

## **Dicionários:**

### **Criação e manipulação de dicionários**

Crie um dicionário chamado produto com os seguintes campos: nome, preco, quantidade. Adicione valores a esses campos e, depois, atualize o preço do produto.

Imprima o dicionário antes e depois da atualização.

### **Acessando valores de dicionários**

Crie um dicionário usuario com as chaves nome, idade, email. Imprima o nome do usuário utilizando a chave correspondente. Use o método get para tentar acessar um campo que não existe.

### **Utilizando fromkeys() para inicializar um dicionário**

Crie um dicionário usando o método fromkeys(), onde as chaves são os nomes de vários produtos e o valor padrão será não disponível. Permita que o usuário altere a disponibilidade de alguns produtos.

### **Usando o setdefault() para garantir a consistência do dicionário**

Crie um dicionário clientes com informações sobre clientes. Adicione um campo de endereço se ele não estiver presente usando o setdefault().

### **Iterando por um dicionário complexo**

Crie um dicionário que tenha uma lista e outro dicionário dentro dele (por exemplo, pessoa que tem nome, idade, endereços, e uma chave contatos que é outro dicionário). Itere sobre as chaves e valores do dicionário complexo.

### **Contagem de ocorrências com o Counter**

Crie uma lista de palavras repetidas e use o Counter para contar quantas vezes cada palavra aparece. Exiba as palavras mais comuns.

## **Funções:**

### **Funções com parâmetros e retorno**

Crie uma função chamada `saudacao` que recebe o nome de uma pessoa como argumento e retorna uma mensagem de boas-vindas. Imprima a saudação para o usuário.

### **Função para calcular área e perímetro de um triângulo**

Crie uma função `calcular_area_perimetro_triangulo` que recebe a base e a altura como parâmetros e retorna a área e o perímetro do triângulo. Teste com valores de sua escolha.

### **Funções com valores padrão**

Crie uma função chamada `multiplicar` que recebe dois números e retorna o produto. Caso o segundo número não seja informado, use 5 como valor padrão.

### **Função lambda para operação matemática**

Crie uma função lambda que receba dois números e retorne o maior deles. Teste a função com diferentes valores.

### **Funções aninhadas**

Crie uma função chamada `calculadora` que tenha funções internas para soma, subtração, multiplicação e divisão. Cada operação deve receber dois parâmetros. A função externa deve chamar e retornar o resultado da operação desejada.

## **Módulos e Pacotes:**

### **Importando e utilizando módulos**

Crie dois arquivos Python: `operacoes.py` com uma função `soma(a, b)` e `diferenca(a, b)`, e `main.py` que importe as funções de `operacoes.py` e as use.

### **Organizando um pacote Python**

Crie um pacote chamado `matematica`, com dois módulos dentro: `area.py` e `perimetro.py`. Cada módulo deve ter uma função que calcula a área e o perímetro de um quadrado e de um círculo. No arquivo principal, importe essas funções e calcule a área e o perímetro para um círculo de raio 5 e um quadrado de lado 4.

### **Criando um pacote para utilitários de strings**

Crie um pacote chamado `utils_strings` com dois módulos: `formata.py` (que tem funções para formatar texto) e `conta.py` (que tem funções para contar caracteres). Importe essas funções no arquivo principal e use-as para formatar e contar caracteres de uma string.

### **Utilizando o `__init__.py` em pacotes**

Crie um pacote chamado `geometria`, com os módulos `area.py` e `perimetro.py`. Em `__init__.py`, importe as funções de cálculo de área e perímetro e use-as em seu script principal.

**Extras:****Desafio: Criando um mini sistema de cadastro de clientes**

Crie um sistema de cadastro de clientes usando dicionários. O sistema deve permitir ao usuário cadastrar nome, idade, endereço e telefone. Depois, imprima uma lista com todos os cadastros.

**Desafio: Calculadora com funções e módulos**

Crie um módulo calculadora.py com funções para soma, subtração, multiplicação e divisão. Em um script principal, importe essas funções e crie uma interface simples para o usuário escolher a operação desejada.