**Resolução Relatório Python 04 – Exercício 03**

1. Using the Python interpreter, interrogate the difference between these two ways to copy a list. Careful! One of these is NOT what you might expect.
   * Method 1
     + Create a list. For example: my\_list = ['a', 'bb', 'ccc']
     + Make a copy using the = assignment operator: list\_copy = my\_list
     + Print the original list print(my\_list)
     + Alter the list\_copy by adding a new element using append()
     + Print the original list again print(my\_list)
   * Method 2
     + Create a list. For example: my\_list2 = ['a', 'bb', 'ccc']
     + Make a copy with the copy() method list\_copy2 = my\_list2.copy()
     + Print the original list print(my\_list2)
     + Alter the list\_copy2 by adding a new element using append()
     + Print the original list again print(my\_list2)

# Resolução pelo Método 1

>>> my\_list = ['a' , 'bb' , 'ccc']

>>> list\_copy = my\_list

>>> print (my\_list)

['a', 'bb', 'ccc']

>>> list\_copy.append('dddd')

>>> print (my\_list)

['a', 'bb', 'ccc', 'dddd']

# Resolução pelo Método 2

>>> my\_list2 = ['a' , 'bb' , 'ccc' ]

>>> list\_copy2 = my\_list2.copy()

>>> print (my\_list2)

['a', 'bb', 'ccc']

>>> list\_copy2.append('dddd')

>>> print (my\_list2)

['a', 'bb', 'ccc']

A diferença é que ao usar o método copy(), conforme a resolução do segundo exemplo, a lista original não é alterada. Já ao usar o sinal de atribuição ' = ', a lista original é alterada ao se fazer modificações na “lista cópia”. Isso ocorre, pois foi copiado um ponteiro para a lista original.