

Python Crashkurs: Aufgabenzettel

Aufgabe 1: Grundlegende Ein-/Ausgabe, Variablen und if/else

Ziel der Übung: Konsolenausgabe und -eingabe, Variablenzuweisung, einfache bedingte Logik.

1. Erstelle ein Python-Skript `tage_im_monat.py`
2. Frage den Benutzer nach einem Monat (als Zahl 1–12) und einem Jahr (z.B. 2024) mittels Konsoleneingabe. Achte darauf, dass das Programm beendet werden soll, wenn eine invalide Eingabe getätigt wird.
3. Speichere die Eingaben in Variablen und wandle sie in eine Ganzzahl um.
4. Verwende eine `if/elif/else`-Struktur, um die Anzahl der Tage im jeweiligen Monat zu bestimmen (Januar = 31, Februar abhängig von Schaltjahr, ...).
Hinweis: Ein Jahr ist Schaltjahr, wenn es durch 4 teilbar ist, aber nicht durch 100, außer es ist durch 400 teilbar. Benutze den Modulo % Operator
5. Gib die Anzahl der Tage in der Konsole aus, z.B.: „Februar 2024 hat 29 Tage.“
6. Setze die gleiche Logik mit einer `match-case`-Struktur (Switch-Case) statt `if/elif`, um die Monatszuordnung zu implementieren.

Aufgabe 2: Schleifen, Dictionaries und Funktionen

Ziel der Übung: Verwendung von `while`, `for`- und `for each`-Schleifen, Dictionaries, und Funktionen.

1. Erstelle ein neues Skript `notendurchschnitt.py`.
2. Definiere eine Funktion `def eingaben_sammeln()` – sie fragt in einer Schleife Noten (0–100) ab, bis der Benutzer `q` eingibt.
 - a. Speichere jede Note in einer Liste.

Hinweis: Der Hinweis ist unsichtbar markiere, wenn du nicht mehr weiter weißt neben diesem Text ->

3. Definiere eine Funktion `berechne_statistiken(noten)`
 - a. Berechne Minimum, Maximum und Durchschnitt.

Hinweis: Benutze nicht die `math` Bibliothek

- b. Speichere die Ergebnisse in einem Dictionary mit den Schlüsseln 'min', 'max', 'avg'.
 - c. Gebe die Statistik als Dictionary zurück
- 4. Rufe beide Funktionen im Skript auf und gib die Statistiken formatiert aus.
- 5. Erweitere das Programm um weitere Funktionen:
 - a. Verwende eine for-Schleife, um jede Note zusammen mit ihrem Index (beginnend bei 1) auszugeben.
 - b. Gebe eine Liste an Noten zusammen mit den Statistiken aus

Aufgabe 3: Standardbibliotheken, NumPy und Git-Integration

1. Erstelle eine Requirements.txt für die Version von NumPy die du verwenden möchtest (Recherchiere die neuste Version)
2. Installiere NumPy über die Requirements.txt
3. Erstelle ein neues Skript `datenanalyse.py`
4. Importiere NumPy und erzeuge ein NumPy-Array aus zufälligen Ganzzahlen zwischen 0 und 100 in der Größe 1×10 .
 - a. Verwende NumPy-Funktionen, um min, max und avg zu berechnen.
 - b. Vergleiche die Ergebnisse mit den eigenen Funktionen aus Aufgabe 2.
 - c. Gib alles in der Konsole aus.
5. Git-Integration:
 - a. Erstelle ein Git-Repository (GUI oder CLI)
 - b. Lasse dir das Git-Repository in VS-Code anzeigen
 - c. Lade deinen Code in dein Remote Repository

Lösungen befinden sich in den entsprechenden Dateien

Bonusaufgabe:

Wenn du diese Aufgabe lösen kannst, bist du über den Grundlagen von Python hinaus.

Analysiere die `bonus.py` Datei und erkläre in einem Kommentar, was genau jeder Codeblock ist. Erkundige dich über die verwendete Syntax für die Deklaration der Variablen. Benenne dabei die Arten der Schleifen, erkläre Funktionen und wo diese benutzt werden. Informiere dich, wofür die verwendeten Bibliotheken genutzt werden.

Bei diesem Skript handelt es sich um eine alte Uniabgabe von mir, ich habe alle Variablen und Funktionen umbenannt, sodass es nicht gleich zu erahnen ist, worum es sich bei diesem Skript handelt. Finde irgendwie heraus worum es sich bei diesem Skript handelt. Schicke mir die Lösung zu wir sprechen darüber, die Bonusaufgabe ist nicht in den Lösungen.

Nach lösen dieser Aufgabe und wenn du Interesse hast, erstelle ich eine kurze Präsi zu Neuroinformatik und erkläre so ein wenig, wie KI funktioniert.