CISCO- Python Project

****

Disclaimer:

This Document Contains the Project Approach, Problem Statement, and Solution Submitted by Me for CISCO NCH Python Project.

Email: ***devangsharma419@gmail.com***

****

Problem Statement

DESCRIPTION

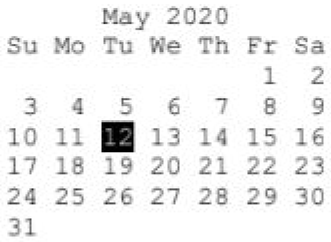
Create a desktop python app that, can provide the advanced functionalities that are not generally available in the default calendar module.

When run, the application should be able to ascertain the current day, date, month and year using the existing modules (Ex. Calendar) but the rest of the functionalities should be purely computed based on the algorithm and the flow of thoughts captured in the problem statement.

The application expects the learners to compute a future calendar (future month, for that date) or a past calendar (past month) of the given date (today). The application building expects the users to use their knowledge of Python using features such as Python Shell, Python list and sets, dictionary, formatting, utilities, calendar (for verification), geocoder (for lat and long), ‘os ’and ‘sys ’ and usual program control flow structures.

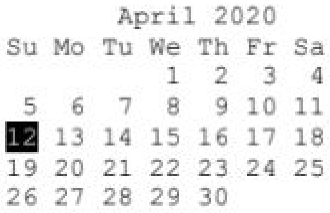
The application design should follow the below guidelines:

- The application should be able to run with or without command line parameters.
- When the application is run without command line parameters, it should simply print the current month’s calendar with current date highlighted.
- The Month and Year should appear on the top (for example, May 2020)
- The Days should appear in the next row (for example: Su Mo Tu We Th Fr Sa)
- The full calendar of the month should appear as follows:



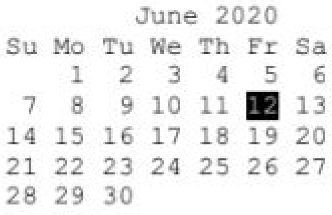
```
      May 2020
Su Mo Tu We Th Fr Sa
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
31
```

- Note that current date should be highlighted. (in the above example, the app was run on 12 May, 2020)
- The application is run with command line parameter, it can be an integer only.
- For example, it can be +5 or -10.
- If, for example, the application is run with -1, the output should be the following:



```
     April 2020
Su Mo Tu We Th Fr Sa
          1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30
```

- Note that date 12 is still highlighted, showing “last month, this day” concept.
- If the application is run with +1, the following should be the output:



```
     June 2020
Su Mo Tu We Th Fr Sa
    1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30
```

- Note that 12th is still highlighted, showing, “this day, next month” concept.
- If the application is run without any other command line parameter (+12 through -12, including 0), or with a “-help” as the command line parameter, the app should show the help for running the app, along with the student credentials.
- Help: This application should be run with the following command line parameters: a positive integer, a negative integer or zero. The positive integer cannot be more than 12 or less than -12.

Verification of the data can be done using the inbuilt libraries.

Assumptions

The student is expected to create the app in such a way that it can compute up to ±12 months.

Time Allocated: 3 Hours

Solution

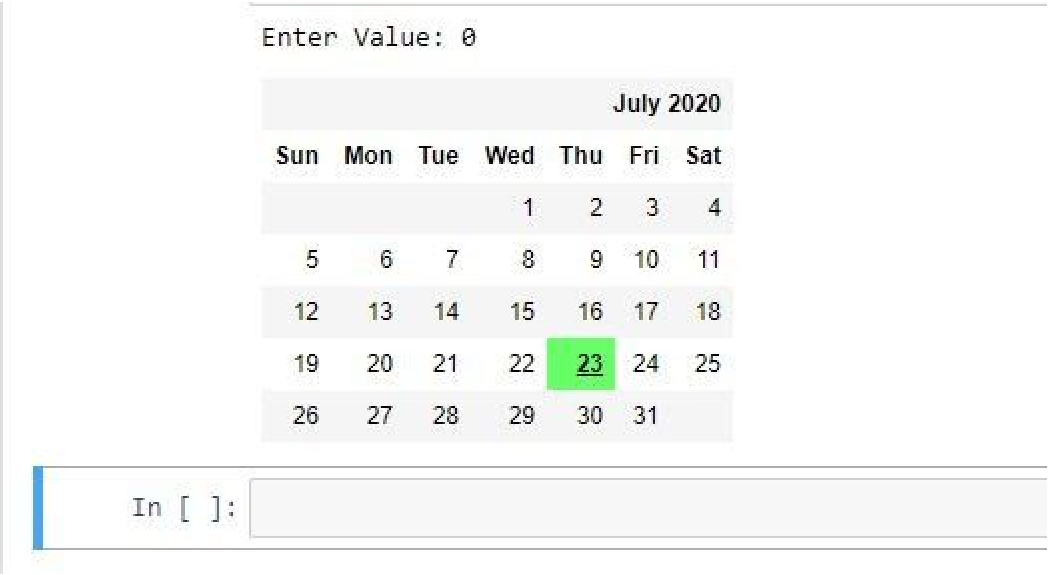
```
# Solution Submitted By: DEVANG SHARMA
# Email: devangsharma419@gmail.com
# Alternate Email: devangsharma25398@gmail.com
# T: +91-9953027469


# Code Starts Here


import calendar
import datetime
from IPython.display import display,HTML

def function(num):

    # Entered Value Must Lie Between -12 and 12
    if num>12 or num<-12:
        print("Enter a Number between -12 to +12")
    else:
        now = datetime.datetime.now()
        yy=now.year
        mm=now.month+num

        if mm>12:
            yy=yy+1
            mm=mm%12

        if mm<1:
            yy=yy-1
            mm=mm+12

        # create HTML Calendar month
        cal = calendar.HTMLCalendar(calendar.SUNDAY)
        s= cal.formatmonth (yy,mm)

        # add cell's backgroundcolor, bold and underling html tags
        # around today's date
        ss = s.replace('>%i<'%t.day, ' bgcolor="#66ff66"><b><u>%i</u></b><'%t.day)
        display(HTML(ss))

print("Enter Value: ")
val=int(input())
function(val)
```

NOTE:

Run With: ***Jupyter Notebook***

Language Used : ***PYTHON 3***

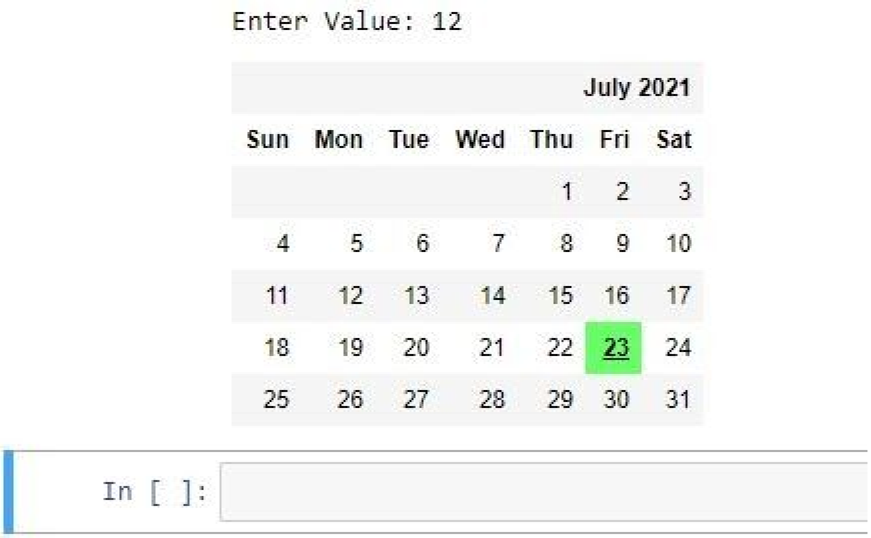
SCREENSHOTS



```
Enter Value: 0
```

July 2020

Sun	Mon	Tue	Wed	Thu	Fri	Sat
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	**23**	24	25
26	27	28	29	30	31	



```
Enter Value: 12
```

July 2021

Sun	Mon	Tue	Wed	Thu	Fri	Sat
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	**23**	24
25	26	27	28	29	30	31