

## Lendo arquivos

Com os dados "crus" em mãos, nosso primeiro passo é abrir o arquivo e ler sua informação para ver sua estrutura.

Comando para entrar no REPL do python:

```
$ python3
```

Código para ser usado no terminal:

```
>>> import json
>>>
>>> with open("video_games.json", mode="r") as file:
>>>     video_games = json.load(file)
>>>
```

Lembre-se que o modo leitura é padrão quando utilizamos o open, portanto, podemos retirá-lo tranquilamente.

Reparem que aqui estamos lendo um arquivo json, por isso estamos usando a função load. Caso estivéssemos lendo o json a partir de um texto, teríamos que usar o loads.

Com os jogos em mãos, podemos fazer algumas análises.

Quantos jogos temos analisados?

```
# >>> import json
# >>>
# >>> with open("video_games.json", mode="r") as file:
# >>>     video_games = json.load(file)
# >>>
# >>>     len(video_games)
```

Quantos gêneros diferentes?

Cada jogo tem um metadado que é uma lista de gêneros. Esta lista é apresentada como uma string separada por vírgula.

```
>>> video_games[0]["Metadata"]["Genres"].split(",")
```

Vamos anotar então todos os gêneros que aparecem na nossa coleção de dados:

```
>>> video_games[0]["Metadata"]["Genres"].split(",")
```

Vamos anotar então todos os gêneros que aparecem na nossa coleção de dados:

**OE** Digite o código abaixo, linha a linha, também no terminal, demonstrando os gêneros apresentados:

```
>>> game_genres = set()
>>> for game in video_games:
...     for genre in game["Metadata"]["Genres"].split(","):
...         game_genres.add(genre)
...
>>>
>>> game_genres
>>> len(game_genres)
```

Observem que, para coletar os gêneros, utilizamos um set, que é uma estrutura de dados de *conjunto*, que garante que vamos coletar apenas gêneros únicos.

Utilizando set, não é possível colocar dados repetidos.

E o número de plataformas que existem no arquivo?

```
>>> consoles = set()
>>> for game in video_games:
...     consoles.add(game["Release"]["Console"])
...
>>>
>>> consoles
>>> len(consoles)
```

E qual a média de notas por gênero de jogo?

```
>>> scores_by_genre = {genre:[] for genre in game_genres}
>>> for game in video_games:
...     for genre in game["Metadata"]["Genres"].split(","):
...         scores_by_genre[genre].append(game["Metrics"]["Review
Score"])
...
>>>
>>> mean_review_score_by_genre = {}
>>> for genre, scores in scores_by_genre.items():
...     mean_review_score_by_genre[genre] = sum(scores) / len(scores)
>>>
>>>
>>> mean_review_score_by_genre
>>>
```

No código que acabamos de executar, primeiro criamos um dicionário (`scores_by_genre`), que a chave é o gênero e o valor uma lista vazia para armazenarmos as notas das reviews. Em seguida, para cada métrica de um jogo adicionamos sua review ao gênero, fazemos isso dentro do laço de repetição `for`. Por fim, criamos um novo comentário (`mean_review_score_by_genre`) em que a chave é o gênero e o valor a média das notas.

Qual console arrecadou mais dinheiro com a venda de jogos?

```
>>> sales_by_console = {console: 0.0 for console in consoles}
>>> for game in video_games:
```

```

...     console = game["Release"]["Console"]
...     sales = game["Metrics"]["Sales"]
...     sales_by_console[console] += sales
...
>>>
>>> sales_by_console
>>>

```

## Escrevendo arquivos

Agora que fizemos algumas análises, que tal gerarmos relatórios com esses dados?

Vamos abrir um arquivo, no modo de escrita, e utilizaremos o escritor do módulo csv para criarmos o relatório.

```

>>> import csv
>>>
>>>
>>> with open("reviews_by_genre.csv", "w") as file:
...     writer = csv.writer(file)
...     # escreve o cabeçalho do arquivo
...     writer.writerow(["genre", "mean"])
...     for genre, mean in mean_review_score_by_genre.items():
...         writer.writerow([genre, mean])
...
>>>

```

Vamos fazer o mesmo passo agora para as vendas por console.

Também poderíamos expor nossas análises em arquivos **JSON**.

Criamos um arquivo no formato **JSON**, e utilizamos o método dump do módulo json para escrever o conteúdo já formatado.

```

>>> import json
>>>
>>>
>>> with open("reviews_by_genre.json", "w") as file:
...     json.dump(mean_review_score_by_genre, file)
...
>>>

```

Vamos fazer a mesma coisa, mas agora vamos fazer com as vendas por console!

```

>>> import json
>>>
>>>
>>> with open("sales_by_console.json", "w") as file:
...     json.dump(sales_by_console, file)
...
>>>

```

É importante ressaltar que a resposta de uma requisição web pode ser considerada um arquivo e poderíamos escrever estes valores no formato **JSON** diretamente sobre ela, sem precisar criar e manipular um arquivo em disco. Basta substituímos `file` pela resposta e logicamente não precisaríamos do gerenciador de contexto (*with*).