

Aula 4: Estruturando Dados com JSON

Eduardo Ribeiro Silva de Oliveira

07 de Outubro de 2024

O Que é JSON

- ▶ Introdução ao JSON e sua utilidade na organização de dados.
- ▶ JSON (JavaScript Object Notation) é um formato leve para troca de dados.
- ▶ É amplamente utilizado por ser de fácil leitura tanto para humanos quanto para máquinas.

Estrutura de Dados em JSON

- ▶ JSON é composto por estruturas de chave-valor.
- ▶ Pode representar dicionários e listas:
 - ▶ ****Dicionários****: Estruturas de chave-valor, similares aos dicionários em Python.
 - ▶ ****Listas****: Conjuntos ordenados de valores, podendo conter outros objetos JSON.

Prática com JSON

- ▶ Como criar e manipular dados em JSON.
- ▶ Exemplo de um objeto JSON:

```
{  
  "nome": "Eduardo",  
  "idade": 22,  
  "habilidades": ["Python",  
    "Scraping",  
    "Data Analysis"]  
}
```

Uso de JSON no Scraping

- ▶ JSON é muito útil para armazenar dados coletados durante o scraping.
- ▶ Estruturar os dados em JSON facilita o armazenamento e posterior análise dos dados extraídos.
- ▶ A padronização do formato torna o processamento mais eficiente.

Exemplos de Armazenamento com JSON

- ▶ Salvando dados coletados em um arquivo JSON:

```
import json
```

```
dados = {  
    "nome": "Eduardo",  
    "idade": 22  
}
```

```
with open("dados.json", "w") as arquivo:  
    json.dump(dados, arquivo)
```

Trabalhando com APIs e JSON

- ▶ Muitas APIs retornam dados no formato JSON.
- ▶ O JSON facilita a extração e manipulação dos dados coletados através de uma API.
- ▶ Exemplo: Utilizando a biblioteca 'requests' para acessar uma API e obter dados em JSON.

Exemplo de Uso de API com JSON

```
import requests

resposta = requests.get("https://api.exemplo.com/dados")
dados = resposta.json()
print(dados["nome"])
```


Conversão para JSON

- ▶ Como converter dados coletados em diferentes formatos para JSON.
- ▶ Garantir a padronização dos dados extraídos facilita seu uso em aplicações posteriores.
- ▶ Ferramentas e bibliotecas, como 'json' em Python, ajudam na conversão de dados.

Benefícios de Usar JSON

- ▶ Formato leve e fácil de ler e escrever.
- ▶ Amplamente suportado por várias linguagens de programação.
- ▶ Facilita a integração entre diferentes sistemas e a análise de dados extraídos.

Prática Final com JSON

- ▶ Criação de um arquivo JSON a partir de dados coletados.
- ▶ Manipulação e leitura dos dados armazenados em JSON.
- ▶ Exemplo:

```
with open("dados.json", "r") as arquivo:  
    conteudo = json.load(arquivo)  
    print(conteudo["idade"])
```