



# COMO MANIPULAR JSON COM PYTHON

Nesse ebook você vai aprender a manipular dados em JSON utilizando Python de uma maneira simples e eficiente!

# Crie Ebooks técnicos incríveis em minutos com IA

Conheça a 1ª IA Especializada na criação de Ebooks **com código!**



Chega de formatar código no Google Docs



Deixe que nossa IA faça o trabalho pesado

 Syntax Highlight

 Adicione Banners Promocionais

 Edite em Markdown em Tempo Real

 Infográficos feitos por IA

**TESTE AGORA** 

Salve salve Pythonista!

O JSON (JavaScript Object Notation) é um formato de dados muito popular na web, que permite a troca de informações estruturadas entre diferentes sistemas.

Ele é amplamente utilizado em aplicações RESTful, em bancos de dados NoSQL, em APIs e em muitos outros contextos.

No Python, a biblioteca padrão `json` oferece suporte para a manipulação de JSON de forma fácil e eficiente!

Neste artigo, vamos aprender como manipular JSON com Python. Você vai aprender a escrever dados em um arquivo JSON, ler dados de um arquivo JSON, converter uma string JSON em um objeto Python e também a converter um objeto Python em uma string JSON.

Então vamos nessa, que temos bastante conteúdo!

## Escrevendo dados em um arquivo JSON

A primeira competência que vamos aprender é como escrever dados em um arquivo JSON usando Python.

Para isso, precisaremos da biblioteca `json` e do método `dump()`. Veja o exemplo abaixo:

```
import json

dados = {
    "nome": "João",
    "idade": 30,
    "cidade": "São Paulo"
}

arquivo = open("dados.json", "w")
json.dump(dados, arquivo)
arquivo.close()
```

No exemplo acima, criamos um dicionário chamado `dados`, que contém informações pessoais.

O arquivo `dados.json` será criado e as informações contidas no dicionário serão escritas nesse arquivo.

Note que utilizamos o modo `"w"` para abrir o arquivo em modo de escrita.

## Lendo dados de um arquivo JSON

Agora que já sabemos como escrever dados em um arquivo JSON, vamos aprender como lê-los de um arquivo para dentro do Python.

Para isso, utilizamos o método `load()` da biblioteca `json`. Veja o exemplo a seguir:

```
import json

arquivo = open("dados.json", "r")
dados = json.load(arquivo)
arquivo.close()

print(dados)
```

No exemplo acima, abrimos o arquivo `dados.json` em modo de leitura (`"r"`) e utilizamos o método `json.load()` para carregar os dados desse arquivo em uma variável chamada `dados`.

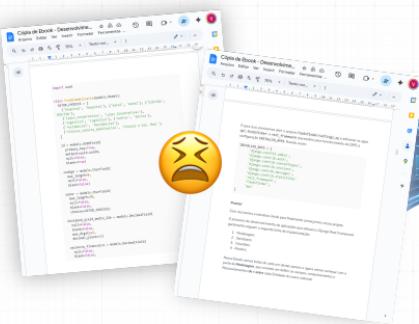
Por fim, exibimos os dados na tela com a função `print()`.



*Estou construindo o **DevBook**, uma plataforma que usa IA para criar e-books técnicos — com código formatado e exportação em PDF. Te convido a conhecê!*

## Crie Ebooks técnicos incríveis em minutos com IA

Conheça a 1ª IA Especializada na criação de Ebooks **com código!**



Chega de formatar código no Google Docs



Deixe que nossa IA faça o trabalho pesado

 Syntax Highlight

 Adicione Banners Promocionais

 Edite em Markdown em Tempo Real

 Infográficos feitos por IA

**TESTE AGORA! PRIMEIRO CAPÍTULO 100% GRÁTIS** 

# Convertendo uma string JSON em um objeto Python

Além de escrever e ler dados de um arquivo JSON, podemos também converter uma string JSON em um objeto Python.

Para isso, utilizamos o método `loads()` da biblioteca `json`. Veja o exemplo abaixo:

```
import json

string_json = '{"nome": "Ana", "idade": 25, "cidade": "Rio de Janeiro"}'
dados = json.loads(string_json)

print(dados)
```

No exemplo acima, temos uma string chamada `string_json`, que contém os mesmos dados do exemplo anterior.

Utilizando o método `json.loads()`, convertemos essa string em um objeto Python. Em seguida, exibimos os dados na tela.

## Convertendo um objeto Python em uma string JSON

Da mesma forma que podemos converter uma string JSON em um objeto Python, também podemos converter um objeto Python em uma string JSON.

Para isso, utilizamos o método `dumps()` da biblioteca `json`. Veja o exemplo a seguir:

```
import json

dados = {
    "nome": "Pedro",
    "idade": 35,
    "cidade": "Belo Horizonte"
}

string_json = json.dumps(dados)

print(string_json)
```

No exemplo acima, temos um dicionário chamado `dados`, que contém as mesmas informações dos exemplos anteriores.

Utilizando o método `json.dumps()`, convertemos esse dicionário em uma string JSON.

Por fim, exibimos a string na tela com a função `print()`.

## Conclusão

Neste artigo, aprendemos como manipular JSON com Python!

Vimos como escrever dados em um arquivo JSON, como ler dados de um arquivo JSON, como converter uma string JSON em um objeto Python e como converter um objeto Python em uma string JSON.

A manipulação de JSON é uma habilidade essencial para qualquer programador Python que trabalha com dados estruturados.

Com a biblioteca `json`, podemos realizar essas tarefas de forma simples e eficiente.

Espero que este artigo tenha te ajudado a entender melhor essa biblioteca e como utilizá-la em seus projetos.

Continue praticando e explore outras funções e métodos oferecidos pela biblioteca `json`.

Com o tempo e a prática, você se tornará um especialista em manipulação de JSON com Python!

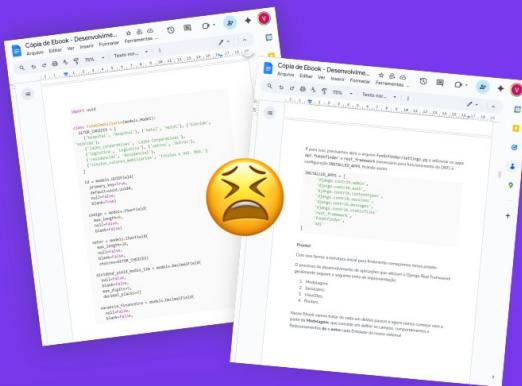
Não se esqueça de conferir!



# DevBook

# Crie Ebooks técnicos em minutos com IA

Conheça a 1ª IA Especializada na criação de Ebooks **com código!**



## Chega de formatar código no Google Docs



## Syntax Highlight



*Adicione Banners Promocionais*



• Infográficos feitos para...

Deixe que nossa IA faça o trabalho pesado



 Edite em Markdown em Tempo Real

**TESTE AGORA**



 PRIMEIRO CAPÍTULO 100% GRÁTIS