



Aula Prática 9 – Listas

- 1) (ex1.py) A função integrada `range()` pode ser utilizada para criar uma lista com uma sequência de números. Crie um arquivo no formato .py (ou use o Shell do Python) e teste algumas chamadas da função `range` para você entender seu funcionamento. Por exemplo, teste os seguintes comandos:

```
1 print(list(range(10)))
2 print(list(range(15)))
3 print(list(range(0, 10)))
4 print(list(range(5, 10)))
5 print(list(range(10, 15)))
6 print(list(range(0, 10, 2)))
7 print(list(range(0, 10, 3)))
8 print(list(range(3, 16, 3)))
```

Com base na função `range`, vamos definir algumas funções para gerar uma lista com uma sequência de números.

- a) A função abaixo, tem por objetivo criar e retornar uma lista com os números inteiros de 0 a n . No entanto, ela possui um pequeno erro. Corrija-o e use a função para criar uma lista L com os elementos de 0 a 10.

```
1 def myRange1(n, L=[], i=0):
2     if i == n:
3         return L + [i]
4     else:
5         myRange1(n, L + [i], i+1)
6
7 L = myRange1(10)
8 print(L) #Saída esperada: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

O que aconteceria se no lugar de $L + [i]$ (linhas 3 e 5) fosse $[i] + L$? Implemente e teste para perceber a diferença.

- b) Crie uma função, chamada `myRange2`, que retorne uma lista com os números de a a b . Se $a > b$, a função deve retornar uma lista vazia.
- c) Crie uma função, chamada `myRange3`, que retorne uma lista com os números de a a b com salto s . Por exemplo, a chamada `myRange3(2, 10, 2)` deve retornar a lista `[2, 4, 6, 8, 10]`. Se $a > b$, a função deve retornar uma lista vazia. Note que `myRange3` gera o mesmo resultado de `myRange2` se $s = 1$.

d) Crie uma função, chamada `myRange`, que:

- Se for passado um único parâmetro, chame e retorne a lista gerada por `myRange1`;
- Se for passado dois parâmetros, chame e retorne a lista gerada por `myRange2`;
- Se for passado três parâmetros, chame e retorne a lista gerada por `myRange3`;

Sua função deve ser de tal forma que o código abaixo funcione e gere os resultados esperados:

```
1 def main():
2     print(myRange(10)) #Saida esperada: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
3     print(myRange(2, 10)) #Saida esperada: [2, 3, 4, 5, 6, 7, 8, 9, 10]
4     print(myRange(10, 2)) #Saida esperada: []
5     print(myRange(0, 10, 2)) #Saida esperada: [0, 2, 4, 6, 8, 10]
6     print(myRange(5, 30, 5)) #Saida esperada: [5, 10, 15, 20, 25, 30]
7
8 main()
```

2) (ex2.py) Faça uma função recursiva, chamada `semRepetidos`, que receba que receba uma lista `l1` e retorne outra lista sem os elementos repetidos de `l1`. Por exemplo, o resultado/retorno de `semRepetidos([1, 2, 1, 3, 2, 1, 4, 0])` deve ser a lista `[1, 2, 3, 4, 0]`.

Dica: use o operador `in`;

3) (ex3.py) Faça uma função que receba duas listas (`l1` e `l2`) e retorne uma lista com a união de `l1` e `l2` de forma que a nova lista não contenha elementos repetidos. Por exemplo, se `l1 = [1, 2, 3, 3]` e `l2 = [1, 5, 3]`, a união de `l1` e `l2` deve ser `[1, 2, 3, 5]` (a ordem dos elementos na lista não importa).

4) (ex4.py) A função `choice` (do módulo `random`) pode ser utilizada para escolher um dos elementos de uma lista. Ela recebe uma lista de valores e retorna aleatoriamente um dos elementos. Por exemplo:

```
1 print(random.choice([0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]))
2 print(random.choice(myRange(1, 60)))
```

Faça uma função, chamada `sorteioMegaSena`, que retorne uma lista com 6 números sorteados da MegaSena. Lembrando que:

- Os valores devem ser entre 1 e 60;
- A ordem dos valores sorteados não importa;
- Não pode haver repetição dos valores sorteados, ou seja, se um valor aparece na lista de sorteados, ele não pode aparecer novamente.