

کتابخانه `json` در پایتون برای کار با داده‌های فرمت JSON (JavaScript Object Notation) استفاده می‌شود. JSON یکی از فرمت‌های محبوب برای تبادل داده‌ها بین سیستم‌ها است و به‌ویژه در تعامل با API‌ها و ذخیره داده‌ها در فایل‌ها کاربرد دارد.

1. تبدیل داده‌های پایتون به فرمت JSON با استفاده از `json.dumps()`

داده‌های پایتون (مانند دیکشنری‌ها، لیست‌ها، اعداد و غیره) می‌توانند به فرمت JSON تبدیل شوند تا برای ذخیره‌سازی یا ارسال به دیگر سیستم‌ها استفاده شوند.

استفاده از `json.dumps()`

- `json.dumps()` برای تبدیل داده‌های پایتون به فرمت JSON به‌کار می‌رود.
- این متد داده‌ها را به رشته‌ای از JSON تبدیل می‌کند.

```
import json

# داده‌های پایتون
data = {
    "name": "Ali",
    "age": 25,
    "city": "Tehran"
}

# تبدیل داده‌های پایتون به JSON
json_data = json.dumps(data)

print(json_data)
# خروجی: {"name": "Ali", "age": 25, "city": "Tehran"}
```

2. تبدیل داده‌های JSON به اشیاء پایتون با استفاده از `json.loads()`

در زمان دریافت داده‌ها از یک فایل یا API، معمولاً داده‌ها به فرمت JSON هستند. برای تبدیل داده‌های JSON به اشیاء پایتون (مانند دیکشنری یا لیست)، می‌توان از متد `json.loads()` استفاده کرد.

استفاده از `json.loads()`

- `json.loads()` برای تبدیل داده‌های JSON به ساختارهای پایتون به‌کار می‌رود.

```
import json

# داده JSON
json_data = '{"name": "Ali", "age": 25, "city": "Tehran"}'

# پایتون (دیکشنری) تبدیل داده‌های JSON
data = json.loads(json_data)

print(data)
# خروجی: {'name': 'Ali', 'age': 25, 'city': 'Tehran'}
```

3. ذخیره و بارگذاری داده‌ها در/از فایل‌های JSON

پایتون به راحتی می‌تواند داده‌ها را در فرمت JSON ذخیره کرده و مجدداً از آن‌ها بارگذاری کند. برای این کار از متدهای `json.dump()` و `json.load()` استفاده می‌شود.

استفاده از `json.dump()` برای ذخیره داده‌ها به فایل

- داده‌های پایتون را به‌طور مستقیم در یک فایل JSON ذخیره می‌کند. `json.dump()`

```
import json

# داده‌های پایتون
data = {
    "name": "Ali",
    "age": 25,
    "city": "Tehran"
}

# JSON ذخیره داده‌ها به فایل
with open("data.json", "w") as file:
    json.dump(data, file)

print("Data saved to file.")
```

استفاده از `json.load()` برای بارگذاری داده‌ها از فایل

- داده‌های JSON ذخیره شده در فایل را به‌طور مستقیم به اشیاء پایتون تبدیل می‌کند. `json.load()`

```
import json

# JSON بارگذاری داده‌ها از فایل
with open("data.json", "r") as file:
    data = json.load(file)

print(data)
# خروجی: {'name': 'Ali', 'age': 25, 'city': 'Tehran'}
```

4. گزینه‌های پیشرفته برای فرمت‌بندی بهتر فایل JSON

برای فرمت‌بندی بهتر و قابل‌خواندن‌تر داده‌های JSON، می‌توان از گزینه‌های پیشرفته‌ای مانند `indent` و `sort_keys` در متد `json.dumps()` و `json.dump()` استفاده کرد.

استفاده از `indent` برای زیبا کردن فرمت JSON

- به شما امکان می‌دهد تا خروجی JSON را به‌صورت منظم و با فاصله‌های بیشتر چاپ کنید. `indent`

```
import json

# داده‌های پایتون
data = {
    "name": "Ali",
```

```
"age": 25,
"city": "Tehran"
}

# فاصله گذاری JSON تبدیل به فرمت
json_data = json.dumps(data, indent=4)
print(json_data)
# خروجی:
# {
#   "name": "Ali",
#   "age": 25,
#   "city": "Tehran"
# }
```

استفاده از `sort_keys` برای مرتب سازی کلیدهای JSON

- باعث می شود که کلیدهای JSON به صورت مرتب شده (به ترتیب حروف الفبا) ذخیره شوند.

```
import json

# داده های پایتون
data = {
    "name": "Ali",
    "age": 25,
    "city": "Tehran"
}

# با مرتب سازی کلیدها JSON تبدیل به فرمت
json_data = json.dumps(data, sort_keys=True, indent=4)
print(json_data)
# خروجی:
# {
#   "age": 25,
#   "city": "Tehran",
#   "name": "Ali"
# }
```

5. استفاده از JSON برای ارتباط بین سیستم ها

JSON به عنوان فرمت داده ای برای ارتباطات میان سیستم ها، به ویژه در API ها (Application Programming Interfaces) و انتقال داده ها از طریق HTTP استفاده می شود.

ارسال و دریافت داده های JSON از API ها

در تعامل با API ها، داده ها معمولاً به فرمت JSON ارسال و دریافت می شوند. برای این کار، می توان از کتابخانه `requests` برای ارسال درخواست های HTTP به سرور و دریافت پاسخ های JSON استفاده کرد.

```
import requests

# API به GET ارسال یک درخواست
response = requests.get("https://api.example.com/data")

# از پاسخ JSON دریافت داده‌های
data = response.json()

print(data)
```

امنیت داده‌های JSON

هنگام کار با داده‌های JSON، به‌ویژه زمانی که از APIها داده دریافت می‌کنید یا آن‌ها را ارسال می‌کنید، باید به امنیت توجه ویژه‌ای داشته باشید. برای جلوگیری از آسیب‌پذیری‌ها:

- اطمینان حاصل کنید که داده‌ها از منابع معتبر و ایمن دریافت می‌شوند.
- از احراز هویت و رمزگذاری (مانند HTTPS) برای محافظت از داده‌ها استفاده کنید.
- اعتبارسنجی داده‌های JSON ورودی قبل از پردازش آن‌ها برای جلوگیری از حملات تزریق داده (Injection) بسیار مهم است.

نتیجه‌گیری

کتابخانه `json` در پایتون ابزاری قدرتمند برای کار با داده‌های JSON است. این ابزار به‌ویژه در زمان تعامل با APIها، ذخیره و بارگذاری داده‌ها، و تبدیل داده‌های پایتون به JSON و بالعکس مفید است. استفاده از گزینه‌هایی مانند `sort_keys` و `indent` کمک می‌کند تا فرمت داده‌های JSON خوانا و مرتب شود.