JSON

`JSON` は（名前は紛らわしいが）
プログラミングで言うところの `オブジェクトではなく` `データ` である

- JSON（ジェイソン、JavaScript Object Notation）
- JSONは「データ記述形式」である（つまり、XML・CSVデータなどのようなデータの一種）
- JSONテキストをプログラムで扱う際に、各言語の「オブジェクト」に変換することがある
- JSONの中の `{}` で囲まれた部分は「JSONオブジェクト」と呼ばれるが、
これはプログラミング言語のオブジェクトとは異なるものである

JSONの形式

- `{}`：オブジェクト(Object)
  - 【キー：値】のペアが入り、【カンマ(,)】で区切る
  - **キーは必ずダブルクォーテーション(`"`)で囲まれた文字列でなければならない**
  - 値には、許可されている様々なデータが入る
- `[]`：配列(Array)
  - 値を【リスト形式】で持つ、【カンマ(,)】で区切る
  - 空の配列 `[]` も有効
- **許可されているデータ型**
  - 文字列 (`"text"`) - 必ずダブルクォート
  - 数値 (`123` , `3.14`) - 指数表記も可 (`1.23e+2`)
  - 真偽値 (`true` , `false`) - 小文字のみ
  - Null値 (`null`) - 小文字のみ
  - オブジェクト (`{}`)
  - 配列 (`[]`)
- **カンマ `,` についての厳格なルール**
  - JSONでは `末尾のカンマが許可されていません`

  例えば、以下は有効なJSONではありません：

```
{
  "name": "John",
  "age": 30,  // 末尾にカンマがあるためエラー（コメントも本来使えない）
}
```

また、以下も無効です：

```
[
  1,
  2,  // 末尾にカンマがあるためエラー（コメントも本来使えない）
]
```

これはJavaScriptやPythonなど多くの言語では末尾カンマが許容されるようになっているため、
よく混同されるポイントです。

- **その他、JSONについての補足：**
  - コメントは許可されていない（`//` や `/* */` は使えない）
  - シングルクォート（`'`）は使えない、必ずダブルクォート（`"`）
  - キー名も必ずダブルクォートで囲む必要がある
  - トップレベルは配列かオブジェクトのいずれかである必要がある（単一の文字列や数値は不可）

これらの厳格なルールがあるため、JSONは単純ですが制限も多い形式となっています。

- **こんなへんなjsonでも、文法的にはOK**

```
[{"hello":1},["world",2],{"Null":null}]
```

その他覚えたいこと

```
基本的なJSON操作：
    json.load() - JSONファイルからPythonオブジェクトに読み込み
    json.loads() - JSON文字列をPythonオブジェクトに変換
    json.dump() - PythonオブジェクトをJSONファイルに書き込み
    json.dumps() - PythonオブジェクトをJSON文字列に変換


JSONとPythonデータ型の対応関係：
    JSON の {} → Python の辞書
    JSON の [] → Python のリスト
    JSON の文字列 → Python の文字列
    JSON の数値 → Python の整数または浮動小数点数
    JSON の true/false → Python の True/False
```

```
    JSON の null → Python の None


基本的なパラメータ：
    indent - 出力時の整形
    ensure_ascii - ASCII文字のみ使用するかどうか（前の質問で触れた部分）
    sort_keys：
        出力時にキーをアルファベット順にソートするかどうかを指定します。
        デフォルトはFalse（ソートしない）です。

jsonモジュールのインポート：
    import json（モジュールというのはおおむねファイル指し、その中のメソッドなどを使えるようにする）
```

メソッド

読み込み

`json.load()`：ファイルオブジェクトからJSONデータを読み込みます。

game.json

```
{
  "ゲーム": "トランプ",
  "例": [
    "ババ抜き",
    "七並べ",
    "スピード"
  ]
}
```

j_load.py

```
import json

with open("game.json", "r", encoding="utf-8") as f:
    game_data = json.load(f)

print(game_data)  # Pythonオブジェクトとして表示する

"""実行結果：
{'ゲーム': 'トランプ', '例': ['ババ抜き', '七並べ', 'スピード']}
"""
```

`json.loads()`：JSON形式の文字列からPythonオブジェクトに変換します。

```
import json

# 大枠の囲みは「'」にすべき
json_string = \
    '[{"商品名": "文房具セット", "内容物": ["鉛筆", "消しゴム", "定規"]}]'
data = json.loads(json_string)

# dataはリスト形式なので、最初の要素にアクセスしてから内容物を取得する
#   リスト内に、要素は辞書が一つあるのみ
#   そして、キーでアクセスしている
print(data[0]["内容物"])  # ['鉛筆', '消しゴム', '定規']
```

書き込み

`json.dump()`：辞書などのPythonオブジェクトをJSONファイルとして保存する

```
import json

data = {
    'name': '花子',
    'age': 25,
    'city': '東京'
}

with open("hanako.json", "a", encoding="utf-8") as f:
    json.dump(data, f, ensure_ascii=False)  # ensure_ascii: 後述

""" 実行結果： hanako.json：' → "(jsonは必ず2重引用符である)
{"name": "花子", "age": 25, "city": "東京"}
"""
```

`json.dumps()`：Pythonオブジェクトを JSON形式の文字列に変換します。

```
import json

data = {'name': 'Bob', 'python_test': [60, 65, 90]}
json_obj = json.dumps(data)

print(json_obj)

"""
# json形式なので、2重引用符である
```

```
{"name": "Bob", "python_test": [60, 65, 90]}
"""
```

オプション引数

`indent` :
出力されるJSONデータの整形（インデント）を行います。
インデント幅を指定することでデータを読みやすく表示できます。

```
import json

data = {'name': '山田', 'hobbies': ['読書', '旅行', '料理']}
formatted_json = json.dumps(data, indent=4, ensure_ascii=False)

print(formatted_json)

""" 実行結果：
{
    "name": "山田",
    "hobbies": [
        "読書",
        "旅行",
        "料理"
    ]
}
"""
```

`ensure_ascci`
日本語などの非ASCII文字をエスケープするかどうかを指定します。
デフォルトはTrue（エスケープする）です。

環境に応じた文字コードに合わせる
つまり、Trueで行った場合は、人間が見るには分かりにくいけれども、安全な方法であり、
Falseの場合は、必ず環境に適応しているという自信がある時だけになります

```
import json

data = {'message': 'こんにちは世界'}
# ASCII文字にエスケープ
json_ascii = json.dumps(data)
print(
    json_ascii)  # {"message": "\u3053\u3093\u306b\u3061\u306f\u4e16\u754c"}
```

```
# 非ASCII文字をそのまま出力
json_utf8 = json.dumps(data, ensure_ascii=False)
print(json_utf8)  # {"message": "こんにちは世界"}
```

`sort_keys`
出力時に`キーを`アルファベット順にソートするかどうかを指定します。
デフォルトはFalse（ソートしない）です。

```
import json

data = {'c': 3, 'a': 1, 'b': 2}
# ソートなし
json_default = json.dumps(data)
print(json_default)  # {"c": 3, "a": 1, "b": 2}

# キーをソート
json_sorted = json.dumps(data, sort_keys=True)
print(json_sorted)  # {"a": 1, "b": 2, "c": 3}
```

END