



Dicionário em Python

< thefutureisblue.me />





Agenda

O que vamos aprender?

O que é um dicionário.

Mapeamento dos elementos de um dicionário.

Acesso aos itens de um dicionário.

Adicionando valores ao dicionário.



O que são **Dicionários**?

Em Python um **dicionário** é um tipo de estrutura de dados do **tipo coleção**, ou seja, um objeto que contém mais que um valor.

Lista de Compras.

Agenda Telefônica.

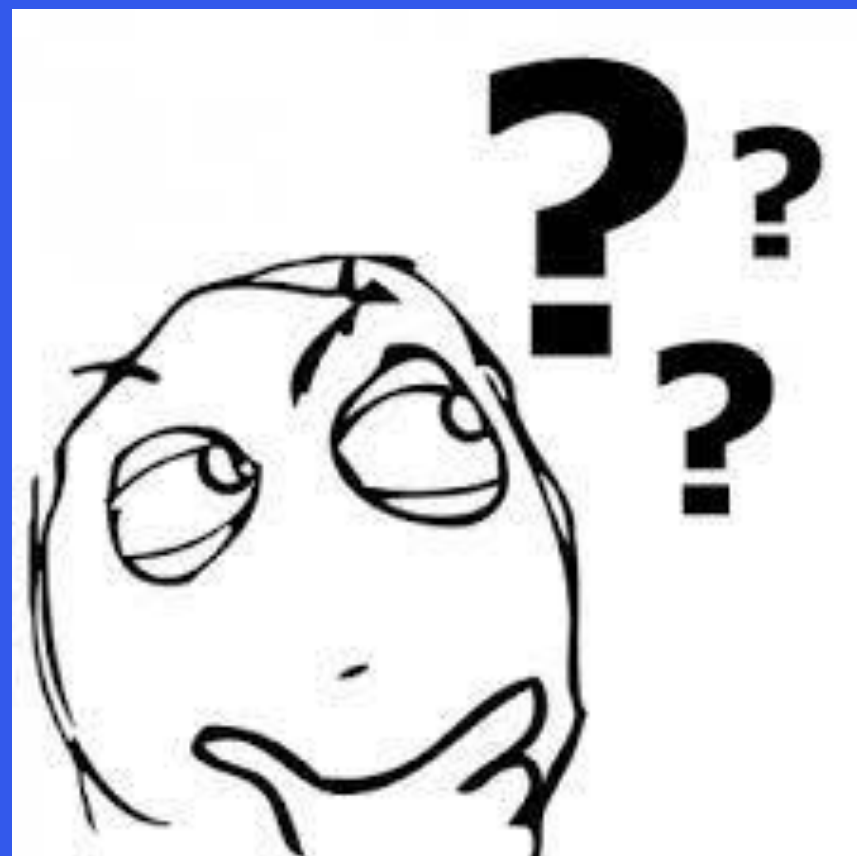
Lista de Chamada dos Alunos.

Trabalhando com **Dicionários**

Vamos pensar em uma agenda de contatos telefônicos, onde necessitamos salvar o número dos nossos contatos.

A princípio podemos pensar em usar uma lista:

👉 telefones = ['1234-5678', '9999-8888', '8765-4321', '7788-8877']



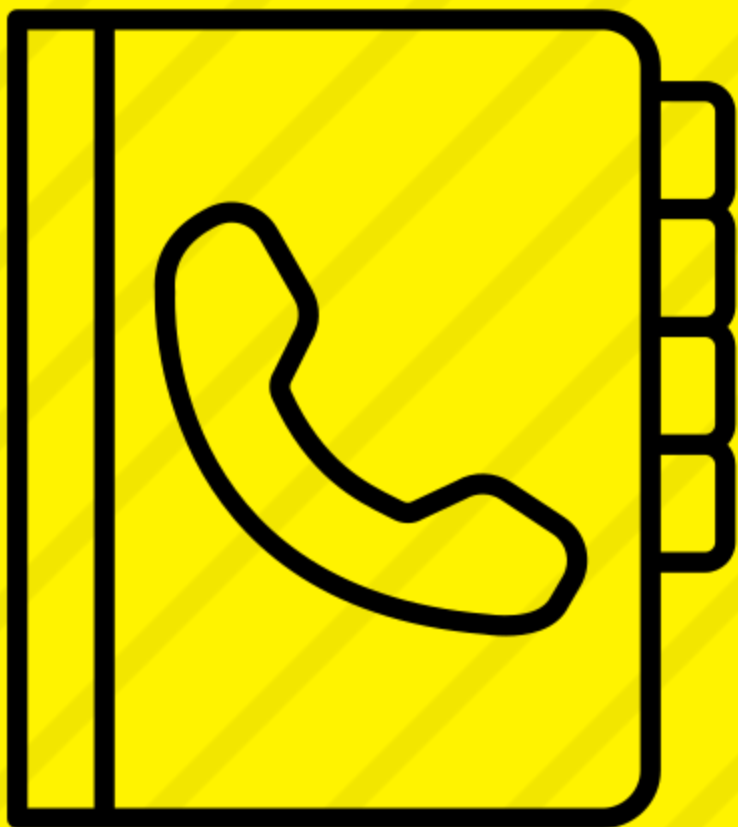
De quem é o telefone:
9999-8888 ?

Trabalhando com Dicionários

Para não precisarmos de uma variável para cada contato, podemos colocá-los direto em uma lista de contatos:



```
contatos_lista = [('Gustavo', '1234-5678'), ('Pedro', '9999-9999'),  
                  ('Ana', '8765-4321'), ('Marina', '7788-8877')]
```



Botando para rodar!

Crie sua lista de contatos com 5 contatos.



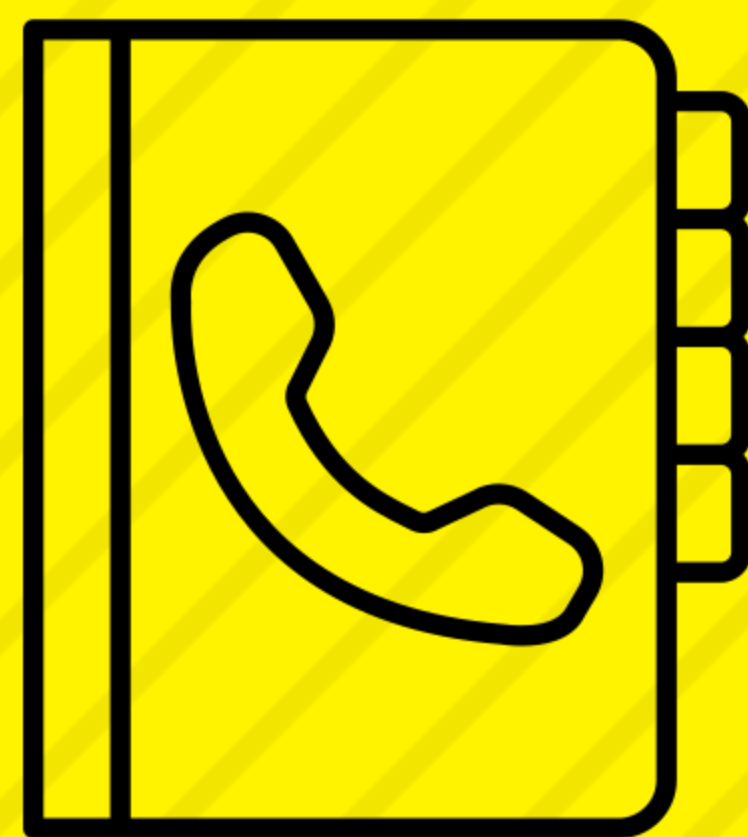
Trabalhando com **Dicionários**

```
contatos_lista = [('Gustavo', '1234-5678'), ('Pedro', '9999-9999'),  
                  ('Ana', '8765-4321'), ('Marina', '7788-8877')]
```

Se quisermos acessar o número de telefone da Marina:



```
print(contatos_lista[3][1])
```



Botando para rodar!

Acesse o número de telefone
de todos os contatos
cadastrados.

Mapeando contatos em um Dicionário

Até agora temos uma lista de contatos em que, ao menos, cada contato tem o seu nome e telefone conectados. Até esse momento, só conseguimos acessar um contato individualmente pela sua posição na lista e não pelo próprio nome.

O que precisamos fazer agora é mapear os nossos contatos, dessa forma, será possível acessá-los pelo nome.

Mapeando contatos em um Dicionário

No Python, usamos chaves ({}) para construir nossos dicionários.

```
➡ contatos = {'Gustavo': '1234-5678'}
```

```
➡ print (type(contatos))
```


Como resultado temos:

```
➡ <class 'dict'>
```

Mapeando contatos em um Dicionário

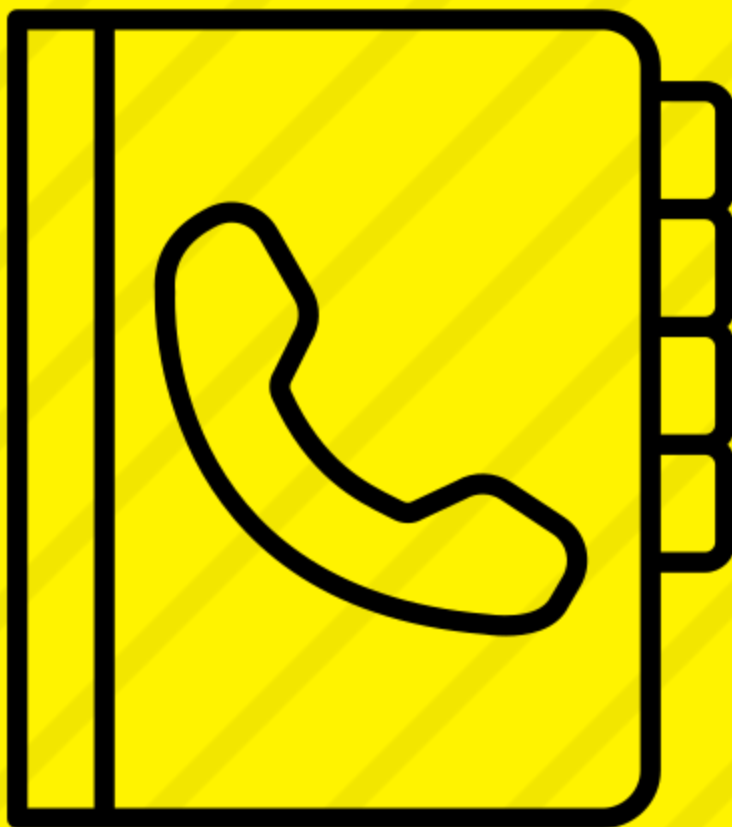
Será que agora vamos ter que redigitar toda a nossa lista de contatos?

Podemos utilizar a função construtora `dict()` e passar como parâmetro `contatos_lista`.

```
 contatos_lista = [('Gustavo', '1234-5678'), ('Pedro', '9999-9999'),  
                    ('Ana', '8765-4321'), ('Marina', '7788-8877')]
```

```
 contatos = dict(contatos_lista)
```

```
 print(contatos)
```



Botando para rodar!

Vamos implementar a função `dict()` em nossa lista de contatos.

Acessando contatos em um Dicionário

Podemos acessar os valores de um dicionário de forma similar a como acessamos os valores de uma lista, por exemplo, com a diferença de que usamos as chaves que definimos no lugar dos índices numéricos:

```
 print(contatos['Gustavo'])
```

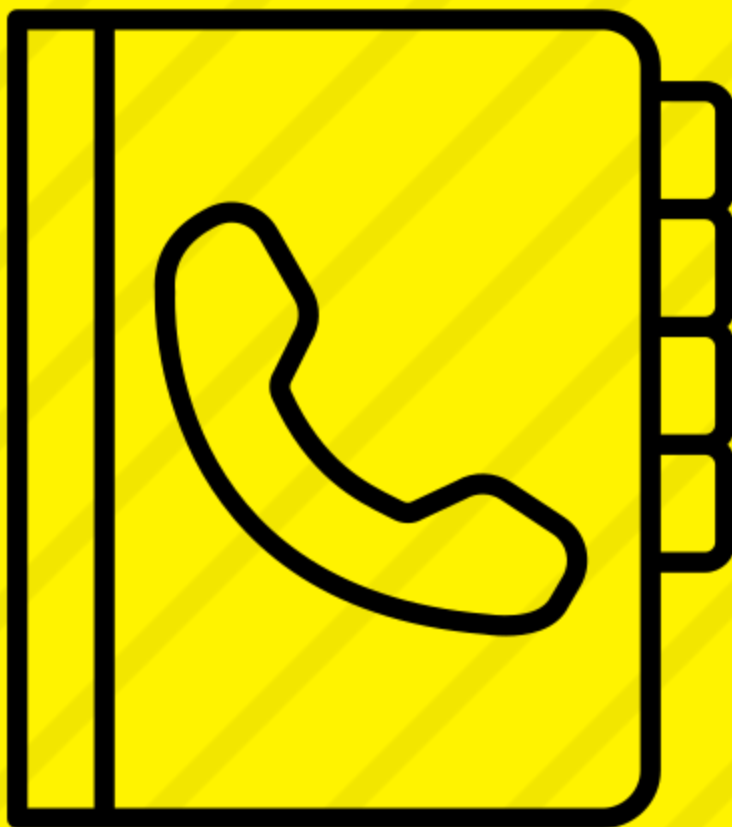
E o resultado seria:

```
 1234-5678
```

Acessando contatos em um Dicionário

O que será que acontece se tentarmos acessar os dados de um contato que não existe em nossa agenda?





Botando para rodar!

Vamos tentar acessar os dados de um contato que não existe em nossa lista.

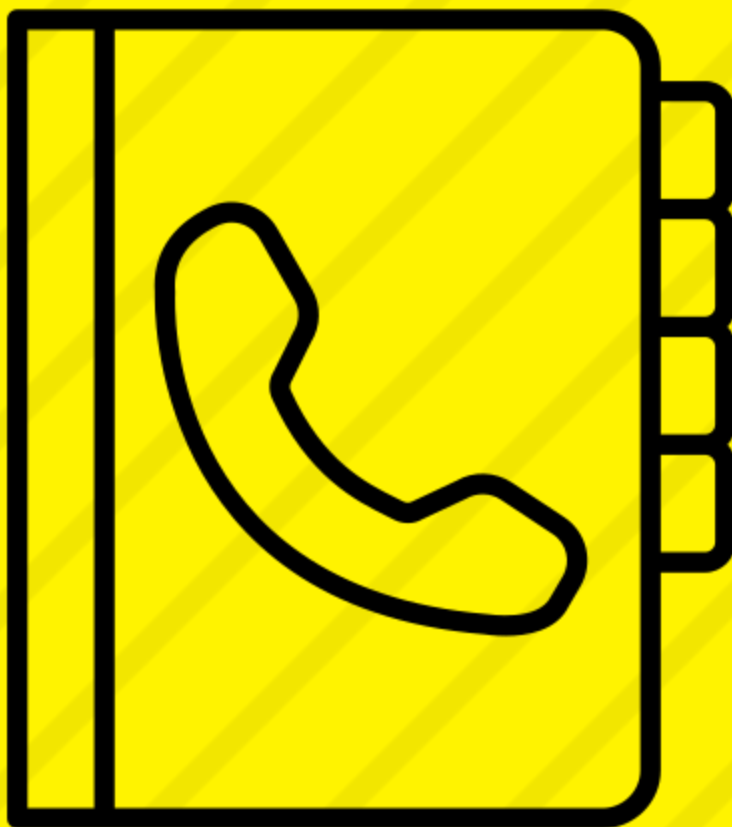
Exceção `KeyError` em um Dicionário

A exceção `KeyError` ocorre quando buscamos uma chave (no nosso caso um contato) que não existe.

Os dicionários possuem um **método específico para busca de valores**, o **método `get()`**, no qual passamos como parâmetros a chave que queremos e um valor padrão para retornar caso essa chave não seja encontrada

```
 print(contatos.get('Chuck Norris', 'Contato não encontrado'))
```

```
 print(contatos.get('Bruce Lee', 'Contato não encontrado'))
```

Botando para rodar!

Vamos implementar o método
`get()` em nossa agenda
telefônica



Botando para rodar 2!

Vamos criar um dicionário com os nomes dos atores de Vingadores e seus respectivos personagens.

Chris Evans – Capitão América

Mark Ruffalo – Hulk

Tom Hiddleston – Loki

Chris Hemworth – Thor

Robert Downey Jr – Homem de Ferro

Scarlett Johansson – Viúva Negra

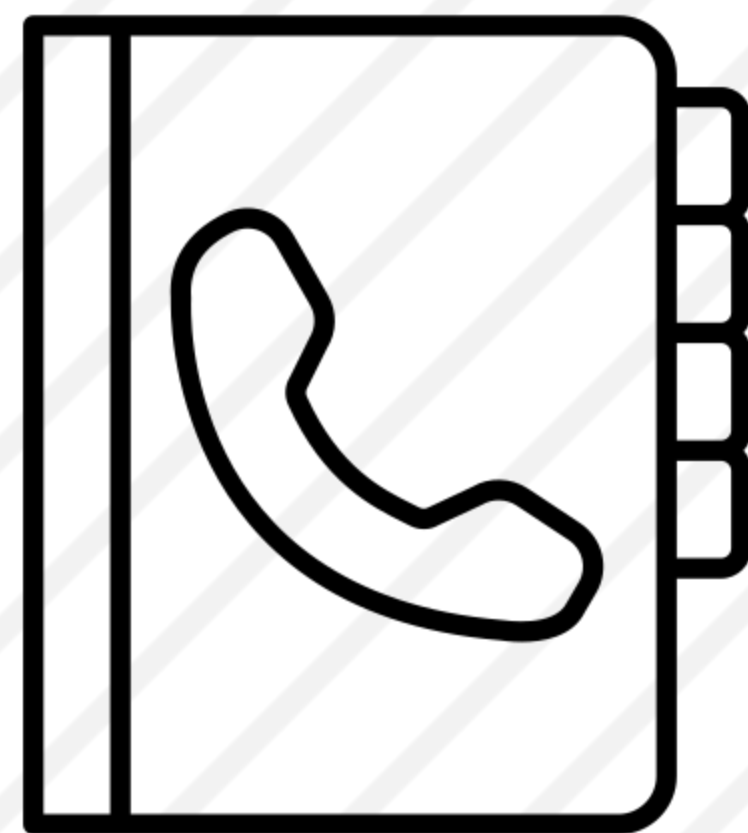


Buscando Elementos em um Dicionário

Na aula anterior vimos o método get (), que é utilizado para busca de valores, no qual passamos como parâmetros a chave que queremos e um valor padrão para retornar caso essa chave não seja encontrada.

Uma outra forma de fazermos a mesma coisa é utilizando o método values (), para verificar se determinado valor está presente em nosso dicionário.

 `print('1234-5678' in contatos.values())`



Botando para rodar!

Vamos ver o funcionamento do método `values()` em nossa agenda

Adicionando valores em um Dicionário

Ao contrário das listas, quando trabalhamos com dicionários o método `append()` NÃO FUNCIONA.

Para adicionarmos um elemento utilizamos a seguinte sintaxe:

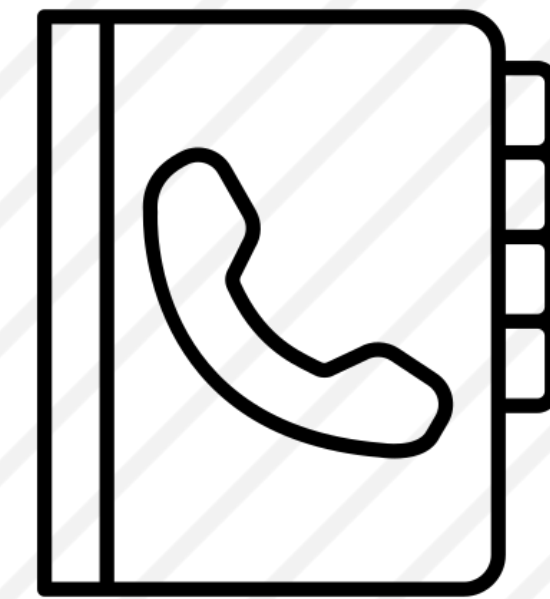
➡ `contatos ['Mulher Maravilha'] = '4356-8855'`

➡ `contatos ['Michael Myers'] = '6666 - 2424'`



Botando para rodar!

Vamos adicionar 5 contatos na
nossa agenda telefônica



Vamos adicionar 5 vingadores
no nosso time de heróis

Removendo itens em um Dicionário

Para excluirmos itens do nosso dicionário, basta utilizarmos o *statement del*:

 `del contatos['Mulher Maravilha']`

 `del contatos['Michael Myers']`

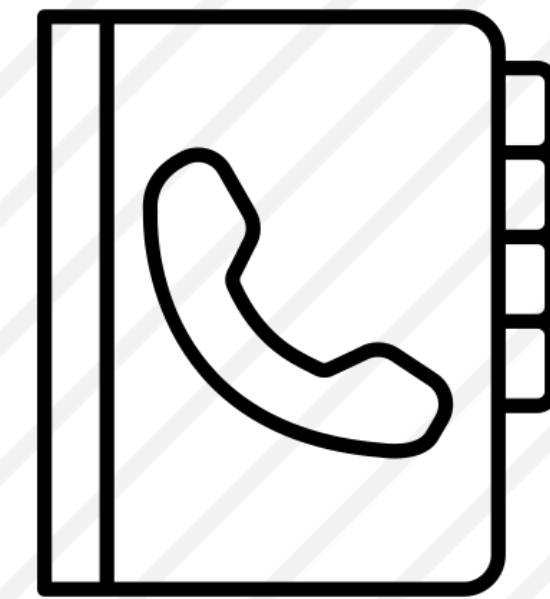
 `print(contatos)`

E se tentarmos excluir um contato que não existe??



Botando para rodar!

Vamos remover 3 contatos da nossa agenda telefônica



Vamos remover 2 vingadores do nosso time de heróis



Removendo itens em um Dicionário – pop ()

O método pop(), além de remover o elemento com a chave especificada no dicionário, nos retorna o valor desse elemento. Também podemos definir um valor padrão de retorno para caso a chave não seja encontrada.

```
contatos = {'Yan': '1234-5678', 'Pedro': '9999-9999', 'Ana': '8765-4321',  
            'Marina': '8877-7788', 'João': '8887-7778'}  
  
print(contatos.pop('Marina', 'Contato não encontrado'))  
print(contatos.pop('Catarina', 'Contato não encontrado'))  
print()  
print(contatos)
```

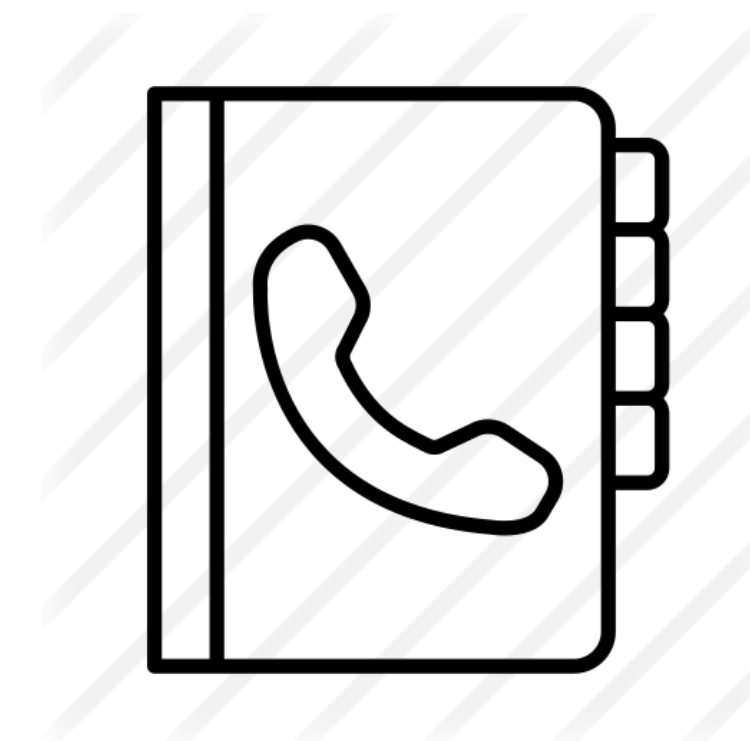


Botando para rodar!

Vamos praticar o método pop ()
em nossos 2 exemplos



+



Unindo Dicionários

Para unirmos dois dicionários podemos fazer **um loop (for)** para percorrer todos os elementos do dicionário e ir adicionando um a um:

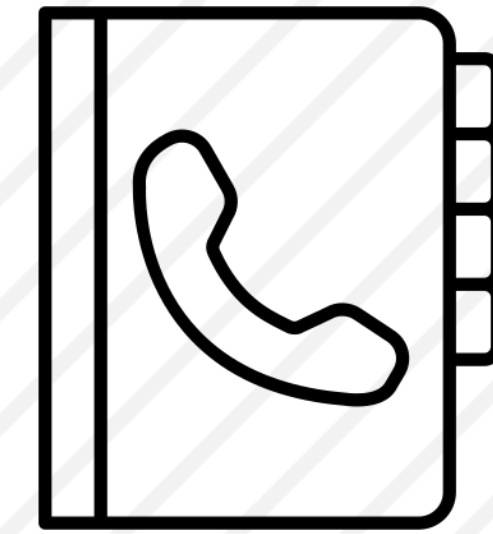
```
meus_contatos = {'Yan': '1234-5678', 'Pedro': '9999-9999',  
                 'Ana': '8765-4321', 'João': '8887-7778'}  
  
contatos_do_pedro = {'Yan': '1234-5678', 'Fernando': '4345-5434',  
                     'Luiza': '4567-7654'}  
  
for nome in contatos_do_pedro:  
    meus_contatos[nome] = contatos_do_pedro[nome]  
  
print(meus_contatos)
```





Botando para rodar!

Vamos criar uma outra lista telefônica e juntá-la com a nossa lista criada anteriormente utilizando o laço for.



Crie um dicionário chamado Vingadores com o nome dos heróis. Crie outro Dicionário chamado vilões com o nome dos vilões. Una os dois dicionários com o laço for.



Unindo Dicionários – update ()

Se quisermos atualizar o valor do primeiro dicionário para que contenha também os valores do segundo, podemos utilizar o método update () ao invés de um laço for:

```
meus_contatos.update(contatos_pedro)
print(meus_contatos)
```



Por hoje é só!

Obrigado! =)

Até a próxima aula.
Gustavo Molina.
gmolina@thefutureisblue.me