Introdução ao JSON em Python

# O que é JSON?

JSON (JavaScript Object Notation) é um formato leve de troca de dados. É fácil de ler e escrever por humanos e fácil de interpretar e gerar por máquinas. É amplamente utilizado para armazenar e transportar dados entre sistemas, especialmente via APIs na web.

# Estrutura do JSON

JSON trabalha com dois tipos principais de estrutura:  
- Objetos: representados por chaves {} (equivalente a dicionários em Python);  
- Arrays: representados por colchetes [] (equivalente a listas em Python).

# Características do JSON

- Baseado em texto (formato string);  
- Suporta apenas tipos de dados simples: strings, números, booleanos, listas, objetos e null;  
- Leve, portátil e amplamente compatível com linguagens de programação.

# Usando JSON no Python

Para usar JSON em Python, você precisa importar o módulo json:  
import json  
  
Principais funções:  
- json.dump(objeto, arquivo): salva os dados no arquivo JSON;  
- json.load(arquivo): carrega os dados de um arquivo JSON;  
- json.dumps(objeto): converte o objeto Python para uma string JSON;  
- json.loads(string): converte uma string JSON para um objeto Python.

# Exemplo simples

import json  
  
dados = {'nome': 'João', 'idade': 30}  
with open('dados.json', 'w', encoding='utf-8') as f:  
 json.dump(dados, f, indent=4)  
  
# Leitura  
with open('dados.json', 'r', encoding='utf-8') as f:  
 conteudo = json.load(f)  
 print(conteudo['nome'])

# Dicas úteis

- Sempre use 'utf-8' como codificação ao trabalhar com arquivos JSON no Python.  
- Utilize indent=4 no json.dump para deixar o arquivo mais legível.  
- Faça tratamento de exceções ao abrir arquivos (try/except).  
- Para atualizar dados: carregue o arquivo, altere o conteúdo e salve novamente.  
- Mantenha os arquivos JSON com nomes simples e extensões .json.