National College of Ireland

Database and Analytics Programming

Tutorial 2 - Python Skills

Prepared by Noel Cosgrave *Associate Faculty, National College of Ireland*

This tutorial is designed to develop and test your skills with the Python language. We will explore some basic Python data structures and programming constructs, including lists, tuples, sets, dictionaries, loops and functions.

Keywords: Python, Data Types, Loops, Functions

Introduction

To get started, open Anaconda Navigator and launch Jupyter Notebook. Create a new notebook and start working on the tasks below. Use a new notebook cell for each task.

Task 1 - Lists

1. Create a list to store the range of values between 10 and 25, assign it to a variable called *my_list*.
2. Print the contents *my_list*.
3. Print the contents of *my_list* in reverse using index slicing.
4. Print the contents of *my_list* in reverse using a loop.
5. Append the following list [11, 19, 26] to *my_list*. Note the difference between the *extend* and *append* methods.
6. Extend *my_list* with 10 random integers.
7. Store the last number in the list in a variable called *last_val* and remove that value from *my_list*.
8. Convert *my_list* to a set named *my_set*. Try to extract the second element in the set using indexing. Does it work? If not, why not?
9. Convert *my_list* to a tuple named *my_tuple*. Assign the value 5 to the fourth element in the tuple? Does it work? If not, why not?

Task 2 - Dictionaries

1. Create an empty dictionary object called *my_dict*.
2. Using a loop construct, add an entry to *my_dict* for each value in *my_list*. The keys for entries in *my_dict* should be the English-language representation of integer values in *my_list*.

   Note: to convert numbers to their English-language representation, you will need to install a module named inflect. You can do this by adding the following code to a cell in your notebook:

   ```
   !conda install -y inflect
   ```

Task 3 - Loops

1. Read the linked Wikipedia article about Pascal's Triangle.
2. Without accessing one of the Python implementations available on the web, write a piece of code to generate the first 8 rows of Pascal's triangle.
3. For this task you should implement the necessary loop constructs and use appropriate data-structures.
4. When printing each row of the triangle, ensure that values are separated by a tab character.

Task 4 - Functions

1. Investigate the chr() and randint() functions.
2. Create a function that takes one argument, the length of a password to generate.
3. Your function should generate and return a password of the required length, ensuring that you use only characters with decimal values between 33 and 126 in the ASCII table . Note that some characters in the table are non-printing!