

# Project 1

## Part A:

In this project, we calculate the weekday given a date.

Python has a module `datetime` and a module `calendar` that provide the same functionality. Your code has to be independent of these modules, though in fact, I will give you an implementation using these modules so that you can test your code easily.

First, implement a function that returns a Boolean value depending on whether its only input, a year, is a leap year. Your program should behave just like `calendar.isleap(year)`.

```
import calendar
def is_leap_year(year):
    return calendar.isleap(year)
```

Second, you are to develop a function that calculates the number of days in the year that precede a given date. The function should have result 0 for January 1st.

Third, you are to develop a function that calculates the number of days since January 1st 2016. It only needs to cover years 2016 to 2020.

The test-function is

```
import calendar

def days_in_year(day, month, year):
    dt = datetime.datetime(year, month, day)
    newyear = datetime.datetime(2016, 1, 1)
    return dt.toordinal()-newyear.toordinal()
```

Fourth, given that the first of January 2016 is a Friday, use the preceding function to develop a function that calculates the day of the week for a date in 2016 — 2020. You can do this by calculating the number of days since January 1, 2016 and then use the modulo 7 operation suitably shifted. The test code is below:

```
def dow(day, month, year):
    code = datetime.datetime(year, month, day).isoweekday()
    if code == 1:
        return "Monday"
    elif code == 2:
        return "Tuesday"
    elif code == 3:
        return "Wednesday"
    elif code == 4:
        return "Thursday"
    elif code == 5:
        return "Friday"
    elif code == 6:
        return "Saturday"
    elif code == 7:
        return "Sunday"
```

```
else:  
    return "error"
```

Fifth, you are to write a function that takes a string such as "6/15/2016 11:31:00 PM" and changes it to "Wednesday 23:31:00", i.e. that replaces the date with the day of the week and the time to military time.

## **Part B:**

Download the "Denver Crime Data" from Kaggle:

<https://www.kaggle.com/paultimothymooney/denver-crime-data/>

Store the unzipped .csv file. Filter the .csv file for entries that have "traffic accident" for their offense category. In a new .csv file, write down the weekday of the accident and the time of the accident rounded to the nearest hour.

## **Part C:**

Once we know dictionaries, you can order the data in the new .csv file to give us a weekday and an hour and count the number of traffic accidents in that hour in Denver during the reporting period.