Dealing with data with plain Python - exercises

|  |  |
| --- | --- |
| http://www.clker.com/cliparts/M/Q/8/m/N/Z/warning-sign-hi.png | **NOTE**: Code in this document has been tested with Python 2.7.7. However, it may work also with previous and newer versions. |

Tabla de contenido

Dealing with data with plain Python - exercises 1

1. Numbers 2

Examples 2

Proposed exercises 2

2. Strings 2

Proposed exercises 2

3. Functions 3

Proposed exercises 3

3. Lists and loops 3

Examples 3

Proposed exercises 3

# 1. Numbers

## Examples

You can use “j” with numbers to specify the imaginary part of complex numbers.

>>> c = 4j

>>> type(c)

<type 'complex'>

>>>

And then mix it with real parts in a very simple way:

>>> c = c + 25

>>> c

(25+4j)

>>> c.imag

4.0

>>> c.real

25.0

>>> c.conjugate()

(25-4j)

The standard module cmath contains functions to deal with complex numbers.

There is a lot of functions for random numbers and randomizing in general in Python. For example, to generate random even numbers:

>>> for i in range(1,100):

... print randrange(1,100, 2)

...

67

71

97

…

## Proposed exercises

1. Search for the divmod built-in function and find out what it does.
2. Given a complex number in Python, find the way to get its absolute value and to convert it to polar coordinates.
3. Find a way in the random module to generate random integers and use it. Is it similar to randrange()?

# 2. Strings

## Examples

You can define multi-line string with triple quotation marks. There are many methods that you can call on strings. For example to get the position of a word in a text:

>>> st1 = """

... The red lights on the Red Night were more red than the most beautiful red.

... Red is the future of reds.

... """

>>> st1

'\nThe red lights on the Red Night were more red than the most beautiful red.\nRed is the future of reds.\n'

>>> st1.upper()

'\nTHE RED LIGHTS ON THE RED NIGHT WERE MORE RED THAN THE MOST BEAUTIFUL RED.\nRED IS THE FUTURE OF REDS.\n'

>>> st1.upper().find("RED")

5

And then if you want to create a new string substituting that string:

st2 = st1[0:st1.upper().find("RED")]+st1[st1.upper().find("RED")+len("RED"):]

>>> st2

'\nThe lights on the Red Night were more red than the most beautiful red.\nRed is the future of reds.\n'

## Proposed exercises

1. Write a program to replace all the occurrences of “red” (in capitals or not) with another colour.
2. Now do the same with the replace method.
3. Take a string with a fragment of text from a book. Split the words using the split method.
4. Take a string with the fragment of text starting Don Quixote. Find all the words that start with “m” or “l”.

# 3. Functions

## Proposed exercises

1. Implement as a function the Commodity Channel Index: <http://www.investopedia.com/terms/c/commoditychannelindex.asp> Test the function with different calls.
2. Create a function that takes a tweet and returns a list with the hashtags used as a list.

# 3. Lists

## Examples

Python has a compact syntax for creating lists, called list comprehensions.

>>> [i\*\*2 for i in range(10)]

[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]

This is useful to make code more compact when selecting or transforming lists.

## Proposed exercises

1. Find a function in the random module to aleatorize the position of elements in a list, i.e. doing permutations of them.
2. Fill a list of 100 variables from a gamma distribution. You will probably find a way of getting random values following a gamma in the random module.
3. Implement with list comprehensions a function that returns a list with “Pass” or “Fail” strings taking as input a list of scores of students in a class.
4. Implement a random Rock-paper-scissors game (<http://en.wikipedia.org/wiki/Rock-paper-scissors>). Use a list representing the three elements (and do exercise the choice() method from the random module) and a function try() that determines if player A or player B wins based on the rules of the game.
5. Implement the “guess a number” game: <http://www.funbrain.com/guess/>
6. Create a list of lists of integers, and initialize them using random integers in the range 1 to 20. Compute the mean, mode and mean of these integers using functions.
7. Implement with functions the computation of the Money Flow Index(MFI): <http://www.investopedia.com/terms/m/mfi.asp>