



MÓDULO 5. UNIDADE 3

Exercícios propostos de
JSON



DIRETRIZES GERAIS

- Guardar o documento de soluções com o seguinte formato para entrega:
M2_01_nome_apelido1_apelido2
- Software recomendado: **Anaconda** – Jupyter. Junto a este documento está um padrão de Jupyter com os enunciados
- **Comentar o código**
- Utilizar nomes de variáveis apropriados. Se vamos guardar uma nota, chamamos a essa variável nota, não n ou x.



EXERCÍCIOS DE JSON

EXERCÍCIO 1

Tenha em consideração o seguinte JSON String:

```
{"chefe_projeto": {"Nome": "João", "Idade": 28, "Experiência": ["Gestão", "Finanças", "Bases de dados"], "Residência": "Porto", "HorasProjeto": 3500}, "funcionários": [{"Nome": "Helena", "Idade": 26, "Experiência": ["JavaScript", "Python"], "Residência": "Faro", "HorasProjeto": 500}, {"Nome": "Luis", "Idade": 31, "Experiência": ["Django", "Flask", "Pyramid"], "Residência": "Lisboa", "HorasProjeto": 1100}]}
```

- Se necessário, utilizar um visualizador de **JSON** para o entender.
- O objetivo é criar um dicionário com todos os dados e estruturas internas necessárias para que seja igual ao **JSON String**.
- Criar estruturas mais pequenas até que se chegue a formar o mesmo JSON string que vemos acima.
- Armazenar esta estrutura (o dicionário) numa variável.
- Comprovar o seu tipo e mostrar por ecrã.
- Criar duas variáveis: **horas_funcionários** e **horas_chefe**
- Extrair as horas do chefe e introduzir na sua variável.
- Criar um ciclo que percorra a totalidade de funcionários que tem na estrutura e para cada funcionário somar as suas horas na sua variável correspondente.
- Somar todas as horas e imprimir



EXERCÍCIOS DE JSON

EXERCÍCIO 2

No exercício anterior trabalhou-se com um dicionário que nós mesmos/as criamos.

Converter para um formato **JSON String**, imprimir o seu tipo e os seus dados .

Para que serve este formato?



EXERCÍCIOS DE JSON

EXERCÍCIO 3

Por vezes encontramos **JSON** que se tem de modificar. Para isso, temos de descodificá-los, fazer as modificações pertinentes e voltar a codificá-los para os deixar como JSON de novo.

No seguinte exemplo podemos ver que há alguns erros:

- * Ao Superman falta-lhe como poder "Voar"
- * Em Batman, a idade é 35, não 350
- * Em Batman, sobra-lhe o poder de "Raios nos olhos"
- * Em Wonder Woman falta-lhe o poder "Laço da verdade"

Depois de corrigir, sem criar um novo, voltar a transformar estes dados num **JSON String**



EXERCÍCIOS DE JSON. EXERCÍCIO 3

```
superherois = {  
    "nomedaEquipa": "Super Hero Squad",  
    "cidade": "Metro City",  
    "formado": 2016,  
    "baseSecreta": "Super Tower",  
    "activo": "Sim",  
    "membros": [  
        {  
            "nome": "SuperMan",  
            "idade": 29,  
            "identidadeSecreta": "Clart Kent",  
            "poderes": [  
                "Super força",  
                "Super velocidade",  
                "Raio laser"  
            ]  
        },  
        {  
            "nome": "Batman",  
            "idade": 350,  
            "identidadeSecreta": "Bruce Wayne",  
            "poderes": [  
                "Detective",  
                "Dinheiro",  
                "Raio laser"  
            ]  
        },  
        {  
            "nome": "Wonder Woman",  
            "idade": 900,  
            "identidadeSecreta": "Diana de Temiscira",  
            "poderes": [  
                "Super força",  
                "Super velocidade"  
            ]  
        }  
    ]  
}
```



EXERCÍCIOS DE JSON

EXERCÍCIO 4

Com base no exercício anterior, modificar a estrutura de **super_herois** para conseguir que **membros** tenham dois ramos: "membrosAtivos" e "membrosInativos", onde cada um destes ramos, armazenem os heróis que estão no ativo e os que não.

Neste caso, introduzimos SuperMan e WonderWoman na lista de ativos e Batman na de Inativos

Esta modificação podemos fazê-la como quisermos, ou programando as estruturas de dados e adicionar ao JSON ou fazer a alteração diretamente no JSON.

Ao finalizar, podemos comprovar o nosso JSON num visualizador *online*.

(<http://jsonviewer.stack.hu/>)



EXERCÍCIOS DE JSON

EXERCÍCIO 5

No seguinte código, aceder a um JSON de forma remota. A partir da resposta, fazer o seguinte:

- Mostrar o tipo de dados que se recebeu.
- Expor os dados recebidos.
- Mostrar o número de pessoas que se encontram atualmente na estrutura..
- Fazer um ciclo que percorra todas essas pessoas e mostre os seus detalhes (nome, armazém, etc) no qual se encontra.



EXERCÍCIOS DE JSON

```
from pip._vendor import requests
```

```
# API que nos comunica quantas pessoas se encontram atualmente no espaço
```

```
response = requests.get("http://api.open-notify.org/astros.json")
```

```
data = response.json()
```

```
# Completar o exercício.
```