COMP 204

For loops, nested loops

Yue Li
based on materials from Mathieu Blanchette, Christopher Cameron and Carlos Oliver Gonzalez

Recap: `for` loop and nested `for` loop

```
for someVariable in someList:
    # body of the loop

#rest of code
```

```
for someVariable1 in someList1:
    # start of body of outer loop

    for someVariable2 in someList2:
        # body of inner loop

    # rest of body of outer loop

# rest of program
```

for loop - example

```
1 L = ["Andrea","Oliver","Chris","Malika"]
2
3 for name in L:
4     print("Dear", name, ", welcome to COMP 204.")
```

Prints:

```
Dear Andrea, welcome to COMP 204.
Dear Oliver, welcome to COMP 204.
Dear Chris, welcome to COMP 204.
Dear Malika, welcome to COMP 204.
```

Equivalent with while loop:

```
1 L = ["Andrea","Oliver","Chris","Malika"]
2
3 index = 0
4 while index < len(L):
5     print("Dear", L[index], ", welcome to COMP 204.")
6     index = index + 1
```

The for loop - example

Goal: Calculate the average, minimum, and maximum of a list of grades.

```
grades = [68,86,77,79,73,89,56,68]

sumGrades = 0  # will keep track of
               # sum of grades seen so far
maxGrade = -1  # will keep track of max grade seen so far
               # Why do we initialize to -1 ?
minGrade = 100 # will keep track of min grade seen so far
               # Why do we initialize to 100 ?

for g in grades:
    sumGrades = sumGrades + g
    if g > maxGrade:
        maxGrade = g
    if g < minGrade:
        minGrade = g

print("Average: ",sumGrades/len(grades))
print("Minimum grade: ", minGrade)
print("Maximum grade: ", maxGrade)
```

What is an iterable?

For loops are used to *iterate over* a sequence, i.e. to visit each element of the sequence, and do some operation for each element

In Python, we say that an object is an **iterable** when your program can *iterate over* it

- Or an **iterable** is an object that represents a sequence of one or more values

All instances of Python's sequence types are iterables

- Lists
- Strings
- Tuples
- More later...

Iterating through a list of tuples

```
1  periodicTable = [ ( "H" , 1 ) , ( "C" ,12) , ( "N" , 14) ]
2
3  # iterating through periodicTable with for loop
4  for element in periodicTable:
5      print(element[0],"has mass",element[1],"g/mol")
6  print()
7
8  # Alternate, better way to do this:
9  for element,mass in periodicTable:
10     print(element,"has mass",mass,"g/mol")
11 print()
12
13 # same thing with while loop
14 index = 0
15 while index < len(periodicTable):
16     element = periodicTable[index]
17     print(element[0], "has mass", element[1],"g/mol")
18     index = index + 1
```

Iterating through a list of lists

Recall from last lecture, when assigning a list to a new variable, we assign the address of the list. This implies that any changes made to the new list variable *will* also modify the original list variable.

```
1  periodicTable = [ ["H" , 1], ["C" ,12] , ["N" , 14] ]
2  # iterating through periodicTable with for loop
3  for element in periodicTable:
4      print(element[0],"has mass",element[1],"g/mol")
5      if element[0]=="H" and element[1]!=1.0004:
6          element[1]=1.0004
7
8  print("\nNew table after update H mass:\n")
9
10 for element in periodicTable:
11     print(element[0],"has mass",element[1],"g/mol")
```

Nested for loops - Iterating through complex lists

Nested for loops are useful to iterate through lists where each item is itself a compound type:

```
moleculesWithNames = [ ("carbon dioxyde", ["C","O","O"]),
                       ("nitrous oxyde", ["N","O"]) ]

for moleculeTuple in moleculesWithNames:
    print("Molecule name:",moleculeTuple[0])
    print("Molecule composition: ")
    for atom in moleculeTuple[1]:
        print(atom)

print("\n####Alternative way to print####\n")

# or an alternate, more elegant way:
for moleculeName,composition in moleculesWithNames:
    print("Molecule name:",moleculeName)
    print("Molecule composition: ")
    for atom in composition:
        print(atom)
```

Advanced topic - Enumerate function

Often, it is useful to iterate through a sequence while keeping track of the index of the item we're looking at. This can be done with the enumerate function.

```
countriesSorted=[("China",140951739),
                 ("India",1339180127),
                 ("United States", 324459463),
                 ("Indonesia",263991379)]

for rank,countryTuple in enumerate(countriesSorted):
    name = countryTuple[0]
    population = countryTuple[1]
    print(name,"has population", population)
    if rank == 0:
        print(name," is the most populous country")
    if rank == len(countriesSorted)-1:
        print(name," is the least populous country")
```

Revisit molecule example

```
# Correcting atom name based on the molecule names:
moleculesWithNames = [ ("carbon dioxyde", ["X","O","O"]),
                       ("nitrous oxyde", ["Y","O"]) ]

for moleculeName,composition in moleculesWithNames:
    for index,atom in enumerate(composition):
        if moleculeName=="carbon dioxyde" and atom != "C"
    and index==0:
            composition[index] = "C"
        elif moleculeName=="nitrous oxyde" and atom != "N"
    and index==0:
            composition[index] = "N"

for moleculeName,composition in moleculesWithNames:
    print("Molecule name:",moleculeName)
    print("Molecule composition: ")
    for index,atom in enumerate(composition):
        print(composition[index])
```

Nested for loops: Diameter of a molecule

Goal: Given information about the 3D structure of a molecule (in the form of xyz coordinates for each of the atoms), find the diameter of the molecule, i.e. the distance between the two most distant atoms in the molecule.

Data: a list tuples, where each tuple contains a string describing the atom and a tuple containing the xyz coordinates of that atom.

Algorithm: Use nested for loops to over each pair of atoms in the molecule. For each firstAtom

- for each secondAtom
  - Calculate distance between first and second atom
  - If distance is larger than max distance seen so far, update max distance seen so far.
- Report max distance found

Nested for loops: Diameter of a molecule

```
import math
aceticAcid = [ ("C", (0.053,    0.206,    0.000 )),
("O", (0.053,    1.576,    0.000 )),
("O", (1.274,   -0.499,    0.000)),
("C", (-1.269,   -0.557,    0.000 )),
("H", (1.840,   -0.238,    0.882 )),
("H", (-1.840,   -0.294,   -0.890 )),
("H", (-1.840,   -0.294,    0.890 )),
("H", (-1.070,   -1.629,    0.000 )) ]

maxDistance = -1
for firstAtom, firstXYZ in aceticAcid:
    for secondAtom, secondXYZ in aceticAcid:
        distance = math.sqrt((firstXYZ[0]-secondXYZ[0])**2 +
                             (firstXYZ[1]-secondXYZ[1])**2 +
                             (firstXYZ[2]-secondXYZ[2])**2)
        if distance > maxDistance:
            maxDistance = distance

print("Diameter of molecule is",maxDistance," Angstrom")
```