Lista de Exercícios

1º. Desafio: Crie uma lista chamada minha_lista com os seguintes itens: “Flash”, 100, “oi”, False, [], -1, 100. Para esta lista, escreva os seguintes comandos:

- a) Inserir “comida” e 100 no final da lista.
- b) Inserir “cachorro” na posição de índice 2.
- c) Inserir 99 no início da lista.
- d) Encontrar o índice de “oi”.
- e) Contar o número de ocorrências de 100 na lista.
- f) Remover a primeira ocorrência do número 100 da lista.

In [ ]:

```
```

2º. Desafio: Escreva uma função calcula_minimo_2 que receba como entrada uma lista de números inteiros (tamanho mínimo da lista de 3 elementos) e retorna soma dos dois menores valores na lista. Por exemplo, calcula_minimo([4,3,6,1,2]) deve retornar 3 (1+2). !

In [ ]:

```
```

3º. Desafio: Escreva uma função soma_quadrados que receba uma lista de números e retorne a soma dos quadrados dos números na lista. Por exemplo, soma_dos_quadrados([2, 3, 4]) deve retorna 4+9+16 que é 29

In [ ]:

```
```

4º. Desafio: Crie uma função seleciona_alunos, que recebe uma lista de alunos e um valor inteiro correspondente a sua nota (Ex: [[‘Pedro’, 8], [‘Maria’, 9.5], ...]), e retira todos os alunos da lista que possuem nota menor que 7.!

In [ ]:

5º. Desafio: Escreva uma função chamada espelho que recebe uma strings como parâmetro e anexa o conteúdo da string a si mesma em ordem inversa. Por exemplo: [a, b, c] -> [a, b, c, c, b, a]

In [ ]:

6º. Desafio: Escreva uma função quatidade_palavras_5 que receba uma lista de palavras e retorne o número de palavras na lista que tenham comprimento 5. Por exemplo, quantidade_palavras_5(['Brasil','peralta','mesa','lance','teste']) deve retornar 2.!b

In [ ]:

7º. Desafio: Escreva uma função radical que receba um radical e uma lista de palavras e retorne o número de palavras na lista que tenham o radical fornecido. Por exemplo, radical(‘part’, ['partiu','parceiro','mesa','partida','parente']) deve retornar 2!

In [ ]:

8º. Desafio: Escreva uma função palavras_inicio_fim_iguais que receba como entrada uma lista de palavras e retorne a quantidade de palavras da lista que possuem 2 ou mais caracteres e cujos primeiro e últimos caracteres sejam iguais. Por exemplo, palavras_inicio_fim_iguais(['aba', 'xyz', 'aa', 'x', 'bbb']) retorna 3.!

In [ ]:

```
```

9º. Desafio: Um triângulo retângulo pode ter lados cujos comprimentos são todos inteiros. O conjunto de três valores inteiros para os comprimentos dos lados de um triângulo retângulo é chamado de Tripla de Pitágoras. Os comprimentos dos três lados devem obedecer à relação de que a soma dos quadrados de dois dos lados é igual ao quadrado da hipotenusa. Escreva um programa para identificar todas as triplas de Pitágoras para lado1, lado2 e hipotenusa, não maiores que 500. Utilize um método de força bruta, com um loop for triplamente aninhado que tenta todas as possibilidades.!

In [ ]:

```
```

10º. Desafio: Escreva uma função troca que receba três strings str1, velho e novo e troca em str1 todas as ocorrências de velho por novo. Por exemplo, troca('Um aluno, dois alunos, tres alunos.','aluno','estudante') deve retornar a string 'Um estudante, dois estudante, tres estudantes.'. Observação: existe a função replace que faz isso, mas você não deve usá-la. (Dica: use os métodos split e join).!

In [ ]:

```
```

11º. Desafio: Escreva uma função soma_impares que receba uma lista de números inteiros e retorne a soma dos números ímpares na lista. Por exemplo, soma_impares([11,20,36,41,55,6]) deve retornar 107.

In [ ]:

12º. Desafio: Escreva uma função quantidade_negativos que receba uma lista de números inteiros e retorne a quantidade de números negativos na lista. Por exemplo, quantidade_negativos([-1,-2,3,4,-5,-6]) deve retornar 4.

In [ ]:

13º. Desafio: Qual o resultado de cada um dos comandos seguintes:!

- a) 'Python'[1]
- b) 'Strings são sequências de caracteres.'[5]
- c) len('maravilhoso')
- d) 'Misterio'[:4]
- e) 'p' in 'Pineapple
- f) 'apple' in 'Pineapple
- g) 'pear' not in 'Pineapple
- h) 'apple' > 'pineapple'
- i) 'pineapple' < 'Peach'

In [ ]:

14º. Desafio: Escreva uma função nao_eh_ruim que receba como entrada uma string e encontre as primeiras ocorrências de 'nao eh' e de 'ruim'. Se 'nao' vier antes de 'ruim', deve-se substituir a expressão 'nao eh...ruim' por 'eh bom', retornando a nova string. Por exemplo, nao_eh_ruim('Figado nao eh tao ruim') deve retornar 'Figado eh bom'.

In [ ]:

15º. Desafio: Calcule o valor de π usando a série infinita:

Construa um dicionário que guarde a quantidade de termos da série e o valor de π correspondente. Por exemplo: {1:4, 2: 2.666666666666667, 3: 3.466666666666667 , ... } Quantos termos da série são necessários para obter 3,14? 3,141? 3,1415? 3,14159?

In [ ]: