

# Python Grundlagen - Basisfunktionen

## For-Schleifen

Definition: Eine for-Schleife wird verwendet, um ueber eine Sequenz (z. B. Listen, Strings oder Zahlenbereiche) zu iterieren. Dabei wird eine Anweisung fuer jedes Element ausgefuehrt.

Beispiel:

```
for zahl in range(1, 11):  
    print(zahl) # Ausgabe: 1, 2, 3, ... , 10
```

## While-Schleifen

Definition: Eine while-Schleife fuehrt Code aus, solange eine Bedingung wahr ist. Die Bedingung wird vor jedem Durchlauf ueberprueft.

Beispiel:

```
counter = 0  
  
while counter < 3:  
    print('Zaehler:', counter)  
    counter += 1
```

## If-Anweisungen

Definition: if-Anweisungen erlauben bedingte Codeausfuehrung. Mit elif und else koennen weitere Bedingungen geprueft werden.

Beispiel:

```
x = 10  
  
if x > 5:  
    print('Groesser als 5')  
  
elif x == 5:  
    print('Gleich 5')
```

else:

```
print('Kleiner als 5')
```

## Zufallszahlen

Definition: Mit dem Modul random koennen Zufallszahlen generiert werden, z. B. fuer Spiele oder Simulationen.

Beispiel:

```
import random

zufallszahl = random.randint(1, 10)

print(zufallszahl) # Ausgabe: z. B. 7
```

## Funktionen und Hauptprogramm

Definition: Funktionen helfen dabei, wiederverwendbaren Code zu schreiben. Eine Funktion wird mit dem Schluesselwort def definiert und kann Argumente akzeptieren und Werte zurueckgeben.

Das Hauptprogramm (\_\_\_main\_\_\_) wird verwendet, um sicherzustellen, dass ein Skript nur ausgefuehrt wird, wenn es direkt gestartet wird - und nicht, wenn es als Modul importiert wird.

Beispiel:

```
def addiere(a, b):

    return a + b


if __name__ == '__main__':

    import sys

    a, b = int(sys.argv[1]), int(sys.argv[2])

    ergebnis = addiere(a, b)

    print('Ergebnis:', ergebnis) # Beispielausgabe: Ergebnis: 8
```

## Verzeichnisse und Dateien

Definition: Python bietet Module wie `os` und `os.path`, um Verzeichnisse zu durchsuchen und Dateien zu öffnen.

Beispiele:

- Verzeichnisse auflisten:

```
import os
```

```
print(os.listdir('.')) # Gibt alle Dateien im aktuellen Verzeichnis zurück
```

- Datei öffnen und lesen:

```
with open('beispieldatei.txt', 'r') as datei:
```

```
    inhalt = datei.read()
```

```
    print(inhalt)
```

## Uebungen

### 1. Generiere ein Einmaleinstafel

**Beschreibung:** Erstelle ein kleines Einmaleins mit einer schoenen Überschrift.

**Eingabe:** Keine

**Ausgabe:**

\*\*\*\*\*kleines Einmaleins\*\*\*\*\*

1 x 1 = 1

1 x 2 = 2

...

10 x 10 = 100

### 2. Benutzerinteraktion mit einer Laenderliste

**Beschreibung:** Frage den Namen des Benutzers und sein Herkunftsland ab. Überpruefe, ob das Land in einer gegebenen Liste ist (Link per QR-Code bereitgestellt). Wenn nicht, fuege es hinzu.

**Eingabe:** Benutzername: Peter, Land: Narnia

**Ausgabe:**

- Falls das Land neu ist: Peter, Dein Land ist jetzt neu auf die Liste eingeschrieben.
- Falls es schon existiert: Peter, Dein Land ist bereits auf der Liste.

### 3. Argumente verarbeiten und Dateien auflisten

**Beschreibung:** Akzeptiere ein Benutzerargument, konvertiere es in einen String und speichere es. Liste anschliessend alle Dateien im aktuellen Verzeichnis auf.

**Eingabe:** Argument: my\_argument

**Ausgabe:**

**Argument gespeichert: my\_argument**

**Dateien im aktuellen Verzeichnis: ['datei1.txt', 'ordner1']**

#### **4. Dateien und Unterverzeichnisse auflisten**

**Beschreibung:** Liste alle Dateien und Unterverzeichnisse (rekursiv) im aktuellen Verzeichnis auf.

**Eingabe:** Keine

**Ausgabe:**

**./datei1.txt**

**./ordner1/datei2.txt**

#### **5. Datei öffnen**

**Beschreibung:** Öffne eine Datei im Ordner Talente und zeige ihren Inhalt.

**Eingabe:** Datei: talente.txt

**Ausgabe:** Inhalt der Datei wird angezeigt.