Regular Expressions

Application: Date Format Converter

***Make sure that you have read and understood***

* ***module week4***
* ***Text pp. 1405-1427***
* ***Helpful Python3 tutorial links:***
  + [*re - Regular Expression Operations*.](https://docs.python.org/3/library/re.html)
  + [*Regular Expression HOWTO*.](https://docs.python.org/3/howto/regex.html)
  + [*Text Processing Services*.](https://docs.python.org/3/library/text.html)
  + [*Built-in Exceptions*.](https://docs.python.org/3/library/exceptions.html)

before submitting this assignment.  Hand in only one submission.

**Format Conversion**

Often, in working with big data, the task of converting document formats to alternate formats either for data storage and retrieval or document presentation presents itself.

**Understand the Application**

For this lab, the task at hand is to create a date format converter.  Your program will convert a date in the format “mm/dd/yyyy” to the format “month day, year”.

**The Program Spec**

Prompt the user to obtain a date from the user.  Specify the required input format: **mm/dd/yyyy**  Use a regular expression to validate the user input date format.  If the format is incorrect raise a **SystemExit**.

Split the input string into respective month, day, and year components.  Using a list to hold the month format as a string, convert the month input to the correct string month name.  You will need to calculate an appropriate index to retrieve the correct month name from the month list.

Use the Gregorian calendar for valid **dd**:

**Note:**For February dd can be 28 (i.e. do not need to factor in leap years).

The 12 Months

The [Gregorian calendar.](https://www.timeanddate.com/calendar/gregorian-calendar.html) consists of the following 12 months:

1. [January.](https://www.timeanddate.com/calendar/months/january.html) - 31 days
2. [February.](https://www.timeanddate.com/calendar/months/february.html) - 28 days in a [common year.](https://www.timeanddate.com/date/common-year.html) and 29 days in [leap years.](https://www.timeanddate.com/date/leapyear.html)
3. [March.](https://www.timeanddate.com/calendar/months/march.html) - 31 days
4. [April.](https://www.timeanddate.com/calendar/months/april.html) - 30 days
5. [May.](https://www.timeanddate.com/calendar/months/may.html) - 31 days
6. [June.](https://www.timeanddate.com/calendar/months/june.html) - 30 days
7. [July.](https://www.timeanddate.com/calendar/months/july.html) - 31 days
8. [August.](https://www.timeanddate.com/calendar/months/august.html) - 31 days
9. [September.](https://www.timeanddate.com/calendar/months/september.html) - 30 days
10. [October.](https://www.timeanddate.com/calendar/months/october.html) - 31 days
11. [November.](https://www.timeanddate.com/calendar/months/november.html) - 30 days
12. [December.](https://www.timeanddate.com/calendar/months/december.html) - 31 days

Below is an example conversion:

Enter a date (mm/dd/yyyy): 01/01/2018  
The converted date is: January 01, 2018

**Testing Requirements:**  Use a loop construct to display 5 date conversions.  The 1st 4 test cases should consist of valid user input.  The last 5thtest case should consist of invalid user input.

Here are some tips and REQUIREMENTS:

1.    A list is used to store the month string names.

2.    An index is calculated to retrieve the correct month from the list based on the user input.

3.    A user prompt instructs the user of the expected input date format: mm/dd/yyyy

4.    Obtain and validate user input. Display an error message and raise a SystemExit if invalid.

5.    The converted date output is in the format: month day, year

6.    The output display: month is spelled out, the day is 2 digits dd and the year is 4 digits dddd.

7.    Use named constants instead of literal values (i.e. NUM\_DATES = 5).