Date Manipulation Functions

As part of our application, we often need to deal with dates. Let us get an overview about dealing with dates in Python.

- `datetime` is the main library to deal with dates.
- `datetime.datetime` and `datetime.date` are the classes as part of `datetime` library that can be used to deal with dates.
- `datetime.datetime` is primarily used for date with timestamp and `datetime.date` can be used for date with out timestamp.
- When we try to print the date it will print as below (for datetime). It is due to the implementation of string representation functions such as `__str__` or `__repr__`.

```
datetime.datetime(2020, 10, 7, 21, 9, 1, 39414)
```

- We need to format the date using format string to display the date the way we want. These are typically used along with functions such as `strptime` and `strftime`.
  - `%Y` - 4 digit year
  - `%m` - 2 digit month
  - `%d` - 2 digit day with in month
  - There are quite a few other format strings, but these are the most important ones to begin with.
- Also, `datetime` library provides functions such as `strptime` to convert strings to date objects.
- Other important modules to manipulate dates.
  - `calendar` - to get the calendar related information for dates such as day name, month name etc.
  - `datetime.timedelta` - to perform date arithmetic

```
# Importing datetime
import datetime as dt
```

```
# Getting Current date with timestamp
dt.datetime.now()
```

```
datetime.datetime(2020, 12, 23, 3, 50, 16, 892023)
```

```
# Getting Current date without timestamp
from datetime import date
date.today()
```

```
datetime.date(2020, 12, 23)
```

```
# Converting date to a string in the form of yyyy-MM-dd (2020-10-07)
date.today().strftime('%Y-%m-%d')
```

```
'2020-12-23'
```

```
# Converting date to a string in the form of dd-MM-yyyy (07-10-2020)
date.today().strftime('%d-%m-%Y')
```

```
'23-12-2020'
```

```
# Converting date to a string in the form of yyyy/MM/dd (2020/10/07)
date.today().strftime('%Y/%m/%d')
```

```
'2020/12/23'
```

```
# Converting date to an integer in the form of yyyyMMdd (20201007)
int(date.today().strftime('%Y%m%d'))
```

```
20201223
```

```
# Converting time to a string in the form of yyyy-MM-dd HH:mm:SS (2020-10-08 19:25:31)
dt.datetime.now().strftime('%Y-%m-%d %H:%M:%S')
```

```
'2020-12-23 03:50:17'
```

```
# Converting date to a string in the form yyyyMMdd (20201007)
# We can represent this date as integer and hence we can convert the data type
int(date.today().strftime('%Y%m%d'))
```

```
20201223
```

```
# Converting string which contains date using format yyyy-MM-dd as date
dt.datetime.strptime('2020-10-07', '%Y-%m-%d')
```

```
datetime.datetime(2020, 10, 7, 0, 0)
```

```
dt.datetime.strptime('2020-10-07', '%Y-%m-%d').date()
```

```
datetime.date(2020, 10, 7)
```

```
# Converting number which contains date using format yyyyMMdd as date
# strptime expects first argument to be string which contain date
# so we need to convert datatype of number to string
dt.datetime.strptime(20201007, '%Y%m%d')
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-12-af2fd67730c2> in <module>
      2 # strptime expects first argument to be string which contain date
      3 # so we need to convert datatype of number to string
----> 4 dt.datetime.strptime(20201007, '%Y%m%d')

TypeError: strptime() argument 1 must be str, not int
```

```
# Converting number which contains date using format yyyyMMdd as date
# strptime expects first argument to be string which contain date
# so we need to convert datatype of number to string
dt.datetime.strptime(str(20201007), '%Y%m%d')
```

```
datetime.datetime(2020, 10, 7, 0, 0)
```

```
# Converting string which contains timestamp using format yyyy-MM-dd HH:mm:ss as date
dt.datetime.strptime('2020-10-07 21:09:10', '%Y-%m-%d %H:%M:%S')
```

```
datetime.datetime(2020, 10, 7, 21, 9, 10)
```

```
import calendar, datetime as dt
```

```
d = dt.date.today()
```

```
d
```

```
datetime.date(2020, 12, 23)
```

```
list(calendar.day_name)
```

```
['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']
```

```
calendar.weekday?
```

```
Signature: calendar.weekday(year, month, day)
Docstring:
Return weekday (0-6 ~ Mon-Sun) for year (1970-...), month (1-12),
day (1-31).
File:      /opt/anaconda3/envs/beakerx/lib/python3.6/calendar.py
Type:      function
```

```
type(d)
```

```
datetime.date
```

```
d.year
```

```
2020
```

```
d.month
```

```
12
```

```
d.day
```

```
23
```

```
calendar.weekday(d.year, d.month, d.day)
```

```
2
```

```
calendar.day_name[calendar.weekday(d.year, d.month, d.day)]
```

```
'Wednesday'
```

By Durga Gadiraju