# Data Structures and Algorithms with Python

## Homework 1

**Instructions**. Download the Python file `hw1.py` in the homework directory on the Hub. Complete the functions in file following the instructions below and in the file. Submit the file using the Hub. Your submission filename should be exactly the same as the current filename, that is `hw1.py`.

The homework is intended as practice for your skills in working with the basic elements of Python, including variables, types, conditionals, and functions. The last part of the exercise also asks you to deal with errors and bad user input.

The homework is graded mainly on the correctness of your functions for different input values, but you may receive extra credit for clean and expressive code. The homework is worth 15% of your total course grade. Each of the functions below that works correctly for at least some inputs will earn you partial credit.

You can discuss approaches to solving the problem with your classmates but please complete the code for the homework individually. It is due on 20/9 at 10am.

**Your task: string conversions.** There are a number of ways dates are expressed as strings, which can sometimes lead to confusion. For example, `6/12/85; 6 December, 1985; 6DEC85; 06-12-1985; 06.12.85` all refer to the same date.[1] Your task is to write a Python script that converts dates from one format to another. Specifically, your program will take a date in 'slash' format (6/12/85) and convert it to 'dash' format (06-12-1985).

The formats follow the following rules.

Strings in the slash format contain slashes and digits in the form `day/month/year`. The day-part and the month-part may contain either one or two digits depending on the date. The year-part contains either two or four digits representing the year (6/12/85 or 6/12/1985).

Strings in the dash format similarly contain dashes and digits in the form `day-month-year`. The day-part and the month-part always contain exactly two digits. The year-part contains exactly four digits representing the year.

Complete the following five functions:

[1] Furthermore, different countries have different conventions on the order of months, days and years, which we'll ignore here...

- `dateConversion` (**3 points**): takes as input a string in slash format and returns a string in dash format. You may assume that the input is always a valid date. Do not use any Python libraries for dealing with dates. Instead, your implementation should use the following three helper functions:

- `convertDay` (**2 points**): converts the day-part of a slash-format string into the day-part of a dash format string.

- `convertMonth` (**2 points**): converts the month-part of a slash-format string the month-part of a dash format string.

- `convertYear` (**2 points**): converts the year-part of a slash-format string into the year-part of a dash format string. Assume that any two-digit input years are from the past century (1917 - 2016).

- `dateConversionRobust` (**6 points**): once you have the date conversion working for valid dates, complete a robust version of it. In the event of a non-valid date, it should print out the string `Not a valid date` and return the value `None`. Otherwise it should return the converted dash date. Note that there are several possible things that could be wrong with the input string...

You will most likely need at least type conversions, conditional statements, and string concatenation. You may want to start with the simpler partial conversion functions and then combine their outputs in `dateConversion`. In the last part, you may want to consider a `try...except` structure. Be careful that your code returns the right type of object as specified in the functions.

You can test your implementation with the provided functions:

- `testDateConversion()`

- `testDateConversionRobust()`

Extra questions to think about (not graded): A slightly more challenging task would be identifying a date format from a string (and then perhaps changing it). How you would achieve this given the different date formats above? How would you parse a long document to find all dates in any format within it?