EEB 177 Lecture 10

Thursday Feb 10th, 2017

Topics

- for, if, else, while statements
- in class challenge!

Office Hours

Tues 11-12

Wednesday 10-11

Hacking Sessions

Tuesday from 6-8:30 in LS 3209

Homework #5

Due Wednesday by 5:00 PM

Projects Deadline #2

Due Friday by 5:00 PM. To complete this assignment:

- Add a README to your Git repository
- Add two files of pseudocode for project. One should describe file input-output. The other should describe data manipulation over a loop.
- due Friday Week 5: Feb 10th
  (4%)

collections recap

```
# round brackets --> tuple
In [1]: type((1, 2))
Out[1]: tuple
# square brackets --> list
In [2]: type([1, 2])
Out[2]: list
# curly brackets, sequence of values --> set
In [3]: type({1, 2})
Out[3]: set
# curly brackets, key-value pairs --> dictionary
In [4]: type({1: "a", 2: "b"})
Out[4]: dict
```

for

for loops

imagine you wanted to print out each element of this list

```
apes = ["Homo", "Pan", "Gorilla"]
```

you could print these out seperately

```
print(apes[0] + " is an ape")
print(apes[1] + " is an ape")
print(apes[2] + " is an ape")
```

A better way is to use the for loop syntax

```
apes = ["Homo sapiens", "Pan troglodytes", "Gorilla goril
for ape in apes:
    print(ape + " is an ape")
```

for loops have the following structure:

```
for x in y:
    do something
```

y is a list (or list-like object)

x is a variable names

the colon sets off an indented block of code (the body of the loop)

a more complex loop

```
apes = ["Homo", "Pan", "Gorilla"]
for ape in apes:
    name_length = len(ape)
    first_letter = ape[0]
    print(ape + " is an ape. Its name starts with " + firs
    print("Its name has " + str(name_length) + " letters"
```

the range function

```
for number in range(6):
    print(number)
```

more on range

With two numbers, range will count up from the 1st number (inclusive) to the second (exclusive):

```
for number in range(3, 8):
    print(number)
```

range and step

With three numbers, range will count up from the 1st to the second with the step size given by the third:

```
for number in range(2, 14, 4):
    print(number)
```

while loops

a while loop runs until some condition is met.

```
count = 0
while count<10:
    print(count)
    count = count + 1
```

program flow in python

In its simplest form, a program is just a series of instructions (statements) that the computer executes one after the other. In Python, each statement occupies one line (i.e., it is terminated by a newline character). Other programming languages use special characters to terminate statements (e.g., ; is used in C).

Lets demonstrate statements that control flow in python with a simple program.

conditional tests

A condition is simply a bit of code that can produce a true or false answer.

```
print(3 == 5)
print(3 > 5)
print(3 <=5)
print(len("ATGC") > 5)
print("GAATTC".count("T") > 1)
print("ATGCTT".startswith("ATG"))
print("ATGCTT".endswith("TTT"))
print("ATGCTT".isupper())
print("ATGCTT".islower())
print("V" in ["V", "W", "L"])
```

we use conditional tests to control the flow of our program

```
expression_level = 125
if expression_level > 100:
   print("gene is highly expressed")
```

if, elif, else

if and else create branching points in your program resulting in the execution of different blocks of code depending on a condition.

```
x=4
if x % 2 == 0:
    print("Divisible by 2") #body of loop
```

note indentation to designate loop body

We can use else to specify an action when a condition is not met.

```
x=4
if x % 2 == 0:
    print("Divisible by 2")
else:
    print("Not divisible by 2")
```

```
expression_level = 125
if expression_level > 100:
   print("gene is highly expressed")
else:
    print("gene is lowly expressed")
```

If we have multiple cases that need to be evaluated, elif is useful...

```
x = 17
if x % 2 == 0:
    print("Divisible by 2")
elif x % 3 == 0:
    print("Divisible by 3")
elif x % 5 == 0:
    print("Divisible by 5")
elif x % 7 == 0:
    print("Divisible by 7")
else:
    print("Not divisible by 2, 3, 5, 7")
```

Challenge

change the program below so that it reports the temperature in Fahrenheit or Celsius depending on user input.

```
xx = input("What is the temperature in Fahrenheit?")
yy = (float(xx) - 32) * 5 / 9
print("The temperature in Celsius is {}".format(yy)
```

if, elif, and else part II

we can handle complex flow with these statements.

```
file1 = open("one.txt", "w")
file2 = open("two.txt", "w")
accs = ['ab56', 'bh84', 'hv76', 'ay93', 'ap97', 'bd72']
for accession in accs:
        if accession.startswith('a'):
                file1.write(accession + "\n")
        else:
                file2.write(accession + "\n")
```

use and, or and not to specify complex conditionals

```
accs = ['ab56', 'bh84', 'hv76', 'ay93', 'ap97', 'bd72']
for accession in accs:
        if accession.startswith('a') and accession.endswi
                print(accession)
```

```
accs = ['ab56', 'bh84', 'hv76', 'ay93', 'ap97', 'bd72']
for accession in accs:
        if accession.startswith('a') or accession.startsw
                print(accession)
```

```
accs = ['ab56', 'bh84', 'hv76', 'ay93', 'ap97', 'bd72']
for acc in accs:
        if acc.startswith('a') and not acc.endswith('6'):
                print(acc)
```

```
file1 = open("one.txt", "w")
file2 = open("two.txt", "w")
file3 = open("three.txt", "w")
accs = ['ab56', 'bh84', 'hv76', 'ay93', 'ap97', 'bd72']
for accession in accs:
        if accession.startswith('a'):
                file1.write(accession + "\n")
        elif accession.startswith('b'):
                file2.write(accession + "\n")
        else:
                file3.write(accession + "\n")
```

Reading and writing files

To work with text file contents you need to first open the file and then read the contents in as a string

```
# open the file
my_file = open("dna.txt")
# read the contents
my_dna = my_file.read()
# calculate the length
dna_length = len(my_dna)
# print the output
print("sequence is " + my_dna +  " and length is " + str(
```

In Class Challenge