

Python para Ciência de Dados - Aula 03

Sumário

1. Matrizes	1
1.1. Lista	1
1.1.1. indexação negativa	2
1.1.2. Faixa de índices	2
1.1.3. Alterar valor do item	2
1.1.4. O item existe na lista?	3
1.1.5. Como adicionar um item a lista	3
1.1.6. Como remover um item da lista?	4
1.1.7. Como juntar duas listas?	4
1.2. Tupla	4
1.3. Dicionário	5
1.3.1. Usando o método chamado get()	5
1.3.2. Alterando o valor de um item específico	5
1.3.3. Determinado se uma chave especificada está presente em um dicionário	5
1.3.4. Removendo itens do dicionário usando pop()	6
1.4. Set	6
1.5. Resumo	7

1. Matrizes

Lembrando um pouco o que já vimos em outros cursos...

Uma matriz é um coleção de dados, todos do mesmo tipo, armazenados em endereços de memória sequencias e vinculados a um único nome de variável.

Em python temos basicamente 4 tipos de matrizes, que você deve entender cada tipo para poder escolher melhor:

- List: coleção ordenada e **modificável**. Permite membros duplicados.
- tupla: coleção que é ordenada e **imutável**. Permite membros duplicados.
- Dicionário: coleção que é desordenada, alterável e **indexada**. Nenhum membro duplicado.
- Set: coleção desordenada e **não indexada**. Nenhum membro duplicado.

1.1. Lista

Como dissemos anteriormente, uma lista é uma coleção ordenada e modificável, que devem ser escritas entre colchetes. Você pode acessar um determinado item da lista usando o número do índice:

```

print("Lista dos principais jogos para PC")
minhaLista = ["League Of Legends",
              "Counter Strike Global Offensive",
              "World of Warcraft"]
#mostrando toda a lista
print(f"Toda a lista: {minhaLista}")

#mostrando um item específico
print(f"Apenas no segundo termo: {minhaLista[1]}")

#modificando a lista
minhaLista = ["League Of Legends",
              "Counter Strike Global Offensive ",
              "Fre fire"]
print(f"Toda a lista: {minhaLista}")

```

Lista dos principais jogos para PC
 Toda a lista: ['League Of Legends', 'Counter Strike Global Offensive', 'World of Warcraft']
 Apenas no segundo termo: Counter Strike Global Offensive
 Toda a lista: ['League Of Legends', 'Counter Strike Global Offensive ', 'Fre fire']

1.1.1. indexação negativa

Refere-se ao item do final da lista para o início.

```

print("Lista dos principais jogos para PC")
minhaLista = ["League Of Legends", "Counter Strike Global Offensive", "World of Warcraft"]
print(f"{minhaLista[-1]}")
print(f"{minhaLista[-2]}")

```

Lista dos principais jogos para PC
 World of Warcraft
 Counter Strike Global Offensive

1.1.2. Faixa de índices

Se você precisar mostrar ou utilizar apenas uma faixa específica de itens de uma lista, pode especificar um intervalo de índices dentro dos colchetes, especificando onde começar e onde encerrar o intervalo, separados por :.

```

print("Lista de personagens de Dragon Ball")
personagens = ["Son Goku", "Bulma", "Yamcha", "Kuririn", "Tenshinhan",
               "Piccolo", "Son Gohan", "Vegeta", "Trunks"]
print(f"Lista inicial: {personagens}")
listaMenor = personagens[2:5]
print(f"Lista menor: {listaMenor}")
print(f"Lista inicial intacta: {personagens}")

```

Lista de personagens de Dragon Ball
 Lista inicial: ['Son Goku', 'Bulma', 'Yamcha', 'Kuririn', 'Tenshinhan', 'Piccolo', 'Son Gohan', 'Vegeta', 'Trunks']
 Lista menor: ['Yamcha', 'Kuririn', 'Tenshinhan']
 Lista inicial intacta: ['Son Goku', 'Bulma', 'Yamcha', 'Kuririn', 'Tenshinhan', 'Piccolo', 'Son Gohan', 'Vegeta', 'Trunks']

1.1.3. Alterar valor do item

Você pode alterar o valor de um item específico.

```
print("Roteiro de Viagem")
roteiro = ["Havaí", "Miami", "Nova York"]
print(f"Próximas viagens: {roteiro}")

#mudando um item do roteiro
roteiro[1] = "Las Vegas"

#Mostrando novamente a lista
print(f"Próximas viagens: {roteiro}")
```

Roteiro de Viagem
Próximas viagens: ['Havaí', 'Miami', 'Nova York']
Próximas viagens: ['Havaí', 'Las Vegas', 'Nova York']

1.1.4. O item existe na lista?

Para determinar se um item específico está na lista pode usar o `in` dentro de um `if`, por exemplo.

```
print("Roteiro de Viagem")
roteiro = ["Havaí", "Miami", "Nova York"]
print(f"Próximas viagens: {roteiro}")

if "Las Vegas" in roteiro:
    print("Não vou passar por Las Vegas")

if "Miami" in roteiro:
    print("Tenho que fazer compras")
```

Roteiro de Viagem
Próximas viagens: ['Havaí', 'Miami', 'Nova York']
Tenho que fazer compras

1.1.5. Como adicionar um item a lista

Você pode adicionar um item ao final da lista utilizando o método `append()`.

```
print("Roteiro de Viagem")
roteiro = ["Havaí", "Miami", "Nova York"]
print(f"Próximas viagens: {roteiro}")
roteiro.append("Las Vegas")
print(f"Próximas viagens: {roteiro}")
```

Roteiro de Viagem
Próximas viagens: ['Havaí', 'Miami', 'Nova York']
Próximas viagens: ['Havaí', 'Miami', 'Nova York', 'Las Vegas']

Ou usando um índice específico com o método `insert()`.

```
print("Roteiro de Viagem")
roteiro = ["Havaí", "Miami", "Nova York"]
print(f"Próximas viagens: {roteiro}")
roteiro.insert(1, "Las Vegas")
print(f"Próximas viagens: {roteiro}")
```

Roteiro de Viagem
Próximas viagens: ['Havaí', 'Miami', 'Nova York']
Próximas viagens: ['Havaí', 'Las Vegas', 'Miami', 'Nova York']

1.1.6. Como remover um item da lista?

Também podemos remover um item da lista usando o método `remove()`.

```
print("Roteiro de Viagem")
roteiro = ["Havaí", "Miami", "Nova York"]
print(f"Prótimas viagens: {roteiro}")
roteiro.remove("Miami")
print(f"Prótimas viagens: {roteiro}")
```

Roteiro de Viagem

Prótimas viagens: ['Havaí', 'Miami', 'Nova York']

Prótimas viagens: ['Havaí', 'Nova York']

1.1.7. Como juntar duas listas?

Podemos concatenar uma ou mais listas usando o sinal `+`.

```
print("Roteiro de Viagem")
comedyCentral = ["Thiago Ventura", "Fabiano Cambota"]
print(f"Humoristas do Comedy Central: {comedyCentral}")
mm = ["Welder Rodrigues", "Ricardo Pipo"]
print(f"Humoristas dos Melhores do Mundo: {mm}")
varios = comedyCentral + mm
print(f"Varios humoristas: {varios}")
```

Roteiro de Viagem

Humoristas do Comedy Central: ['Thiago Ventura', 'Fabiano Cambota']

Humoristas dos Melhores do Mundo: ['Welder Rodrigues', 'Ricardo Pipo']

Varios humoristas: ['Thiago Ventura', 'Fabiano Cambota', 'Welder Rodrigues', 'Ricardo Pipo']

1.2. Tupla

As tuplas são criadas usando parênteses e são imutáveis.

```
print("Desculpas mais comuns")
minhaTupla = ("Minha mãe passou mal", "Vou viajar com urgência", "Estou muito cansado hoje")
print(minhaTupla)
minhaTupla[1] = "não quero me envolver"
print(minhaTupla)
```

Desculpas mais comuns

('Minha mãe passou mal', 'Vou viajar com urgência', 'Estou muito cansado hoje')

TypeError

Traceback (most recent call last)

<ipython-input-5-9c5ed95eb5ca> in <module>()

2 minhaTupla = ("Minha mãe passou mal", "Vou viajar com urgência", "Estou muito cansado hoje")

3 print(minhaTupla)

----> 4 minhaTupla[1] = "não quero me envolver"

5 print(minhaTupla)

TypeError: 'tuple' object does not support item assignment

Coisas semelhantes com a lista:

- acessar os itens da tupla consultando o número do índice;
- indexação negativa significa a partir do final;
- especificar um intervalo de índices;
- percorrer os itens da tupla usando um `for` loop;
- determinar se um item especificado está presente usando a palavra-chave `in`;

- determinar quantos itens uma tupla usando len();
- juntar duas tuplas usando o +.

1.3. Dicionário

É uma coleção não ordenada, modificável e indexada, usando chaves para definí-las.

```
: print("Ford Mustang Mach 1")
ano1969 = {
    "fabricante": "Ford",
    "corpo": "Coupé",
    "portas": 2,
    "cilindros": "V8",
    "potência": "250cv"
}
print(ano1969)
print(ano1969["potência"])
```

Ford Mustang Mach 1
{'fabricante': 'Ford', 'corpo': 'Coupé', 'portas': 2, 'cilindros': 'V8', 'potência': '250cv'}
250cv

1.3.1. Usando o método chamado get()

```
: print("Ford Mustang Mach 1")
ano1969 = {
    "fabricante": "Ford",
    "corpo": "Coupé",
    "portas": 2,
    "cilindros": "V8",
    "potência": "250cv"
}
ano1969.get("fabricante")
print(f"Quem fabrica o Mustang é a {aux}")
```

Ford Mustang Mach 1
Quem fabrica o Mustang é a Ford

1.3.2. Alterando o valor de um item específico

```
: print("Ford Mustang Mach 1")
ano1969 = {
    "fabricante": "Ford",
    "corpo": "Coupé",
    "portas": 2,
    "cilindros": "V8",
    "potência": "250cv"
}
ano1969["potência"] = "280cv"
print(f"A potência do Mustang é "+ano1969["po
```

Ford Mustang Mach 1
A potência do Mustang é 280cv

1.3.3. Determinado se uma chave especificada está presente em um dicionário

```
: print("Ford Mustang Mach 1")
ano1969 = {
    "fabricante": "Ford",
    "corpo": "Coupé",
    "portas": 2,
    "cilindros": "V8",
    "potência": "250cv"
}
if "fabricante" in ano1969:
    print(f"Quem fabrica o Mustang é a {aux}")
```

Ford Mustang Mach 1
Quem fabrica o Mustang é a Ford

1.3.4. Removendo itens do dicionário usando pop()

```
print("Ford Mustang Mach 1")
ano1969 = {
    "fabricante": "Ford",
    "corpo": "Coupé",
    "portas": 2,
    "cilindros": "V8",
    "potência": "250cv"
}
print(ano1969)
ano1969.pop("cilindros")
print(ano1969)
```

Ford Mustang Mach 1
{'fabricante': 'Ford', 'corpo': 'Coupé', 'portas': 2, 'cilindros': 'V8', 'potência': '250cv'}
{'fabricante': 'Ford', 'corpo': 'Coupé', 'portas': 2, 'potência': '250cv'}

Coisas semelhantes com a lista:

- percorrer os itens do dicionário usando um for loop;
- determinar se um item especificado está presente usando a palavra-chave in;
- determinar quantos itens uma dicionario usando len();
- juntar dois dicionários usando o +.

1.4. Set

É uma coleção não ordenada, modificável e não indexada, usando chaves para definí-las. Portanto, como eles não são ordenados, você não pode ter certeza sobre em qual ordem os itens serão exibidos. Ou seja, você não pode acessar itens em um set referindo-se a um índice, uma vez que os itens são desordenados e não têm índice.

```
print("Comidas")
alimentos = {"lasanha", "pizza", "churrasco"}
print(alimentos)
```

Comidas
{'churrasco', 'pizza', 'lasanha'}

```
print("Set")
setCarros = {"Honda", "Toyota", "Toyota"}
print(setCarros)#nenhum repetido
```

Set
{'Toyota', 'Honda'}

Coisas semelhantes com a lista:

- percorrer os itens do dicionário usando um for loop;
- determinar se um item especificado está presente usando a palavra-chave in;
- determinar quantos itens uma dicionario usando len().

Não aceita usar a concatenação com o +.

1.5. Resumo

```
print("Lista: ordenada e modificável, pode ter itens duplicados")
listaCarros = ["Honda","Toyota","Fiat"]
print(listaCarros)

print("\nTupla: ordenada e NÃO modificável, pode ter itens duplicados")
tuplaCarros = ("Honda","Toyota","Fiat")
print(tuplaCarros)

print("\nDicionario: NÃO ordenada e modificável, NÃO pode ter itens duplicados")
dicionaCarros = {"Honda:Civic","Toyota:Hilux","Fiat:Toro"}
print(dicionaCarros)

print("\nSet: NÃO ordenada e NÃO indexada, modificável, NÃO pode ter itens duplicados")
setCarros = {"Honda","Toyota","Fiat"}
print(setCarros)
```

Lista: ordenada e modificável, pode ter itens duplicados
['Honda', 'Toyota', 'Fiat']

Tupla: ordenada e NÃO modificável, pode ter itens duplicados
('Honda', 'Toyota', 'Fiat')

Dicionario: NÃO ordenada e modificável, NÃO pode ter itens duplicados
{'Fiat:Toro', 'Honda:Civic', 'Toyota:Hilux'}

Set: NÃO ordenada e NÃO indexada, modificável, NÃO pode ter itens duplicados
{'Fiat', 'Toyota', 'Honda'}