

Woche 2: Verzweigungen und Schleifen und ihr Einsatz in Python

Einleitung

Willkommen zur zweiten Woche unseres spannenden Python-Kurses! Nachdem Sie in der ersten Woche die Grundlagen von Python kennengelernt haben, werden Sie nun in die faszinierende Welt der Verzweigungen und Schleifen eintauchen. Diese Woche steht ganz im Zeichen des dynamischen Programmierens - einer Fähigkeit, die für jeden angehenden Programmierer unerlässlich ist.

Verzweigungen und Schleifen sind die Bausteine, um reaktive und effiziente Programme zu erstellen. Sie ermöglichen es uns, auf unterschiedliche Eingaben und Situationen angemessen zu reagieren (Verzweigungen) und wiederholte Aktionen effizient durchzuführen (Schleifen). Diese Konzepte sind nicht nur in Python, sondern in fast allen Programmiersprachen von zentraler Bedeutung.

Stellen Sie sich vor, Sie könnten ein Programm schreiben, das intelligente Entscheidungen trifft - zum Beispiel unterschiedliche Aktionen basierend auf Benutzereingaben ausführt oder automatisch bestimmte Aufgaben wiederholt, ohne dass Sie jeden Schritt einzeln programmieren müssen. Das ist die Kraft von Verzweigungen und Schleifen!

Diese Woche wird Ihnen zeigen, wie Sie Ihre Ideen in Python-Programme umsetzen können, die nicht nur funktionieren, sondern auch smart und effizient sind. Wir werden gemeinsam durch praktische Beispiele und Übungen gehen, die Ihnen helfen, diese Konzepte zu verstehen und anzuwenden.

Freuen Sie sich auf eine Woche voller Lernerfolge, in der Sie Ihre Programmierfähigkeiten auf das nächste Level heben werden. Am Ende dieser Woche werden Sie in der Lage sein, Python-Programme zu schreiben, die logische Entscheidungen treffen und Aufgaben automatisiert ausführen können. Lassen Sie uns gemeinsam die Möglichkeiten von Python entdecken und nutzen!

Gesamtüberblick

Hier ein detaillierter Überblick über die geplanten Inhalte und Aktivitäten dieser Woche:

- Selbststudium:
 - Verzweigungen:
 - Grundlagen der if-, else- und elif-Anweisungen
 - Erstellen bedingter Ausdrücke
 - Praktische Beispiele für Entscheidungsstrukturen
 - Schleifen:
 - Einführung in die while-Schleife
 - Einführung in die for-Schleife
 - Anwendung von Kontrollstrukturen wie break und continue
 - Einsatz von Schleifen in praktischen Beispielen
- Aufgaben:
 - Verzweigungen: Erstellen einfacher if-else-Programme
 - Verzweigungen: Anwendung von elif-Anweisungen für mehrfache Bedingungen
 - Schleifen: Einsatz von while-Schleifen für wiederholte Aktionen
 - Schleifen: Einsatz von for-Schleifen zur Iteration über Listen und andere Datenstrukturen
- Übungstag 2:
 - Wiederholung und Vertiefung: Verzweigungen und ihr Einsatz
 - Wiederholung und Vertiefung: Schleifen und ihr Einsatz
 - Ergänzung: Fehlerbehandlung mit try - except
 - Ausblick: Funktionen und Datenstrukturen, die in den kommenden Wochen behandelt werden

Inhalte und thematische Abgrenzung

Die folgende Auflistung zeigt detailliert auf, welche Themen in der zweiten Woche behandelt und bearbeitet werden. Diese Inhalte bauen auf den Grundlagen der ersten Woche auf und sind essentiell für das Verständnis weiterführender Konzepte. Sollten Sie Fragen oder Unsicherheiten haben, zögern Sie nicht, diese am Übungstag anzusprechen. Denken Sie daran, dass es keine „dummen“ Fragen gibt und dass wir hier sind, um gemeinsam zu lernen!

1. Verzweigungen in Python

- Was sind Verzweigungen und warum sind sie wichtig?
- Grundlagen der if-, elif- und else-Anweisungen
- Erstellen und Verstehen bedingter Ausdrücke
- Praktische Anwendungsbeispiele für Verzweigungen

2. Schleifen in Python

- Einführung in die while-Schleife: Konzept und Anwendung
- Einführung in die for-Schleife: Nutzung für Iterationen
- Kontrollstrukturen innerhalb von Schleifen: break und continue
- Erstellen von Schleifen für praktische Anwendungsfälle

3. Kombinierte Anwendung von Verzweigungen und Schleifen

- Einsatz von Verzweigungen innerhalb von Schleifen
- Erstellung komplexerer Programme durch Kombination von Verzweigungen und Schleifen
- Fallbeispiele und Übungen zur Vertiefung

4. Best Practices und häufige Fehler

- Gängige Fehler beim Einsatz von Verzweigungen und Schleifen und wie man sie vermeidet
- Tipps zur Verbesserung der Lesbarkeit und Effizienz von Code
- Richtlinien für die Einrückung und Strukturierung von Codeblöcken

5. Praktische Übungen und Herausforderungen

- Verschiedene Übungen zur Anwendung von Verzweigungen und Schleifen in Python
- Entwicklung kleiner Projekte oder Skripte, die das Gelernte anwenden
- Gelegenheiten, eigene Lösungsansätze zu entwickeln und zu diskutieren

Lernpfad

Der Lernpfad für die zweite Woche dient als Leitfaden, um die neuen Konzepte schrittweise zu erarbeiten. Sie können ihn als Checkliste nutzen, um sicherzustellen, dass Sie alle wichtigen Themen abdecken und gut vorbereitet in die dritte Woche starten.

1. Kurze Wiederholung der wichtigsten Konzepte aus Woche 1: Variablen, Datentypen und grundlegende Python-Syntax.
2. Lernen Sie die Grundlagen der if-, elif- und else-Anweisungen. Verstehen Sie, wie Bedingungen in Python formuliert und genutzt werden.
3. Machen Sie sich mit der Struktur und Funktionsweise von while-Schleifen vertraut. Einführung in for-Schleifen und deren Anwendung auf Listen und andere Datentypen.
4. Üben Sie die Erstellung komplexerer Bedingungen und den Einsatz von verschachtelten if-Anweisungen. Erarbeiten Sie praktische Beispiele, um das Gelernte zu festigen.
5. Erfahren Sie, wie man mit Schleifen repetitive Aufgaben löst. Lernen Sie den Einsatz von break und continue innerhalb von Schleifen.
6. Experimentieren Sie mit der Kombination von Verzweigungen und Schleifen
7. Lösen Sie Übungen, die das Zusammenspiel von if-Anweisungen und Schleifen in realen Szenarien zeigen. Entwicklung kleiner Projekte, in denen beide Konzepte integriert werden.
8. Erfahren Sie mehr über gängige Fehler mit Schleifen und Verzweigungen und wie man sie vermeidet. Lernen Sie, wie man Code klar und wartbar strukturiert.
9. Lösen Sie spezifische Aufgabenstellungen, um Ihr Verständnis zu überprüfen und zu vertiefen. Arbeiten Sie an kleinen Projekten, die praktische Anwendungsfälle simulieren.
10. Wiederholen Sie alle gelernten Konzepte und bereiten Sie Fragen für den Übungstag vor. Nutzen Sie die Gelegenheit, Unklarheiten zu klären und tiefer in die Materie einzutauchen.

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.

- Einfaches Entscheidungsprogramm mit if-else-Strukturen

Erstellen Sie ein Programm, das den Benutzer nach seinem Alter fragt. Das Programm soll dann entscheiden, ob der Benutzer jugendlich (unter 18 Jahren), erwachsen (zwischen 18 und 65 Jahren) oder im Rentenalter (über 65 Jahre) ist. Verwenden Sie dazu if-, elif- und else-Anweisungen.

- Entwicklung eines Zählers mit einer while-Schleife

Schreiben Sie ein Programm, das Zahlen von 1 bis 10 auf dem Bildschirm ausgibt. Verwenden Sie dazu eine while-Schleife. Das Programm soll jede Zahl in einer neuen Zeile ausgeben.

- Anwendung einer for-Schleife zur Durchlaufung einer Liste

Erstellen Sie eine Liste mit mindestens fünf verschiedenen Obstsorten. Verwenden Sie dann eine for-Schleife, um jede Obstsorte in der Liste nacheinander auszugeben.

- Kombination von Verzweigungs- und Schleifenstrukturen in einem Programm

Entwickeln Sie ein Programm, das Zahlen von 1 bis 30 ausgibt. Für jede Zahl soll das Programm überprüfen:

- Wenn die Zahl durch 3 teilbar ist, soll "Fizz" ausgegeben werden.
- Wenn sie durch 5 teilbar ist, soll "Buzz" ausgegeben werden.
- Wenn sie sowohl durch 3 als auch durch 5 teilbar ist, soll "FizzBuzz" ausgegeben werden.
- In allen anderen Fällen soll die Zahl selbst ausgegeben werden.

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. Diese und das nächste Abschlussquiz sind etwas umfangreicher, als die dann später im Kurs folgenden. Der Grund dafür ist, dass die hier zu legenden Grundlagen so wichtig sind, um Ihnen einen guten Erfolg und viel Freude an der Programmierung zu sichern.

Teil 1 - Grundfragen zu Verzweigungen und Schleifen

1. Was ist der Zweck einer Verzweigung in der Programmierung?
 - a) Daten zu speichern
 - b) Eine Bedingung zu überprüfen und basierend darauf unterschiedliche Aktionen auszuführen
 - c) Einen Codeblock wiederholt auszuführen
 - d) Die Ausführung des Programms zu beenden
2. Was beschreibt eine Schleife in der Programmierung am besten?
 - a) Eine einmalige Ausführung eines Codeblocks
 - b) Eine Anweisung, die das Programm beendet
 - c) Eine Struktur, die einen Codeblock wiederholt ausführt
 - d) Eine Fehlermeldung im Programm
3. In welchem Fall wird der else-Block einer if-else-Struktur ausgeführt?
 - a) Wenn die Bedingung der if-Anweisung wahr ist
 - b) Nach jeder Ausführung des if-Blocks
 - c) Wenn die Bedingung der if-Anweisung falsch ist
 - d) Der else-Block wird immer ausgeführt
4. Welche Aussage über eine while-Schleife ist richtig?
 - a) Sie führt einen Codeblock basierend auf einer festen Anzahl von Durchläufen aus
 - b) Sie wird mindestens einmal ausgeführt, unabhängig von der Bedingung
 - c) Sie wiederholt einen Codeblock, solange eine Bedingung wahr bleibt
 - d) Sie kann nur für Zählvorgänge verwendet werden
5. Was ist der Hauptunterschied zwischen einer while-Schleife und einer for-Schleife?
 - a) for-Schleifen sind in der Programmierung nicht bekannt
 - b) while-Schleifen sind schneller als for-Schleifen
 - c) while-Schleifen eignen sich besser für eine unbekannte Anzahl von Durchläufen

d) for-Schleifen können keine Bedingungen überprüfen

6. Welches Konzept beschreibt das vorzeitige Beenden einer Schleife?

a) Unterbrechung

b) Abbruch

c) Überlauf

d) Rückkehr

7. Was passiert, wenn die Bedingung einer while-Schleife immer wahr bleibt?

a) Die Schleife stoppt automatisch nach einer bestimmten Zeit

b) Die Schleife wird unendlich oft ausgeführt

c) Die Schleife verwandelt sich in eine for-Schleife

d) Die Schleife wird vom Compiler ignoriert

8. Welcher Zweck wird typischerweise mit einer for-Schleife verfolgt?

a) Einen Codeblock für jedes Element einer Sammlung auszuführen

b) Eine unendliche Schleife zu erzeugen

c) Bedingungen zu überprüfen

d) Globale Variablen zu deklarieren

9. Was ist eine typische Verwendung für eine verschachtelte Schleife?

a) Die Erstellung von komplexen Algorithmen zu vermeiden

b) Die Ausführungsgeschwindigkeit des Programms zu erhöhen

c) Datenstrukturen wie Matrizen zu verarbeiten

d) Fehler im Code zu generieren

10. Was ist eine typische Anwendung für eine if-Anweisung?

a) Einen Codeblock kontinuierlich zu wiederholen

b) Verschiedene Aktionen basierend auf bestimmten Bedingungen auszuführen

c) Einen Codeblock ohne Bedingungen auszuführen

d) Daten ohne Überprüfung zu speichern

Teil 2 - Verzweigungen und Schleifen in Python

1. Wie beginnt man eine if-Anweisung in Python?

- a) if (Bedingung):
- b) if [Bedingung]:
- c) if Bedingung:
- d) if {Bedingung}:

2. Wie erstellt man in Python eine Schleife, die eine bestimmte Anzahl von Malen durchlaufen wird?

- a) while i < n:
- b) for i < n:
- c) for i in range(n):
- d) while i in range(n):

3. Welches Schlüsselwort wird in Python verwendet, um den aktuellen Durchlauf einer Schleife zu beenden und mit dem nächsten fortzufahren?

- a) pass
- b) continue
- c) break
- d) next

4. Wie wird in Python eine Bedingung geschrieben, die prüft, ob eine Variable x gleich 10 ist?

- a) if x = 10:
- b) if x == 10:
- c) if x === 10:
- d) if x != 10:

5. In Python, wie wird eine while-Schleife initialisiert, die so lange läuft, bis die Variable x größer als 5 ist?

- a) while x > 5:
- b) while x <= 5:
- c) while x != 5:
- d) while x == 5:

6. Was macht die else-Anweisung in einer for-Schleife in Python?

- a) Sie wird nach jeder Iteration der Schleife ausgeführt.
- b) Sie wird ausgeführt, wenn die Schleife normal beendet wird, ohne dass ein break-Statement erreicht wird.

- c) Sie wird immer ausgeführt, unabhängig davon, ob ein break-Statement erreicht wird.
- d) Sie wird nur ausgeführt, wenn die Schleife keine Iterationen hat.

7. Welches Schlüsselwort wird verwendet, um in Python eine unendliche Schleife zu erstellen?

- a) forever
- b) repeat
- c) while True:
- d) infinite

8. Wie kann in Python eine for-Schleife über eine Liste namens my_list iterieren?

- a) for item in my_list:
- b) for item = my_list:
- c) for item <= my_list:
- d) for item != my_list:

9. Was bewirkt das break-Schlüsselwort in einer Schleife in Python?

- a) Es hält die Schleife vorübergehend an.
- b) Es setzt die Schleife zurück zum Anfang.
- c) Es beendet die gesamte Schleife sofort.
- d) Es überspringt den Rest des aktuellen Schleifendurchlaufs.

10. Wie überprüft man in Python, ob eine Liste my_list leer ist?

- a) if len(my_list) == 0:
- b) if my_list is None:
- c) if my_list:
- d) if not my_list:

Teil 3 - Vertiefende Fragen zu Python

1. Was passiert, wenn die Einrückung in einem Python-Codeblock inkonsistent ist?

- a) Der Code wird normal ausgeführt, aber langsamer.
- b) Der Code wird ohne Fehlermeldung ignoriert.
- c) Es wird ein SyntaxError ausgelöst.
- d) Die Einrückung wird automatisch vom Python-Interpreter korrigiert.

2. Wie kann man in Python eine else-Anweisung mit einer for-Schleife kombinieren?

- a) Die else-Anweisung wird nach jedem Durchlauf der for-Schleife ausgeführt.
- b) Die else-Anweisung wird ausgeführt, wenn die for-Schleife normal endet, ohne durch ein break unterbrochen zu werden.
- c) Die else-Anweisung in Kombination mit einer for-Schleife ist in Python nicht zulässig.
- d) Die else-Anweisung wird vor der for-Schleife ausgeführt.

3. Welches Ergebnis liefert der folgende Python-Code:

Python-Code:

```
if not (True or False):  
    print("A")  
elif not (True and False):  
    print("B")  
else:  
    print("C")
```

- a) A
- b) B
- c) C
- d) Keine Ausgabe

4. In Python, wie kann man mehrere Bedingungen in einer if-Anweisung prüfen?

- a) if condition1 and condition2:
- b) if condition1 plus condition2:
- c) if condition1 & condition2:
- d) if condition1 then condition2:

5. Was bewirkt das continue-Statement in einer Schleife in Python?

- a) Es beendet sofort die gesamte Schleife.
- b) Es startet die nächste Iteration der Schleife.
- c) Es pausiert die aktuelle Iteration der Schleife.
- d) Es überspringt alle folgenden Schleifen.

6. Wie viele Male wird der Body einer while-Schleife in Python ausgeführt, wenn die Bedingung anfangs False ist?

- a) Einmal
- b) Zweimal
- c) Gar nicht
- d) Unendlich oft

7. Was ist der korrekte Weg, um in Python eine unendliche Schleife zu erstellen?

- a) while True:
- b) for ever:
- c) while 1 == 1:
- d) Sowohl a) als auch c) sind korrekt.

8. Welcher Fehler tritt auf, wenn in Python eine Schleife oder Verzweigung ohne entsprechenden Codeblock (nur mit einem Pass-Statement) erstellt wird?

- a) SyntaxError
- b) IndentationError
- c) NameError
- d) Kein Fehler tritt auf

9. Wie kann man in Python mehrere Bedingungen nacheinander prüfen?

- a) Mit verschachtelten if-Anweisungen
- b) Mit mehreren if-Anweisungen hintereinander
- c) Mit einer if-elif-else-Kette
- d) Sowohl a), b) als auch c) sind korrekt.

10. Was ist der korrekte Weg, um in Python eine for-Schleife vorzeitig zu beenden?

- a) exit()
- b) break
- c) stop
- d) return

Ressourcen

Hier nun die Verweise auf Lernquellen, die uns für diese Woche und ihre Inhalte geeignet erscheinen. Wie sie sicherlich bemerken wählen wir dazu die gleichen Medien aus, wie letzte Woche, da diese die Themen ebenfalls weiterführen. 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:** Get Programming: Learn to Code with Python

Grundlagen der Programmierung mit Hilfe von Python erklärt. Dieses Buch passt sehr gut zu den ersten beiden Wochen des Kurses, zeigt aber auch darüber hinausgehende Konzepte in verständliche Form.

- **Buch:** Python Crash Course

Verzweigung (if-Anweisungen): Dieses Thema wird in "Kapitel 5: if-Anweisungen" behandelt.

Schleifen: While-Schleifen werden in "Kapitel 7: Benutzereingabe und while-Schleifen" eingeführt. Für andere Arten von Schleifen, wie zum Beispiel for-Schleifen, möchten Sie

vielleicht "Kapitel 4: Arbeiten mit Listen" konsultieren

- **Video:** CodeAcademy

Codecademy: Python for Programmers – Ein erster Einstieg

Codecademy: Learn Python 3 – Ein umfassender Kurs mit 27h Dauer

- **Lab:**

Python Novice – Einsteiger

Python for Developers – Grundlegendes Python inklusive Datenstrukturen