

Basic Python workshop

Noufal Ibrahim

12 May 2018

Contents

1	Introduction	2
2	Functions	2
2.1	Writing our own functions	2
3	Conditionals	3
4	More types	3
4.1	Looping over lists	3
5	Methods	3
6	User input	4
7	Standard library	4
8	Indefinite looping using while	4
9	Exercises	4
10	Creating our own modules	5
11	More Exercises	5
12	Introducing boolean types	6
13	Exercises again!	6
14	Files	6

15 Exercise	7
16 Dictionaries	7
17 Exercise	7
18 References	7
18.1 Practice problems	7
--org--	

1 Introduction

- We'll be using python3
- typing `python` and hitting enter at the command prompt will start the interpreter.
- You can quit the interpreter using Ctrl-D (linux) or Ctrl-Z enter(windows)
- REPL - Read Eval Print Loop

2 Functions

```
print ("Hello, world!")
```

- `print` is the function name
- `()` is the function call operator
- `"Hello, world"` is an argument

2.1 Writing our own functions

- Use the `def` keyword
- Indent properly
- Difference between tabs and spaces. (**watch out for mistakes here**)
- Functions can return values using the `return` keyword
- **Exercise:** Write a function `avg` which takes 3 numbers as arguments and returns their average.

3 Conditionals

- The `if` statement can be used for conditional branching
- The `else` clause can be used for code if the condition is false.
- Multiple branches can be done using the `elif` statement

4 More types

- We've seen string, integers and floats
- lists
 - Created using the `[]` operator
 - The size can be obtained using the `len` builtin function

4.1 Looping over lists

- The `for` statement
- Can loop over lists
- The `range` function can be used to create a range of numbers to loop over.
- Can loop over any iterable (we've already seen strings. These are iterable).
- **Exercise:** Write a function `maximum` which takes a list of numbers as an argument and returns the largest number in the list.

5 Methods

- Methods are functions attached to objects (e.g. the `format` method of a string)
- Examples of methods of strings - `upper`, `lower`, `title`, `capitalize`, `center`, `ljust` etc.
- Examples of methods of lists - `append`, `extend`, `pop`, `sort`, `reverse` etc.

6 User input

- You can use the `input` function to get input from the user.
- e.g. `t = input("Enter your name ")`

7 Standard library

- We use the `import` statement to use **modules** from the standard library

8 Indefinite looping using `while`

- The `while` statment can be used to loop as long a condition is True

9 Exercises

- Write a function `c2f` that when given a temperature in degrees Centigrade, returns it in Farenheit.

$$F = 9 \cdot C / 5 + 32$$

- Write a function called `temp_tables` which will print a table of temperatures from 0 to 100 degrees centigrade and the corresponding temperature in Farenheit in increments of 10.

0 32 10 . 20 . 30 . 40 . . . 90 100

Use the function you wrote above inside this.

- Write a function called `fizzbizz` which takes a single argument `n` and prints numbers from 1 to `n` subject to the following rules.
 - If the number is divisible by 3, print fizz
 - If the number is divisible by 5, print bizz
 - If the number is divisible by 15, print fizzbizz
 - Otherwise, just print the number

So, `fizzbizz(20)` will print

1 2 fizz 4 bizz fizz 7 8 fizz bizz 11 fizz 13 14 fizzbizz 16 17 fizz 19 bizz

10 Creating our own modules

- A file with a `.py` extension with python code in it is a "module". It can be `=import=`ed by your interpreter and functions inside it can be called.
- Modules are not reloaded if you import them again. You have to quit the interpreter and restart it.

11 More Exercises

- Write a function called `num_digits` which will return the number of digits in a given number.
- Write a function called `num_words` which will return the number of words in a string.
- Write a function called `num_sentences` which will return the number of sentences in a given paragraph.
- Write a function called `longest_word` which will return the longest word in a sentence.
- (Bonus) Write a function called `abbreviate` which will take a string as input and abbreviate it according to the following rule. It will take the first letter of all words in the string that start with a capital letter. e.g. "Indian Institute of Management" will be abbreviated to "IIM" (the "of" will be skipped since it doesn't start with a capital letter).

Hint: You can use `x[0]`, `x[1]` etc. to refer to individual letters of a string. Also, strings have a `.isupper()` method to check whether they are uppercase or not and a `.istitle()` method to check if they are title cased or not.

- (Bonus) Write a function called `sq_tables` which takes a number `n` as input and prints the tables in a square format from `1x1` to `nxn` e.g. `sq_tables(5)`

```
1 2 3 4 5 2 4 6 8 10 3 6 9 12 15 4 8 12 16 20 5 10 15 20 25
```

The value in each cell will be the product of the row and column number.

12 Introducing boolean types

- There are two literal booleans in PYthon **True** and **False**. Comparison operators return this. These can be combined with logical **and**, **or**, and inverted with **not**.

13 Exercises again!

- A panagram is a sentence which contains all letters of the alphabet. e.g. "sphinx of black quartz, judge my vow".

Write a function called **panagram** which will take a sentence as input and then return **True** if the sentence is a panagram, **False** if not.

- Try to estimate the value of pi using a monte carlo simulation.
set darts inside circle to 0
for n iterations select a random point
if point inside circle, increment darts inside circle by 1
estimate will be ((darts inside circle) / iterations) * 4

14 Files

- Files can be opened using the **open** function. It returns a **file** type.
- `=file=s` have several methods like
 - **.read(n)** - To read **n** bytes of data out of the file
 - **.tell()** - To get information on where the internal file pointer is
 - **.seek(n)** - Moves the file pointer to position **n**
 - Files can be looped over using **for**
 - **.close()** - To close a file
- They can also be opened for writing. This is done by passing **"w"** to the **open** function as a second argument (e.g. **open("sample.txt", "w")**)
 - These have a usable **.write()** method which can be used to write data into the file.

15 Exercise

Write a function called `ari` which will take a filename as an argument. It will open the file and then use the formula described at https://en.wikipedia.org/wiki/Automated_readability_index to calculate the ARI for the contents of the file.

16 Dictionaries

- Dictionaries are hash tables. They are used to hold key, value pairs. Given a key, you can get the value out.
- They are constructed using the `{}` operator (similar to `[]` for lists).
- e.g. `d = {"name" : "Noufal", "place" : "Kozhikode"} d['name']`
Will return "Noufal"

17 Exercise

Write a function called `freq` which will take a string `s` as input and return a dictionary with the counts of each letter in the input.

e.g. `freq("abbaac")` will return `{"a" : 3, "b" : 2, "c" : 1}`
don't use the `.count()` method of strings to do this

- Logic: initialise an empty table for each letter in the input string if the letter is not in the table Add it as a key to table with value 1 else if it is in the table Increment the count by 1 return the table

18 References

- <http://learnpythonthehardway.org/>
- <https://docs.python.org/3/tutorial/index.html>
- Udacity Python course

18.1 Practice problems

- <https://www.hackerrank.com/>
- <http://projecteuler.net/>