

Woche 4: Funktionen und Module

Einleitung

Herzlich willkommen zur vierten Woche Ihres Python-Selbststudiums! Sie haben bereits einen soliden Einblick in die Grundlagen von Python und seine Datenstrukturen erhalten. Jetzt ist es an der Zeit, die Dinge auf die nächste Stufe zu heben und in die Welt der Funktionen und Module einzutauchen.

In dieser Woche werden wir uns auf Funktionen konzentrieren, die es Ihnen ermöglichen, wiederverwendbaren Code zu schreiben und Ihre Programme besser zu organisieren. Sie lernen, wie man Parameter und Rückgabewerte verwendet, um flexible und robuste Funktionen zu erstellen.

Darüber hinaus werden wir die Nutzung von Modulen erforschen. Module sind eine Möglichkeit, Ihre Python-Programme zu organisieren und wiederverwendbaren Code über verschiedene Programme hinweg zu nutzen. Sie lernen, wie man Module importiert und installiert, einschließlich der Nutzung wichtiger Module aus der Python-Standardbibliothek.

Ihre Aufgaben für diese Woche beinhalten das Aufzeigen der Gliederung von Programmen mit Funktionen, die Anwendung von Parameterlisten und Datentypen und das Testen des Scopes von Variablen. Sie werden auch Module von PyPI installieren und lernen, wie man eigene Module erstellt und einsetzt.

In den Übungseinheiten werden wir die Konzepte der Woche wiederholen und vertiefen.

Am Ende dieser Woche erhalten Sie einen Ausblick auf die Grundlagen der objektorientierten Programmierung, ein mächtiges Konzept, das Sie in den kommenden Wochen erforschen werden.

Freuen Sie sich auf eine spannende und herausfordernde Woche voller neuer Konzepte und Fähigkeiten. Lassen Sie uns diese Reise fortsetzen und Ihre Python-Kenntnisse auf die nächste Stufe heben!

Gesamtüberblick

Hier ein Überblick über die Inhalte und Aktivitäten der aktuellen Woche:

- Selbststudium:
 - Funktionen
 - Parameter und Rückgabewerte
 - Module
 - Modulimport
 - Modulinstallation
 - Wichtige Module der Standardbibliothek
- Aufgabe:
 - Gliederung von Programmen mit Funktionen aufzeigen
 - Parameterlisten und Datentypen anwenden
 - Scope von Variablen testen
 - Module von pypi installieren
 - Eigene Module einsetzen
- Übungseinheiten 4:
 - Wiederholung
 - Vertiefung: Funktionen
 - Vertiefung: Module
 - Ergänzung: Einsatz von virtuellen environments bzw. von Jupyter Notebooks
 - Ausblick: Grundlagen der Objektorientierten Programmierung

Inhalte und thematische Abgrenzung

Die folgende Auflistung zeigt detailliert, welche Themen Sie in der Woche behandeln und bearbeiten. Sie sind eine Voraussetzung für die folgenden Wochen und sollten gut verstanden worden sein. Wenn es Verständnisprobleme gibt, machen Sie sich Notizen und fragen Sie am Präsenztage nach, so dass wir gemeinsam zu Lösungen kommen können. Und denken Sie bitte immer daran: es gibt keine „dummen“ Fragen!

- Funktionen
 - Definition und Verwendung von Funktionen
 - Parameter und Argumente
 - Rückgabewerte
 - Standardargumente und Schlüsselwortargumente
 - Funktionen als Objekte
 - Anonyme Funktionen (Lambda)
 - Dekoratoren
 - Filter
 - Generatoren
 - Fehlerbehandlung und Ausnahmen
- Module
 - Importieren von Modulen
 - Benennen und Umbenennen von Modulen
 - Von Modulen importieren
 - Erstellen eigener Module
 - Pakete
- Python Standardbibliothek
 - Module der Standardbibliothek (math, datetime, os, sys, json, collections etc.)
 - Dateioperationen (open, read, write, close)
 - Pfadoperationen (os.path)
 - Zeit- und Datumoperationen (datetime)
 - Mathematische Operationen (math)
 - Systemoperationen (sys)
 - JSON-Operationen (json)
 - Sammlungstypen (collections)

Lernpfad

Der Lernpfad ist ein Vorschlag, in welcher Reihenfolge Sie die Inhalte der Woche angehen können. Betrachten Sie ihn gerne als eine Todo-Liste, die Sie von oben nach unten abhaken. So können Sie sicher sein, dass Sie alle wichtigen Themen bearbeitet haben und sind gut vorbereitet für die folgenden Wochen.

1. Vorbereitung

- Empfohlene Lernmaterialien bereitstellen (Bücher, Online-Kurse, Videos)
- Auffrischen der bisherigen Python-Kenntnisse

2. Grundlagen der Funktionen

- Verstehen, was eine Funktion ist und warum sie verwendet wird
- Lernen, wie man eine Funktion definiert und aufruft
- Arbeiten mit Parametern und Rückgabewerten von Funktionen
- Verstehen des Scopes und der Lebensdauer von Variablen in Funktionen

3. Erweiterte Funktionen

- Einführung in anonyme Funktionen (Lambda-Funktionen)
- Verstehen von Funktionen als Objekte und First-Class-Funktionen
- Arbeiten mit Funktionen, die Funktionen zurückgeben (Higher-Order-Funktionen)

4. Module und Pakete

- Einführung in Module und warum sie verwendet werden
- Lernen, wie man Module importiert und benutzt
- Erstellen eigener Module und das Verständnis der Dateistruktur

5. Python-Standardbibliothek

- Einführung in die Python-Standardbibliothek
- Verwendung von Standardbibliotheksmodulen wie os, sys, math, datetime, json
- Erkunden weiterer hilfreicher Module in der Standardbibliothek

6. Übungsaufgaben

- Funktionen erstellen, die auf spezifische Anforderungen reagieren
- Arbeiten mit Modulen und der Python-Standardbibliothek, um praktische Probleme zu lösen

7. Selbstbewertung

- Multiple-Choice-Bewertung absolvieren
- Wichtige Konzepte und Fähigkeiten überprüfen
- Schwierige Themen gegebenenfalls erneut durcharbeiten

8. Online-Schulungstag

Programmieraufgaben

Die folgenden Programmieraufgaben sollen Ihnen eine Anregung geben. Haben Sie eigene Ideen und Themen, die Sie ausprobieren wollen, dann sollten Sie diesen nachgehen. Wichtig ist vor allem, dass Sie „Dinge ausprobieren“. Und auch, dass Sie Fehler machen, sowohl syntaktische als auch semantische. Versuchen Sie diese Fehler zu finden und aufzulösen, dann gerade aus den Fehlern lernen Sie am Ende am meisten.

1. Erstellen Sie eine Funktion, die eine Liste von Zahlen nimmt und die Summe zurückgibt.
2. Erstellen Sie eine Funktion, die einen Textstring als Parameter nimmt und die Anzahl der Wörter im String zurückgibt.
3. Erstellen Sie eine Funktion, die eine Liste von Strings nimmt und eine neue Liste mit den Längen der einzelnen Strings zurückgibt.
4. Schreiben Sie eine Funktion, die eine Liste von Zahlen nimmt und die kleinste und größte Zahl in der Liste zurückgibt.
5. Verwenden Sie das math Modul aus der Python Standardbibliothek, um eine Funktion zu erstellen, die den Umfang eines Kreises berechnet, wenn der Radius gegeben ist.
6. Erstellen Sie eine Funktion, die das datetime Modul verwendet, um die aktuelle Zeit und das aktuelle Datum zurückzugeben.
7. Erstellen Sie eine Funktion, die den Namen einer Textdatei als Parameter nimmt und die Anzahl der Zeilen in der Datei zurückgibt.
8. Schreiben Sie eine Funktion, die eine .json Datei einliest und deren Inhalt als Python-Datenstruktur zurückgibt.
9. Schreiben Sie eine Funktion, die eine Liste von Zahlen nimmt und eine Liste von Tupeln zurückgibt, wobei jedes Tupel eine Zahl und ihren Quadratwert enthält.
10. Schreiben Sie eine Funktion, die eine verschachtelte Liste (eine Liste von Listen) nimmt und sie in eine einzige, flache Liste umwandelt.

Abschluss-Quiz

Das Quiz soll Ihnen einen ersten Hinweis auf Ihren Lernfortschritt geben. Nach unserer Einschätzung sollten Sie diese Fragen alle beantworten können, wenn Sie den Stoff der Woche durchgearbeitet und verstanden haben. Natürlich gibt es noch sehr viel mehr mögliche Fragen, dazu wollen wir auf die Literatur und das Internet verweisen. Geben Sie gerne einmal „python quizzes“ bei Google ein.

1. Was ist das Ergebnis dieser Funktion in Python?

```
def add(x):  
    return x + 2  
  
print(add(5))
```

- a) 7
- b) 5
- c) 2
- d) 8

2. Was ist das Ergebnis dieser Funktion in Python?

```
def multiply(x, y=2):  
    return x * y  
  
print(multiply(3))
```

- a) 5
- b) 6
- c) 9
- d) 1

3. Welche Funktion aus der Python Standardbibliothek kann verwendet werden, um das aktuelle Datum und die aktuelle Uhrzeit zu ermitteln?

- a) time.time()
- b) datetime.datetime()
- c) datetime.now()
- d) date.today()

4. Wie importiert man ein Modul in Python?

- a) import module_name
- b) require 'module_name'
- c) include module_name
- d) use module_name

5. Was ist das Ergebnis des folgenden Codes?

```
import math  
  
print(math.sqrt(9))
```

- a) 3.0
- b) 3
- c) 9.0

d) 81.0

6. Was macht die folgende Funktion?

```
def greet(name="User") :  
    return f"Hello, {name}"
```

- a) Sie begrüßt den Benutzer mit seinem Namen.
- b) Sie gibt "Hello, User" aus, wenn kein Name angegeben wird.
- c) Sie gibt "Hello, {name}" aus.
- d) Alle der oben genannten.

7. Wie können Sie die Quadratwurzel einer Zahl in Python berechnen?

- a) Verwenden Sie die Funktion `sqrt()` aus dem Modul `math`.
- b) Multiplizieren Sie die Zahl mit sich selbst.
- c) Verwenden Sie die Funktion `power()` aus dem Modul `math`.
- d) Keine der oben genannten.

8. Wie definiert man eine Funktion in Python?

- a) `def function_name():`
- b) `function function_name():`
- c) `function_name() = def`
- d) Keine der oben genannten

9. Was gibt `print` aus?

```
def add(x, y):  
    return x + y  
  
def do_it(func, a, b):  
    return func(a, b)  
  
print(do_it(add, 4, 5))
```

- a) 9
- b) `<function object>`
- c) nichts.
- d) Syntaxfehler

10. Wie rufen Sie eine Funktion in Python auf?

- a) `function_name()`
- b) `call function_name`
- c) `function_name.call()`
- d) `run function_name`

Ressourcen

Hier nun die Verweise auf Lernquellen, die uns für diese Woche und ihre Inhalte geeignet erscheinen. Je nachdem, welcher Lerntyp Sie sind, wählen Sie sich ihre bevorzugte Quelle, es ist nicht zwingend notwendig alle durchgearbeitet zu haben. Allerdings sollten die Inhalte des Lernpfads angesprochen und verstanden worden sein.

- **Buch:** Python Crash Course:

Funktionen: Dieses Thema wird in "Kapitel 8: Funktionen" behandelt.

- Video:
 - Codecademy: Python for Programmers – Ein erster Einstieg
 - Codecademy: Learn Python 3 – Ein umfassender Kurs mit 27h Dauer
- Lab:
 - Python Novice – Einsteiger
 - Lab: Python for Developers – Grundlegendes Python inklusive Datenstrukturen