

Python-Grundlagen für Jugendliche: Dictionarys, JSON und Dateiverarbeitung

Hier findest du eine einfache Erklärung zu wichtigen Konzepten in Python, die dir beim Programmieren helfen.

1. Was ist ein Dictionary?

- **Dictionarys** (deutsch: Wörterbücher) sind spezielle Datentypen in Python, die Daten als „Schlüssel-Wert-Paare“ speichern.
- Jeder Schlüssel (key) ist eindeutig und zeigt auf einen Wert (value).

Beispiel:

```
person = {  
    "name": "Anna",  
    "alter": 15,  
    "stadt": "Friedrichshafen"  
}  
print(person["name"]) # Ausgabe: Anna
```

- Du kannst auf Werte zugreifen, indem du den Schlüssel in eckigen Klammern angibst.

2. Was ist JSON und wie funktioniert es in Python?

- **JSON** steht für „JavaScript Object Notation“ und ist ein Format, um Daten als Text zu speichern und auszutauschen.
- In Python kannst du das `json`-Modul verwenden, um Dictionarys in JSON umzuwandeln oder umgekehrt.

Beispiel: Dictionary zu JSON machen:

```
import json  
  
person = {"name": "Anna", "alter": 15}  
json_text = json.dumps(person)  
print(json_text) # Ausgabe: {"name": "Anna", "alter": 15}
```

Beispiel: JSON zu Dictionary machen:

```
import json  
  
json_text = '{"name": "Anna", "alter": 15}'  
person = json.loads(json_text)  
print(person["alter"]) # Ausgabe: 15
```

3. Datei speichern (File Write)

- Um Daten in eine Datei zu schreiben, benutzt du `open()` mit dem Modus `"w"` (write).
- Mit `write()` schreibst du Text in die Datei.

Beispiel:

```
text = "Hallo Welt!"
with open("beispiel.txt", "w") as datei:
    datei.write(text)
# Jetzt steht "Hallo Welt!" in der Datei beispiel.txt
```

- Das `with` sorgt dafür, dass die Datei nach dem Schreiben automatisch geschlossen wird.

4. Datei laden (File Open/Read)

- Um eine Datei zu lesen, benutzt du `open()` mit dem Modus `"r"` (read).
- Mit `read()` liest du den Inhalt als Text ein.

Beispiel:

```
with open("beispiel.txt", "r") as datei:
    inhalt = datei.read()
print(inhalt) # Ausgabe: Hallo Welt!
```

5. JSON in eine Datei speichern und laden

Speichern:

```
import json

daten = {"name": "Anna", "alter": 15}
with open("daten.json", "w") as datei:
    json.dump(daten, datei)
```

Laden:

```
import json

with open("daten.json", "r") as datei:
    daten = json.load(datei)
print(daten["name"]) # Ausgabe: Anna
```

Tipps für den Einstieg

- Probiere die Beispiele selbst aus.
- Ändere Werte und beobachte, was passiert.
- Wenn du Fehler bekommst: Keine Panik! Lies die Fehlermeldung und versuche, den Fehler zu finden.

Mit diesen Grundlagen kannst du schon viele kleine Programme schreiben und Daten speichern oder laden!

Python: Dictionaries, JSON und Dateien

Hier findest du eine einfache Erklärung zu wichtigen Python-Konzepten – und Aufgaben, mit denen du direkt üben kannst!

1. Dictionaries

- Ein Dictionary speichert Daten als „Schlüssel-Wert-Paare“.
- Jeder Schlüssel ist eindeutig und zeigt auf einen Wert.

Beispiel:

```
person = {  
    "name": "Anna",  
    "alter": 15,  
    "stadt": "Friedrichshafen"  
}  
print(person["name"]) # Ausgabe: Anna
```

Aufgaben:

- Erstelle ein Dictionary mit deinen Lieblingsfächern in der Schule und deren Noten.
- Füge einen neuen Schlüssel „Lieblingssport“ mit deinem Lieblingssport hinzu.
- Gib den Wert zu einem bestimmten Schlüssel aus (z.B. „Lieblingssport“).

2. JSON und Python

- JSON ist ein Textformat, um Daten zu speichern und zu übertragen.
- Mit dem `json`-Modul kannst du Dictionaries in JSON umwandeln und umgekehrt.

Beispiel: Dictionary zu JSON

```
import json  
  
person = {"name": "Anna", "alter": 15}  
json_text = json.dumps(person)  
print(json_text) # Ausgabe: {"name": "Anna", "alter": 15}
```

Beispiel: JSON zu Dictionary

```
import json

json_text = '{"name": "Anna", "alter": 15}'
person = json.loads(json_text)
print(person["alter"]) # Ausgabe: 15
```

Aufgaben:

- Wandle dein Dictionary aus Aufgabe 1 in einen JSON-Text um und gib ihn aus.
- Schreibe einen JSON-Text mit deinen Hobbys und lade ihn als Dictionary.

3. Datei speichern

- Mit `open()` und dem Modus "w" schreibst du Text in eine Datei.
- Mit `write()` wird der Text gespeichert.

Beispiel:

```
text = "Hallo Welt!"
with open("beispiel.txt", "w") as datei:
    datei.write(text)
```

Aufgaben:

- Speichere einen Text mit deinem Namen und deinem Lieblingsessen in eine Datei.
- Öffne die Datei und überprüfe, ob der Text richtig gespeichert wurde.

4. Datei lesen

- Mit dem Modus "r" liest du den Inhalt einer Datei als Text ein.

Beispiel:

```
with open("beispiel.txt", "r") as datei:
    inhalt = datei.read()
print(inhalt) # Ausgabe: Hallo Welt!
```

Aufgaben:

- Lies den Inhalt der Datei aus Aufgabe 3 und gib ihn auf dem Bildschirm aus.
- Ändere den Text in der Datei und lies ihn erneut ein.

5. JSON in Datei speichern und laden

Speichern:

```
import json

daten = {"name": "Anna", "alter": 15}
with open("daten.json", "w") as datei:
    json.dump(daten, datei)
```

Laden:

```
import json

with open("daten.json", "r") as datei:
    daten = json.load(datei)
print(daten["name"]) # Ausgabe: Anna
```

Aufgaben:

- Speichere dein Dictionary aus Aufgabe 1 als JSON-Datei.
- Lade die JSON-Datei wieder und gib einen Wert daraus aus.

Tipps

- Probiere alle Beispiele und Aufgaben direkt im Python-Interpreter oder in einer IDE aus.
- Ändere Werte und beobachte, was passiert.
- Wenn ein Fehler auftritt: Lies die Fehlermeldung und versuche, den Fehler zu finden.

Viel Spaß beim Ausprobieren!