

Estrutura de Dados Tuplas

Tupla é uma Lista imutável. O que diferencia da Lista é que a Lista pode ter elementos adicionados a qualquer momento, enquanto que a Tupla após estrutura definida, não permite a adição ou remoção de elementos.

Exemplos de Tupla:

```
tupla_numeros = (1,2,56,45)  
print(type(tupla_numeros))  
print(tupla_numeros)
```

```
tupla_numeros = 1,2,56,45  
print(type(tupla_numeros))  
print(tupla_numeros)
```

Isto não é Tupla

```
tupla_numeros = (1)
```

Tupla é sempre identificada pelo dado e a virgula após o mesmo.

Estrutura de Dados Tuplas

Isto é Tupla

```
tupla_numeros = (1,)  
print(type(tupla_numeros))  
print(tupla_numeros)
```

```
tupla_numeros = 1,  
print(type(tupla_numeros))  
print(tupla_numeros)
```

Tamanho da Tupla

```
tupla_numeros = 1, 45, 56, 12  
print(len(tupla_numeros))
```

Mostrar a posição dos dados na tupla

```
tupla_numeros = 1, 45, 56, 12  
print(tupla_numeros[2])
```

Estrutura de Dados Tuplas

Podemos executar calculo em Tuplas de valores numéricos:

```
print(sum(tupla_numeros))  
print(max(tupla_numeros))
```

Tuplas são imutáveis não podemos alterar o valor delas, somente conseguimos concatenar

```
tupla_numeros = 1, 45, 56, 12  
tupla_numeros2 = 2,23,78,0  
print(tupla_numeros + tupla_numeros2)
```

Para alterar esse valores e somar o dados dessas tuplas precisamos criar uma 3 tuplas

```
tupla_numeros3 = tupla_numeros + tupla_numeros2  
print(tupla_numeros3)
```

Estrutura de Dados Tuplas

Tuplas não pode ser alteradas porém nada impede de ser reescritas ou substituídas

```
tupla_numeros = tupla_numeros + tupla_numeros2  
print(tupla_numeros)
```

Verificar se um dados está contido na Tupla.

```
print(45 in tupla_numeros)
```

O resultado será True ou False.

Iterando em Tuplas

```
for i in tupla_numeros:  
    print(i)
```

Estrutura de Dados Tuplas

Para copiar os dados de uma tupla basta a nova tupla receber o valor da tupla existente.

```
copia_tupla_numeros = tupla_numeros
```

```
print(copia_tupla_numeros)
```

Exercício Tuplas

Escreva uma tupla e valide se a mesma é um tipo de tupla utilizando o `type`.

Escreva uma tupla e alterne as posições de memória da mesma com o comando `print`.

Escreva uma tupla com valores numéricos e retorne o menor valor da tupla.

Utilize as tuplas existentes e concatene os dados das tuplas em uma 3 tupla.

Verifique se um valor está contido dentro da tupla utilizando condicionais `if/else`.

Estrutura de Dados Dicionários

Dicionários são um coleção desordenada de objetos representados na forma de chave, valor onde a chave é usada para referenciar um determinado valor. As chaves de um dicionário só podem ser de um tipo imutável como inteiros, floats e strings. Tuplas também podem ser aceitas desde que não contenham direta ou indiretamente um tipo mutável como listas.

Dicionários não possuem uma noção de índice e não podem ser fatiados.

Dicionários são mutáveis de forma que a qualquer momento você pode inserir ou remover itens.

Dicionários são listas de associações compostas por:
Uma chave e um valor correspondente

```
dicionario = { 'chave' : 'valor' }
```

Estrutura de Dados Dicionários

Exemplo de Dicionários:

```
frutas_iniciais = { 'A' : 'Abacate', 'M' : 'Melancia' , 'L' : 'Limao', 'B' : 'Banana' }  
print(frutas_iniciais)
```

Trazer uma parte do dicionário, utilizamos o valor de chaves do dicionário

```
frutas_iniciais = { 'A' : 'Abacate', 'M' : 'Melancia' , 'L' : 'Limao', 'B' : 'Banana' }  
print(frutas_iniciais["M"])
```

Iterando em dicionários:

```
for dic in frutas_iniciais:  
    print(dic + " " + frutas_iniciais[dic])
```

```
for dic in frutas_iniciais.items():  
    print(dic)
```


Estrutura de Dados Dicionários

```
for dic in frutas_iniciais.values():  
    print(dic)
```

```
for dic in frutas_iniciais.keys():  
    print(dic)
```

Utilizando Tuplas com dicionários:

```
n_ruas = {  
    (100,200) : 'Rua do Braz',  
    (1,100): 'Rua das Arvores',  
    (50,230): 'Rua dos Atletas',  
}  
print(n_ruas)
```

Estrutura de Dados Dicionários

Adicionando Dados há um dicionário:

```
n_ruas[(200,500)] = 'Rua das Camélias'  
print(n_ruas)
```

Removendo itens do dicionário:

```
del n_ruas[(1,100)]  
print(n_ruas)
```

Atualizando um dicionário através de outro dicionário

```
nova_rua = { (200,500) : 'Rua José da Silva' }
```

```
n_ruas.update(nova_rua)  
print(n_ruas)
```

Estrutura de Dados Dicionários

```
n_ruas.update({(200,1000) : 'Rua Abilio Costa' })  
print(n_ruas)
```

Extrair informações de dicionários com pop

```
print(frutas_iniciais.pop('A'))
```

Limpar dados de um dicionário

```
frutas_iniciais.clear()  
print(frutas_iniciais)
```

Podemos copiar um dicionário dentro de outro dicionário

```
novas_ruas = n_ruas.copy()  
novas_ruas = n_ruas  
print(novas_ruas)
```

Exercícios Dicionários

Crie um dicionário com índices de chaves e valores e faça uma iteração na mesma com o for.

Imprima a saída de alguns valores e chaves do dicionário em tela.

Crie um dicionário contendo tuplas e adicione novos dados e remova dados desse mesmo dicionário.

Crie um segundo dicionário e adicione o mesmo dentro do dicionário criado anteriormente.

Extraia a informação de um dicionário criado com o comando pop.

Crie um novo dicionário efetuando a cópia de um dicionário Existente.

Trabalhando com Módulos

Na linguagem Python, os módulos são nada mais que códigos-fonte que podemos importar em nossos scripts. Estes arquivos podem ter qualquer estrutura do Python e são executados no momento que são importados

Exemplo importar o modulo Random. Para gerar saída aleatórias:

```
import random
```

```
numero = random.randint(0,100)  
print(numero)
```

```
lista = [1, 45, 56, 43,21]  
numero = random.choice(lista)  
print(numero)
```

Trabalhando com Módulos

```
import math  
print(math.pi)
```

```
print(math.sqrt(45))
```

Vimos até agora alguns módulos ou bibliotecas built-in do Python.
Nesse link temos a lista das bibliotecas padrões:

<https://docs.python.org/pt-br/3/library/>

Vamos agora importar uma biblioteca externa do python com o comando pip.
Vamos analisar um arquivo de excel via python e gerar um gráfico.

Para isso vamos instalar via linha de comando as módulos.

Trabalhando com Módulos

```
C:\> Prompt de Comando  
  
C:\Users\joao.cavichiolli>pip install pandas
```

Pip install pandas
Pip install matplotlib
Pip install xlrd
Pip install openpyxl

Vamos copiar para o computador o arquivo teste.xlsx e importar os módulos.

Trabalhando com Módulos

```
import pandas as pd  
import matplotlib.pyplot as plt  
x = pd.read_excel(r"C:\temp\teste.xlsx")
```

```
plt.hist(x["Quantidade"])  
plt.hist(x["Material Escolar"])
```

```
plt.show()
```


Exercícios Módulos

Importar o módulo padrão do Python chamado `datetime` e testar na IDE se o mesmo carregou as opções do módulo. Utilize a opção `datetime.now()` e `print` a data e hora Na saída.

Importar a biblioteca externa chamada `pypdf2` com comando `pip`, e importar a mesma Na IDE e testar as opções da mesma com o atributo `PdfFileReader` e `print` na tela o resultado desse atributo.