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']

#criando uma lista

>>> my\_list = ['aa', 'bb', 'cc', 'dd']

* + - Make a copy using the = assignment operator: list\_copy = my\_list
    - Print the original list print(my\_list)

#criando cópia com sinal de ‘=’

>>> list\_copy = my\_list

>>> print (my\_list)

['aa', 'bb', 'cc', 'dd']

* + - Alter the list\_copy by adding a new element using append()
    - Print the original list again print(my\_list)

#adicionando elemento à lista

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

>>> print (my\_list)

['aa', 'bb', 'cc', 'dd', 'ee']

#o elemento foi adicionado também a lista original, já que quando utiliza-se o sinal de ‘=’, as alterações acontecem em ambas as listas, pois é copiado um ponteiro para a lista original.

* + Method 2
    - Create a list. For example: my\_list2 = ['a', 'bb', 'ccc']

>>> my\_list2 = ['aa', 'bb', 'cc', 'dd']

* + - Make a copy with the copy() method list\_copy2 = my\_list2.copy()

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

* + - Print the original list print(my\_list2)

>>> print (my\_list2)

['aa', 'bb', 'cc', 'dd']

* + - Alter the list\_copy2 by adding a new element using append()

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

* + - Print the original list again print(my\_list2)

>>> print (my\_list2)

['aa', 'bb', 'cc', 'dd']

#Nesse caso, a lista original continua sem alterações, pois as alterações apenas são feitas na cópia.