

Introduction to programming

Reusing code with functions and modules

Exercise 1 – encrypted_login.py

You have been asked to update your login program that you have been refining over the previous weeks tasks. Passwords entered in to the login program need to be encrypted using a [substitution cipher](#). It has been suggested to create a function which will encrypt the user's password when they enter it.

Create a function which takes a string and then encrypts it with a simple cipher such as [ROT-13](#) (where letters in the alphabet get shifted 13 places) or another simple cipher if you'd prefer. You might want to print the encrypted password out to test that the function works correctly. Save your file as **encrypted_login.py** in your Week 6 folder.

Exercise 2 – position_in_alphabet.py

You are working on some text analysis software that examines text for certain patterns and letters. Write a function that takes a single letter and returns a value that represents the position it occurs in the alphabet. For example

- a = 1
- b = 2
- ...
- z = 26

Test that your program returns the correct values for some example letters and save your file as **position_in_alphabet.py** in your Week 6 folder.

Exercise 3 – average_length.py

You have been asked to create an additional function for the text analysis software mentioned earlier. The function needs to take a list of words, determine the length of each word and then return the average length of the list.

So for example with the words: **The, quick, brown, fox** the average length would be **4**.

Test your program with some different lists and then save the program as **average_length.py** in your Week 6 folder.

Exercise 4 – last_word.py

Another part of the text analysis software requires a function that returns the last word of a sentence. Write a function that takes a string argument and returns the last word of the sentence.

Test your program and save it in as **last_word.py** in your Week 6 folder.

Extension: add a second argument to the function that specifies which word in the sentence should be returned.

Exercise 5

In order to hand over your code to the rest of the team that are working on the text analysis software you have been asked to put all of your functions in to a module so that it can be easily imported in to the application.

Create a module that includes all of the functions for the text analysis software above (and any other that you feel are relevant) and then check that this is accessible by importing into a test program and trying some of the function. You can label your module and test file however you like and save these in your Week 6 folder.

Extension task – document_date.py

In our text analysis software, there are dates included in the documents. These are included as the software analyses the quality of the document depending on what time of the day they were written.

As part of the software, there needs to be a function that can read in the date of the document and then return a result that represents whether the document was created in the morning, afternoon or evening.

For example in the text below the function should return **morning** or something similar.

Title: New Document

Document time: Tue Jan 13 10:17:09 2015

Sustainable taxidermy iPhone, kitsch ennui tilde beard kogi photo booth skateboard shabby chic bitters cray. Gluten-free dreamcatcher flexitarian, leggings migas DIY narwhal fixie food truck shoreditch pour-over. +1 health goth 8-bit, tote bag pork belly intelligentsia meh post-ironic polaroid try-hard austin church-key. Pug hammock four dollar toast cred, raw denim pitchfork distillery.

