

Python-Projekt: Ordnergröße berechnen mit Python

Lernziele

- Mit Python durch Ordner und Dateien laufen (os.walk)
- Dateigrößen ermitteln (os.path.getsize)
- Gesamtspeicherplatz eines Verzeichnisses berechnen
- Ergebnisse benutzerfreundlich anzeigen (MB/GB)
- Eigenes Analyse-Tool für Speicherplatz entwickeln

Schritt 1 – „Hallo Ordnergröße“

Theorie:

Wir beginnen mit einer einfachen Ausgabe.

Beispielcode:

```
print("Programm zur Berechnung der Ordnergröße gestartet!")
```

Übung:

1. Schreibe ein Programm, das „Berechnung gestartet“ und am Ende „Berechnung beendet“ ausgibt.
2. Ergänze das Programm so, dass dein Name und das aktuelle Datum angezeigt werden.

Schritt 2 – Dateigröße einer einzelnen Datei

Theorie:

Mit `os.path.getsize` können wir die Größe einer Datei in Bytes herausfinden.

Beispielcode:

```
import os
```

```
file_path = "/home/user/daten/test.txt" # oder  
"C:\\Users\\Rico\\Documents\\daten\\test.txt"
```

```
print(f"Größe der Datei: {os.path.getsize(file_path)} Bytes")
```

Übung:

1. Erstelle eine Testdatei und prüfe ihre Größe.
2. Gib die Größe nicht in Bytes, sondern in Kilobytes aus (Tipp: durch 1024 teilen).

Schritt 3 – Durch einen Ordner laufen

Theorie:

Mit `os.walk` können wir durch alle Unterordner und Dateien iterieren.

Beispielcode:

```
import os  
  
folder = "/home/user/daten"  
  
for dirpath, dirnames, filenames in os.walk(folder):  
    for f in filenames:  
        print(os.path.join(dirpath, f))
```

Übung:

1. Liste alle Dateien in deinem Testordner auf.
2. Zähle die Anzahl der Dateien und gib sie am Ende aus.

Schritt 4 – Gesamte Ordnergröße berechnen

Theorie:

Wir summieren die Dateigrößen, um die Gesamtgröße zu erhalten.

Beispielcode:

```
import os

def get_folder_size(path):
    total_size = 0
    for dirpath, dirnames, filenames in os.walk(path):
        for f in filenames:
            fp = os.path.join(dirpath, f)
            if os.path.exists(fp):
                total_size += os.path.getsize(fp)
    return total_size

folder = "/home/user/daten" # oder
"C:\\Users\\Rico\\Documents\\daten"
size = get_folder_size(folder)
print(f"Größe von '{folder}': {size} Bytes")
```

Übung:

1. Berechne die Ordnergröße in Bytes.
2. Ändere die Ausgabe so, dass die Größe in Kilobytes angezeigt wird.

Schritt 5 – Benutzerfreundliche Ausgabe (MB/GB)

Theorie:

Große Zahlen sind schwer lesbar – besser in MB oder GB anzeigen.

Beispielcode:

```
print(f"Größe von '{folder}': {size / (1024*1024):.2f} MB")
```

Übung:

1. Stelle die Größe in MB dar (zwei Nachkommastellen).
2. Ergänze eine zweite Ausgabe in GB.
3. Teste dein Programm mit einem größeren Ordner.

Erweiterungsideen

- Ergänze eine Eingabe (input()), sodass der Benutzer den Ordnerpfad selbst eingeben kann.
- Gib zusätzlich die Anzahl der Dateien und Unterordner aus.
- Schreibe die Ergebnisse in eine Logdatei (ordnergroesse.log).