CS109
Lists

Let’s say that we would like to make a trivia game where we ask the player questions. Each question has a point value, with harder questions worth more points. We could start a program with some variables such as the following:

```
question1 = "What is the answer to life the universe and everything?"
answer1 = "42"
value1 = 1
question2 = "What does the J in Jython stand for?"
answer2 = "java"
value2 = 2
question3 = "What age do children start kindergarten?"
answer3 = "5"
value3 = 1
```

etc.

This works fine, but what if you had hundreds of questions? It would be too much work to explicitly set each one. It would be nice if we could programmatically access each separate variable.

For example, if we wanted to output all the questions, it would be nice if we could do the following:

```
for i in range(1, 1001):
    print question_i
```

Of course, this will not work because question_i is considered a single variable; python won’t insert the number value for i at the end of the variable.

However, the construct that allows us to do what we want is called **a list**. We’ve played with this a little bit when accessing samples in sounds and pixels in a picture.

A list is a consecutive group of variables that all have the same name. To refer to a particular *subscript* or index, we specify the name of the list variable and then the positive index into the list. The index is specified by square brackets. The first index is 0.

Here is an example where we’ve made a list of six numbers in the variable x:

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

List Index	**Contents**
x[0]	1
x[1]	2
x[2]	3
x[3]	4
x[4]	5
x[5]	6

We can refer to the list like we have defined six variables. x[0] refers to the first variable in the list. If we try to access x[6] in this case, then we will be trying to access memory beyond the bounds of the list and this will cause an error to occur.

Here are some simple ways we can access the array just like it was a variable:

```
x[3]) = 54                  # Stores 54 into list element 3
x[0] = x[3]                 # Copies 54 into list element 0
x[5] = x[2+1]               # Copies contents of x(3) to x(5)
i=5
x[i]=x[i]+1;                # Increment value in x[5]

for i in range(0,6):        # Print each list value out
        print x[i]

for i in range(0,len(x)):   # Also prints each list value out, more general
        print x[i]
```

The flexible thing here is we can programmatically access each variable, based on the index, instead of hard-coding the access by hand.

We can add a new element to the end of the list by using + and the new element in square brackets:

```
x  = x + [100]
x  = x + [300]
```

The list would now have 100 and 300 tacked onto the end:

List Index	**Contents**
x[0]	1
x[1]	2
x[2]	3
x[3]	4
x[4]	5
x[5]	6
x[6]	100
x[7]	300

We can store any type of data in a list. This includes strings, integers, floats, and even other lists! We'll use this later to make more complex data structures. For a short example, consider our problem of storing questions, answers, and point values. One technique is to use three separate lists:

```
questions = ["What is…", "Who was…", "How many…"]
answers = ["stuff", "kibo", "42"]
values = [1, 2, 1]
```

Another approach is to make lists of lists. Each sublist contains the question, answer, and value for that question:

```
question1 = ["What is ...", "stuff", 1]
question2 = ["Who was ...", "kibo", 2]
question3 = ["How many ..." "42", 1]
trivia = [question1, question2, question3]

>>> print trivia
[['What is...', 'stuff', 1], ['Who was ...', 'kibo', 2], ['How many...', '42', 1]]
>>> print trivia[0]
['What is...', 'stuff', 1]
>>> print trivia[0][0]
What is...
```

We will look at this in more detail shortly to build a trivia game program.

Let's look at a few example programs that use lists. The first one inputs 10 numbers from the user and then calculates the average:

```
def addNums():
  listOfNums = []

  for i in range(1,11):
      num = int(getInput("Enter number " + str(i)))  # str(i) to turn i into a string
      listOfNums = listOfNums + [num]                # Tack num onto end of list

  total = 0
  for i in range(0,10):
      total = total + listOfNums[i]

  print "The average is " , (total / 10)
```

In this example, we loop over the array twice. Once to input each value, and another time to generate the average. Note that we could compute the average while also entering the numbers, if we wished, but this method does allow us to save all input numbers for future processing.

Exercise:

Write a program that inputs 10 integer, numeric grades into a list and finds the average and the standard deviation.

The standard deviation is a measure of how spread out the grades are around the average. Formally, if X_1, X_2, … X_n are n numbers then:

$$s\tan dard\_deviation = \sqrt{\frac{(x_1 - ave)^2 + (x_2 - ave)^2 + ...(x_n - ave)^2}{n-1}}$$

Passing Lists To Functions

When a “normal” variable is passed as a parameter to a function, the general behavior is that changes to the variable are local to the function. For example:

```
def test(nonlist):
   nonlist = 3
```

We could make a variable x and set it to 10:

```
>>> x = 10
```

If we invoke the function, x is still 10, because the variable nonlist is local to test:

```
>>> test(x)
>>> print x
10
```

Lists behave differently because what is passed is really a reference to the place in memory where the list is stored. This means that any changes inside a function to a list passed as a parameter are reflected in the caller:

```
def test(somelist):
    somelist[0] = "foo"
```

We could make a list x and set it to [1,2,3]:

```
>>> x = [1,2,3]
```

If we call the function its first element will be changed:

```
>>> test(x)
print x
["foo", 2, 3]
```

This behavior arises because it is often easer to pass just the address of where the list is stored in memory. One reason for this is efficiency – if lists were passed by value, it would mean copying the entire contents of the list. If the list was very large, this would take a long time.

Note that if you change a list in a function that is passed as a parameter in such a way as to create a new list, then the original list is not changed:

```
def listConfusion(somelist):
  # This really makes a new list with "HI!" on the end
  somelist = somelist + ["HI!"]
  print somelist

>>> x = [1,2,3]
>>> listConfusion(x)
[1, 2, 3, 'Hi!']
>>> print x
[1,2,3]
```

If you want to change the original list where the function creates a new list, then you should **return** the changed list and save the returned value in a variable:

```
def listConfusion(somelist):
  # This really makes a new list with "HI!" on the end
  somelist = somelist + ["HI!"]
  print somelist
  return somelist

>>> x = [1,2,3]
>>> x = listConfusion(x)
[1, 2, 3, 'Hi!']
>>> print x
[1,2,3, 'Hi!']
```

List/Function Exercise:
Re-do the program that inputs 10 integer, numeric grades into an array and finds the average and the standard deviation, except this time create the following functions:

```
def findAverage(listGrades)
def findStDev(listGrades, average)
```

Recall that the standard deviation is a measure of how spread out the grades are around the average. Formally, if X_1, X_2, … X_n are n numbers then:

$$standard\_deviation = \sqrt{\frac{(x_1 - ave)^2 + (x_2 - ave)^2 + ...(x_n - ave)^2}{n-1}}$$

List Example – Trivia Game

Going back to the trivia game scenario, let's actually write a program to play the trivia game. First we need a database of trivia questions. Let's say we have come up with the following questions followed by the answer followed by the question point value:

```
The possession of more than two sets of chromosomes is termed?
polyploidy
2

Age of Amelia Earhart when she disappeared.
39

3
Actor whose real name was Marion Morrison
john wayne
1

Study of ancient inscriptions
epigraphy
2

I am the geometric figure most like a lost parrot
polygon
3
```

For a real game we would probably have a lot more questions!

First, let's make a function that takes as input separate lists to store the questions, answers, and point values and set them up with the proper values.

```
def setupQuestions():
  questions = []
  answers = []
  values = []

  questions = questions + ["The possession of more than two sets of chromosomes is
termed?"]
  answers = answers + ["polyploidy"]
  values = values + [2]

  questions = questions + ["Age of Amelia Earhart when she disappeared."]
  answers = answers + ["39"]
  values = values + [3]

  questions = questions + ["Actor whose real name was Marion Morrison"]
  answers = answers + ["john wayne"]
  values = values + [1]

  questions = questions + ["Study of ancient inscriptions"]
```

```
    answers = answers + ["epigraphy"]
    values = values + [2]

    questions = questions + ["I am the geometric figure most like a lost parrot"]
    answers = answers + ["polygon"]
    values = values + [3]

    # Return the lists in a list
    return [questions, answers, values]
```

This function returns a list where the first element is a list with the questions, the second is a list with the answers, and the third is a list with the values:

```
[['The possession of more than two sets of chromosomes is termed?', 'Age of Amelia Earhart when she disappeared.', 'Actor whose real name was Marion Morrison', 'Study of ancient inscriptions', 'I am the geometric figure most like a lost parrot'], ['polyploidy', '39', 'john wayne', 'epigraphy', 'polygon'], [2, 3, 1, 2, 3]]
```

Next, let's make the game so it simply asks all questions, outputs if the player is correct or not, and then outputs the total score. To keep track of these we need some new variables:

```
def triviaGame():
  score = 0              # Track player's score
  questionNum = 0        # Track if question 0 to 4
```

score an integer that tracks our score.   questionNum keeps track of which question we are asking, and will start at 0 and go up to 4, the last question.   When we reach 5 then we have asked all the questions and can quit the game.

Now let's fill in the rest:

```
def triviaGame():
  score = 0              # Track player's score
  questionNum = 0    # Track if question 0 to 4

  data = setupQuestions()
  questions = data[0]   # Returns list of three separate arrays
  answers = data[1]
  values = data[2]

  # Now loop through each question, from 0 to 4
  while (questionNum <> 5):
     guess = getInput(questions[questionNum])
     if (guess.lower() == answers[questionNum].lower()):
        score = score + values[questionNum]
        printNow("Thats right!  Your score is " + str(score))
     else:
        printNow("That is incorrect.  The correct answer is " +
answers[questionNum])
     questionNum = questionNum + 1
  print "The game is over.  Your total score is " , score
```

Let's refine our trivia game, and say that we would like to randomly select four questions out of all of the questions that we loaded, ask each to the player, output if the player is correct or not, and then output the total score. This doesn't make a lot of sense with just 5 questions, but if we had a lot more questions it would be make the game somewhat different every time we played.

To keep track of this new wrinkle we'll make some new variables:

```
questionsUsed              # List that tracks which questions are used
indexRandQuestion          # Index to randomly selection question
numQuestionsAsked          # Track how many questions have been asked
```

The questionsUsed list will be used to hold an integer indicating if we have asked the question before or not. This is so we won't randomly pick the same question to ask twice. We should initialize this in the setupQuestions subroutine:

```
def setupQuestions():
    questions = []
    answers = []
    values = []
    questionsUsed = []

    questions = questions + ["The possession of more than two sets of
chromosomes is termed?"]
    answers = answers + ["polyploidy"]
    values = values + [2]
    questionsUsed = questionsUsed + [0]   # 0 means not used, 1 means used

    questions = questions + ["Age of Amelia Earhart when she disappeared."]
    answers = answers + ["39"]
    values = values + [3]
    questionsUsed = questionsUsed + [0]

    questions = questions + ["Actor whose real name was Marion Morrison"]
    answers = answers + ["john wayne"]
    values = values + [1]
    questionsUsed = questionsUsed + [0]

    questions = questions + ["Study of ancient inscriptions"]
    answers = answers + ["epigraphy"]
    values = values + [2]
    questionsUsed = questionsUsed + [0]

    questions = questions + ["I am the geometric figure most like a lost parrot"]
    answers = answers + ["polygon"]
    values = values + [3]
    questionsUsed = questionsUsed + [0]

    # Return the lists in a list
    return [questions, answers, values, questionsUsed]
```

We just added a new list that contains all 0's to indicate no question has been used yet. Next let's write a function to pick a random question out of the array of questions. Here is some code to do that, and make sure it hasn't been picked before:

```
def pickRandomQuestion(questionsUsed):
   # Get a random index into the array
   q = random.randint(0, len(questionsUsed) - 1)
   # Keep picking until we find one that is unused
   while (questionsUsed[q] == 0):
       q = random.randint (0, len(questionsUsed) - 1)
   # Mark this question as being used
   questionsUsed[q] = 1
   # Return the index
   return q
```

Here we put this together now into the modified trivia game:

```
import random
def triviaGameRandom():
  score = 0              # Track player's score
  numQuestionsAsked = 0  # Tracks how many questions asked

  data = setupQuestions()
  questions = data[0]   # Returns list of four separate arrays
  answers = data[1]
  values = data[2]
  questionsUsed = data[3]

  # Now loop through questions, from 0 to 3 (4 total)
  while (numQuestionsAsked < 4):
     indexRandQuestion = pickRandomQuestion(questionsUsed)
     guess = getInput(questions[indexRandQuestion])
     if (guess.lower() == answers[indexRandQuestion].lower()):
        score = score + values[indexRandQuestion]
        printNow("Thats right!  Your score is " + str(score))
     else:
        printNow("That is incorrect.  The correct answer is " +
answers[indexRandQuestion])
     numQuestionsAsked = numQuestionsAsked + 1
  print "The game is over.  Your total score is " , score
```

One modification that would be good to make is to read the questions from a file. This will shorten the amount of code in the setupQuestions function, and also make it easier to maintain the database of trivia questions so we don’t have to go mess with our code when we change or add new questions.

Let’s say that this is the format of our file, C:\TRIVIA.TXT. The first line indicates how many questions there are:

```
5
The possession of more than two sets of chromosomes is termed?
polyploidy
2
Age of Amelia Earhart when she disappeared.
39
3
Actor whose real name was Marion Morrison
john wayne
1
Study of ancient inscriptions
epigraphy
2
I am the geometric figure most like a lost parrot
polygon
3
```

Here is the modified setupQuestions subroutine to load in the file:

```
def setupQuestions():
  questions = []
  answers = []
  values = []
  questionsUsed = []

  filevar = open(r"c:\trivia.txt","rt")
  line = filevar.readline().strip()  # Read in first line with num of q's
  numQuestions = int(line)
  for i in range(0,numQuestions):
    questions = questions + [filevar.readline().strip()]
    answers = answers + [filevar.readline().strip()]
    values = values + [int(filevar.readline())]
    questionsUsed = questionsUsed + [0]
  filevar.close()

  # Return the lists in a list
  return [questions, answers, values, questionsUsed]
```

Let's do one more variant of the trivia game. The way we set the program up with four separate lists is a little undesirable because all of the related information is scattered among different lists. Let's instead make a structure of sublists within the main list that store the question, answer, value, and used information all in one place. The master list will look like this:

```
data:   [
                [
                 "The possession of more than two sets of chromosomes is termed?",
                 "polyploidy",
                2,
                0
                ],

                [
                 "Age of Amelia Earhart when she disappeared ",
                 "39",
                3,
                0
                ],

                [
                 "Actor whose real name was Marion Morrison ",
                 "john wayne",
                1,
                0
                ],

                [
                 "Study of ancient inscriptions ",
                 "epigraphy",
                2,
                0
                ],

                [
                 "I am the geometric figure most like a lost parrot ",
                 "polygon",
                3,
                0
                ]
        ]
```

The item in data[0] is a list, and the first element of this list is the question, the second is the answer, the third is the value, and the last is whether it is used or not.

Here is a complete modified program to operate on this new list structure. The tradeoff is additional complication in accessing the variables, but now we only have one list instead of four:

```
import javax.swing as swing
 def getInput(message):
  return swing.JOptionPane.showInputDialog(message)

def setupQuestions():
   data = []
   filevar = open(r"c:\trivia.txt","rt")
   line = filevar.readline().strip()  # Read in first line
   numQuestions = int(line)
   for i in range(0,numQuestions):
     question = filevar.readline().strip()
     answer = filevar.readline().strip()
     value = int(filevar.readline())
     questionUsed = 0
     data = data + [[question,answer,value,questionUsed]]
   filevar.close()
   # Return the lists in a list
   return data

def pickRandomQuestion(data):
  # Get a random index into the array
  q = random.randint(0, len(data) - 1)
  # Keep picking until we find one that is unused
  while (data[q][3] <> 0):
    q = random.randint(0, len(data) - 1)
  # Mark this question as being used and return index
  data[q][3] = 1
  return q

import random
def triviaGameRandom():
  score = 0              # Track player's score
  numQuestionsAsked = 0  # Tracks how many questions asked
  data = setupQuestions()
  # Now loop through each question, from 0 to 4
  while (numQuestionsAsked < 4):
     indexRandQuestion = pickRandomQuestion(data)
     guess = getInput(data[indexRandQuestion][0])
     if (guess.lower() == data[indexRandQuestion][1].lower()):
        score = score + data[indexRandQuestion][2]
        printNow("Thats right!  Your score is " + str(score))
     else:
        printNow("That is incorrect.  The correct answer is " +
data[indexRandQuestion][1])
     numQuestionsAsked = numQuestionsAsked + 1
  print "The game is over.  Your total score is " , score
```