Link da resolução no Github: <https://github.com/rominoru/GRA0944_A03>

Universidade Anhembi Morumbi

Curso Big Data e Inteligência Analítica

Disciplina GRA0944 PROGRAMAÇÃO AVANÇADA EM BIG DATA

**\*\*Atividade A3\*\***

**### Resolução**

# importar biblioteca json

import json

# carregar o arquivo .txt e guardá-lo uma variável

with open('textodesafio.txt', 'r') as file:

    json\_dict = json.load(file)

# uma das soluções possíveis é por meio da função json.dump(), conforme está sugerido no enunciado

# passamos como argumentos o dicionário json original, o arquivo de saída e o nível de indentação desejado (argumento indent)

# por meio dessa função, atribuímos uma indentação ao json e gravamos o resultado em um arquivo de texto

with open('jsonhierarquizado.txt', '+w') as file:

    json.dump(json\_dict, file, indent=4)

# a estrutura json resultante é a mesma que pode ser visualizada por meio da função json.dumps()

print(json.dumps(json\_dict, indent=4))

>>>

{

"menu": {

"header": "SVG Viewer",

"items": [

{

"id": "Open"

},

{

"id": "OpenNew",

"label": "Open New"

},

null,

{

"id": "ZoomIn",

"label": "Zoom In"

},

{

"id": "ZoomOut",

"label": "Zoom Out"

},

{

"id": "OriginalView",

"label": "Original View"

},

null,

{

"id": "Quality"

},

{

"id": "Pause"

},

{

"id": "Mute"

},

null,

{

"id": "Find",

"label": "Find..."

},

{

"id": "FindAgain",

"label": "Find Again"

},

{

"id": "Copy"

},

{

"id": "CopyAgain",

"label": "Copy Again"

},

{

"id": "CopySVG",

"label": "Copy SVG"

},

{

"id": "ViewSVG",

"label": "View SVG"

},

{

"id": "ViewSource",

"label": "View Source"

},

{

"id": "SaveAs",

"label": "Save As"

},

null,

{

"id": "Help"

},

{

"id": "About",

"label": "About Adobe CVG Viewer..."

}

]

}

}

# para visualizarmos os itens do dicionário no formato proposto pelo enunciado, sem os símbolos de chaves, colchetes etc,

# podemos utilizar o seguinte código, que percorre por iteração todos os itens do dicionário json e retornar apenas os

# valores que cumprem com os requisitos das condicionantes

for k1, v1 in json\_dict.items():

    print(k1)

    for k2, v2 in v1.items():

        print('  ', k2)

        if type(v2) == str:

            print('    ', v2)

        elif type(v2) == list:

            for item in v2:

                if type(item) == dict:

                    try:

                        print('    ', item['label'])

                    except:

                        print('    ', item['id'])

                elif item == None:

                    print('     ---------')

>>>

menu

header

SVG Viewer

items

Open

Open New

---------

Zoom In

Zoom Out

Original View

---------

Quality

Pause

Mute

---------

Find...

Find Again

Copy

Copy Again

Copy SVG

View SVG

View Source

Save As

---------

Help

About Adobe CVG Viewer...

**### Enunciado**

O desenvolvimento de aplicações para análise de Big Datas não é tão simples, entretanto, algumas tecnologias de desenvolvimento/programação podem reduzir consideravelmente a complexidade dessa atividade. A linguagem de programação Python é uma das tecnologias que oferece todo suporte necessário para que o desenvolvimento de aplicações que realizam coleta e manipulação de dados para Big Data seja o mais fácil e produtivo possível. A linguagem Python é atendida com bibliotecas que permitem trabalhar com matrizes, arquivos de texto, de Excel, cvs, JSON, bancos de dados e outros tipos de arquivos de dados.

Para desenvolver uma aplicação que seja capaz de realizar análise de dados, deve-se, antes, ser capaz de desenvolver aplicações mais simples, aplicações que deverão fazer coleta e manipulação de dados. Esse é o desafio que a empresa na qual você pleiteia uma vaga de emprego destinou a você. Para conquistar sua vaga de programador de aplicações de data science, você deve ser capaz de cumprir um desafio:

Primeiramente crie, em seu computador, um arquivo-texto com o seguinte conteúdo:

```{"menu": {"header": "SVG Viewer", "items": [ {"id": "Open"}, {"id": "OpenNew", "label": "Open New"},null,{"id": "ZoomIn", "label": "Zoom In"},{"id": "ZoomOut", "label": "Zoom Out"},{"id": "OriginalView", "label": "Original View"},null,{"id": "Quality"},{"id": "Pause"},{"id": "Mute"},null,{"id": "Find", "label": "Find..."},{"id": "FindAgain", "label": "Find Again"},{"id": "Copy"},{"id": "CopyAgain", "label": "Copy Again"},{"id": "CopySVG", "label": "Copy SVG"},{"id": "ViewSVG", "label": "View SVG"},{"id": "ViewSource", "label": "View Source"},{"id": "SaveAs", "label": "Save As"},null,{"id": "Help"},{"id": "About", "label": "About Adobe CVG Viewer..."}]}}```

Salve o arquivo como “textodesafio.txt”. O conteúdo é uma estrutura JSON.

Recupere todo esse conteúdo para dentro de uma estrutura JSON que utiliza a linguagem Python.

Após carregar os dados, apresente-os de forma hierárquica, como apresentado abaixo:

menu

&emsp;header

&emsp;items

&emsp;&emsp;Open

&emsp;&emsp;Open New

&emsp;&emsp;Zoom In

e assim por diante. Nada complexo. A intenção é simplesmente avaliar sua capacidade de realizar manipulação de dados utilizando arquivos de texto e estruturas JSON.

A utilização da função ```json.dump()```, com alguns argumentos, será um diferencial