

Determine the dates of DST and EST. 4/28/2020

This python function, *dst(utc)*, calculates the start date of Daylight savings time and the start date of Eastern standard time, for the date entered.

DST begins on the second Sunday in March and Ends on the first Sunday in November.

Today's date is entered as a string in the format of yyyy-mm-dd.

The year, month and date are extracted and assigned to the variable *tday*.

The algorithm to find the start and end date of DST is as follows:

Find the number of the day of January first *jan1=date(yr,1,1)* of the current year.

Find the day of the week for March 1.

Find the date of the second Sunday.

Find the number of the day for the second Sunday in March, DST start.

Find the day of the week for November 1.

Find the date of the first Sunday.

Find the number of the day for the first Sunday in November, EST start.

If the number of the day of today falls within DST start and EST start, today is on DST.

Otherwise, today is on EST

See Appendix 1 for all of the options of when DST starts and when EST starts.

The function imports the date functions from the standard library.

In Linux each day is assigned a number beginning on January 1, 1970

The number of the January 1 is assigned the variable *jan1*.

```
jan1 = date(yr,1,1)
```

It generates three dictionaries. The *dst{}* dictionary and the *est{}* dictionary contain the offset of the first day of the month to the start of DST in March and the start of EST in November. See Appendix 1.

The *dow{}* dictionary shows the days of the week based on the numeric value from datetime. 0 is Sunday and 6 is Saturday.

The current date *gpsd.utc* is in the format of yyyy-mm-dd.

The *yr*, *mon*, and *da* variables are extracted from *gpsd.utc*.

```
yr = int(gpsd.utc)[0:4]
```

As an example, if `gpsd = 2020-03-17`, `yr = 2020`

```
mon = int(gpsd.utc)[5:7])
```

```
da = int(gpsd[8:10])
```

Today is the variable `tday`.

```
tday = date(yr,min,da)
```

Today's day of the week is `dayofweek`

The number for the day of January 1 of the year above is assigned to the variable `jan1`.

```
Jan1 = date(yr,1,1)
```

The day of the week of March 1, of the year above is assigned to the variable `mar1`.

```
mar1 = datetime.date(yr, 3, 1).weekday() # days 0-6
```

Similarly, for November 1.

From the `dst{}` and `est{}` dictionaries the starting dates for daylight savings time and eastern standard time are the variables `dst_start` and `est_start`.

```
dst_start = dst.get(mar1) # This is the offset from Mar 1
```

```
est_start = est.get(nov1)
```

The number of the starting dates for daylight savings time and eastern standard times are:

```
start_dst= int((date(yr,3,dst_start)-jan1).days)
```

```
start_est = int((date(yr,11,est_start)-jan1).days)
```

Finally the `timetag`, either DST or EST is determined by:

```
if ttday-start_dst <0 or start_est - ttday <=0:
```

```
    tag= 'EST'
```

```
else:
```

```
    tag = 'DST'
```

```
print 'Our time zone is on ',tag
```

```
timetag = tag
```

```
return (timetag)
```

An interesting observation regarding calendars is that March 1 and November 1, always occur on the same day of the week, regardless of the year.

Listing of *dst(x)*

```
def dst(utc):
    #####
    ###Finds the start of EST and DST from today's date
    import datetime
    from datetime import date

    dst = {0:14,1:13,2:12,3:11,4:10,5:9,6:8} # off-set from Mar 1 to start
of DST
    est = {0:7,1:6,2:5,3:4,4:3,5:2,6:1} # off_set from Nov 1 to start of
EST
    dow =
    {0:'Monday',1:'Tuesday',2:'Wednesday',3:'Thursday',4:'Friday',5:'Saturday'
,6:'Sunday'}

    xxx = gpsd.ut
    yr = int(xxx[0:4])
    mon = int(xxx[5:7])
    da = int(xxx[8:10])
    tday =date(yr,mon,da)

    dayofweek = datetime.date(yr, mon, da).strftime("%A")

    # find Jan 1 of year
    jan1 = date(yr,1,1)

    #find mar1
    mar1 = datetime.date(yr, 3, 1).weekday()# days 0-6
    nov1 = datetime.date(yr, 11,1).weekday()
    print 'March 1 is on a :', dow.get(mar1)
    print 'November 1 is on a :', dow.get(nov1)
    dst_start = dst.get(mar1) # This is the offset from Mar 1
    est_start = est.get(nov1)

    print 'Today is :',xxx
    print 'Today is ',dayofweek,' ',tday
    print 'Daylight savings time starts on Sunday March' ,
dst.get(mar1),'th in ',yr
    print 'Estern Standard time starts on Sunday
November',est.get(nov1),'th in ',yr

    print xxx
    yr= int(xxx[0:4])
    mon = int(xxx[5:7])
    da = int(xxx[8:10])
    tday =date(yr,mon,da)
    dayofweek = datetime.date(yr, mon, da).strftime("%A")

    # find Jan 1 of year
    jan1 = date(yr,1,1)
```

```

#find mar1
mar1 = datetime.date(yr, 3, 1).weekday()# days 0-6
nov1 = datetime.date(yr, 11,1).weekday()
print 'March 1 is on a :', dow.get(mar1)
print 'November 1 is on a :', dow.get(nov1)
dst_start = dst.get(mar1) # This is the offset from Mar 1
est_start = est.get(nov1)

print 'Today is :',xxx
print 'Today is ',dayofweek,' ',tday
print 'Daylight savings time starts on Sunday March' ,
dst.get(mar1),'th in ',yr
print 'Eastern Standard time starts on Sunday
November',est.get(nov1),'th in ',yr
ttday =int((tday-jan1).days) # Number of days between today and 1/1
print dst_start
print est_start
print 'Number of days from Jan 1 to today ;',ttday
print 'tday: ',tday
print date(yr,3,dst_start)
print date(yr,11,est_start)
start_dst= int((date(yr,3,dst_start)-jan1).days)
start_est = int((date(yr,11,est_start)-jan1).days)

print start_dst
print start_est

if ttday-start_dst <0 or start_est - ttday <=0:
    tag= 'EST'
else:
    tag ='DST'
print 'Our time zone is on ',tag
timetag = tag
return (timetag)

```

Appendix 1

The start of Daylight Savings Time is the second Sunday in March

	Sun	Mo n	Tu e	We d	Thu	Fri	Sat
Python Day Number	6	0	1	2	3	4	5
If March 1 is Mon		1	2	3	4	5	6
	7	8	9	10	11	12	13
DST starts	14						
If March 1 is Tue			1	2	3	4	5
	6	7	8	9	10	11	12
DST starts	13	14					
If March 1 is Wed				1	2	3	4
	5	6	7	8	9	10	11
DST starts	12	13	14				
If March 1 is Thu					1	2	3
	4	5	6	7	8	9	10
DST starts	11	12	13	14			
If March 1 is Fri						1	2
	3	4	5	6	7	8	9
DST starts	10						
If March 1 is Sat							1
	2	3	4	5	6	7	8
DST starts	9						
If March 1 is Sun	1	2	3	4	5	6	7
DST starts	8	9	10	11	12	13	14

The start of Eastern Standard Time is the first Sunday in November

Python Day Number	Sun	Mo n	Tu e	We d	Thu	Fir	Sat
	6	0	1	2	3	4	5
If Nov 1 is Mon		1	2	3	4	5	6
EST starts	7						
If Nov 1 is Tue			1	2	3	4	5
EST starts	6	7					
If Nov1 is Wed				1	2	3	4
EST starts	5	6	7				
If Nov 1 is Thu					1	2	3
EST starts	4	5	6	7			
If Nov 1 is Fri						1	2
EST starts	3	4	5	6	7		
If Nov 1 is Sat							1
EST starts	2	3	4	5	6	7	
If Nov 1 is Sun							
EST starts	1	2	3	4	5	6	7