

Python

Full stack Skills Bootcamp

Introducing JSON

■ What is JSON?

JSON (JavaScript Object Notation) is a lightweight format for storing and transporting data.

■ Why use JSON in Python?

- It's easy to read and write, and it's widely used for APIs and data exchange.
- It's popular in web applications, and Python json module makes it easy to work with JSON data.
- It supports data types like strings, numbers, objects (dictionaries), arrays(lists), true/false/null.



JSON in Python

■ Key points

Python comes with `json` module, which makes it easy to parse JSON data and convert Python objects to JSON.

■ Common Functions

- `json.loads()` : Converts a JSON string into a Python dictionary.
- `json.dumps()` : Converts a Python dictionary into a JSON string.
- `json.load()` : Reads JSON data from a file and converts it to a Python object.
- `json.dump()` : Writes Python object as JSON data to a file.

A stylized logo for JSON, featuring the word "JSON" in a bold, blue, sans-serif font, enclosed within a pair of large, grey, curly braces.

Parsing JSON Data

■ Key Points

Example showing how to parse json data into a dictionary.

```
python

import json

# JSON string
json_data = '{"name": "Alice", "age": 25, "city": "New York"}'

# Parsing JSON string to Python dictionary
python_dict = json.loads(json_data)
print("Parsed JSON into Python Dictionary:", python_dict)
```

- The `json.loads()` method converts the JSON string into a python dictionary.
- Accessing values is like accessing dictionary elements.

Converting Python to JSON

■ Key Points

Example showing how to convert dictionary into JSON data.

```
python

# Converting Python dictionary to JSON format
json_output = json.dumps(python_dict, indent=4)
print("Python Dictionary Converted to JSON:")
print(json_output)
```

- The `json.dumps()` method converts a Python Dictionary back to JSON format.
- The `indent = 4` argument formats the output nicely for readability.

Reading JSON from a File

■ Key Points

Example showing how to read JSON from a file.

```
python

# Reading JSON data from a file
with open('data.json', 'r') as file:
    file_data = json.load(file)
    print("Data Read from JSON File:", file_data)
```

- The `json.load()` function reads JSON data from a file and converts it to a Python object.
- Use `with open()` to ensure the file is properly opened and closed.

Writing JSON data to a File

■ Key Points

Example showing how to write JSON data into a file.

```
python

# Writing JSON data to a file
with open('output.json', 'w') as file:
    json.dump(python_dict, file, indent=4)
    print("Data Written to output.json")
```

- The `json.dump()` function writes JSON object i.e. the python dictionary to a file.

Handling Exceptions in JSON Operations

■ Key Points

File Handling: Always handle file reading and writing carefully. Files might not exist or could be corrupted.

■ Key Differences

- Use try-except blocks to handle common error, like missing files or improperly formatted JSON.

```
python

try:
    with open('data.json', 'r') as file:
        file_data = json.load(file)
except FileNotFoundError:
    print("Error: File not found.")
except json.JSONDecodeError:
    print("Error: Invalid JSON format.")
```

Will cover exception handling later separately.....

Conclusion

■ Key Points

- JSON is widely used in web and API development for data exchange.
- Python's json module makes it easy to parse and manipulate JSON data.
- JSON can store complex, nested structures.
- Access nested data using key/index chaining.



JAVASCRIPT OBJECT NOTATION