Week 9 - Assessed Exercises

Data Programming with Python

This week we learnt about regular expressions and manipulating strings. In this assessed exercise you will create regular expressions to extract particular parts of a string or set of strings. You may find it helpful to go through the non-assessed exercises first as they contain some similar questions

Each question asks you to write a function with a specific set of input arguments. The *.py* template defines the function name and inputs for each question, **do not** change these. Be sure you test your functions before you submit your code to make sure that they are outputting the correct answer. Unless otherwise stated, all functions must have a return value.

Include the import statements for all packages used within your code. Additionally, please include the package prefixes (`pd`, `np`, etc.) for functions/methods from these packages, even if the command runs in Canopy without the prefix. .

1. Suppose I want to generate a password of length n using a random combination of all of the letters (a-z). Write a function that takes n and a seed value s as inputs and returns a string containing the password. Now all you have to remember for your password is the seed value you used to create it.

2. Suppose I have a list of phone number and I wish to extract the area codes of each number. Write a function that takes a list of phone numbers as input and extracts the area code (assuming that the area code is enclosed in parentheses, e.g. the area code for (08) 03 49 98, would be 08).

3. I have a list of strings consisting of email addresses and I want to find the domains (the part after the @). Write a function that extracts the characters after the @ sign for each email address and returns them as a list.

4. I have a list of strings, each of which contains an email address. Write a function that finds and returns all of the email addresses in a given list of strings. You may assume that all email addresses consist of a set of characters (from a-z) and digits (from 0-9), followed by an @ symbol, followed by another set of characters, followed by a full stop and finally a third set of characters (none of the email addresses will have special characters, such as ? and !).

All of your code should be written into the *.py* template. Save your filled *.py* file with the following name structure *SurnameFirstname_Week9.py* (where *Surname* and *Firstname* should be replaced with your name) and upload it to Brightspace. You must also upload a PDF of your code.