

Introduction to programming, Lesson 2 : Lists and functions

Warm-up : variables, cycles, if

1. Does this script print out « She said, "Hello world" » ?

```
prin("She said, "Hello world")
```

2. Change this script so that it prints out integers 0, 2, 4.

```
i = 0
while i < 5:
    i = i + 1
    if i % 2 = 0:
        print i
```

3. Does this script stop? Change it so that it prints out numbers from 0 to 4.

```
firstnumber = 0
secondnumber = 2
while firstnumber < 5:
    print firstnumber
    secondnumber = secondnumber + 1
```

4. Which result do you expect? How to fix this script?

```
#V = Voltage in volts, I = Current in amps, R = Resistance in ohms
#Ohm's Law: V = I R
V = 5
I = 2
print "R = ", V/I
```

New: else and elif, lists, functions

1. A condition is a piece of text that can produce two answers, True and False. The best method to understand how to conditions work is to try several examples.

```
print (3 == 5)
print (3 < 5)
print ("banana" == "banana")
print ("banana" == "banana" and 3 < 5)
print ("banana" == "banana" and 3 == 5)
print ("banana" == "banana" or 3 == 5)
```

2. The most simple conditional instruction is if. Using if, we can make a simple yes/no decision: if the answer is True, do something. The clause else is closely related to if. Often, we need an instruction of either/or type, where we have two possible actions to take. To make it possible, we can add the clause else right after the body of the if clause.

```
expression_level = 125
if expression_level > 100:
```

Introduction to programming, Lesson 2 : Lists and functions

```
print("gene is highly expressed")
else :
    print("gene is lowly expressed")
```

3. Write a script that prints out “10 has remainder 1 modulo 3, 11 has remainder 2 modulo 3, 12 has remainder 0 modulo 3, ..., 20 has remainder 2 modulo 3”. Use if/else and while.
4. Given three integers entered by a user from a keyboard, print them in the increasing order. To enter an integer you can use:

```
a = raw_input("Enter a: ")
```

5. Functions is a practical way to divide you code into blocks to make it more readable, and also to reuse some parts of your code and save time. Functions are defined by a keyword “def” followed by the name of the function.

```
def my_function():
    print("Hello From My Function!")
```

Functions can take arguments.

```
def my_function_with_args(username, greeting):
    print("Hello, %s , From My Function!, I wish you %s"%(username, greeting))
```

Functions can return a value to a caller of the function.

```
def sum_two_numbers(a, b):
    return a + b
```

To call a function write its name followed by () and arguments if needed.

```
c = sum_two_numbers(3, 4)
```

6. Write a function that receives the voltage V and the current I and returns the resistance R. Compute the resistance for V = 5, I = 2 ; V = 1, I = 0.2 ; V = 3, I = 0.5 ; V = 0.4, I = 5.
7. Similar to string, which is a sequence of characters, sometimes we need a sequence of int or a sequence of something else. In Python, sequences are implemented using lists.

```
animals = ['giraffe', 'lion', 'monkey', 'cat']
taille = [5.0, 1.0, 0.7, 2.0]
mix = ['giraffe', 5.0, 'monkey', 2]
```

Lists can be also created by using a function append.

```
animals = []
animals.append('giraffe')
animals.append('lion')
animals.append('monkey')
animals.append('cat')
```

We can call elements of a list by using indexes. Indexes of a list of length n change from 0 to n-1.

```
print animals[0], animals[1], animals[-1]
```

len gives you the length of a list.

```
print len(animals)
```

We can use for to iterate over elements of a list.

Introduction to programming, Lesson 2 : Lists and functions

```
l = ['a', 'b', 'c', 'd']
for element in l:
    # Do something with the element
```

8. Write a function that takes a list and returns True if it contains a duplicate.
9. You can consider a string as a list of its characters. Write a script that prints out all characters of a string, one per line.

Harder exercises

1. Cryptography

- Create a string 'this is my message to encode'. Using a cycle, encode the string with a shift of 3 (Caesar cipher). For example, letter a will be encoded by a letter d, and x by a.

You can convert a letter to an integer using `ord(letter)`, and back by `chr(my_integer)`. More information : <http://www.asciitable.com/>

- Write a script that encodes (or decodes, according to the choice of a user) a string entered from the keyboard using a key which is also chosen by a user. We define the functions as `encode` and `decode`. Be careful, the message can contain non-alphabetic characters that you must not change. You can verify whether a character `c` is a letter using `c.isalpha()`.

```
$ python caesar_cipher.py
Enter a text to encode: ceci est un autre message a chiffrer
Enter a shift (0 to 26): 12
Enter 'e' to encode or 'd' to decode: e
oqou qef gz mgfdq yqeemsq m oturrdqd

$ python caesar_.py
Enter a text to encode: oqou qef gz mgfdq yqeemsq m oturrdqd
Enter a shift (0 to 26): 12
Enter 'e' to encode or 'd' to decode: d
ceci est un autre message a chiffrer
```

- Modify the previous script so that it implements the Vigenere cipher: the key is a string and we encode the text by shifting the *i*-th character using the *i*-th character of the message (we restart from the beginning of the key after having read it), where A=1, B=2, ... , Z=0.
- Modify the previous script so that it encodes/decodes the file the path to which is given by a user. You can use the following novel of Edgar Poe to test the script: <http://bit.ly/2Cvpf15>. To obtain the content of the file given its path you can use

```
nom_fichier = "Poe.txt"
contenu_fichier = open(nom_fichier).read()
```

2. Cryptanalysis

As explained in the novel of E. Poe, encodings by shift can be easily decoded by frequency analysis. For example, if the text is written in French, the most frequent letter is "E" and since Caesar cipher does not change the frequencies, the most frequent letter in the encoded text must be the encoding of the letter "E". Therefore, the difference between the encoding of "E" and "E" can give us the key of the encoding.

Introduction to programming, Lesson 2 : Lists and functions

- Write a script that computes frequencies of the characters in the text.
- Modify the script so that it prints out the most frequent character.
- Use this function to write another function that receives a text encoded by Caesar cipher (but not the key) and returns the original text.