

Computação I - Python

Laboratório 7

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 **dados 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 no IDLE.**

Vamos lá!

1. Faça uma função que simule um jogo de dois dados. A função deve contar quantas vezes os dados foram jogados até que saiam números repetidos. Use a função *randint* do módulo *random* para simular a jogada de um dado. *Atenção*: essa é uma função que não tem dados de entrada.
2. Seguimos com o desenvolvimento do *contatinhosApp*, que está sendo desenvolvido em Python por uma equipe. Este é um aplicativo de agenda de contatos. O projeto de desenvolvimento do *contatinhosApp* prevê o desenvolvimento da interface e de funções de operação, ou seja, as funções que implementarão as funcionalidades do aplicativo. A modelagem de dados do *contatinhosApp* é a seguinte:
 - As informações de cada um dos contatos são armazenadas em uma lista.
 - A única informação obrigatória nesse aplicativo para que um contato seja cadastrado é o nome. Além destas, a agenda comporta também as seguintes informações: um ou mais números de telefone, email e instagram.
 - Um detalhe: O número de telefone (ou telefones) de um contato é armazenado em uma lista, pois cada contato poder ter mais de um telefone.
 - A lista com as informações de cada contato segue o seguinte padrão:

<i>índice na lista</i>	<i>informação</i>
0	nome
1	lista de telefones
2	email
3	instagram

- Exemplo dos dados de um contato no formato do modelo de dados do *contatinhosApp*: ['Bruno Campos', ['2199112233', '2133992211'], 'brunoc91@emailquente.com.br', '@brunocampos91'].
- **NOVIDADE: As listas com as informações de todos os contatos salvos são armazenadas em uma lista chamada *contatos*. Isto é, *contatos* é uma lista e cada um dos seus elementos é também do tipo lista.**
- Exemplo da lista de contatos de um usuário do *contatinhosApp*: `contatos = [['Bruno Campos', ['2199112233', '2133992211'], 'brunoc91@emailquente.com.br', '@brunocampos91'], ['Julia Fields', ['2198145233'], '', '@juju.fields']]`

Você é um dos programadores da equipe de serviço do *contatinhosApp*. Nesta semana, sua lista tarefas é:

- (a) Buscar os dados de um contato salvo: sua função recebe como entrada a lista de contatos e uma string referente ao nome buscado, e deve retornar uma lista de contatos que tem o nome buscado. Mais detalhes:
 - Como as informações de cada contatinho estão em uma lista, sua função irá retornar uma lista de listas, no mesmo formato que a lista *contatos*.
 - Na lista retorno, deverão constar todos os contatinhos cujo nome corresponde ao nome buscado (o nome buscado seja substring do nome do contato, desconsiderando diferenças entre letras maiúsculas e minúsculas).
 - Por exemplo, se foi buscado 'field', seria retornado os dados da "Julia Fields" e também do "Fieldsnei Campos Jr", caso ambos estivessem salvos no aplicativo.
 - A lista de retorno **não** deve ter cópias das informações de cada contatinho, e sim aliases (referências) para as informações dos contatinhos na lista de contatos original. Ou seja, se identifiquei que o elemento índice 1 da lista de contatos original (os dados da "Julia Fields", por exemplo) deve fazer parte da lista de retorno, vou acrescentar na lista de retorno `contatos[1]`, e não `contatos[1][:]`.
 - Se o nome buscado não corresponder a nenhum contatinho, deve ser retornada uma lista vazia.