

JSON

Um dos formatos de transmissão de dados mais comum, utilizado na maioria das APIs é o formato JSON, que significa em inglês JavaScript Object Notation, ou notação de objeto javascript, e é um formato que lembra muito bem as estruturas de objetos JavaScript, apesar de ser completamente válido com arrays por exemplo.

Estrutura de um JSON

Um JSON é estruturado de maneira semelhante a um dicionário, ou a uma lista, com a diferença que sets e tuplas não são permitidos na notação original, nestes casos, o ideal é converter estes tipos de estruturas para listas. As barras para a esquerda servem para não finalizar a declaração da string.

```
# JSON com lista
"[\"foo\", {\"bar\": [\"baz\", null, 1.0, 2]}]"

# JSON com dicionário
"{\"a\": 0, \"b\": 0, \"c\": 0}"
```

Para começarmos a trabalhar com JSON dentro dos nossos scripts, podemos importar um módulo convenientemente intitulado json.

```
import json
```

Criando JSON

Para criar um JSON a partir de uma lista ou dicionário basta usar o comando json.dumps:

```
obj_json = json.dumps({"a": 1, "b": 2, "c": 3})
print(obj_json)
# {"a": 1, "b": 2, "c": 3}

array_json = json.dumps(["nome", None, {"a": "c"}])
print(array_json)
# ["nome", null, {"a": "c"}]
```

Decodificando um JSON:

Para converter um JSON em uma lista ou dicionário, podemos utilizar o método json.loads:

```
obj = json.loads('{"a": 1, "b": 2, "c": 3}')
print(obj)

array = json.loads('["nome", null, {"a": "c"}]')
print(array)
```



Importante! Arquivos JSON são sempre arquivos de **texto**! Apesar de possuírem uma estrutura bastante semelhante a estrutura de dicionários e listas, os JSON são sempre lidos como se fossem **strings**, isso é especialmente importante porque facilita a troca de informações utilizando requisições HTTP