

FIAP GRADUAÇÃO

TSC

Database Application Development e Data Science
Coleções em Python

PROFA. PATRICIA ANGELINI profpatricia.angelini@fiap.com.br

✓ Coleções

- ✓ Lista
- ✓ Tuplas
- ✓ Conjuntos
- ✓ Dicionários

✓ Exercícios

Coleções

Coleções

Python

MinhaLista = ['café', 'açúcar', 'água', 'café']

Lista com itens do tipo **str**

FilhaTupla = tuple('água', 342, 'meia', False, -2)

Tupla com itens do tipo **str, int, boolean**

- As coleções são usadas quando queremos armazenar múltiplos itens em uma única variável
- Esses itens podem ser do mesmo tipo, mas podem ser de tipos diferentes

Listas

Listas Python

```
MinhaLista = ['café', 'açúcar', 'água', 'café']
```

Lista com itens do tipo `str`

```
print(type(MinhaLista))
```

```
print(len(MinhaLista))
```

```
OutraLista = list( ('tomate', 2, 'banana', 98, True ))
```

```
PequenaLista = MinhaLista[1:3]
```

- Tipo mais comum em Python
- Flexível! Podemos incluir e remover itens
- Ordenado, Indexado, Mutável, Permite Duplicado
- Os itens podem ser do mesmo tipo, mas podem ser de tipos diferentes
- Por ser indexado, cada item pode ser acessado por um número. A lista começa no número 0. O último item pode ser acessado com o índice -1
- Podemos acessar uma parte da lista de um item até outro item usando seu índice

Listas Python

```
MinhaLista = ['café', 'açúcar', 'água', 'café']
```

#Substituindo

```
MinhaLista [0:1] = ['canela', 'adoçante' ]
```

#Adicionando

```
MinhaLista.insert(4, 'chantilly')
```

#Adicionando ao final

```
MinhaLista.append('baunilha')
```

#Removendo de uma determinada posição

```
MinhaLista.pop(0) ou del MinhaLista[0]
```

```
MinhaLista.pop() #Remove o último
```

#Removendo um item específico

```
MinhaLista.remove('chantilly')
```


Listas Python

```
MinhaLista = ['café', 'açucar', 'agua', 'café']
```

#Apagando

```
del MinhaLista
```

#Limpando

```
MinhaLista.clear() #limpa todos itens da lista
```

#Juntando Listas

```
OutraLista = list( ('tomate', 2, 'banana', 98, True ))
```

```
NovaLista=MinhaLista+OutraLista
```

#Extendendo Lista

```
OutraLista.extend(MinhaLista)
```

Listas Python

```
MinhaLista = ['café', 'açúcar', 'agua', 'café']
```

```
#Achando um item
```

```
TemAcucar = index('açúcar')
```

```
if 'café' in MinhaLista:
```

```
    TemCafe = True
```

```
#Colocando em ordem
```

```
MinhaLista.sort()
```

```
#Em reversa
```

```
MinhaLista.sort(reverse = True)
```

```
#Dica especial – o sorted não altera a lista original
```

```
MinhaListaRequinte = ['café', 'açúcar', 'canela',  
    'agua', 'chantilly', 'raspas limão']
```

```
MinhaListaChique = sorted(MinhaListaRequinte)
```

```
print(MinhaListaRequinte)
```

```
print(MinhaListaChique)
```

Tuplas

Tuplas Python

```
MinhaTupla = ('sono', 'comida', 'sol', 'sono')
```

Tupla com itens do tipo `str`

```
print(type(MinhaTupla))
```

```
print(len(MinhaTupla))
```

```
OutraTupla = tuple(('respirar', 100000, 'verdade', True))
```

```
PequenaTupla = MinhaTupla[0:2]
```

- Menos comum em Python
- Inflexível! Não é permitido incluir, substituir ou remover itens
- Ordenado, Indexado, IMUTÁVEL, Permite Duplicado
- Os itens podem ser do mesmo tipo, mas podem ser de tipos diferentes
- Por ser indexado, cada item pode ser acessado por um número. A lista começa no número 0. O último item pode ser acessado com o índice -1
- Podemos acessar uma parte da lista de um item até outro item usando seu índice

Tuplas Python

```
MinhaTupla=('sono', 'comida', 'sol', 'sono')
```

#Apagando

```
del MinhaTupla
```

#Juntando Tuplas

```
OutraTupla=tuple(('dançar', 'pintar', 6, 'meias  
quentinhas', 3))
```

```
NovaTupla = MinhaTupla + OutraTupla
```

#Tupla de 1 elemento

#Atenção a virgula

```
TuplaUnica=('felicidade', )
```

Tuplas Python

```
MinhaTupla=('sono', 'comida', 'sol', 'sono')
#Achando um item
TemSol = MinhaTupla.index('sol')
if 'comida' in MinhaTupla:
    TemComida = True

#Colocando em ordem
#Como a tupla é IMUTAVEL, o único jeito é copiar usando a função
sorted
BemEstar=sorted(MinhaTupla) #Quando faz o SORTED vira LISTA

print('Bem Estar (virou lista!!!)', BemEstar)
Saude = tuple(sorted(MinhaTupla))
print('Saude (com a transformacao)', Saude)
```

Tuplas Python

```
MinhaTupla=('sono', 'comida', 'sol', 'sono')
```

#ATENCAO

#Para qualquer outra operação, tem que transformar em lista, fazer a operação e voltar para Tupla

#Adicionando ao final

```
MinhaLista = list(MinhaTupla)
MinhaLista.append('mensagem relaxante')
MinhaTupla = tuple(MinhaLista)
print(MinhaTupla)
```

#Vale para todas operações: adicionar ao final (append), inserir em determinada operação (insert), apagar item (pop), remover um item(remove), limpar(clear), extender(extend)

Tuplas Python

```
MinhaTupla=('sono', 'comida', 'sol', 'sono')
```

#Dica especial – podemos duplicar tuplas

```
MaisSaude = MinhaTupla * 2
```


Conjuntos

Conjuntos

Python

```
MeuConjunto={'rosa', 'camélia', 'geranio', 'artemisia'}  
#Conjunto com itens do tipo tipo str
```

```
print(type(MeuConjunto))  
print(len(MeuConjunto))
```

```
OutroConjunto = set(('pinheiro', -8, 'capuchinha',  
False))
```

- Bastante comum em Python
- Imutável, não pode alterar os itens mas é permitido incluir, e remover itens
- Desordenado, Não é Indexado
- Os itens podem ser do mesmo tipo, mas podem ser de tipos diferentes
- NÃO Permite Duplicado

Conjuntos Python

```
MeuConjunto={'rosa', 'camélia', 'geranio', 'artemisia'}
```

#Adicionando

```
MeuConjunto.add('lírio')
```

#Adicionando ao final

#Removendo um item específico

```
MeuConjunto.remove('artemisia')
```

```
MeuConjunto.discard('geranio')
```

```
MeuConjunto.discard('margarida') #Quando o item não existe, gera  
um erro
```

```
MeuConjunto.pop() #Remove o último
```

Conjuntos Python

```
MeuConjunto={'rosa', 'camélia', 'geranio', 'artemisia'}
```

```
#Apagando
```

```
del MeuConjunto
```

```
#Limpando
```

```
MeuConjunto.clear()
```

```
#Juntando Conjuntos
```

```
OutroConjunto = set(('pinheiro', -8, 'capuchinha',  
False))
```

```
#Criando um novo conjunto a partir de 2
```

```
NovoConjunto=OutroConjunto.union(MeuConjunto)
```

```
#Adicionando um conjunto no outro!
```

```
OutroConjunto.update(MeuConjunto)
```

Conjuntos

Python

```
MeuConjunto={'rosa', 'camélia', 'geranio', 'artemisia'}
```

#Somente os duplicados

```
SeuConjunto=set('pinheiro', 'camélia', 'rosa')
```

#Criando um novo conjunto a partir dos elementos comuns em 2 conjuntos

```
NossoIgual=SeuConjunto.intersection(MeuConjunto)
```

#Somente os duplicados entre os dois conjuntos

```
SeuConjunto.intersection_update(MeuConjunto)
```

#Somente os direntes

```
SeuConjunto=set(('pinheiro', 'camélia', 'rosa'))
```

#Criando um novo conjunto a partir dos elementos DIFERENTES em 2 conjuntos

```
NossoDiferente=SeuConjunto.symmetric_difference(MeuConjunto)
```

#Somente o diferente entre os dois conjuntos

```
SeuConjunto.symmetric_difference_update(MeuConjunto)
```

Conjuntos Python

```
MeuConjunto={'rosa', 'camélia', 'geranio', 'artemisia'}
```

#Achando um item – os elementos num conjunto não são indexados por isso `index()` não funciona

#Achando um item

```
if 'rosa' in MeuConjunto:  
    TemRosa = True
```

Dicionários

Dicionários Python

```
meuDicionario={'nome':'Patrícia', 'altura':'1,65',  
'peso':'60'}
```

#Dicionários com itens do tipo tipo `str`

```
print(type(meuDicionario))  
print(len(meuDicionario))
```

```
outroDicionario = dict((( 'nome','Patrícia'), ('idade',49),  
( 'vacina_covid',False),(49,['risco', False])))
```

- Bastante comum em Python
- Sempre do tipo `chave:valor` (`key:value`)
- Flexível! Podemos incluir e remover itens
- Mutável
- Ordenado (a partir da versão 3.7 do Python) mas NÃO Permite Duplicado
- Os itens podem ser do mesmo tipo, mas podem ser de tipos diferentes

Dicionários

Python

```
meuDicionario={'nome':'Patrícia','altura':'1,65','peso': '60'}
```

#Recuperando um valor de um item

```
meuValor = meuDicionario['nome']  
meuValor = meuDicionario.get('nome')
```

#Recuperando todos itens

```
meusItens = meuDicionario.items()
```

#Recuperando todas chaves

```
minhasChaves = meuDicionario.keys()
```

#Recuperando todos valores

```
meusValores = meuDicionario.values()
```

Dicionários

Python

```
meuDicionario={'nome':'Patrícia','altura':'1,65','peso': '60'}
```

#Recuperando de maneira iterativa

#Chaves

```
for chave in meuDicionario:
    print(chave)
for chave in meuDicionario.keys():
    print(chave)
```

#Valores

```
for chave in meuDicionario:
    print(meuDicionario[chave])
for valor in meuDicionario.values():
    print(valor)
```

#Itens

```
for chave, valor in meuDicionario.items():
    print(chave, valor)
```

Dicionários Python

```
meuDicionario={'nome':'Patrícia','altura':'1,65','peso':'60'}
```

#Adicionando

```
meuDicionario['tipo sangue'] = 'O'
```

#Substituindo

```
meuDicionario.update({'idade':49}) # se o item não existir vai incluir
```

```
meuDicionario['nome'] = 'Lis' #Ao tentar incluir um item que já existe (tem uma chave igual) ele substitui o valor do item
```

#Removendo um item específico

```
meuDicionario.pop('tipo sangue') #Remove um item de acordo com a chave
```

```
del meuDicionario['tipo sangue']
```

```
meuDicionario.popitem() #Remove o último inserido #python 3.7
```

Dicionários Python

```
meuDicionario={'nome':'Patrícia','altura':'1,65','peso':'60'}
```

```
#Apagando  
del meuDicionario
```

```
#Limpando  
meuDicionario.clear()
```

```
#Copiando dicionários  
novoDicionario= meuDicionario.copy()  
novoDicionario= dict(meuDicionario)
```

Dicionários Python

```
meuDicionario={'nome':'Patrícia','altura':'1,65','peso':  
'60'}
```

#Achando um item

```
if 'peso' in meuDicionario:  
    TemPeso = True
```

EXERCÍCIOS

EXERCÍCIOS

- Acabou a pandemia, chegou o dia e você está ajudando a montar a lista de uma pequena reunião no seu apartamento. Conversando com o seu síndico ele proibiu que houvesse mais de 15 pessoas no seu apartamento. Faça um algoritmo que peça a quantidade de pessoas da sua reunião. E utilizando a função FOR peça o nome dos convidados um a um. Certifique-se que seu melhor amigo João está na sua lista
- Você vai precisar dos nomes em ordem alfabética para poder enviar a portaria. Altere seu algoritmo e ordene a sua lista. Depois apresente o resultado
- Falando com a portaria, foi pedido que adicionasse o número do RG em frente cada um dos nomes dos convidados. Altere seu algoritmo para colocar esses documentos na ordem solicitada. Exemplo ['Pat', 345453, 'Hamilton', 924504]
- Você percebeu que esse primeira reunião vai passar de 20 pessoas e você alugou o salão de festas do seu condomínio. Sendo assim você vai precisar alterar seu algoritmo e agora não vai mais utilizar o comando FOR, vai utilizar o comando WHILE e enquanto a resposta da pergunta “Tem mais convidados” for igual a SIM você cadastrará um novo amigo na sua lista.
- A cada dia mais perto da sua festa, alguns amigos ainda estão meio noiados com a transmissão do virus (e não é para menos com tantas variantes de depois de tantos mortos). Seu amigo João é um deles. Infelizmente ele pediu para ser retirado da lista. Remova o João a Lista

EXERCÍCIOS

- Com esse grande evento você está planejando agora o cardápio. Você deve inicialmente montar uma lista de entradas para que sua irmã possa te ajudar. Monte uma lista com usando o WHILE com todas as delicias gourmets. Não se esqueça de colocar o queijo Roquefort.
- Sua irmã não quer providenciar mais nada e para não chateá-la você transformou essa lista em uma coleção imutável. Apresente a coleção
- Você esqueceu das azeitonas e não tem como pedir para sua irmã providenciar mais nada. A não ser que ela não note que você colocou mais esse item na sua coleção. Para garantir isso coloque a azeitona logo após o queijo Roquefort. Continue o seu algoritmo anterior
- A sua prima resolveu ajudar quando sua irmã já estava dando um PITI. Altere seu algoritmo para fornecer uma coleção com os 3 últimos itens da coleção da sua irmã.
- Sua prima é muito nova ainda e não tem noção da quantidade de cada item! Você deve ajuda-la criando pequenas listas com as quantidades dentro da coleção imutável dela.
- A sua reunião se tornou festa! Você vai ter que providenciar mais comida. Duplique todos os itens já listados.

EXERCÍCIOS

- Chegou o dia e você, sua irmã e sua prima estão preparando o salão de festas. Para isso tem que descer de carrinho no elevador com todas as coisas para arrumar a festa. Prepare uma coleção de coisas que precisa descer. Você está livre para usar o FOR ou o WHILE
- Coloque no carrinho a coleção que você montou no exercício anterior. Não se esqueça que no carrinho os itens não tem ordem para serem colocados
- Sua prima quer saber se alguém já colocou guardanapo no carrinho. Verifique e responda a ela
- Sua irmã saiu pegando itens para serem usados no lavabo e montou um conjunto de coisas. Ela quer adicionar essas coisas no carrinho. O que ela deve fazer?
- Alguém colocou por engano o talher de prata da sua vó. Remova – o rapidamente do carrinho
- Já está perto da hora do almoço e vocês estão ansiosos para uma pausinha. Mas sua prima está cheio de itens de cozinha na mão: jarra, pegador de silicone, copos, pratos. Antes dela colocar no carrinho monte uma coleção do que está duplicado e retire da mão dela antes de colocar no carrinho. Use a função DE DUPLICADOS
- Você se perdeu e tem 3 itens na sua mão: pano de prato, copos, fósforo. Deixe na sua mão somente aquilo que não está no carrinho antes de colocar nele. Use a função DE DIFERENTES

EXERCÍCIOS

- Como faz muito tempo que vocês não se veem (voce e os convidados) você está organizando uma agenda colocando a informação de cada pessoa. Utilize a colecao que imita formulario para pegar itens como nome, endereco, telefone, idade
- Acrescente os dados estado civil na colecao. Você vai montar um novo tinder em breve
- Crie uma colecao que organize todos os itens anteriores numa agenda.
- Exclua aquela amigo que colocou um apelido antigo
- Localize no dicionario todas as idades e ordene do maior para o menor
- Sua Irmã é muito desorganizada, quando ela foi anotar os dados dos convidados ela anotou em listas separadas e voce tera que juntar. Crie tres jeitos diferentes de juntar# [nome, endereco, telephone, idade]# [Andre, Rua do Alfaiates 10, 9887342, 22]
- Para facilitar voce resolveu criar um dicionario cuja a chave é o nome da pessoa e no valor voce ira armazenar outro dicionario com todos os dados da pessoa.

REFERÊNCIAS



- OLIVEIRA, Jayr Figueiredo de; MANZANO, José Augusto N. G. **Algoritmos: Lógica para Desenvolvimento de Programação de Computadores**. 23ª Edição. São Paulo: Érica, 2010.
- MIZRAHI, Victorine Viviane. **Treinamento em Linguagem C**. 2ª Edição. São Paulo: Pearson, 2008.

Copyright © 2023 Profa. Patrícia Angelini

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).