JSON
jsn
kurffylue
balue
jsn
json
json
value
blufn
json
key-value pair
vallue
json

JSON

WHAT IS JSON?

- JSON was first specified in the early 2000s as a lightweight data **interchange format** designed to be easy for humans to read and write, and easy for machines to parse and generate.
- Also, many configuration files and data files use JSON to store settings and information.
- JSON is simple and compact, which makes it easy to read and write.
- JSON supports arrays and nested objects, which are not available in CSV, which makes it more expressive and flexible.

JSON AND DICTONARY

- JSON and dictionaries in Python have a similar syntax, as they both use curly braces to enclose the key-value pairs, and commas to separate them.
- JSON and dictionaries in Python can be easily converted to each other using the built-in `json` module in Python.
- BUT:
  - JSON only supports strings as keys, while dictionaries in Python can use any immutable types.
  - JSON requires double quotes for strings, while in dictionaries we can use either single or double quotes.

EXAMPLE

See example: W10E05.py

JSON FILE

A JSON file containing multiple JSON objects is typically formatted as an array of JSON objects.

Each JSON object is enclosed in curly braces {} and separated by commas within square brackets [].

This structure makes it a "list" of JSON objects.

Example file: `books.json`

LOADING A JSON FILE TO PYTHON

```
import json


# Open the JSON file
file = open("books.json")
# Load the file into a list of dictionaries
books = json.load(file)
file.close()
```

Example: `json_file_read.py`