

Basic CSV Interview Questions & Answers in Python

Q1. What is a CSV file?

- CSV (Comma-Separated Values) is a simple file format used to store tabular data.
- Each line in the file represents a row, and values are separated by commas (,) or other delimiters (e.g., ; or \t).

Q2. How do you read a CSV file in Python?

- Using csv module:

```
import csv
```

```
with open("data.csv", newline="") as file:  
    reader = csv.reader(file)  
    for row in reader:  
        print(row)
```

- Using Pandas:

```
import pandas as pd
```

```
df = pd.read_csv("data.csv")  
print(df.head())
```

Q3. How do you write data into a CSV file in Python?

```
import csv
```

```
data = [{"Name", "Age"}, ["Alice", 25], ["Bob", 30]]
```

```
with open("output.csv", "w", newline="") as file:  
    writer = csv.writer(file)  
    writer.writerows(data)
```

Q4. Difference between `csv.reader()` and `csv.DictReader()`?

- `csv.reader()` → Reads rows as lists.
- `csv.DictReader()` → Reads rows as dictionaries, using the first row as keys.
- Example:

```
import csv
```

```
with open("data.csv") as file:  
    reader = csv.DictReader(file)  
    for row in reader:  
        print(row["Name"], row["Age"])
```

Q5. Difference between `csv.writer()` and `csv.DictWriter()`?

- `csv.writer()` → Writes data as lists.
- `csv.DictWriter()` → Writes data as dictionaries, with keys matching fieldnames.

Q6. How can you handle different delimiters in CSV?

Use the delimiter parameter in `csv.reader()` or `csv.writer()`.

```
import csv
```

```
with open("data.tsv") as file:  
    reader = csv.reader(file, delimiter="\t")  
    for row in reader:  
        print(row)
```

Q7. How do you skip headers when reading a CSV file?

```
import csv
```

```
with open("data.csv") as file:  
    reader = csv.reader(file)  
    next(reader) # Skip header row  
    for row in reader:  
        print(row)
```