

Dicionários

quarta-feira, 17 de agosto de 2022 19:24

Existem 2 métodos para criação de um dicionário:

Primeiro método:

```
dicionario = dict(chave_ = "valor", chave2 = 12, chave3 = 50, chave4 = "valor4")
```

Segundo método:

```
dicionario = {"chave": "valor", "Chave2": "valor2", "Chave3": "valor3"}  
  
dicionario["chavenova"] = "valor novo"
```

ou então podemos usar a função update:

```
dicionario = {1: "Valor", 2: "Valor2", 3: "Valor3", (1, 2, 3, 4): "valor"}  
  
dicionario.update({"Nova_chave": "valornovo"})  
  
print(dicionario)
```

A função update também serve para unir dois dicionários.

```
d1 = {"Geronimo": "nome", "Eunice": "nome"}  
d2 = {"IP": "1033329", "IP2": "1524586"}  
d1.update(d2)  
print(d1)
```

quando temos chaves com o mesmo nome e chamamos o print o resultado da chave será apenas o ultimo valor dela.

```
dicionario = {"Chave": "valor", "Chave2": "Valor2", "Chave": "Valor_novo"}  
  
print(dicionario)
```

```
{'Chave': 'Valor_novo', 'Chave2': 'Valor2'}
```

```
Process finished with exit code 0
```

podemos usar também números para criar chaves:

```
dicionario = {1:"Valor", 2:"Valor2", 3:"Valor3"}

print(dicionario)
```

Uma tupla também pode ser usada como chave dentro de um dicionário:

```
dicionario = {1:"Valor", 2:"Valor2", 3:"Valor3", (1,2,3,4):"valor"}

print(dicionario)
```

porém se precisar acessar essa chave em formato de tupla devemos usar a seguinte sintaxe:

```
print(dicionario[(1,2,3,4)])
```

se por acaso a chave não existir dentro do dicionário, e usar o print para chamar essa chave, o python encontra uma exceção e para o programa:

```
dicionario = {1:"Valor", 2:"Valor2", 3:"Valor3", (1,2,3,4):"valor"}

print(dicionario["Chaveinexistente"])
```

```
File "C:\Users\geron\PycharmProjects\Cursoudemy\Cursoudemy\dicionários.py", line 4, in <module>
    print(dicionario["Chaveinexistente"])
KeyError: 'Chaveinexistente'

Process finished with exit code 1
```

porém existe outro método para saber se existe tal chave no dicionário usando o `get()`:

```
print(dicionario.get(1))
```

se por acaso não existir o valor dentro do dicionário o valor retornado é `none`, então podemos usar um método para ver se o valor não for `none` na lista:

```
if dicionario.get("nomedachave") is not None:
    print(dicionario.get("nomedachave"))
```

então se essa chave existir dentro do dicionário será imprimido na tela o valor da chave caso contrário não imprime nada.

Quando queremos mudar o valor dentro do dicionário basta usar nome do dicionário a chave e o novo valor.

```
dicionario = {1:"Valor", 2:"Valor2", 3:"Valor3", (1,2,3,4):"valor"}

dicionario[1]="geronimo"

print(dicionario)
```

para deletar algo dentro do dicionário usamos o metodo del

```
dicionario = {1:"Valor", 2:"Valor2", 3:"Valor3", (1,2,3,4):"valor"}

dicionario.update({"Nova_chave": "valornovo"})

del dicionario[1]

print(dicionario)
```

dicionário agora:

```
{2: 'Valor2', 3: 'Valor3', (1, 2, 3, 4): 'valor', 'Nova_chave': 'valornovo'}

Process finished with exit code 0
```

Se quiser verificar se existe um valor de uma chave ou valor dentro do dicionário podemos usar 2 métodos:

```
print("chave1" in dicionario)
print("chave1" in dicionario.keys())
print("valor" in dicionario.values())
```

Resultado da expressão é True pois existe no dicionário a chave e o valor

Iterando sobre o dicionário:

```
dicionario = dict(chave1 = "valor", chave2 = "valor2")

for c in dicionario.items():
    print(c[0], c[1])
```

dessa forma obtemos o resultado da chave e depois o do valor:

ou podemos usar também outro método:

```
dicionario = dict(chave1 = "valor", chave2 = "valor2")

for c,v in dicionario.items():
    print(c,v)
```

Criando chaves como valor e iterando sobre ela:

```

dicionario = {"cliente1":{
    "nome": "geronimo",
    "Endereçoip": "10332996425",
},
    "cliente1":{
    "nome": "Eunice",
    "Endereçoip": "10332996425",
},
}

for cliente_chave, cliente_valor in dicionario.items():
    print(cliente_chave)
    for dados_cliente, dados_valor in cliente_valor.items():
        print(f"\t{dados_cliente}: {dados_valor}")

```

Quando vinculamos uma variável a um dicionário estamos ligando um dicionário ao outro e se fizer alteração em 1 fazemos no outro, porém existe um método usado `copy()` para os dicionários que é praticamente uma cópia do outro porém se alterado não interfere no primeiro

Exemplo:

```

dicionario = {"chave1": "valor", "2": "valor"}

v = dicionario.copy()

v["chave1"] = 50
print(v)

```

porém se existir uma lista dentro do dicionário para mudar o valor assim que mudar em um muda no outro tbm então existe o módulo `copy`

```

import copy
dicionario = {"chave1": "valor", "2": "valor", "chave3": [1, 2, 3, 4]}

v = copy.deepcopy(dicionario)

v["chave3"][0] = 50
print(v)

```

se ao invés de listas forem tuplas so serão alterados se você buscar a chave e determinar outra tupla

Exemplo:

```

import copy
dicionario = {"chave1": "valor", "2": "valor", "chave3": (1, 2, 3, 4)}

v = copy.deepcopy(dicionario)

v["chave3"] = (3, 2, 5, 4, 6, 7)

print(v)

```

podemos também converter listas ou tuplas para dicionários contanto que tenham valores em pares.

Exemplo:

```
lista = [
    [1,2],
    [3,4],
    [4,5]
]

dic = dict(lista)
print(dic)
```

O mesmo serve para tuplas

podemos também excluir algum item dentro do dicionario usando o pop()

```
lista = [
    [1,2],
    [3,4],
    [4,5]
]

dic = dict(lista)
dic.pop(4)
print(dic)
```

Procura o valor da chave que é 4 poderia ser qualquer outro tipo "chave1"

Exercicio de dicionários

Sistemas de perguntas:

```
perguntas = {
    'Pergunta 1': {
        'pergunta': 'Quanto é 2+2? ',
        'respostas': {'a': '1', 'b': '4', 'c': '5',},
        'resposta_certa': 'b',
    },
}

respostas_certas = 0
for pk, pv in perguntas.items():
    print(f'{pk}: {pv["pergunta"]}')
    print('Respostas: ')
    for rk, rv in pv['respostas'].items():
        print(f'[{rk}]: {rv}')
    resposta_usuario = input('Sua resposta: ')

    if resposta_usuario == pv['resposta_certa']:
        print('EHHHHH!!!! Você acertou!!!!')
        respostas_certas += 1
    else:
        print('IXIIIII!!! Você ERROU!!!!')

qtd_perguntas = len(perguntas)
porcentagem_acerto = respostas_certas / qtd_perguntas * 100

print(f'Você acertou {respostas_certas} respostas.')
print(f'Sua porcentagem de acerto foi de {porcentagem_acerto}%.')
```