Arrays in Python

Arrays in python come in many forms. They are called Lists, Tuples and Dictionaries. This document works mainly with Lists...but first:

Strings, which we have already used, are in reality, arrays as well.

An array, in a simple form, is using one variable to store many different values.

For example, a string stores many characters under one string variable.

The List

Lists are what Python calls one form of an array. It is, as the name implies, a listing of values associated with one variable.

Defining a List

A list syntax is enclosed in [square brackets] and each element or value is separated by a comma.

For example:
marks = [76, 68, 92, 35, 71, 87]

Accessing Elements

Similar to strings, we can “slice” the array:

```
print (marks[2])
92
print (marks[1:3])
[68,92]
```

Remember, elements start counting at 0 and that [n:m] starts at n and continues to, but does not include, m.

Note that displaying an entire list or a portion will also display the values separated by a comma and with the square brackets on the sides.

If you wanted to display an entire list without the brackets, you would need to use a loop:

```
myList = [1,2,3,4,5,6]
for i in myList:
    print (i, end="\t")
```

Modifying Elements of a List

```
myList[2] = 7
print (myList)

[1, 2, 7, 4, 5, 6]
```

As shown above, each individual element can be changed by using variable[elementNumber] = newValue

Also, multiple elements can be modified at once:

```
myList = [1,2,3,4,5,6]

myList[2:4] = [7,8]
print (myList)

[1, 2, 7, 8, 5, 6]
```

Adding a New Element to the List

The following shows a number of ways of adding to the list.

```
myList = [1,2,3,4,5,6]

myList.append(7)
myList = myList + [8]
myList += [9]
myList += [10,11,12]

print (myList)

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
```

Removing Elements from a List

The following shows a number of ways of removing elements from a list. All methods use the keyword **del** and we can “slice” the List as needed.

```
myList = [1,2,3,4,5,6]

del myList[-1]
print (myList)
del myList[2:4]
print (myList)
del myList[0]
print (myList)

[1, 2, 3, 4, 5]
[1, 2, 5]
[2, 5]
```

Example #1
Find the average of the marks of a student.

```
marks = [76, 68, 92, 87, 79, 88, 94, 56]
total = 0

for mark in marks:
    total += mark

average =total / len(marks)
print ("Average : ", average)
```

Note: len(marks) - what does this do?

Example #2

Write a program that asks the user to input a course code before the program displays the student's current mark in that course.

```
classes = ["CHC2P", "CHV2OH", "ENG2DN", "GLC2OH", "ICS3M",
                "MPM2D", "SNC21", "SNC2D", "HFN2O"]
marks =      [87, 76, 74, 83, 88, 79, 92, 65, 78]

course = input("What class do you want the mark for?")
if course in classes:
    spot = classes.index(course)
    print ("Your mark in", course, "is", marks[spot])
else:
    print ("Umm, you don't have", course)
```

Note: Both classes and marks are "related" lists. Each element in classes relates to the same element number in marks. Index will return the element of a List that corresponds to the course entered.

Example #3

Write a program that creates a deck of playing cards.

```
suits = ["Clubs", "Diamonds", "Hearts", "Spades"]
values = ["Ace", "2", "3", "4", "5", "6", "7", "8", "9", "10", "Jack", "Queen", "King"]
deck = []  # creates an empty List
for s in suits:
    for v in values:
        deck.append(v + " of " + s)
print (deck)
```

Note: Each “card” is appended to what starts off as an empty list. We could then write a program that will deal cards randomly from this deck.

Example #4

Write a program that creates a deck before dealing out five cards.

```
from random import *

suits = ["Clubs", "Diamonds", "Hearts", "Spades"]
values = ["Ace", "2", "3", "4", "5", "6", "7", "8", "9", "10", "Jack",
"Queen", "King"]
deck = []
for s in suits:
    for v in values:
        deck.append(v + " of " + s)

shuffle(deck)
hand = deck[:5]
print (hand)
```

Note: What does shuffle do? How nice is that?

Some list methods that we can use:

COMMAND	EXAMPLE	OUTPUT
sort	myList = [3,2,6,4,5] myList.sort() print (myList) myList.sort(reverse = True) print (myList)	[2, 3, 4, 5, 6] [6, 5, 4, 3, 2]
reverse	myList = [3,2,6,4,5] myList.reverse() print (myList)	[5, 4, 6, 2, 3]
remove	myList = [3,2,6,4,5] myList.remove(6) print (myList)	[3, 2, 4, 5]
index	myList = [3,2,6,4,5] spot = myList.index(6) print (spot)	2
count	myList = [5,5,6,4,5] count = myList.count(5) print (count)	3
pop	myList = [1,2,3,4,5] last = myList.pop() print (last) print (myList)	5 [1, 2, 3, 4]

Using the previous information as well as loops and if statements, complete the following:

1) Write a program that asks the users to input any 10 numbers. Your program should then display the numbers in order from largest to smallest as well as the median.

2) Write a program that asks the user to input five names. Your program should output the names alphabetically.

3) Write a program that merges two sorted Lists into one new sorted List (without using sort). You may use any values you want of any length. But it should work for any length of List.

4) Write a program that deals two “hands” of five random playing cards. Repeats not allowed.

5) Make a list of five of your friends and a list of five emotions or action words (e.g. “loves”, “hates”, “misses” ....). For each friend in the list randomly choose a word and another friend in the list. Don't worry if “Jimmy loves Jimmy”, he probably does anyways.

6) Make a list of ten fruits and another list of the price of each fruit. Use these two lists to create an attractive “fruit stand” menu.

7) Write a program that simulates the following:

100 lockers in a school all start with their doors closed.

A student walks from the first locker to the last, opening each door.

A second student then walks from the first locker to the last, changing the state of every second door, starting from the second door. In other words, if a door is open, they close it. If the door is closed, they open it.

The process continues, the third student changing the state of every third locker starting at the third locker, etc...until 100 students have passed through.

Output the state of each of the 100 lockers.
Also, display which lockers are open, for example: 1 3 8 etc...

8) Write a program that rolls a pair of 6-sided dice 1000 times. Output how many times each **sum** has occurred.

9) Write a program that plays the game Blackjack or 21 with a user. GIve the user the option to “hit” or “stay”.

10) Write a program that simulates the following game:

6 contestants are each given a key. One of them has a key that unlocks the door to a new car. They win the car if their key unlocks it.

Write a program that simulates this game 1000 times before displaying how many times each contestant won.