

Lista de Exercícios: Fundamentos de Python

1. Sintaxe Básica e Controle de Fluxo

Exercício 1: FizzBuzz Clássico. Escreva um script que itere de 1 a 50. Para cada número:

- Se o número for divisível por 3, imprima "Fizz".
- Se o número for divisível por 5, imprima "Buzz".
- Se o número for divisível por 3 e por 5, imprima "FizzBuzz".
- Caso contrário, imprima o próprio número.

Exercício 2: Validador de Senha Simples. Crie um programa que use um laço while para pedir ao utilizador que digite uma senha. O laço deve continuar a pedir a senha até que o utilizador digite "python123". Quando a senha correta for inserida, imprima "Acesso concedido" e termine o laço com a declaração break.

2. Funções e Funções Lambda

Exercício 3: Calculadora de Área. Defina uma função chamada calcular_area_retangulo que aceite dois parâmetros: comprimento e largura. A função deve calcular a área (comprimento * largura) e retornar o resultado. Chame a função com os valores 10 e 5 e imprima o valor retornado.

Exercício 4: Filtrar Palavras Longas. Dada a lista de palavras: palavras = ['gato', 'elefante', 'cachorro', 'ornitorrinco', 'formiga']. Use a função filter() em conjunto com uma função lambda para criar uma nova lista que contenha apenas as palavras com mais de 5 caracteres.

Exercício 5: Mapear Temperaturas. Dada uma lista de temperaturas em Celsius: celsius = [38, 42, -3]. Use a função map() com uma função lambda para converter cada temperatura para Fahrenheit. A fórmula é: $(C \times 9/5) + 32$. Armazene os resultados numa nova lista.

3. Escopo de Variáveis (LEGB, global, nonlocal)

Exercício 6: Prever a Saída (Regra LEGB). Analise o código abaixo e preveja qual será a saída. Justifique a sua resposta com base na regra LEGB.

```
x = "global"
```

```
def funcao_externa():
```

```
    x = "enclosing"
```

```
    def funcao_interna():
```

```
        x = "local"
```

```
        print(x)
```

```
    funcao_interna()
```

```
funcao_externa()
```

Exercício 7: Contador de Chamadas com global. Crie uma variável global chamada `contador_chamadas` inicializada com 0. Em seguida, defina uma função que, cada vez que é chamada, incrementa `contador_chamadas` em 1 usando a palavra-chave global. Chame a função três vezes e, no final, imprima o valor de `contador_chamadas`.

Exercício 8: Fábrica de Contadores com nonlocal. Crie uma função "fábrica" chamada `criar_contador()`. Esta função deve inicializar uma variável `contagem = 0` e definir uma função interna chamada `incrementar()`. A função `incrementar` deve usar a palavra-chave `nonlocal` para aumentar a `contagem` em 1 e imprimir o seu novo valor. A função `criar_contador` deve retornar a função `incrementar`.

Código para testar a sua solução

```
meu_contador = criar_contador()
```

```
meu_contador() # Deve imprimir 1
```

```
meu_contador() # Deve imprimir 2
```

4. Mutabilidade e Imutabilidade

Exercício 9: Efeitos Colaterais de Objetos Mutáveis. Defina uma função `adicionar_item(carrinho, item)` que recebe uma lista (mutável) e adiciona um item a ela. Crie uma lista `meu_carrinho = ['maçã', 'banana']`. Chame a função para adicionar 'laranja'. Depois da chamada da função, imprima `meu_carrinho`. O que aconteceu com a lista original e porquê?

Exercício 10: O Perigo do Argumento Padrão Mutável. Analise a função abaixo. Chame `adicionar_tarefa("Lavar a louça")` e imprima o resultado. Depois, chame `adicionar_tarefa("Passear o cão")` e imprima o resultado novamente. Explique por que o segundo resultado é `['Lavar a louça', 'Passear o cão']` e como corrigiria este comportamento.

```
def adicionar_tarefa(tarefa, lista_de_tarefas=[]):
```

```
    lista_de_tarefas.append(tarefa)
```

```
    return lista_de_tarefas
```

Exercício 11: Tupla com Conteúdo Mutável. Crie uma tupla `dados_pessoais = ('Ana', 25,)`. Tente modificar o primeiro elemento da tupla (`dados_pessoais = 'Beatriz'`). O que acontece? Agora, tente adicionar uma nova habilidade à lista dentro da tupla (`dados_pessoais[1].append('Machine Learning')`). O que acontece e por que é possível, mesmo a tupla sendo imutável?

5. Importação de Módulos

Exercício 12: Usar o Módulo `math` com Alias. Importe o módulo `math` usando o alias `m`. Em seguida, use o alias para calcular e imprimir o valor de `Pi` (`m.pi`) e a raiz quadrada de 81 (`m.sqrt(81)`).

Exercício 13: Importar um Item Específico. Importe apenas a função `choice` do módulo `random`. Crie uma lista de cores `['vermelho', 'azul', 'verde', 'amarelo']` e use a função `choice` para selecionar e imprimir uma cor aleatória da lista.