Woche 2: Datenstrukturen

Willkommen zur zweiten Woche Ihres Python-Selbststudiums! Sie haben bereits die Grundlagen von Python gelernt und sind bereit, tiefer in die faszinierende Welt der Programmierung einzutauchen. In dieser Woche werden wir uns auf Datenstrukturen konzentrieren, eine Schlüsselkomponente jeder Programmiersprache und besonders wichtig in Python.

In dieser Woche werden wir uns auf die verschiedenen Arten von Datenstrukturen konzentrieren, die in Python verfügbar sind, einschließlich Strings, Listen, Tupeln, Mengen und Dictionaries. Sie werden lernen, wie man diese Strukturen erstellt, manipuliert und in andere Formate umwandelt. Sie werden auch lernen, wie man mit verschachtelten Strukturen arbeitet, was eine starke Fähigkeit in der Welt der Programmierung ist.

Ihre Aufgaben für diese Woche konzentrieren sich auf die praktische Anwendung dieser Konzepte. Sie werden Datenstrukturen manipulieren, sie in verschiedene Formate umwandeln und mit verschachtelten Strukturen arbeiten. Am Ende der Woche sollten Sie sich wohl fühlen, diese Konzepte in Ihren eigenen Projekten anzuwenden.

Am Übungstag werden wir das Gelernte wiederholen und vertiefen, um sicherzustellen, dass Sie ein solides Verständnis der Materie haben. Wir werden auch in die List-, Dictionary- und Set-Comprehensions eintauchen, eine mächtige und effiziente Methode, um mit Datenstrukturen in Python zu arbeiten.

Vor uns liegt eine spannende Woche voller neuer Lernmöglichkeiten. Mit jedem Schritt auf dieser Reise werden Sie sich weiter als Python-Programmierer entwickeln. Bleiben Sie dran und lassen Sie uns diese Woche mit Begeisterung angehen!

Gesamtüberblick

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

- Selbststudium:
  - Strings
  - Listen
  - Tupel
  - Mengen
  - Listenoperationen
  - Dictionaries
  - Dictionary-Operationen
- Aufgabe:
  - Manipulation von Datenstrukturen
  - Umwandlung Liste zu Tupel, Liste zu Set, String zu Liste, Liste zu String
  - Umwandlung Liste zu Dictionary und Dictionary zu Liste
  - Suche und Manipulation der Daten
  - Verschachtelte Strukturen
- Übungstag 2:
  - Wiederholung
  - Vertiefung: Anlage und Bearbeitung
  - Vertiefung: verschachtelte Strukturen
  - Ergänzung: List-/Dict-/Set-Comprehensions
  - Ausblick: Funktionen und Module

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 Präsenztag nach, so dass wir gemeinsam zu Lösungen kommen können. Und denken Sie bitte immer daran: es gibt keine „dummen“ Fragen!

- Strings
    - Erstellung und Zuweisung von Strings
    - Escape-Sequenzen in Strings
    - String-Methoden (z.B. upper(), lower(), strip(), split(), join(), etc.)
    - Formatierung von Strings (f-strings)
    - Indexierung und Slicing von Strings
    - Unveränderlichkeit von Strings
- Listen
    - Erstellen von Listen
    - Zugriff auf Listenelemente
    - Hinzufügen und Entfernen von Elementen
    - Slicing von Listen
    - Listen durchlaufen
- Tupel
    - Erstellen von Tupeln
    - Zugriff auf Tuplelemente
    - Unveränderlichkeit von Tupeln
    - Tupel durchlaufen
- Mengen
    - Erstellen von Mengen
    - Hinzufügen und Entfernen von Elementen in Mengen
    - Set-Operationen (Vereinigung, Schnittmenge, Differenz)
- Listenoperationen
    - Listen sortieren
    - Listen umkehren

    - Anzahl der Elemente in einer Liste finden
    - Prüfung, ob ein Element in einer Liste vorhanden ist
- Dictionaries
    - Erstellen von Dictionaries
    - Zugriff auf Dictionary-Elemente
    - Hinzufügen und Entfernen von Elementen in einem Dictionary
    - Durchlaufen eines Dictionaries
- Dictionary-Operationen
    - Überprüfen, ob ein Schlüssel in einem Dictionary vorhanden ist
    - Abrufen aller Schlüssel und Werte aus einem Dictionary
    - Verschachtelte Dictionaries

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
    - Installation zusätzlicher Python-Bibliotheken (falls erforderlich)
    - Bereitstellung empfohlener Lernmaterialien (Bücher, Online-Kurse, Videos)
2. Strings
    - Verständnis der Struktur und Eigenschaften von Strings
    - Manipulation von Strings (Zugriff auf Zeichen, Slicing, Methoden)
    - Umwandlung zwischen Strings und Listen
3. Listen
    - Verstehen, wie Listen in Python funktionieren
    - Manipulation von Listen (Hinzufügen/Entfernen von Elementen, Slicing, Methoden)
    - Umwandlung zwischen Listen und anderen Datentypen (Tupel, Mengen)
4. Tupel
    - Verständnis der Eigenschaften und Verwendungszwecke von Tupeln
    - Manipulation von Tupeln (Zugriff auf Elemente, Slicing, Methoden)
    - Umwandlung zwischen Tupeln und Listen
5. Mengen
    - Verständnis der Eigenschaften und Verwendungszwecke von Mengen
    - Durchführung von Mengenoperationen (Vereinigung, Schnittmenge, Differenz)
    - Umwandlung zwischen Mengen und Listen
6. Listenoperationen
    - Durchführung verschiedener Operationen auf Listen (Sortierung, Suche, Aggregation)
    - Anwendung von List Comprehensions zur effizienten Manipulation von Listen
7. Dictionaries
    - Verstehen, wie Dictionaries in Python funktionieren
    - Manipulation von Dictionaries (Hinzufügen/Entfernen von Schlüssel-Wert-Paaren, Zugriff auf Werte, Methoden)

- Umwandlung zwischen Dictionaries und Listen

8. Dictionary-Operationen
    - Durchführung verschiedener Operationen auf Dictionaries (Sortierung, Suche, Aggregation)
    - Anwendung von Dictionary Comprehensions zur effizienten Manipulation von Dictionaries
9. Übungsaufgaben
    - Manipulation von Datenstrukturen
    - Umwandlung zwischen verschiedenen Datentypen
    - Suche und Manipulation der Daten
    - Arbeit mit verschachtelten Strukturen
10. Selbstbewertung
    - Durchführung von Multiple-Choice-Bewertungen
    - Überprüfung der wichtigsten Konzepte und Fähigkeiten
    - Nachbearbeitung von schwierigen Themen
11. Online-Schulungstag
    - Wiederholung der gelernten Themen
    - Vertiefung der Arbeit mit Datenstrukturen
    - Einführung in List-/Dict-/Set-Comprehensions
    - Ausblick auf Funktionen und Module

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.

- Strings:
    - Schreiben Sie ein Python-Programm, um einen eingegebenen String umzudrehen.
    - Schreiben Sie ein Python-Programm, das alle Vokale in einem gegebenen String zählt.
    - Schreiben Sie ein Python-Programm, das einen String in eine Liste von Wörtern umwandelt, und dann die Liste wieder in einen String umwandelt.
- Listen:
    - Schreiben Sie ein Python-Programm, das die Elemente einer gegebenen Liste in umgekehrter Reihenfolge ausgibt.
    - Schreiben Sie ein Python-Programm, das das größte und kleinste Element in einer Liste findet.
    - Schreiben Sie ein Python-Programm, das alle Duplikate aus einer Liste entfernt.
- Tupel:
    - Schreiben Sie ein Python-Programm, das ein Tupel in eine Liste umwandelt und umgekehrt.
    - Schreiben Sie ein Python-Programm, das das n-te Element eines gegebenen Tupels findet.
    - Schreiben Sie ein Python-Programm, das überprüft, ob ein gegebenes Element in einem Tupel vorhanden ist.
- Mengen:
    - Schreiben Sie ein Python-Programm, das die Vereinigung und Überschneidung zweier Mengen berechnet.
    - Schreiben Sie ein Python-Programm, das überprüft, ob eine Menge eine Teilmenge einer anderen ist.
    - Schreiben Sie ein Python-Programm, das eine Liste in eine Menge umwandelt und alle Duplikate entfernt.
- Listenoperationen:
    - Schreiben Sie ein Python-Programm, das die Summe aller Zahlen in einer gegebenen Liste berechnet.

    - Schreiben Sie ein Programm, das eine CSV einliest und intern als Liste (Zeilen) von Listen (Felder) darstellt

- Dictionaries:
    - Schreiben Sie ein Python-Programm, das ein Dictionary erstellt, das Zahlen von 1 bis n und ihre Quadrate enthält (n ist eine Eingabe des Benutzers).
    - Schreiben Sie ein Python-Programm, das alle Schlüssel eines gegebenen Dictionary ausgibt.
    - Schreiben Sie ein Python-Programm, das ein Dictionary von einer Liste von Schlüsseln und einer Liste von Werten erstellt.
- Dictionary-Operationen:
    - Schreiben Sie ein Python-Programm, das die Summe aller Werte in einem gegebenen Dictionary berechnet.
    - Schreiben Sie ein Python-Programm, das ein Dictionary in eine Liste von Tupeln umwandelt (jedes Tupel besteht aus einem Schlüssel und seinem zugehörigen Wert).
    - Schreiben Sie ein Python-Programm, das überprüft, ob ein gegebener Schlüssel in einem Dictionary vorhanden ist.

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. Welcher der folgenden Datentypen in Python ist unveränderlich?
    1. List
    2. Dictionary
    3. Set
    4. Tuple
2. Wie greifen Sie auf das letzte Element einer Liste namens 'my_list' zu?
    1. my_list[0]
    2. my_list[-1]
    3. my_list[end]
    4. my_list[1]
3. Wie erstellen Sie eine neue Menge in Python?
    1. set = {1, 2, 3}
    2. set = Set(1, 2, 3)
    3. set = [1, 2, 3]
    4. set = (1, 2, 3)
4. Wie entfernen Sie ein Element aus einem Dictionary?
    1. del dict[element]
    2. dict.remove(element)
    3. dict.pop(element)
    4. dict.delete(element)
5. Wie fügen Sie ein Element zu einer Liste hinzu?
    1. list.add(element)
    2. list.insert(element)
    3. list.extend(element)
    4. list.append(element)
6. Welche Methode wird verwendet, um ein Element aus einer Liste zu entfernen?

    1. remove()
    2. pop()
    3. delete()
    4. discard()

7. Wie erstellen Sie ein Dictionary, das Schlüssel-Wert-Paare speichert?
    1. {'key': 'value'}
    2. ('key', 'value')
    3. {key: value}
    4. ["key", "value"]
8. Wie überprüfen Sie, ob ein Schlüssel in einem Dictionary vorhanden ist?
    1. 'key' in dict
    2. dict.has_key('key')
    3. dict.contains('key')
    4. 'key' in dict.keys()
9. Wie erstellen Sie ein leeres Tuple in Python?
    1. tuple = ()
    2. tuple = []
    3. tuple = {}
    4. tuple = empty()
10. Wie überprüfen Sie, ob ein Element in einer Menge enthalten ist?
    1. set.contains(element)
    2. element in set
    3. set.has(element)
    4. element is set

Lösungen:

1/4 2/2 3/1 4/3 5/4 6/1 7/1 8/1 9/1 10/2

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 erstanden worden sein.

- Buch: Python Crash Course:
  - **Strings**: Dieses Thema wird noch in "Kapitel 2: Variablen und einfache Datentypen" mit behandelt.
  - **Listen**: Eine Einführung in die Listen erfolgt in "Kapitel 3: Einführung in Listen". Die Listenoperationen werden in "Kapitel 4: Mit Listen arbeiten" weiter ausgeführt. In diesem werden auch die Tupel mit abgehandelt.
  - **Dictionaries**: "Kapitel 6: Dictionaries" führt diese Datenstruktur und die mit ihr zusammenhängenden Operationen ein.

- Video: CodeAcademy
  - 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