Description of dst.py 3/25/20

This program calculates the start date of Daylight savings time and the start date of Eastern standard time, for the date entered. This program is very similar the ***find\_dst.py*** program. The difference between the two is that the ***dst.py*** version use ***Dictionaries*** instead of an ***if-elif*** sequence.

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(y,1,1)***} of the current year.

Find the number of the day DST starts of that year.

Find the number of the day EST starts of that year.

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 the options of when DST start and when EST starts.

The program imports the datetime and date function needed

The program creates two dictionaries, ***dst{}*** and ***est{}*** for the offsets of the first of the month to the second Sunday in March and the first Sunday in November. (It is interesting that 03/01/yyyy and 11/01/yyyy always occur on the same day of the week!) See Appendix 1 for the details on the offset days.

It creates a third dictionary ***dow{}*** to convert the day of the week calculation (0 to 6), to a string ( Monday to Sunday.

User enters today’s date

The program parses the date into yr, mon and da variables.

***jan1*** is the number that Linux assigns to 1/1/yr

***mar1*** is the day of the week of March 1.

***nov1*** is the day of the week of November 1.

The start date of DST, is extracted from the dictionary with the command ***dst\_start=dst.get(mar1)***.

The start date of EST, is extracted from the dictionary with the command ***est\_start=est.get(nov1)***.

***ttday*** is the number of days between Jan 1 and today.

***start\_dst*** is the number of days from Jan 1 to the start of day light saving time.

***start\_est*** is the number of days from Jan 1 to the start of eastern standard time.

If today, ***ttday***, is less than day light savings time start, ***start\_dst***, or ***ttday*** is greater than eastern standard time start, ***start\_est***, then today in on EST. Otherwise, today is on DST.

**Import in Python**

Problem: The main program, gps1.py, speedometer.py or another application reads the date from a gps receiver and displays the time of day. The offset from Universal time, utc, must be determined. This requires finding whether the local time is on standard time or daylight savings time, dst.

The program ***dst\_import.py*** will determine this when a date in supplied.

The main program was modified to imported the tag from ***dst\_import.py***. The problem is: How do we pass the UTC variable from the main program, ***speedometer.py*** to the ***dst\_import.py*** program. The solution is to have the main program writes the date to a text file and the ***dst\_import.py*** program reads the date from the text file and generates the tag.

The main program imports the tag and then sets the tagtime and off\_set varibles

To test the program, a file ***get\_tag2.py*** was created to write to the text file. In order to pass the date to the ***dst\_import.py*** file, the date was written to the file ***read\_date.txt*** by the ***get\_tag2.py.***

The file ***dst\_import.py*** is used to get the tag, either EST or DST from the date provided by ***get\_tag2.py.***

The date is retrieved from the file ***read\_date.txt***.

The date was written into the file as **2020-03-17** , but when it is read out it is of the

 format: **[‘2020-03-17’]**, so the parsing for yr, mon and da was changed as shown below:

**From *dst.py***

**yr= int(utc[0:4])**

**mon = int(utc[5:7])**

**da = int(utc[8:10])**

**tday =date(yr,mon,da)**

**From *dst\_import.py***

**yr= int(utc[2:6])**

**mon = int(utc[7:9])**

**da = int(utc[10:12])**

**tday =date(yr,mon,da)**

The file ***get\_tag2.py*** writes the date to the file ***read\_data.txt*** and then imports the tag from the file ***dst\_import.py***.

Is there another way to pass the variable utc from one file to the other without writing it to the third file? TBD!

Tried to import gspd.utc as xyz from the main program to the ***find\_dst\_import.py***, but received the error:

**Traceback (most recent call last):**

 **File "x.py", line 228, in <module>**

 **from find\_dst\_import2 import tag**

 **File "/home/pi/speedometer/find\_dst\_import2.py", line 16, in <module>**

 **from x import xyz**

**ImportError: cannot import name xyz**

Appendix 1

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | The start of Daylight Savings Time |  |  |
|  |  |  is the second Sunday in March |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | Sun | Mon | Tue | Wed | 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 |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | Sun | Mon | Tue | Wed | Thu | Fir | Sat |  |
|  | Python Day Number |  |  | 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 | EST starts |  |  |  |  |  |  |  | 1 |  |
|  |  |  |  | 2 | 3 | 4 | 5 | 6 | 7 |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  | If Nov 1 is Sun | EST starts |  | 1 | 2 | 3 | 4 | 5 | 6 | 7 |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |