktion sind global (Gültigkeit während der gesamten Laufzeit des Programms)
- auf globale Namen kann in Funktionen immer lesend zugegriffen werden
- bei Namensgleichheit überdecken die lokalen Namen die globalen Namen
- ein schreibender Zugriff in Funktionen auf globale Namen mit dem Schlüsselwort `global` – schlechter Programmierstil, weil dadurch viele Fehler und unerwünschte Seiteneffekte auftreten können

- Beispiel:

```
# Funktionen mit globalen und lokalen Namen
def fkt_1():
    print(f"fkt_1 (Zugriff auf globalen Namen): {p:8} {id(p)}")

def fkt_2():
    p = "Java"
    print(f"fkt_2 (Zugriff auf lokalen Namen): {p:8} {id(p)}")

# Definition Name p
p = "Python"
print(f"main: {p:8} {id(p)}")

# Aufruf der Funktionen ohne Parameterübergabe
fkt_1()
fkt_2()

# Werteänderung p
p = "SQL"
print(f"main: {p:8} {id(p)}")

main: Python 2897510747120

fkt_1 (Zugriff auf globalen Namen): Python 2897510747120

fkt_2 (Zugriff auf lokalen Namen): Java 2897510711088

main: SQL 28975110560486
```

Aufgabe

1. Umdrehen

Es sind Funktionen zu definieren, mit denen Reihenfolge der einzelnen Elemente von Zeichenketten, Listen, und Tupel umgedreht werden kann; für Zahlen soll das Vorzeichen umgedreht werden. Für alle genannten Datentypen sind entsprechende Testdaten zu definieren und die Werte vor und nach dem Umdrehen auf der Konsole auszugeben.

2. Visitenkarte

In einer CSV-Datei sind Personendaten zu hinterlegen (Vorname, Nachname, Strasse, Ort und Telefonnr.

Diese Daten sind einzulesen und für jede Person ist eine Visitenkarte in der Form

```
Name:      Heinrich Müller
Strasse:    Luisenstr. 5
Ort:        Düsseldorf
Telefon:    123456
```

auf der Konsole auszugeben.

4. Erweiterung Visitenkarte

Die Visitenkarten sind in eine Textdatei zu schreiben. Auf der Konsole soll die Anzahl der verarbeiteten Personendaten ausgegeben werden.