Würgeschlange 3 - Lesson 9

Tobias Maschek, Viktor Reusch
https://github.com/jemx/wise1920-python

mit Materialien von Felix Döring, Felix Wittwer https://github.com/fsr/python-lessons

7. Januar 2020

Python-Kurs

Gliederung

1. Special Method Names

2. *args und **kwargs

Special Method Names

Hey, i'm different!

```
my_list = ["a", "b", "c"]
print(len(my_list)) #3

##########################

class AudioFile:

    def __init__(self, interpret, name, length):
        self.interpret = interpret
        self.name = name
        self.length = length


song = AudioFile("Darude", "Sandstorm", 232)

print(len(song)) # TypeError: object of type 'AudioFile' has no len()
```

Hey, i'm different!

```python
class AudioFile:

    def __init__(self, interpret, name, length):
        self.interpret = interpret
        self.name = name
        self.length = length

    def __len__(self):
        return self.length

    def __str__(self):
        return f"{self.name} by {self.interpret}"


song = AudioFile("Darude", "Sandstorm", 232)

print(len(song)) #232
print(song) #Sandstorm by Darude
```

Übersicht über Special Method Names

Special Method Name	Benutzung	Erklärung
__str__	str()	informelle Darstellung
__repr__	repr()	erzeugender Python Ausdruck
__lt__, __ge__, __eq__, etc.	$<$, $>=$, $==$	Vergleichsoperatoren
__bool__	bool(), if, while	boolsche Darstellung
__len__	len()	Größe des Container-Objekts
__getitem__, __delitem__	myobj[key], del myobj[key]	ändern von Container-Elementen
__contains__	in	Element in Container enthalten
__add__, __truediv__, __imul__	$+$, $/$, $*=$	Zahlen nachbilden

Dokumentation

Aufgabe 9-1

Programmiert eine Vektor-Klasse. Wir gehen grundsätzlich von dreidimensionalen Vektoren aus. Wird nur eine Koordinate übergeben, sind die anderen Koordinaten 0. Z. B.: vect = Vector(7); entspricht dem Vektor $(7 \quad 0 \quad 0)^T$

- *len(vect)* soll den Betrag des Vektors zurückgeben
- *bool(vect)* soll *False* zurückgeben, falls der Vektor der Nullvektor ist
- ein Vektor ist größer als ein anderer, falls er einen größeren Betrag hat
- Vektoren sollen mithilfe von + addiert werden können
- Sie sollen auch mithilfe von $*$ (Skalarprodukt, $\bullet$) multipliziert werden können

Nutzt die Dokumentation!

*args und **kwargs

Problem

```python
def greet(name0, name1 = None, name2 = None):
    print(f"Hello {name0}")
    if name1 is not None:
        print(f"Hello {name1}")
    if name2 is not None:
        print(f"Hello {name2}")

greet("Alice", "Bob")
# Hello Alice
# Hello Bob
```

Lösungsansatz

Der *- und der **-Operator erlauben das Umwandeln von Argumenten von und zu Listen/Dicts.

- ***args** im Funktionskopf packt alle übergebenen *benötigten Argumente* in eine *Liste*
- ***args** im Funktionsaufruf entpackt die *Liste* args zu *Funktionsargumenten*
- ***kwargs** im Funktionskopf packt alle übergebenen *optionalen Argumente* in ein *Dict*
- ***kwargs** im Funktionsaufruf entpackt das *Dict* kwargs zu *optionalen Funktionsargumenten*

Solution

```python
def greet_at_least_one(name, *names):
    print(f"Hello {name}")
    for n in names:
        print(f"Hello {n}")

def greet_all(*names):
    for n in names:
        print(f"Hello {n}")


greet_at_least_one("Alice", "Bob")
# Hello Alice
# Hello Bob
greet_all("Alice", "Bob")
# Hello Alice
# Hello Bob
```

Beispiel

```
1 def print_squares(*numbers, **kwargs):
2     squares = [number ** 2 for number in numbers]
3     print(*squares, **kwargs)
4
5
6 print_squares(1, 3, 4, sep=", ") # 1, 9, 16
```

Aufgabe 9-2

Es soll eine Funktion zur Vergabe der Zeugnissnoten erstellt werden. Dabei kann ein Schüler beliebig viele Namen haben. Die Funktion soll

- die Namen als **args* entgegennehmen
- die Noten als ***kwargs* entgegennehmen
- den Durchschnitt auf eine Kommastelle gerundet ausgeben.

und diese entsprechend des Beispiels (sieh unten) ausgeben.

```
print_marks("Rainer", "Maria", "Rilke", german=1, english=3)
```

```
Notenuebersicht von Rainer Maria Rilke:
german: 1
english: 3
Durchschnitt: 2.0
```