

# Coleção de Dados

As coleções permitem armazenar múltiplos itens dentro de uma única unidade, que funciona como um container.

Em Python há basicamente 3 tipos de coleções:

Listas

Tuplas

Dicionários

# Coleção de Dados

## LISTA

**Lista** é uma sequência finita de elementos.

A lista uma estrutura de dados amplamente utilizada no desenvolvimento de software.

Estrutura de dados constituída por uma seqüência ordenada e finita de itens (os quais podem, inclusive, ser outras listas, ditas sublistas), e que pode ser modificada com a inserção, exclusão e reordenamento dos itens.

# Coleção de Dados

## LISTA

**Lista** é uma sequência finita de elementos.

A lista uma estrutura de dados amplamente utilizada no desenvolvimento de software.

Estrutura de dados constituída por uma seqüência ordenada e finita de itens (os quais podem, inclusive, ser outras listas, ditas sublistas), e que pode ser modificada com a inserção, exclusão e reordenamento dos itens.

```
lista_inteiros = [12,34,56,67]
```

```
lista_frutas = ['Morango','Uva','Manga','Tomate','Laranja']
```

```
lista_mista = ['Morango',23,'Uva',45,'Tomate']
```

```
print(lista_inteiros)
```

```
print(lista_frutas)
```

```
print(lista_mista)
```

# Coleção de Dados

Para imprimir posições dos índices das listas. Utilizamos print e a posição da Lista

Exemplo:

```
print(lista_mista[2])
```

Ver o tamanho da lista

```
tamanho_lista = len(lista_mista)  
print(tamanho_lista)
```

Adicionar item a uma lista existente.

```
lista_mista.append("Manga")  
print(lista_mista)
```

# Coleção de Dados

Para imprimir posições dos índices das listas. Utilizamos print e a posição da Lista

Exemplo:

```
print(lista_mista[2])
```

Ver o tamanho da lista

```
tamanho_lista = len(lista_mista)  
print(tamanho_lista)
```

Adicionar item a uma lista existente.

```
lista_mista.append("Manga")  
print(lista_mista)
```

# Coleção de Dados

Excluir itens de uma lista.

```
del lista_mista[4:6]  
print(lista_mista)
```

```
del lista_mista[:]  
print(lista_mista)
```

Percorrer a lista com o FOR

```
for lista_frutas in lista_frutas:  
    print(lista_frutas)
```

# Coleção de Dados

## TUPLA

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.

```
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)
```

```
tupla_carros = "Gol","Fusca","Opala","Marea"  
print(type(tupla_carros))  
print(tupla_carros)
```

# Coleção de Dados

Isto não é Tupla

```
tupla_produto = ("Arroz")  
Print(type(tupla_produto))
```

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

Tamanho da Tupla

```
tupla_numeros1 = 1, 45, 56, 12  
print(len(tupla_numeros))
```

Mostrar a posição dos dados na tupla

```
tupla_numeros1 = 1, 45, 56, 12  
print(tupla_numeros[2])
```



# Coleção de Dados

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

```
tupla_numeros = tupla_numeros + tupla_numeros1  
print(tupla_numeros)
```

Verificar se um dados está contido na Tupla.

```
print(100 in tupla_numeros)
```

O resultado será True ou False.

Iterando em Tuplas

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

# Coleção de Dados

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' }
```

# Coleção de Dados

Exemplo de Dicionários:

```
estados_siglas = { 'SC' : 'Santa Catarina', 'PR' : 'Paraná', 'RS' : 'Rio Grande do  
Sul', 'SP' : 'São Paulo' }  
print(estados_siglas)
```

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

```
estados_siglas = { 'SC' : 'Santa Catarina', 'PR' : 'Paraná', 'RS' : 'Rio Grande do  
Sul', 'SP' : 'São Paulo' }  
print(estados_siglas[SC])
```

Iterando em dicionários:

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

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

# Coleção de Dados

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

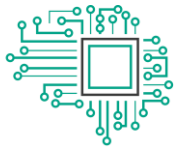
```
for dic in estados_siglas.keys():  
    print(dic)
```

Adicionando Dados há um dicionário:

```
estados_siglas["RJ"] = 'Rio de Janeiro'  
print(estados_siglas)
```

Removendo itens do dicionário:

```
del estados_siglas["SP"]  
print(estados_siglas)
```



**JCAVI**  
TREINAMENTOS EM TI

# Exercício Coleção de Dados

## LISTA

Crie uma lista mista e conte o tamanho dessa lista.

Cria uma lista de dados do tipo string e adicione mais 2 valores nessa lista e printe o resultado dessa lista.

Percorra com laço for a lista mista criada anteriormente.

Remova o ultimo valor de dados da lista mista criada.

## TUPLAS

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

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

## DICIONARIOS

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