

Aufgabe 1: Theoretisches Verständnis (5 Punkte)

Beantworten Sie die folgenden Fragen kurz und präzise.

1. Was ist der Hauptunterschied zwischen dem Lesemodus "**r**" und dem Anhängemodus "**a**" beim Öffnen einer Datei in Python? (1 Punkt)
 2. Warum ist die Verwendung des **with**-Blocks beim Arbeiten mit Dateien eine empfohlene Praxis? Nennen Sie zwei Vorteile. (2 Punkte)
 3. Erklären Sie den Unterschied zwischen den Funktionen `json.dump()` und `json.dumps()`. In welchem Anwendungsfall würden Sie die jeweilige Funktion verwenden? (2 Punkte)
-

Aufgabe 2: Praktische Dateiverwaltung (7 Punkte)

Erstellen Sie ein Python-Skript, das die folgenden Schritte ausführt:

1. **Notizen schreiben (3 Punkte):** Das Skript soll den Benutzer auffordern, drei Zeilen Text als Notizen einzugeben. Jede eingegebene Notiz soll in eine neue Zeile in der Datei `meine_notizen.txt` geschrieben werden. Wenn die Datei bereits existiert, soll ihr Inhalt überschrieben werden.
 2. **Notizen lesen und nummerieren (2 Punkte):** Lesen Sie anschließend den Inhalt der gerade erstellten Datei `meine_notizen.txt` zeilenweise aus. Geben Sie jede Zeile mit einer vorangestellten Zeilennummer auf der Konsole aus (z.B. "**1: Erste Notiz**").
 3. **Datei sicher löschen (2 Punkte):** Zum Schluss soll das Skript prüfen, ob die Datei `meine_notizen.txt` existiert, und sie nur dann löschen. Geben Sie eine Bestätigungsmeldung aus, nachdem die Datei gelöscht wurde, oder eine Meldung, falls sie nicht gefunden wurde.
-

Aufgabe 3: Arbeiten mit JSON (4 Punkte)

Ein Python-Dictionary, das ein Produkt repräsentiert, ist wie folgt gegeben:

```
1 produkt_daten = {  
2     "id": 101,  
3     "name": "Laptop Pro X",  
4     "spezifikationen": {  
5         "CPU": "Intel i7",  
6         "RAM": 16,  
7         "Speicher": "512GB SSD"  
8     },  
9     "verfuegbar": True,  
10    "anhaengsel": None  
11 }
```

1. **In JSON-Datei schreiben (2 Punkte):** Konvertieren Sie dieses Python-Dictionary in einen JSON-String und speichern Sie es in einer Datei namens `produkt.json`. Die JSON-Datei sollte "pretty-printed" sein, also mit einer Einrückung von 4 Leerzeichen formatiert werden.
2. **Aus JSON-Datei lesen (2 Punkte):** Lesen Sie die Daten aus der `produkt.json`-Datei zurück in ein neues Python-Objekt. Greifen Sie auf den Wert des Schlüssels "RAM" innerhalb von "spezifikationen" zu und geben Sie ihn auf der Konsole aus.

Aufgabe 4: Reguläre Ausdrücke (4 Punkte)

Gegeben sei der folgende Text-String in Python:

```
1 text = "Kontakt: info@example.com, Support: support@test.org, Admin:  
        admin@domain.net"
```

1. **E-Mail-Adressen extrahieren (4 Punkte):** Schreiben Sie ein Python-Skript, das die Funktion `re.findall()` verwendet, um alle E-Mail-Adressen aus dem gegebenen `text`-String zu extrahieren. Geben Sie die gefundene Liste von E-Mail-Adressen auf der Konsole aus.