|  |  |
| --- | --- |
|  | **Dicionário** |
| Mutável | sim |
| Definir | dic={“amêndoa”:20,”juropiga”:12} |
| Mostrar um | print(dic[“amêndoa”]) |
| Adicionar | dic[“álcool”]=0,7 |
| Remover | del dic[“álcool”]  ou  valor=dic.pop(“juropiga”,none) |
| Editar | dic[“amêndoa”]=”maracujá” |
| Listar | For chave,valor in dic.items():  Print(chave,valor) |
| Testar se existe | If “maracujá” in dic:  print(“é muito bom”) |
| Igualdade | Operador == e não interessa a ordem dos elementos |
| Funções | Items()  Keys()  Values()  Len() |
| Notas | Chaves tem de ser imutáveis (int,str) |

|  |  |
| --- | --- |
|  | **Conjunto (Set)** |
| Mutável | Sim |
| Definir | Meu\_set={1,2,3,4} |
| Mostrar um | For pos, valor in enumarate(meu\_set,0)  If ops=3:  Print(valor)  Break |
| Adicionar | Meu\_set.add(5) |
| Remover | Meu.set.remove(3) |
| Editar | Remover o elemnto e adicionar o novo, atualizado |
| Listar | for pos, valor in enumerate(meu\_set,0):  print(f"Elemento da posição {pos}: {valor}") |
| Testar se existe | if 2 in meu\_set:  print("O 2 existe no set")  else:  print("O 2 não existe no set") |
| Igualdade | Operador == e a ordem dos elementos ñ interessa |
| Funções | Meu\_set.union(outro\_set)  Interseçao=meu\_set.intersection(outro\_Set)  Diferença=meu\_set.difference(outro\_Set)  Diferenca simétrica=meu\_set.symmetric\_difference(outro\_Set) |
| Notas | Não permite duplicados, coleçao não ordenada e por isso não se pode indexar e só pode ter elemntos imutaveis |

|  |  |
| --- | --- |
|  | **Tuplo** |
| Mutável | Não |
| Definir | Meu\_tuplo=(1,2,3,4,5) |
| Mostrar um | print(meu\_tuplo[0]) |
| Adicionar | N/A |
| Remover | N/A |
| Editar | N/A |
| Listar | for x in meu\_tuplo:  Print(x)  For pos in range(len(meu\_tuplo)):  Print(meu\_tuplo[pos]) |
| Testar se existe | If 2 in meu\_tuplo:  Print(“Otuplo tem o elemento 2”)  Else:  Print(“O tuplo não tem o elemento 2”) |
| Igualdade | Operador == e ordem dos elementos interessa |
| Funções | Meu\_tuplo.count(2) #numero de ocorrências do elmento  Meu\_tuplo.index(3) #posiçao da primeira ocorrência do elemnto  Menor=min(meu\_tuplo) # o maoir elemnto  Maior=max(meu\_tuplo) #soma dos elementos  Soma=sum(meu\_tuplo) #soma dos elementos  T1=(1,2,3)  T2=(4,5,6)  Todos=t1+t2 #todos=(1,2,3,4,5,6)  Sorted() #para ordenar |
| Notas | Podem vários tipos de dados, podem ter valores repetidos  Depois de criada não pode ser alterado |

|  |  |
| --- | --- |
|  | **Lista** |
| Mutável | Sim |
| Definir | minha\_lista**=**[1,2,3] |
| Mostrar um | Print(minha\_lista[3]) |
| Adicionar | append() |
| Remover | remove()  pop() |
| Editar | Minha\_lista[0]=22 |
| Listar | For x in minha\_lista:  Print(x)  For pos in range(len(minha\_lista)):  Print(minha\_lista[pos]) |
| Testar se existe | If 33 in minha\_lista:  Print(“33 existe na lista”)  Else:  Print(“33 não existe na lista”) |
| Igualdade | Operador == e a ordem dos elementos interessa |
| Funções | Insert()  sorted()  sort()  extend()  del  pop() |
| Notas | Permite fatiamento  Perite compreensão de lista |