Analytics Vidhya

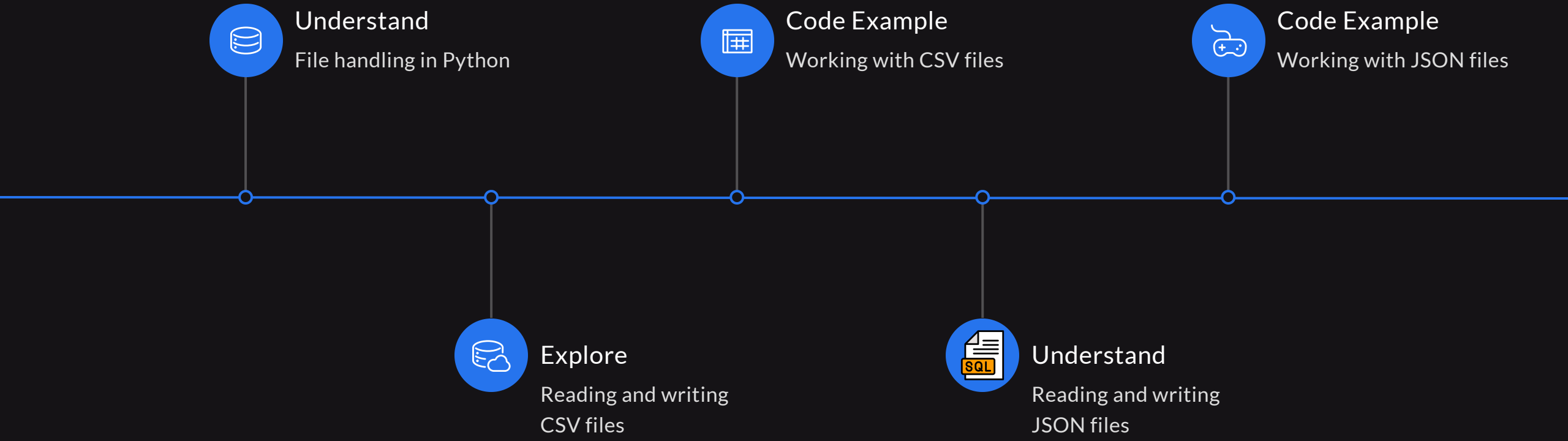
Working with Files and Databases in Python

Handling CSV, and JSON Filetypes

Instructor

Prashant Sahu

Manager (Data Science), Analytics Vidhya


Understand
File handling in Python
Code Example
Working with CSV files
Code Example
Working with JSON files
Explore
Reading and writing
CSV files
SQL
Understand
Reading and writing
JSON files

Agenda

Introduction to File Handling in Python

Why work with files?

Data Storage and Exchange

CSV and JSON are common formats for data storage and transfer

Interoperability

Easily share data between different systems and applications

Data Preprocessing

Prepare data for use in AI models and analytics

Python's Built-in Support

Standard Libraries

csv module for CSV files

json module for JSON files

Third-Party Libraries

pandas for advanced data manipulation

Logging and Monitoring

Store interactions logs for analysis and improvement

Reading and Writing CSV Files with Python

Using the CSV module

Reading CSV files: Use csv.reader() to read data

Writing CSV Files: Use csv.writer() to write data

Analytics Vidhya

Reading and Writing CSV Files with Python

Using pandas for CSV files

Reading CSV files: Use pandas.read_csv() to load into a DataFrame

Writing CSV files: Use DataFrame.to_csv() to write data from a DataFrame to a CSV file.

Handles large datasets efficiently, and provides advanced data manipulation tools

Working with CSV Files

```
## Reading a CSV File Using pandas ##

import pandas as pd

# Read the CSV file into a DataFrame
df = pd.read_csv('data.csv')

# Display the first 5 rows
print(df.head())
```

```
## Writing to a CSV File Using pandas ##

# Perform some data manipulation
df['total'] = df['price'] * df['quantity']

# Write the DataFrame to a new CSV file
df.to_csv('output.csv', index=False)
```

Reading and Writing JSON Files with Python

Using the JSON Module

Reading JSON files: Use json.load()to read data from a file. Use json.loads() to parse JSON strings.

Writing JSON files: Use json.dump() to write data to a file. Use json.dumps() to convert data to a JSON string.

JSON data maps directly to Python dictionaries and lists.

Working with JSON Files

```python
## Reading a JSON File ##
import json

# Open and read the JSON file
with open('data.json', 'r') as file:
    data = json.load(file)

# Access data from the dictionary
print(data['employees'])
```

```python
## Writing to a JSON File ##

# Modify the data
data['employees'].append({'name': 'Jane Doe', 'age': 30})

# Write the updated data back to the JSON file
with open('data.json', 'w') as file:
    json.dump(data, file, indent=4)
```

Summary

Key Takeaways

1. Python provides built-in modules for handling CSV and JSON files.
2. pandas enhances data manipulation capabilities, especially with CSV files.

Analytics Vidhya