

# TRABALHANDO COM ARQUIVOS

Manipulando Arquivos formatado como JSON.

Vamos importar o módulo JSON

Import json

Vamos pegar o conteúdo do arquivo cadastro\_pessoas.json e vamos usar para criar um dicionário. Para assim criarmos o arquivo json.

Iterar o Dicionario

```
for c,v in cadastro_pessoa.items():  
    print(c,v)
```

# TRABALHANDO COM ARQUIVOS

Vamos converter esse dicionario em arquivo json  
Para fazer isso vamos usar o modulo json e a funcao json.dump

```
dados = json.dumps(cadastro_pessoa, indent=4)  
print(dados)
```

Para salvar o arquivo convertido e json vamos manipular o arquivo.

```
with open('arquivos/cadastro_pessoas.json', 'w+') as j:  
    json.dump(cadastro_pessoa, j, indent=4)
```

# TRABALHANDO COM ARQUIVOS

Converter um arquivo JSON para dicionário utilizamos o comando `json.loads`

Abre o arquivo

```
with open('arquivos/cadastro_pessoas.json', 'r') as f:  
    le_json = json.load(f)
```

Iterar o dicionário

```
print(le_json)  
for v in le_json.values():  
    print(v)
```

# TRABALHANDO COM ARQUIVOS

Importando um JSON da Internet manipulando e salvando em JSON  
Vamos importar o módulo urllib que busca dados em UR.

```
import urllib.request
```

```
url = "http://api.open-notify.org/astros.json"
```

Captura os dados do JSON

```
pega_json = urllib.request.urlopen(url).read().decode()
```

Converte os valores do JSON em Dicionário

```
dic_json = json.loads(pega_json)
```

# TRABALHANDO COM ARQUIVOS

Iterando os valores de dicionário

```
for c in dic_json.values():  
    print(c)
```

```
print(dic_json)
```

```
for pessoas in dic_json['people']:  
    print(pessoas['name'])
```

## Criando um arquivo json vindo da internet

```
with open('arquivos/nomes.json', 'w+') as f:  
    json.dump(dic_json, f, indent=4)
```

# EXERCÍCIO DE FIXAÇÃO

Importe um JSON da internet através de uma das fontes descritas no material de aula fonte\_url\_json.txt.

Transforme esse JSON em um dicionários de dados.

Itere o dicionário para testar os valores.

Após testar os valores salve os dados do dicionário em um arquivo .json



# TRABALHANDO COM MÓDULOS E PACOTES

Em Python já sabemos que podemos trabalhar com importação de módulos e instalação de pacotes.

Mas dentro de um projeto em Python podemos criar vários scripts secundários de um programa principal ou até mesmo um pacote com vários módulos. Onde um módulo pode ser importado no programa principal.

# TRABALHANDO COM MÓDULOS E PACOTES

Para Entendermos como funciona a importação de módulos vamos criar uma nova pasta para criar o projeto.

Vamos criar o arquivo principal.py ou main.py. Nesse arquivos iremos chamar os módulos e pacotes no programa principal.

Nesse mesmo projeto iremos criar os módulos e pacotes para ser importado ao programa.



# TRABALHANDO COM MÓDULOS E PACOTES

Vamos criar um arquivo de módulo com algumas funções a serem importadas e pelo programa principal.

Vamos criar o arquivo `modulo_prog.py`

Vamos criar uma função soma

```
def soma(x,y):  
    return x + y
```

Vamos criar uma função de calcular o valor máximo de uma lista

```
def calculamaximo(x,y,z):  
    lista=[x,y,z]  
    print(max(lista))
```

# TRABALHANDO COM MÓDULOS E PACOTES

Existem algumas situações aonde queremos que nosso código seja executado apenas sob condições especiais, é o caso dos módulos principais. Só queremos que nossa função `main()` seja executada se o módulo for o principal. Caso ele tenha sido importado, a aplicação só deverá ser executada se `main()` for chamado explicitamente.

Vamos criar isso no `modulo_prog.py` utilizando o método `__main__`.

```
if __name__ == "__main__":  
    print(soma(2,3))  
    print("Executei sozinho")
```

# TRABALHANDO COM MÓDULOS E PACOTES

Vamos importar o módulo para o programa principal. E vamos chamar as funções criadas dentro do modulo.

```
Import modulo_prog
```

Calcular o maior valor.

```
print(modulos.calculamaximo(1,3,18))
```

Chama a Função de Soma

```
print(modulos.soma(3,2))
```

# TRABALHANDO COM MÓDULOS E PACOTES

Criação de Pacotes em Python.

Quando nossos módulos ficarem maiores, não vamos querer ter cinquenta classes e trezentas funções em um único arquivo. Vamos querer separar em diversos módulos. É para isso que pacotes existem. Para criar um pacote basta criar uma pasta dentro da estrutura do seu programa.

Vamos criar uma pasta chamada pacote dentro da nossa pasta. E vamos criar um arquivo chamado prog1.py dentro da pasta pacote.

# TRABALHANDO COM MÓDULOS E PACOTES

Vamos criar algumas funções e códigos dentro o modulo prog1 e vamos importar no programa principal.

```
var1 = "pacote1"
```

```
listacarros = ["Fusca", "Gol", "Chevete"]
```

```
def calculapercentual(x,y):  
    return ( x + ( x * y /100))
```



# TRABALHANDO COM MÓDULOS E PACOTES

Vamos importar dentro do programa principal com o comando from. Iremos importar do pacote criado o modulo desejado e suas funções.

```
from pacote.prog1 import var1, listacarros, calculapercentual
```

Agora vamos chamar as funções importadas no programa principal.

```
print(var1)
```

```
print(calculapercentual(1000,10))
```

```
print(listacarros)
```



# TRABALHANDO COM MÓDULOS E PACOTES

Vamos criar um novo módulo prog2.py no pacote existente com alguma função e vamos importar ao programa principal.

Criar uma função que tras a saída dos números pares.

```
def listapares(x):  
    for p in range(x):  
        if p % 2 == 0:  
            print(p)
```

# TRABALHANDO COM MÓDULOS E PACOTES

Vamos importar o módulo no programa principal.

```
from pacote.prog2 import listapares
```

Chamando a função importada

```
listapares(40)
```

# EXERCÍCIO DE FIXAÇÃO

Crie uma nova estrutura de pastas para salvar os programas.

Crie um arquivo Python que será o programa principal de execução dos módulos e funções.

Crie um módulo onde o mesmo executa 2 ações. Pode ser da sua escolha.

Importe o módulo criado no programa principal e execute as ações no mesmo.

Crie um pacote e crie um outro módulo dentro desse pacote onde a mesma irá executar 1 função de sua escolha. E também irá executar uma função apenas quando executada como principal.

# EXERCÍCIO DE FIXAÇÃO

Importe e execute no programa principal a função criada dentro do módulo do pacote criado.

Teste o programa principal com todos o módulos e funções importadas do etapa anterior do exercicio.