

Dicionários

Dicionários em python são na verdade um tipo de coleção um pouco diferente, no sentido de que apesar de serem não ordenados, eles também não são indexados, e os dados estão associados a chaves, que podem ser do tipo int, str, ou até mesmo um tupla (desde que essa contenha apenas str ou int), **Dicionários não podem ter chaves duplicadas**. Dicionários são muito semelhantes ao que seriam os objetos em JavaScript ou um Map em C#.

Parece um pouco complicado, mas basta imaginar que os dicionários são listas nos quais os índices podem ser de qualquer tipo, imagina uma lista que em vez de [1], podemos usar ['telefone'] por exemplo, lembre-se sempre que dicionários são coleções de chave e valor. Vamos ver um exemplo de um dicionário em python:

```
phones = {'João': 45617890, 'Maria': 45788832}
```

Para criar um dicionário, basta abrir chave, digitar as combinações de chave e valor desejadas, e fechar essa chave. Também é possível utilizar o construtor dict(), mas não vamos entrar em detalhe nesta aula.

Por padrão, os dicionários trabalham diretamente com as chaves quando utilizamos if, veja um exemplo:

```
phones = {'João': 45617890, 'Maria': 45788832}
if 'João' in phone: # Encontra a chave
    print("True")
if 45617890 in phone: # Não vai encontrar o valor
    print("False")
```

Para acessar o valor de uma chave em um dicionário usamos colchetes informando dentro deles a chave que desejamos buscar:

```
phones = {'João': 45617890, 'Maria': 45788832}
phones['João'] # 45617890
phones['Maria'] # 45788832
```

Iterando sobre dicionários

Podemos utilizar for para listar os valores de um dicionário, com algumas diferenças, vamos ver alguns exemplos.

O primeiro exemplo demonstra como acessar tanto as chaves quanto os valores de um dicionário utilizando um simples for:

```
phones = {'João': 45617890, 'Maria': 45788832}
for chave, valor in phones.items():
    print(chave, valor)

# "João" 45617890
# "Maria" 45788832
```

Adicionando itens

Como os valores das chaves de um dicionário são mutáveis, podemos simplesmente dizer que determinada chave tem determinado valor, exemplo:

```
phones = {'João': 45617890, 'Maria': 45788832}
phones['Ana'] = 45918203

print(phones) # {'João': 45617890, 'Maria': 45788832, 'Ana': 45918203}
```

Removendo itens

Para remover um par de chave-valor podemos utilizar o método `pop`, informando a chave que deseja ser removida:

```
phones = {'João': 45617890, 'Maria': 45788832}
phones.pop('João')

print(phones) # {'Maria': 45788832}
```

Unindo dois dicionários

Podemos juntar dois dicionários diferentes utilizando o método `update()`, lembrando que caso existam chaves duplicadas na união, será mantida apenas a do segundo dicionário.

```
phones = {'João': 45617890, 'Maria': 45788832}
phones_2 = {'Ana': 42212020, 'Maria': 98281234}

phones.update(phones_2)
print(phones) # {'João': 45617890, 'Maria': 98281234, 'Ana': 42212020}
```