



# [DIA 3] Desafio

## O Desafio de Hoje

Assim como no desafio de ontem, temos um **programa que gerencia um dicionário de comidas**. O dicionário possui na key o nome da comida e no value a descrição da comida. Assim:

```
food_list = {  
    'paçoquinha': "Um doce de amendoim brasileiro",  
    'brigadeiro': 'Um doce muito delicioso',  
    'pizza': 'um tipo de comida da italia',  
    'hamburguer': 'comida dos USA'  
}
```

O programa usa **4 funções** que ainda não foram declaradas.

- `add_food`
- `delete_food`
- `update_food`
- `get_food`

Preciso que você **declare as funções com todas as instruções necessárias** para o programa executar as tarefas corretamente.

Bem como trate todos possíveis erros e valores enviados nos testes do programa

Sabemos que um usuário nem sempre envia exatamente o que uma função precisa receber né?

**No dia de hoje vou detalhar para você as 4 funções e o que cada uma recebe e faz:**

## A Função `add_food`

Esta função adiciona uma comida e sua descrição ao dicionário.

A execução correta dela é o seguinte:

```
add_food('hot dog', 'Um pão com salsicha sem purê.')
```

Para a função funcionar, ela precisa receber 2 valores.

- o primeiro sendo a chave (nome da comida - tipo string)
- o segundo sendo o valor (descrição da comida - tipo string)

**CASO** algum dos valores necessários não seja passado para função, a função deve retornar o erro e informar que é necessário passar 2 valores, o nome da comida e descrição.

**CASO** algum dos valores necessários não seja passado com o tipo de dado correto (string), a função deve retornar o erro e informar que o tipo de dado que precisa ser enviado é string.

**CASO** a comida já esteja cadastrada no dicionário (use o valor do nome da comida para verificar se já existe), a função deve informar que já existe uma comida cadastrada com o mesmo nome.

Recebendo os 2 valores corretamente e não existindo outra comida com o mesmo nome na lista, a função deve adicionar a comida no dicionário (food\_list).

Após adicionar, a função deve exibir a mensagem:

"{food\_name} adicionado com sucesso!"

**Exemplo de mensagens de erros e uso da função passando 4 possibilidades de valores:**

```
#### ADD_FOOD - TESTE 1
Usando add_food com valores sendo int
add_food(100, 23)

Ambos os valores precisam ser do tipo string

#### ADD_FOOD - TESTE 2
Usando add_food sem passar a descrição.
add_food('pizza')

Você precisa passar o nome da comida + descrição

#### ADD_FOOD - TESTE 3
Usando add_food com comida já existente.
add_food('brigadeiro', 'Um doce brasileiro')

brigadeiro já está cadastrado no dicionário

#### ADD_FOOD - TESTE 4
Usando add_food adicionando uma comida.
add_food('lasanha', 'Camadas de massa e molho')

lasanha cadastrado com sucesso!

#### DELETE_FOOD - TESTE 1
Usando delete_food com valor sendo int
delete_food(100)

O nome da comida precisa ser do tipo string
```

Todos os testes já estão implementados no código do projeto.

### A Função **delete\_food**

Esta função deleta uma comida do dicionário.

Para a função funcionar, ela precisa receber apenas 1 valor.

- nome da comida (do tipo string)

A execução correta dela é o seguinte:

```
delete_food('hot dog')
```

Recebendo o nome da comida, sendo do tipo string e existindo a comida no dicionário, a função deve remover do dicionário a comida passada.

**Exemplo de mensagens de erros e uso da função passando 4 possibilidades de valores:**

```
#### DELETE_FOOD - TESTE 1
Usando delete_food com valor sendo int
delete_food(100)

O nome da comida precisa ser do tipo string

#### DELETE_FOOD - TESTE 2
Usando delete_food sem nenhum valor.
delete_food()

Você precisa passar o nome da comida para deletar

#### DELETE_FOOD - TESTE 3
Usando delete_food com comida que não existe na lista.
delete_food('massa')

massa não existe no dicionário

#### DELETE_FOOD - TESTE 4
Usando delete_food removendo uma comida.
delete_food('paçoquinha')

paçoquinha foi deletado com sucesso!
```

### A Função `update_food`

Esta função atualiza a descrição de uma comida no dicionário.

Para a função funcionar, ela precisa receber apenas 2 valores.

- o primeiro sendo a chave (nome da comida - tipo string)
- o segundo sendo o valor (descrição da comida - tipo string)

A execução correta dela é o seguinte:

```
update_food('hot dog', 'É muito melhor com purê.')
```

Recebendo o nome da comida e nova descrição e ambos sendo do tipo string, a função deve atualizar a descrição da comida no dicionário.

**Exemplo de mensagens de erros e uso da função passando 4 possibilidades de valores:**

```
#### UPDATE_FOOD - TESTE 1
Usando update_food com valores sendo int
update_food(100, 23)

Ambos os valores precisam ser do tipo string

#### UPDATE_FOOD - TESTE 2
Usando update_food sem passar a descrição.
update_food('pizza')

Você precisa passar o nome da comida + nova descrição

#### UPDATE_FOOD - TESTE 3
Usando update_food com comida não existente.
update_food('sorvete', 'Um doce gelado da italia')

sorvete não existe no dicionário

#### UPDATE_FOOD - TESTE 4
Usando update_food e atualizando uma comida.
update_food('brigadeiro', 'Melhor doce do mundo.')

brigadeiro descrição atualizada para: Melhor doce do mundo.
```

### A Função `get_food`

Esta função pesquisa pelo nome da comida no dicionário e retorna a descrição da comida.

Para a função funcionar, ela precisa apenas 1 valor.

- nome da comida (tipo string)

A execução correta dela é o seguinte:

```
get_food('hamburger')
```

Recebendo o nome da comida e ele sendo do tipo `string`, a função deve pesquisar no dicionário a comida com aquele nome e trazer a descrição dela.

**Exemplo de mensagens de erros e uso da função passando 4 possibilidades de valores:**

```
#### GET_FOOD - TESTE 1
Usando get_food com valor sendo int
get_food(505)

O nome da comida precisa ser do tipo string

#### GET_FOOD - TESTE 2
Usando get_food sem passar a comida.
get_food()

Você precisa passar o nome da comida para consultar a descrição

#### GET_FOOD - TESTE 3
Usando get_food com comida não existente.
get_food('noodle')

noodle não existe no dicionário

#### GET_FOOD - TESTE 4
Usando get_food e pesquisando a descrição uma comida.
get_food('hamburger')

A descrição de hamburger é: comida dos USA
```

**SÃO ESSAS AS 4 FUNÇÕES.**

Sempre que chamadas as funções, os **argumentos/parâmetros** já estão sendo passados.

Você precisa recebê-los e trabalhar com eles na função para fazer/retornar o necessário.

O testes estão todos inclusos no código... Eles enviam vários possíveis valores para função que você vai criar.... Trate todos eles ;)

**Porém muita atenção!**

**Não altere o código que está debaixo da linha de !!!! AVISO !!!!**

Use a parte de cima para escrever o seu código e declarar as funções.

Ter a capacidade de pesquisar, ler erros, ler a documentação e resolver problemas é uma das principais habilidades na programação.

Conte com a comunidade no Slack e bom desafio!