

**LIsTs**

## **Part 1 - Fun with Indexing**

1. Run this block of code in the python shell.

|  |
| --- |
| >>> a = ['she','sells','sea','shells','by','the','sea','shore'] >>> b = "selfish shellfish" >>> c = [1, 1, 2, 3, 5, 8, 13] |

Fill in the blanks below according to the example. Some may have **more than one** correct **answer**. Fill in the **guess** column before you try it on your computer.

|  |  |  |
| --- | --- | --- |
| **List or String Indexing** | **Guess** | **Result** |
| *a[1]* | ‘sells’ | *'sells'* |
| b[3:4] | ‘shells’ | ‘shells’ |
| c[5] | 8 | 8 |
| c[:-2] | [1,1,2,3,5] | [1,1,2,3,5] |
| c[6],c[2] | ‘sea’ | 'sea' |

2. Write **True** or **False** for each expression below. **First** try to **guess**, **then** check in the **shell**:

'by' in a \_\_\_\_\_\_\_True\_\_\_ [1,2,5] in c \_\_True\_\_\_\_\_\_

'self' in b \_\_\_\_\_True\_\_\_ 'sh' in c \_\_\_\_\_False\_\_\_\_\_

a[2] == a[6] \_\_\_\_\_\_True\_\_\_ a[3] == b[8:13] \_\_\_\_\_False\_\_\_\_

3. Try out this code, which uses **len()**. Fill in the blanks with the results that would appear:

|  |
| --- |
| >>> dog = 'dalmatian' >>> len(dog) \_\_\_\_\_9\_\_\_\_\_\_\_\_\_\_\_\_\_ >>> dogs = [dog,'poodle','boxer'] >>> len(dogs)  \_\_\_\_\_\_\_\_\_\_3\_\_\_\_\_\_\_\_ |

## Part 2 - Shelli

Run these two lines of code by **typing** them in the **shell**.

>>> one = [1,2,3,4]

>>> two = [7,6,5,4]

>>> three = ["y1", "friends", "fun"]

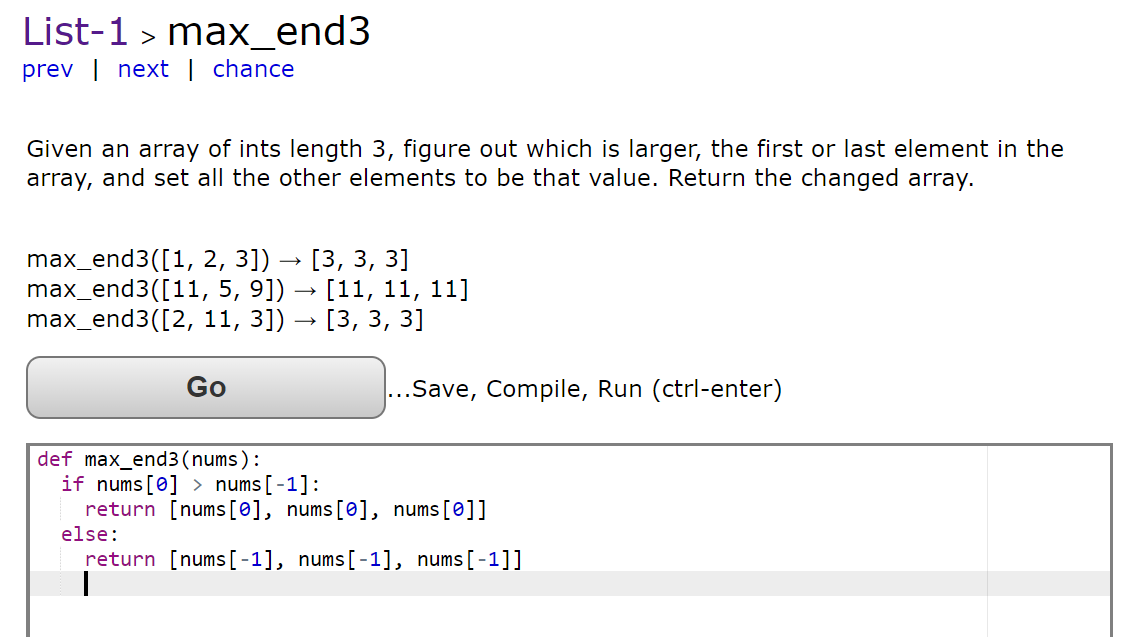
What gets printed to the screen after running the code in each box? (some boxes produce errors). First, you should **guess**, and then check the answer in the **shell**.

|  |  |  |
| --- | --- | --- |
| Input | Guess | Shell Output |
| >>> print (one + two) | [1, 2, 3, 4, 7, 6, 5, 4] | [1, 2, 3, 4, 7, 6, 5, 4] |
| >>> print (one[3]) | 4 | 4 |
| >>> one.remove(4)  >>> print(one) | [1, 2, 3] | [1, 2, 3] |
| >>> one.append(4)  >>> print(one) | [1, 2, 3, 4] | [1, 2, 3, 4] |
| >>> one.pop(1) | 2 | 2 |
| >>> print(one) | [1,3] | [1,3] |
| >>> one.sort()  >>> print(one) | [1,3] | [1,3] |

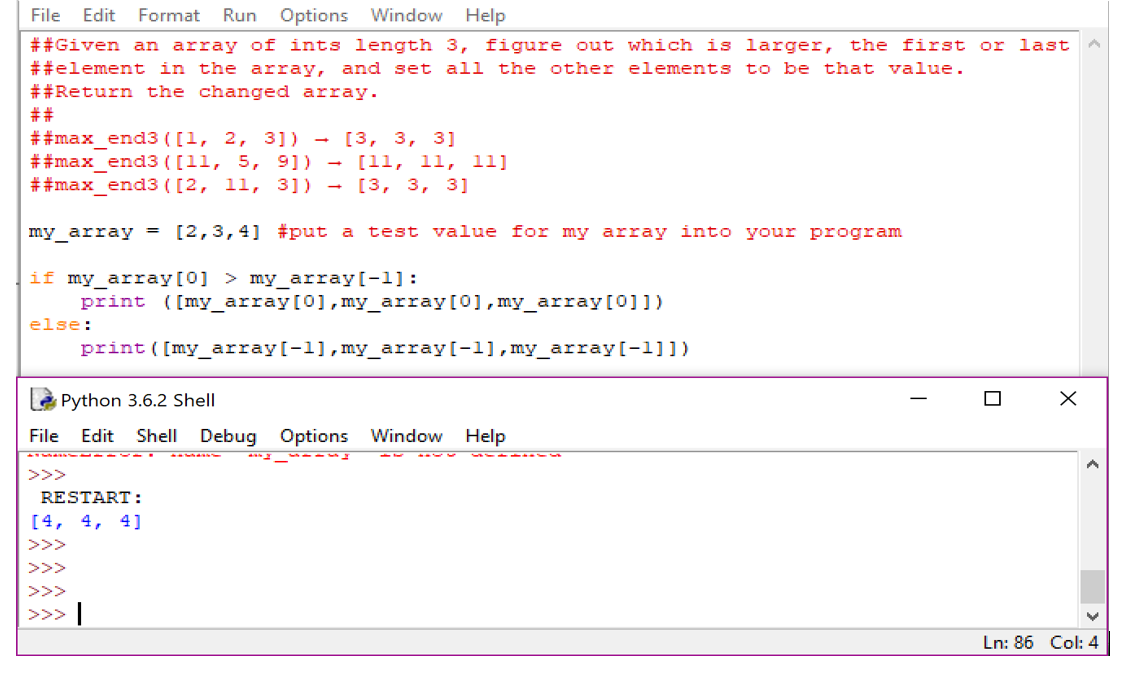
## Part 3 - Fun with LIsTs

**1.Read through the following codingbat example:**

We will now being using codingbat to do some fun exercises. Here is an example of how to use codingbat:



Or if that looks like gibberish (nonsense):



**2. Complete the following codingbat exercises**

List checking:

<https://codingbat.com/prob/p236930>

List matching:

<https://codingbat.com/prob/p245813>

**Bonus One:**

Still unsure about list operations, do this bonus: shelli on your own. Pick your own lists and try out the commands we learned on them. You can also try the following operations we didn't get to.

>>> one = [1,2,3,4]

>>> two = [7,6,5,4]

|  |  |  |
| --- | --- | --- |
| Input | Guess | Shell Output |
| >>> one.reverse()  >>> print(one) |  |  |
| >>> one.sort()  >>> print(one) |  |  |
| >>> one.remove(4)  >>> print(one) |  |  |
| >>> three = [3]  >>> four = three\*3  >>> print(four) |  |  |
| >>> five = one.copy()  >>> one.sort()  >>> print(one)  >>> print(five) |  |  |

You're turn! Come up with your own shelli.

**Bonus Two :**

[**Do more codingbat problems with lists!**](http://codingbat.com/python/List-1)

* You can do the problems in whatever order you want!

**Survey Time:**

[Super Fun Survey](https://docs.google.com/forms/d/e/1FAIpQLSe6vSrShyrutuneets8vJHe_bhchpsBHZNp_RccRIv46K8GsQ/viewform?usp=sf_link)