**1.** For the next series of tasks about lists use the interpreter:

a. Create a list of 5 of your favorite things.

>>> coisasfavoritas = ["videogames", "pizza", "música", "férias", "halloween"]

**Comentários:** aqui foi criada a lista de nome coisasfavoritas.

b. Use the print() function to print your list.

>>> print(coisasfavoritas)

['videogames', 'pizza', 'música', 'férias', 'halloween']

**Comentários:** Foi utilizada a função print para imprimir a lista criada.

c. Use the print() function to print out the middle element.

>>> print(coisasfavoritas[2])

música

**Comentários:** Nesse caso, a função print imprimiu apenas o terceiro elemento da lista (elemento do meio), o qual foi indicado pelo número 2. Isso se dá devido ao fato que a indexação começa a partir do número 0, então para se obter o terceiro elemento é necessário usar o 2.

d. Now replace the middle element with a different item, your favorite song, or song bird.

>>> coisasfavoritas[2] = "Cornelia Street"

**Comentários:** Aqui, o terceiro elemento (indicado pelo número 2) foi alterado para “Cornelia Street”, o nome de uma música.

e. Use the same print statement from b. to print your new list. Check out the differences.

>>> print(coisasfavoritas)

['videogames', 'pizza', 'Cornelia Street', 'férias', 'halloween']

**Comentários:** Foi possível aqui verificar a lista atualizada após a modificação feita na questão anterior, que substituiu o elemento “música” por “Cornelia Street”.

f. Add a new element to the end. Read about append().

>>> coisasfavoritas.append("livros")

**Comentários:** O método append() adiciona um novo elemento ao final da lista. Assim, a lista atualizada é ['videogames', 'pizza', 'Cornelia Street', 'férias', 'halloween', 'livros'].

g. Add a new element to the beginning. Read about insert().

>>> coisasfavoritas.insert(0, "café")

**Comentários:** O método insert() adicionou um novo elemento à lista na posição indicada. Como se desejava adicionar o elemento no começo da lista, se usou o número 0 uma vez que a indexação começa a partir desse número. A lista agora é ['café', 'videogames', 'pizza', 'Cornelia Street', 'férias', 'halloween', 'livros'].

h. Add a new element somewhere other than the beginning or the end.

>>> coisasfavoritas.insert(4, "suco de maracujá")

**Comentários:** O método insert() foi utilizada para adicionar o elemento “suco de maracujá” na quinta posição. A lista agora é ['café', 'videogames', 'pizza', 'Cornelia Street', 'suco de maracujá', 'férias', 'halloween', 'livros'].

i. Remove an element from the end. Read about pop().

>>> coisasfavoritas.pop()

'livros'

**Comentários:** O método pop() remove e destaca o último elemento da lista, que nesse caso era “livros”. Assim, a lista atualmente é ['café', 'videogames', 'pizza', 'Cornelia Street', 'suco de maracujá', 'férias', 'halloween'].

j. Remove an element from the beginning.

>>> coisasfavoritas.pop(0)

'café'

**Comentários:** Ao usar o método pop() especificando uma posição na lista, é possível remover e destacar o elemento que está naquela posição, que nesse caso era “café”. A lista atualizada é ['videogames', 'pizza', 'Cornelia Street', 'suco de maracujá', 'férias', 'halloween'].

k. Remove an element from somewhere other than the beginning or the end.

coisasfavoritas.pop(3)

'suco de maracujá'

**Comentários:** Da mesma forma que na questão anterior, o método pop() foi utilizado para remover um elemento de uma posição especificada, nesse caso sendo a quarta posição. Com o elemento “suco de maracujá” removido, se tem a seguinte lista: ['videogames', 'pizza', 'Cornelia Street', 'férias', 'halloween']

l. Use join to create a string. Join the elements on ', '

>>> coisasfavoritas\_string = ", ".join(coisasfavoritas)

**Comentários:** Foi criada uma nova variável nomeada coisasfavoritas\_string, a qual uniu os elementos da lista coisasfavoritas com uma vírgula entre eles. Para isso, foi utilizado o método join(). Ao se imprimir a nova variável, se tem o seguinte resultado:

>>> print(coisasfavoritas\_string)

videogames, pizza, Cornelia Street, férias, Halloween

**3.** 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']

>>> minha\_lista = ['a', 'bb', 'ccc']

* Make a copy using the = assignment operator: list\_copy = my\_list

>>> copia\_da\_lista = minha\_lista

* Print the original list print(my\_list)

>>> print(minha\_lista)

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

* Alter the list\_copy by adding a new element using append()

>>> copia\_da\_lista.append('dddd')

**Comentários:** Foi adicionado o elemento ‘dddd’ à cópia criada.

* Print the original list again print(my\_list)

>>> print(minha\_lista)

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

**Comentários:** Ao se imprimir a lista original, observou-se que ela incluía o elemento ‘dddd’, que havia sido adicionado à cópia. Isso ocorre pois a cópia criada aponta para a lista original e assim, elas compartilham dos mesmos elementos. Portanto, qualquer alteração feita em uma afetará a outra também.

Method 2

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

>>> outra\_lista = ['a', 'bb', 'ccc']

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

>>> outra\_copia = outra\_lista.copy()

* Print the original list print(my\_list2)

>>> print(outra\_lista)

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

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

>>> outra\_copia.append('dddd')

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

>>> print(outra\_lista)

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

**Comentários:** Ao contrário do método anterior, a mudança feita na cópia da lista não afetou a lista original. Isso ocorre pois usando o método copy(), se cria uma cópia independente da lista, ou seja, que não está ligada à original e assim não a afeta.