

Computação I - Python

Laboratório 6

Seguindo com nossas boas práticas, para cada um dos exercícios a seguir:

- antes de começar a escrever código, faça o estudo do problema e o planejamento de sua solução.
- lembre de botar a **documentação**:

```
'''Calcula e retorna a divisão de a por b;  
int, int -> float'''
```

- escolha **nomes elucidativos** para suas funções e parâmetros;
- pense em **valores de teste** relevantes para testar sua função.
- recorra ao **teste de mesa** quando estiver com dificuldade para entender algum erro de funcionamento ou resultado inadequado de sua função.
- **para fazer a entrega desta atividade prática, escreva suas funções na ferramenta Machine Teaching.**

Vamos lá!

1. Faça uma função chamada **altera_frase** que receba uma frase, uma palavra e uma posição. Caso a palavra já exista na frase, transforme a primeira ocorrência da palavra na frase para maiúscula. Caso a palavra não exista, insira a palavra na frase na posição dada. Assuma que a primeira palavra está na posição 0. Retorne a nova frase.

Exemplo 1:

Frase: “Meu nome é ana”

Palavra: “ana”

Retorno: “Meu nome é ANA”

Exemplo 2:

Frase: “Meu nome é ana”

Palavra: “primeiro”

Posição: 1

Retorno: “Meu primeiro nome é ana”

2. Faça uma função chamada **faltas** que receba uma lista no seguinte formato: `[['Brasil', 'Italia', [10, 9]], ['Brasil', 'Espanha', [5, 7]], ['Italia', 'Espanha', [7,8]]]`. Esta lista indica o número de faltas que cada time fez em cada jogo. Na lista acima, no jogo entre Brasil e Itália, o Brasil fez 10 faltas e a Itália fez 9. Dada essa lista, a função deve retornar o total de faltas do campeonato.

3. Faça uma função `insere(lista_numero, n)` que dada uma lista ordenada (crescente) de números inteiros e um número inteiro n , inclua n na posição correta, ou seja, de tal maneira que a lista continue ordenada.

DICA: Provavelmente, a primeira idéia que vem à cabeça é inserir diretamente na posição correta, verificando os elementos da lista até achar a posição onde o n deve ser inserido. Mas nós ainda não sabemos fazer isso (veremos nas próximas aulas). Pense em outra estratégia de resolução deste problema, usando a função `list.sort(lista)`.

4. Faça uma função `maiores` que, dada uma lista de números inteiros e um número inteiro n , retorna outra lista, que contenha todos os números da lista original maiores que n .
5. Faça uma função `acima_da_media` que dada uma lista com as notas dos alunos de uma turma, retorne uma lista ordenada com as notas que ficaram acima da média.

DICA: Python provê uma função `sum(lista)`. Aproveite também a função desenvolvida no exercício 4.

6. Defina uma função `eh_ordenada` que, dada uma lista não vazia contendo uma quantidade qualquer de valores numéricos, diga se a lista está ordenada em ordem crescente, decrescente ou não ordenada. A retorno da função deve ser uma tupla cujo primeiro elemento é `True`, caso a lista esteja ordenada, e `False`, caso contrário e cujo segundo elemento seja "crescente", "decrescente" ou "desordenada", indicando se a lista está ordenada em ordem crescente, decrescente ou não está ordenada, respectivamente.

DICA: Atenção! Lembre-se que algumas operações sobre listas alteram o valor da lista original ao invés de retornar uma nova lista com as alterações. Esse é o caso da função `list.sort(lista)`, por exemplo. Então, se você precisar preservar o valor da lista original, faça uma cópia antes de alterá-la.