**import** datetime  
**import** pytz

what is Navie date?  
*# Naive date: They are called “naive” because they don’t have a time zone  
# associated with them WORKING WITH DATE ONLY*d = datetime.date(2001, 9, 11)  
print(d) *# 2001-09-11*

How to get today’s date, current year?tday = datetime.date.today()  
print(tday) *# March 07, 2019*print(tday.year) *# 2019  
  
# weekday() - Monday is 0 and Sunday is 6*print(tday.weekday()) *# outputs 3, and today is Thursday  
  
# isoweekday() - Monday is 1 and Sunday is 7*print(tday.isoweekday()) *# outputs 4, and today is Thursday*

How to use timedelta? *# class datetime.timedelta  
# A duration expressing the difference between two date, time, or datetime  
# instances to microsecond resolution.***# datetime.timedelta(days=0, seconds=0, microseconds=0, milliseconds=0,  
# minutes=0, hours=0, weeks=0)**next\_year = datetime.date.today() + datetime.timedelta(days=365)  
print(next\_year) *# 2020-03-06*tdelta = datetime.timedelta(days=12)  
print(tday) *# 2019-03-07*print(tday + tdelta) *# 2019-03-19*print(tday - tdelta) *# 2019-02-23  
  
# if we add or subtract a timedelta from a date we get another date as a*

*# result, if we add or subtract a date from a date then we get a timedelta*

*# date2 = date1 + timedelta  
# timedelta = date1 + date2*bday = datetime.date(2019, 12, 12)  
till\_bday = bday - tday *# this calculation is gonna be a timedelta*print(till\_bday.days) *# 280*print(till\_bday.total\_seconds()) *# 24192000.0*How to get today’s time, current hour?

*# WORKING WITH TIME*t = datetime.time(9, 30, 45, 100000)  
print(t.hour) *# 9*dt = datetime.datetime(2016, 7, 16, 12, 30, 45, 990)  
print(dt) *# 2016-07-16 12:30:45.000990*print(dt.date()) *# 2016-07-16*print(dt.time()) *# 12:30:45.000990*

What is the difference between .today() and .now()? *# .today() - returns current datetime with timezone of none  
# .now() - same as today() save it has the option to pass in the time zone*dtnow = datetime.datetime.now()  
print(dir(datetime.datetime))  
print(dt)  
print(dtnow)

How to use pytz module?  
dt = datetime.datetime(2016, 7, 24, 12, 30, 45, tzinfo=pytz.UTC)  
print(dt) *# 2016-07-24 12:30:45+00:00  
  
# Check the difference of 2 examples below, 2nd one is tz aware but 1st*

*# isn't. Tajikistan is plus 05:00 offset of standard UTC timezone  
  
# e.g#1 Print current time*dt\_utcnow = datetime.datetime.now()  
print(dt\_utcnow) *# 2019-03-07 22:54:50.944731  
  
# e.g#2 Print stand UTC time*dt\_utcnow = datetime.datetime.now(tz=pytz.UTC)  
print(dt\_utcnow) *# 2019-03-07 17:55:48.190005+00:00*How to convert **dt\_utcnow** to a different timezone? *# We can use astimezone() function to get the time into a different timezone.*dt\_mtn = dt\_utcnow.astimezone(pytz.timezone(**'Asia/Dushanbe'**))  
print(dt\_mtn) *# 2019-03-07 23:12:48.788380+05:00*Print all list of pytz**for** tz **in** pytz.all\_timezones:  
 print(tz)  
  
*# COUPLE OF WAYS TO DISPLAY THESE DATETIMES, BEST WAY TO USE IS isoformat*dt\_mtn = datetime.datetime.now(tz=pytz.timezone(**"US/Mountain"**))  
print(dt\_mtn.strftime(**'%B %d, %Y'**))  
*# March 07, 2019*

What is strftime and strptime?

*# strftime – Converts Datetime to String  
# strptime – Converts String to Datetime*

How to use strftime?

dt\_mtn = datetime.datetime.now(tz=pytz.timezone(**"US/Mountain"**))  
print(dt\_mtn.strftime(**'%B %d, %Y'**))  
*# March 07, 2019*

How to use strptime? *# SOMETIMES YOU HAVE A STRING AND WANNA CONVERT THAT TO DATETIME, FOR THAT YOU CAN USE strptime method*dt\_str = **'July 24, 2016'**dt = datetime.datetime.strptime(dt\_str, **'%B %d, %Y'**)  
print(dt) *# 2016-07-24 00:00:00*