What can you do with a list?

- create a list

mylist = range(10)

mylist = [19, ‘a’, 3.14, ‘Tea’, False]

- access an item

1st item: mylist[0] is 19

2nd item: mylist[1] is ‘a’

last item: mylist[-1] is False

- access a slice

mylist[1:4] is [‘a’, 3.14, ‘Tea’]

note range is [inclusive:exclusive]

a slice is always a list:

mylist[1:2] = [‘a’] not ‘a’

What can you do with a list?

- add items to a list

mylist.append(2) -> [19, ‘a’, 3.14, ‘Tea’, False, 2]

mylist.insert(1, ’hi’) -> [19, ‘hi’, ‘a’, 3.14, ‘Tea’, False, 2]

mylist = mylist + [‘bye’] -> [19, ‘hi’, ‘a’, 3.14, ‘Tea’, False, 2, ‘bye’]

- remove items

by position: del(mylist[2])

by value: mylist.remove(‘a’)

from the end: mylist.pop()

What can you do with a list?

- get information

test to see if an item is in the list: if ‘Bye’ in mylist:

len(mylist) gives number of items in the list

mylist.index(‘Tea’) returns the position of item ‘Tea’

mylist.count(‘bye’) counts the number of ‘bye’s in the list

- miscellaneous

mylist.sort()

mylist.reverse()

mylist.extend()

2*mylist

What can you do with a string?

- create a string

name=input('enter your name')

name='John Sixpack'

- access item/s

name[0] is the first character in the string: 'J'

name[7] is the 8th character in the string: 'x'

name[1:7] is a slice: 'ohn S'

- add to a string

strings are immutable: so create a new string and assign to same variable

name = name + ' hello' -> 'John Sixpack hello'

name = name[0:4] + ' hello' + name[4:12] -> John hello Sixpack

What can you do with a string?

- getting information

if ‘a’ in name:

len(name)

name.find(‘ola’) returns position where the first occurrence of ‘ola’

- misc.

name.count(‘p’)

name[2].isalpha()

‘:’.join([‘abc’, ‘def’, ‘ghi’])

"867-395-0892".split('-')

“ John Sixpack “.strip()

5*'+'

Example: Playlist shuffle

Problem: There are N songs in a given playlist. Shuffle this playlist.

example playlist: shuffled playlist: [4,3,11,..., N, ..., 1, 7]

Hint: use list data type and random.randint()

pseudocode?

Example: Playlist shuffle

Problem: There are N songs in a given playlist. Shuffle this playlist.

example playlist: shuffled playlist: [4,3,11,..., N, ..., 1, 7]

```
Initialize playlist (like getting a blank sheet of paper)
Until you have six numbers in the playlist
    Roll the die
    If the value of the die is not in the playlist
        Add it to the playlist
```

Example: Playlist shuffle

Problem: There are N songs in a given playlist. Shuffle this playlist.

example playlist: shuffled playlist: [4,3,11,..., N, ..., 1, 7]

```
1  # playlist_shuffle.py
2
3
4  import random
5
6  nsongs = eval(input("How many songs are in the playlist?"))
7
8  playlist=[]
9  while len(playlist) < nsongs:
10     song = random.randint(1,nsongs)
11     if song not in playlist:
12         playlist.append(song)
13
14 print(playlist)
```

Dice Odds

Imagine that before we go on vacation to Vegas we want to quickly figure out some of the relevant odds. Let's start with the frequency of various outcomes when rolling a pair of dice. We could do this by rolling a pair of fair dice many times and recording (counting!) how many times each outcome occurs. There are 11 possible outcomes because we can observe anywhere from 2 to 12 spots on the pair of dice. This means we need to record how many twos occur, how many threes occur, how many fours occur, and so on up to how many twelves occur.

Hint: use lists in storing outcome and/or occurrences.

Sample output for 1000 dice rolls.

```
====================
Outcome | Occurrences
--------+-----------
    2   |      29
    3   |      66
    4   |      71
    5   |     100
    6   |     137
    7   |     173
    8   |     132
    9   |     122
   10   |      78
   11   |      59
   12   |      33
--------------------
```

Dice Odds (solution)

```
 1 import random
 2 #import time as tm
 3 #start = tm.time()
 4
 5 ROLLS = 1000
 6
 7 # Initialize counters to 0.
 8 counters = [0] * 11
 9
10 # Roll dice many times and record frequency of outcomes.
11 for roll in range(ROLLS):
12     outcome = random.randint(1, 6) + random.randint(1, 6)
13     # Now increment the appropriate counter.
14     counters[outcome-2] = counters[outcome-2] + 1
15
16 # Display results.
17 print("  ======================")
18 print("  Outcome | Occurrences")
19 print("  --------+-------------")
20 for posn in range(11):
21     print("%7d   |%8d" % (posn+2, counters[posn]))
22 print("  ----------------------")
23
24 #end = tm.time()
25 #print("Calculation completed in ",end-start," seconds")
```