

**PPiC 4.1** This question asks me to create a list with the five items given. Using square brackets to include those comma-delimited values in the list, and assign it to `myList`.

```
>>> myList = [7,9,'a','cat',False]
>>> myList
[7, 9, 'a', 'cat', False]
```

**PPiC 4.2** This question asks me to use methods provided by lists in Python.

- (a) Using `append` method to add 3.14 to the end of `myList`, and then add 7 to the end of `myList`. Return `myList`, it now has 3.14 and 7.

```
>>> myList.append(3.14)
>>> myList.append(7)
>>> myList
[7, 9, 'a', 'cat', False, 3.14, 7]
```

- (b) Using `insert` method to insert 'dog' at the 3rd position in `myList`.

```
>>> myList.insert(3,'dog')
>>> myList
[7, 9, 'a', 'dog', 'cat', False, 3.14, 7]
```

- (c) Using `index` method to return the index of the first occurrence of 'cat', which is at 4th position.

```
>>> myList.index('cat')
4
```

- (d) Using `count` method to return the number of occurrence of 7, which is 2.

```
>>> myList.count(7)
2
```

- (e) Using `remove` method to remove the first occurrence of 7.

```
>>> myList.remove(7)
>>> myList
[9, 'a', 'dog', 'cat', False, 3.14, 7]
```

- (f) First, use `index` method to find the position of 'dog' in `myList`, which is 2.

```
>>> myList.index('dog')
2
```

Then, use `pop` method to remove and return the 2nd item in `myList`. Return `myList` with 'dog' removed from `myList`.

```
>>> myList.pop(myList.index('dog'))
'dog'
>>> myList
[9, 'a', 'cat', False, 3.14, 7]
```

**PPiC 4.3** To split the string “the quick brown fox” into a list of words,

First, assign the string to a variable called phrase. Then use split method to take the string as a parameter that indicates the places to break the string into substring.

In this case we want to break the string into a list of words, so we assign no parameter to split, so Python will break the string using the space between words as the delimiter.

```
>>> phrase = "the quick brown fox"
>>> phrase.split()
['the', 'quick', 'brown', 'fox']
```

**PPiC 4.10** Assign an empty list to variable A.

Initialize the list, which is same as `mylist = [[]]*3`, the empty list is repeated 3 times, and assigns the result to `mylist`.

Return `mylist`, the result is also a list consist of three empty lists.

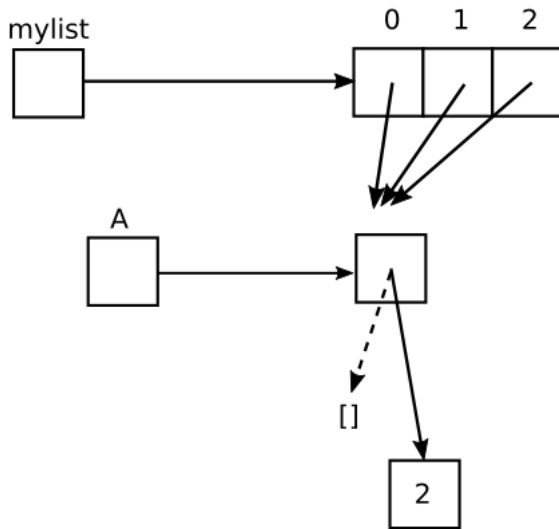
```
>>> A = []
>>> mylist = [[]]*3
>>> mylist
[[], [], []]
```

**PPiC 4.11** As we just did in the previous exercise, `mylist = [[], [], []]`. Index the position 1 in `mylist`, which returns an empty list. Using append method to add 2 to the end of `mylist[1]`, which returns `[2]`.

Since a change to one element in `mylist` shows up in all three occurrences, the repetition result is actually a list of three references to the same list.

```
>>> mylist[1]
[]
>>> mylist[1].append(2)
>>> mylist
[[2], [2], [2]]
>>> mylist[1]
[2]
```

**PPiC 4.12** Draw a reference diagram to illustrate what is happening in the previous two exercises.



By using the append function at the index 1 which is the empty list [], the original empty list [] changes to [2], so `mylist = [A]*3=[[2], [2], [2], 2]`

**PPiC 4.18** Replace the call to the sum function with an iteration that computes the total of the values in `alist`.

- (1) Define function `sum` with the parameter `alist`.
- (2) Set the initial value of `sum` as the value in the 0 position of `alist`
- (3) For each `i` in `alist` start from position 1 to the end
- (4) Increase `sum` by `i`
- (5) Return `sum` that computes the total of the values in `alist`.

```
>>> def sum(alist):
    sum = alist[0]
    for i in alist[1:]:
        sum = sum + i
    return sum
```

To test how the function works, we use `[1,2,3,4,5,6]` as the parameter `alist`.

```
>>> sum([1,2,3,4,5,6])
21
```

**PPiC 4.24** To make a dictionary using the two lists given,

- (1) Define function `makeDictionary` with two parameters: `names`, `scores`, both are lists.
- (2) Assign an empty dictionary as the initial value to `scoreDict`.
- (3) Set the initial of value `v` to 0.
- (4) Write a for statement: for each key `k` in list `names`, the value of the key in `scoreDict` equals the value in list `scores`.
- (5) Increase the value of `v` by 1 to the next `v`.
- (6) Return the dictionary `scoreDict` with the names as the key and the scores as the values.

```
>>> def makeDictionary(names,scores):
    scoreDict={}
    v = 0
    for k in names:
        scoreDict[k] = scores[v]
        v = v + 1
    return scoreDict
```

To test if the function works, use the two lists provided in the question as parameters.

```
>>> names=['joe','tom','barb','sue','sally']
>>> scores=[10,23,13,18,12]
>>> makeDictionary(names,scores)
{'barb': 13, 'joe': 10, 'sue': 18, 'tom': 23, 'sally': 12}
```

**PPiC 4.31** To print out a table of students and their scores with the students listed in alphabetical order,

- (1) Define function `order` with parameter `dictionary`
- (2) Make a list of names from the keys of dictionary
- (3) Modifies names to be sorted, list names is sorted in alphabetical order.
- (4) Write a `for` statement: for each key `k` in names, check if `k` is in dictionary. If it is, print the student name and the corresponding value in dictionary.

```
>>> def order(dictionary):
    names = list(dictionary.keys())
    names.sort()
    for k in names:
        if k in dictionary:
            print(k,dictionary[k])
```

To test how the function works, we take the dictionary created in 4.24 as the parameter dictionary for function `order`.

As we can see from the result, students' names are listed in alphabetical order.

```
>>> order({'barb':13,'joe':10,'sue':18,'tom':23,'sally':12})
barb 13
joe 10
sally 12
sue 18
tom 23
```

**Bonus PPIIC 4.7** To write a function `shuffle` that takes a list and returns a new list with the elements shuffled into a random order.

- (1) Import `random` module so that we can use the `random.randint()` later in the function.
- (2) Define function `shuffle` with parameter `list`
- (3) Decrease the length of list by 1 to get the index of the last element in the list
- (4) Assign an empty list as the initial value to `Newlist`
- (5) Write a `for` statement: for each number `i` in range 0 to the length of list, generate a random integer in the range 0 to `a-i`, assign it to the `idx`.
- (6) `list[idx]` gives the element in the `idx` position in list; use insert method to insert `list[idx]` at the `idx` position in the `Newlist`.
- (7) Remove the first occurrence of `list[idx]` in the list, so the list does not include the `list[idx]` that we have just put into the `Newlist`
- (8) Return `Newlist` with the elements shuffled in list

```
>>> import random
>>> def shuffle(list):
    a = len(list) - 1
    Newlist = []
    for i in range(len(list)):
        idx = random.randint(0,a-i)
        Newlist.insert(idx,list[idx])
        list.remove(list[idx])    return Newlist
```

To test if the function works, we take list `[1, 'dog', 4, 6]` as the parameter.

```
>>> shuffle([1,'dog',4,6])
[4, 'dog', 1, 6]
>>> shuffle([1,'dog',4,6])
[4, 1, 'dog', 6]
>>> shuffle([1,'dog',4,6])
['dog', 1, 6, 4]
>>> shuffle([1,'dog',4,6])
[1, 'dog', 4, 6]
>>> shuffle([1,'dog',4,6])
['dog', 1, 4, 6]
>>> shuffle([1,'dog',4,6])
['dog', 1, 6, 4]
>>> shuffle([1,'dog',4,6])
[1, 6, 4, 'dog']
>>> shuffle([1,'dog',4,6])
[6, 4, 1, 'dog']
>>> shuffle([1,'dog',4,6])
[1, 4, 6, 'dog']
>>> shuffle([1,'dog',4,6])
```