# Aufgabenblatt Teil 2

## Teil 4: Anweisungsfolgen

### Aufgabe 1: Lineare Anweisungsfolge

1. Schreiben Sie ein Python-Programm, das folgende Schritte ausführt:
   * Definieren Sie eine Variable startwert und setzen Sie sie auf 10.
   * Addieren Sie 5 zu startwert und speichern Sie das Ergebnis in einer neuen Variable zwischenwert.
   * Multiplizieren Sie zwischenwert mit 2 und speichern Sie das Ergebnis im Endwert.
   * Geben Sie alle drei Variablen aus.

### Aufgabe 2: Kontrollstrukturen

1. Schreiben Sie ein Programm, das entscheidet, ob eine Zahl gerade oder ungerade ist:
   * Fordern Sie den Benutzer auf, eine Zahl einzugeben.
   * Nutzen Sie eine if-else-Anweisung, um die Zahl zu überprüfen.
2. Schreiben Sie ein Programm, das überprüft, ob eine eingegebene Zahl größer, kleiner oder gleich 100 ist.
3. Schreiben Sie ein Programm, das prüft, ob ein Benutzer ein Passwort korrekt eingibt.
4. Schreiben Sie ein Programm, das überprüft, ob eine Zahl zwischen 50 und 100 liegt.
5. Schreiben Sie ein Programm, das das Alter eines Benutzers abfragt und prüft, ob er Auto fahren darf
   * Ab 18: "Du darfst Auto fahren."
   * Unter 18, aber über 16: "Du darfst einen Führerschein machen."
   * Unter 16: "Du bist zu jung für den Führerschein."
6. Schreiben Sie ein Programm, das zwei Zahlen abfragt und entscheidet
   * Ob die Zahlen gleich sind.
   * Welche der beiden Zahlen größer ist.
   * Ob die Summe der beiden Zahlen gerade oder ungerade ist.

## Teil 5: Print-Ausgaben und Escape-Zeichen

### Aufgabe 3: Grundlegende Print-Ausgabe

1. Schreiben Sie ein Programm, das folgende Informationen ausgibt:
   * Ihren Namen.
   * Ihre Ausbildung.
   * Ein beliebiges Zitat.
   * Verwenden Sie Formatierung.

### Aufgabe 4: Verwendung von Escape-Zeichen

1. Nutzen Sie Escape-Zeichen, um folgende Ausgaben zu erzeugen:
   * Einen Tabulator
   * Einen Zeilenumbruch

## Teil 6: Input und Output

### Aufgabe 5: Benutzereingabe

1. Schreiben Sie ein Programm, das folgende Daten vom Benutzer abfragt:
   * Name
   * Alter
   * Beruf
2. Geben Sie eine Begrüßungsnachricht basierend auf den Eingaben aus

### Aufgabe 6: Eingaben validieren

1. Schreiben Sie ein Programm, das eine Zahl vom Benutzer abfragt und überprüft, ob die Eingabe gültig ist:
   * Verwenden Sie isdigit(), um sicherzustellen, dass die Eingabe eine Zahl ist.
   * Falls die Eingabe ungültig ist, geben Sie eine Fehlermeldung aus.

## Teil 7: Datentypen und Variablen

### Aufgabe 7: Variablen mit verschiedenen Datentypen

1. Erstellen Sie Variablen mit den folgenden Datentypen:
   * Ganze Zahl (int).
   * Gleitkommazahl (float).
   * Text (str).
   * Wahrheitswert (bool).
2. Geben Sie die Variablen zusammen mit ihrem Datentyp aus.

### Aufgabe 8: Typkonvertierung

1. Schreiben Sie ein Programm, das eine Ganzzahl in eine Gleitkommazahl und einen String umwandelt.
2. Geben Sie alle drei Versionen aus.

## Teil 8: Exception Handling

### Aufgabe 9: Grundlegende Fehlerbehandlung

1. Schreiben Sie ein Programm, das eine Zahl vom Benutzer abfragt.
2. Verwenden Sie try-except, um sicherzustellen,
3. Schreiben Sie ein Programm, das den Benutzer auffordert, eine Zahl einzugeben, und basierend auf der Eingabe verschiedene Fehler behandelt:
   * Wenn die Eingabe keine Zahl ist, geben Sie eine Fehlermeldung aus.
   * Wenn die Zahl durch 0 geteilt wird, geben Sie eine Warnung aus.
   * Andernfalls geben Sie das Ergebnis der Division durch 2 aus.
4. Schreiben Sie ein Programm, das eine Zahl prüft:
   * Wenn die Zahl negativ ist, soll ein benutzerdefinierter Fehler ausgelöst werden.