

# Hausaufgabe 1: Grundlagen der Dateibearbeitung in Python

## i Theoriefragen

Beantworte die folgenden Fragen in eigenen Worten. Schreibe deine Antworten in eine separate Text- oder Markdown-Datei.

1. **Was ist der Unterschied zwischen den Dateioffnungsmodi "r", "w" und "a" in Python? Erkläre kurz, was passiert, wenn du eine Datei in jedem dieser Modi öffnest.**

Dateioffnungsmodi „r“: „r“ steht für „read“ also lesen. In dem Modus kann man nur lesen. Es gibt eine Fehlermeldung, falls die Datei nicht existiert.

Dateioffnungsmodi „w“: „w“ steht für „write“ also schreiben. In dem Modus kann man schreiben; wenn es schon Inhalt (Text/...) in der Datei gibt, wird dieser überschrieben. Wenn die Datei noch nicht existiert, wird diese erstellt.

Dateioffnungsmodi „a“: „a“ steht für „append“ also anhängen. Wenn es schon Inhalt in der Datei gibt, wird der neue Text ans Ende von diesem schon bestehenden Inhalt angehängt. Wenn die Datei noch nicht existiert, wird diese erstellt.

2. **Warum ist die Zeile `with open("meinedatei.txt", "r") as f:` eine gute Methode, um Dateien in Python zu öffnen und zu lesen? Nenne einen wichtigen Vorteil.**

Man muss nicht darauf achten bzw. sich nicht darum kümmern, die Datei nachher wieder zu schließen, es wird durch diese Methode automatisch gemacht. Sobald der Code-Block, der die Datei verwendet, verlassen wird, wird die Datei geschlossen. Dies geschieht unabhängig davon, ob ein Fehler oder ein normaler Programmabschluss auftritt. Man muss sich nicht mehr um das manuelle Schließen der Datei kümmern.

3. **Stell dir vor, du hast eine sehr große Textdatei (z.B. ein Logbuch mit vielen Einträgen). Wäre es besser, die Datei Zeile für Zeile zu lesen oder den gesamten Inhalt auf einmal? Begründe kurz.**

Für große Dateien ist es wahrscheinlich besser, die Datei zeilenweise einzulesen, da es weniger Speicherverbrauch bedeutet. Es befindet sich dann jeweils nur die eine Zeile im Speicher „befindet“ bis sie dann z.B. „geprintet“/ausgegeben wird und danach dann die nächste Zeile eingelesen wird und somit die vorherige Zeile im Speicher „ersetzt“. Es befindet sich also nur die neue Zeile im Speicher usw. .

4. **Das os-Modul in Python ist nützlich für Dateioperationen. Nenne eine Funktion aus dem os-Modul und beschreibe, was sie tut.**

Eine Funktion aus dem os\_Modul ist z.B. die mkdir()-Funktion. Sie erstellt ein neues Verzeichnis im aktuellen Verzeichnis. Wenn das Verzeichnis schon existieren sollte, gibt es eine Fehlermeldung( FileNotFoundError). Sie benötigt als Parameter den Pfad zum Verzeichnis, welches erstellt werden soll. Es kann (optional) auch als Argument der Modus (Berechtigungen) mit angegeben werden.