#!/usr/bin/env python3

""" Ed's Calendar - Tkinter ========== :Description: Ed's calendar using Tkinter as user interface and procedural python3 :Author: Edward Birdsall :Date: 2022 March 30 :Version: 0.1

Long Description and Goals: Ulimately this will turn into a calendar program keeping track of appointments and such and displaying the information using a number of different calendars. Currently working with Tkinter and procedural python. Final 'production' edition will be in Object Oriented python3 using the MVC structure.

Goals:

- Display a calendar using either Tkinter or HTML
- Day, Week and Month formats
- Command Line calls or tkinter/HTML interactive
- **In addition to Gregorian a number of other calendars to display such as**
    - Julian
    - Jewish
    - Scientific
    - Shire
    - Oriental
- Use the same call variables for both Tkinter and the Jinja2(HTML) displays

Variables:

Desired input to structuring page:

hd, hdr, cal, pref

Internal:

dts = list of day numbers used to set up tdy[x]["dnum"]

Desired output to display the page Descriptions

hd: Web/TC/Tk page hdr: dictionary with calendar header information

name - name of calendar page - month year Calendar today - day of week and date of current day

days:days of week colorsm - dictionary with colors for background of dates

priormonth, thisbefore, today, thismonth, nextmonth, site, neutral, calSclr

tdy - dictionary array of information for the days to be displayed

bgtclr - background today clear bgeclr - background event clear dnum - day number devt - number of day's events devt1t - day event today first title devt1c - day event today first calendar color devt2t - day event today second title devt2c - day event today second calendar color devt3t - day event today third title devt3c - day event today third calendar color devt4t - day event today fourth title devt4c - day event today fourth calendar color

cal - calendar display and control information

month - Display Month year - Display Year startwk - week number for first display week calrows - number of rows of calendar to be displayed calAt - name of first calendar calBt - name of second calendar calCt - name of third calendar calDt - name of fourth calendar calEt - name of fifth calendar

pref - dictionary of user preferences

startDay - the starting day of the week 1=Monday, 0 or 7 is Sunday calAclr - color for first calendar calBclr - color for second calendar calCclr - color for third calendar calDclr - color for fourth calendar calEclr - color for fifth calendar

""" import calendar import datetime from tkinter import *

class MCalendar(Frame):

f __init__(self):

super().__init__()

self.initUI()

f initUI(self):

#==========

#Begining of definitions. Need to remember how to split this into a seperate area that invokes MCa

#==========

#========== # desired inputs for final version what, when, which event calendars, prefe

#========== inputs = {"name":"Ed", "date":"March 2, 2019", "startD

#========== # some kludges for testing and until I get it all v

#========== #cyear = int(cal.get("year")) cyear = 2019 cmonth = 3 co

cals= [

dict(num=0, app="FALSE", name="Liturgical", color="yellowgreen"), dict(num=1, app="FALSE", name="US Ho color="lightsteelblue"), dict(num=2, app="TRUE", name="Birdsall Family", color="cyan"), dict(num=3, app="TRUE", name= Family",color="magenta"), dict(num=4, app="FALSE", name="", color="purple"), dict(num=5, app="TRUE", name="Site", color="r

]

eventCal = [

dict(num=0, cal="Liturgical", eDate=datetime.date(2019, 2, 24), title="Ordinary 8th Sunday"), dict(num=1, cal="Litu eDate=datetime.date(2019, 3, 3), title="Ordinary 9th Sunday"), dict(num=2, cal="Liturgical", eDate=datetime.date(2019, 3, 6), titl Wednesday"), dict(num=3, cal="US Holidays", eDate=datetime.date(2019, 3, 9), title="DST begins"), dict(num=4, cal="Litu eDate=datetime.date(2019, 3, 10), title="Lent 1st Sunday"), dict(num=5, cal="Liturgical", eDate=datetime.date(2019, 3, 17), title 2nd Sunday"), dict(num=6, cal="Liturgical", eDate=datetime.date(2019, 3, 24), title="Lent 3rd Sunday"), dict(num=7, cal="Litu eDate=datetime.date(2019, 3, 31), title="Lent 4th Sunday"), dict(num=8, cal="Birdsall Family", eDate=datetime.date(2019, title="Bryan Kovas B'Day"), dict(num=9, cal="Birdsall Family", eDate=datetime.date(2019, 3, 3), title="Helen Birdsall B dict(num=10, cal="Birdsall Family", eDate=datetime.date(2019, 3, 4), title="Greg Kovas B'Day"), dict(num=11, cal="Birdsall F eDate=datetime.date(2019, 3, 4), title="Andrew Noyes B'Day"), dict(num=12, cal="Birdsall Family", eDate=datetime.date(2019 title="Brielle Balmer B'Day"),

]

#========== # Working code towards final version #==========

{"loc":"Month Calendar"} hdr = { "name":"Ed", "page":"Calendar", "today":"Saturday March 2, 2019" } hdr["name"] = inputs["

hdr["page"] = datetime.datetime.strptime(inputs["date"], "%B %d, %Y").strftime("%B %Y")+" Calendar" hdr["toda

datetime.datetime.now().strftime("%A %B %d, %Y") #hdr["today"] = "Saturday March 02, 2019" #for testing

cal = {"month":"March", "year":"2019", "startwk":0,"calrows":6, "calAt":"Liturgical", "calBt":"US Holidays",

```
    "calCt":"Birdsall Family", "calDt":"Kirkup Family", "calEt":""}

pref = { "startDay":1, "calAclr": "yellowgreen", "calBclr": "lightsteelblue", "calCclr": "cyan", "calDclr": "magenta", "calEclr": "purple"} pref["startday"] = inputs["startDay"]

if (inputs["startDay"] == 6): # Week starts on Sunday
    cal["startwk"] = int(datetime.date(cyear, cmonth, cday).strftime("%U"))
else: #Week starts on Monday
    cal["startwk"] = int(datetime.date(cyear, cmonth, cday).strftime("%W"))

glcal = calendar.Calendar(inputs["startDay"]) days = [calendar.day_name[i] for i in glcal.iterweekdays()] dts = [] dmt = [] for i in glcal.itermonthdates(cyear, cmonth):

    dts.append( i.day ) dmt.append( i.month )

numdays = len(dts) cal["calrows"] = numdays//7

colorsm = {"priormonth": "Orchid", "thisbefore": "Aqua", "today": "Yellow", "thismonth": "White", "nextmonth": "Lime", "site":"Red" , "neutral": "silver", "calSclr": "red" }

tdy = [] for i in range(0, 42,1):

    tdy.append({"bgtclr":"white","bgeclr":"white", "mnum":0, "dnum":0, "devt":-1, "dev1t":"", "dev1c":"gray85", "dev1e":"", "dev2t":"", "dev2c":"gray85", "dev2e":"", "dev3t":"", "dev3c":"gray85", "dev3e":"", "dev4t":"", "dev4c":"grey85", "dev4e":"", })
for i in range(0, numdays, 1):
    tdy[i]["mnum"] = dmt[i] tdy[i]["dnum"] = dts[i] if (dmt[i] < cmonth):
        tdy[i]["bgtclr"] = colorsm["priormonth"] tdy[i]["bgeclr"] = colorsm["priormonth"]
    elif (dmt[i] > cmonth ):
        tdy[i]["bgtclr"] = colorsm["nextmonth"] tdy[i]["bgeclr"] = colorsm["nextmonth"]
    else:
        if (dts[i] < cday):
            tdy[i]["bgtclr"] = colorsm["thisbefore"] tdy[i]["beeclr"] = colorsm["thisbefore"]
        elif (dts[i] == cday):
            tdy[i]["bgtclr"] = colorsm["today"] tdy[i]["beeclr"] = colorsm["today"]
        else:
            tdy[i]["bgtclr"] = colorsm["thismonth"] tdy[i]["beeclr"] = colorsm["thismonth"]
for c in range(8):
    self.columnconfigure(c, pad=3)
for r in range(38):
    self.rowconfigure(r, pad=3)
for i in range(0, 42, 1):
    for ii in range(0,len(eventCal),1):
        if ((tdy[i]["mnum"] == eventCal[ii]["eDate"].month) and (tdy[i]["dnum"] == eventCal[ii]["eDate"].day)):
            if (tdy[i]["devt"]<1):
                tdy[i]["devt"] = 1
            else:
                tdy[i]["devt"] = tdy[i]["devt"] + 1
            if (tdy[i]["devt"] == 1):
                tdy[i]["dev1t"] = eventCal[ii]["title"] for iii in range(0,len(cals),1):
```

```
                if (cals[iii]["name"] == eventCal[ii]["cal"]):
                    tdy[i]["dev1c"] = cals[iii]["color"] if (cals[iii]["app"] == "TRUE"):
                        tdy[i]["dev1e"] = "Show "+ eventCal[ii]["title"]
            elif (tdy[i]["devt"] == 2):
                tdy[i]["dev2t"] = eventCal[ii]["title"] for iii in range(0,len(cals),1):
                    if (cals[iii]["name"] == eventCal[ii]["cal"]):
                        tdy[i]["dev2c"] = cals[iii]["color"] if (cals[iii]["app"] == "TRUE"):
                            tdy[i]["dev2e"] = "Show "+ eventCal[ii]["title"]
            elif (tdy[i]["devt"] == 3):
                tdy[i]["dev3t"] = eventCal[ii]["title"] for iii in range(0,len(cals),1):
                    if (cals[iii]["name"] == eventCal[ii]["cal"]):
                        tdy[i]["dev3c"] = cals[iii]["color"] if (cals[iii]["app"] == "TRUE"):
                            tdy[i]["dev3e"] = "Show "+ eventCal[ii]["title"]
            elif (tdy[i]["devt"] == 4):
                tdy[i]["dev4t"] = eventCal[ii]["title"] for iii in range(0,len(cals),1):
                    if (cals[iii]["name"] == eventCal[ii]["cal"]):
                        tdy[i]["dev4c"] = cals[iii]["color"] if (cals[iii]["app"] == "TRUE"):
                            tdy[i]["dev4e"] = "Show "+ eventCal[ii]["title"]
            else:
                tdy[i]["devt"] = 4
#=============================================================================================
#Ending of definitions. Need to remember how to split this into a seperate area that invokes MCa
#=============================================================================================

#Window Display self.master.title(hd.get("loc")) ct = Label(self,text=hdr.get("name")+"'s "+hdr.get("page"), justify=CENTER).grid(
columnspan=8, sticky=W+E) ct1 = Label(self, text="Today's date: "+hdr.get("today"), justify=CENTER).grid(row=3, columns
sticky=W+E)

#Header Display

wom = Label(self, text="Week", font='serif, 10', bg="white", fg="black", height=1, width=6, borderwidth=3, relief="raised").grid(
column=0) for x in range(7):

    dow = Label(self, text=days[x], font='serif, 10', bg="white", fg="black", height=1, width=17, borderwidth=3, relief="raised").grid(
    column=x+1)

# Weeks Display for r in range(1, cal.get("calrows")+1,1):


    wom = Label(self, text=cal.get("startwk")+(r-1), background="white", foreground="black", width=8, borderw
    relief="raised").grid(row=5+((r-1)*5), column=0) for d in range(1,8,1):

        dow = Label(self,
        text=tdy[(((r-1)*7)+(d-1))].get("dnum"),background=tdy[(((r-1)*7)+(d-1))].get("bgtclr"),
        width=17, relief="groove").grid(row=(5+((r-1)*5)), column=d) if
        len(tdy[(((r-1)*7)+(d-1))].get("dev1e")) >0:

            dow1 = Button(self, text=tdy[(((r-1)*7)+(d-1))].get("dev1t"),
            bg=tdy[(((r-1)*7)+(d-1))].get("dev1c"), width=15, state=NORMAL,
            command="").grid(row=(6+((r-1)*5)), column=d)
```

else:

dow1 = Label(self, text=tdy[(((r-1)*7)+(d-1))].get("dev1t"), background=tdy[(((r-1)*7)+(d-1))].get("dev1c"), width=17, relief="ridge").grid(row=(6+((r-1)*5)), column=d)

if len(tdy[(((r-1)*7)+(d-1))].get("dev2e")) >0:

dow2 = Button(self, text=tdy[(((r-1)*7)+(d-1))].get("dev2t"), bg=tdy[(((r-1)*7)+(d-1))].get("dev2c"), width=15, state=NORMAL, command="").grid(row=(7+((r-1)*5)), column=d)

else:

dow2 = Label(self, text=tdy[(((r-1)*7)+(d-1))].get("dev2t"), background=tdy[(((r-1)*7)+(d-1))].get("dev2c"), width=17, relief="ridge").grid(row=(7+((r-1)*5)), column=d)

if len(tdy[(((r-1)*7)+(d-1))].get("dev3e")) >0:

dow3 = Button(self, text=tdy[(((r-1)*7)+(d-1))].get("dev3t"), bg=tdy[(((r-1)*7)+(d-1))].get("dev3c"), width=15, state=NORMAL, command="").grid(row=(8+((r-1)*5)), column=d)

else:

dow3 = Label(self, text=tdy[(((r-1)*7)+(d-1))].get("dev3t"), background=tdy[(((r-1)*7)+(d-1))].get("dev3c"), width=17, relief="ridge").grid(row=(8+((r-1)*5)), column=d)

if len(tdy[(((r-1)*7)+(d-1))].get("dev4e")) >0:

dow4 = Button(self, text=tdy[(((r-1)*7)+(d-1))].get("dev4t"), bg=tdy[(((r-1)*7)+(d-1))].get("dev4c"), width=15, state=NORMAL, command="").grid(row=(9+((r-1)*5)), column=d)

else:

dow4 = Label(self, text=tdy[(((r-1)*7)+(d-1))].get("dev4t"), background=tdy[(((r-1)*7)+(d-1))].get("dev4c"), width=17, relief="ridge").grid(row=(9+((r-1)*5)), column=d)

#Bottom of Page brow = ((cal.get("calrows")+1)*5) cb = Label(self,text="Legend").grid(row=brow+1, column=4)

day1 = Label(self, text="Prior Month", background=colorsm["priormonth"], relief="raised").grid(row=brow+2, column=0) day2 = Label(self, text="This Month Prior to today", background=colorsm["thisbefore"], width=30, relief="raised").grid(row=brow+2, column=1, columnspan=2) day3 = Label(self, text="Today", background=colorsm["today"],width=17, relief="raised").grid(row=brow+2, column=3) day4 = Label(self, text="This Month after today", background=colorsm["thismonth"],width=30, relief="raised").grid(row=brow+2, column=4, columnspan=2) day5 = Label(self, text="Next Month", background=colorsm["nextmonth"],width=17, relief="raised").grid(row=brow+2, column=6) day6 = Label(self, text="Site Down", background=colorsm["site"], width=17, relief="raised").grid(row=brow+2, column=7)

ca = Label(self, text="Calendars in Use") ca.grid(row=brow+3, column=4)

cal1 = Label(self, text=cal.get("calAt"), background=pref["calAclr"], width=17, relief="raised").grid(row=brow+4, column=1) cal2 = Label(self, text=cal.get("calBt"), background=pref["calBclr"], width=17, relief="raised").grid(row=brow+4, column=2) cal3 = Label(self, text=cal.get("calCt"), background=pref["calCclr"], width=17, relief="raised").grid(row=brow+4, column=3) cal4 = Label(self, text=cal.get("calDt"), background=pref["calDclr"], width=17, relief="raised").grid(row=brow+4, column=4) cal5 = Label(self, text=cal.get("calEt"), background=pref["calEclr"], width=17, relief="raised").grid(row=brow+4, column=5) cal6 = Label(self, text="Site", background="red", width=17, relief="raised").grid(row=brow+4, column=6)

```
        cc = Label(self) cc.grid(row=brow+5, columnspan=8, sticky=W+E) lbl0 = Label(self, text="
        ").grid(row=brow+6,column=0) brt = Button(self,text="Return", command="").grid(row=brow+6,
        column=1) lbl1 = Label(self, text=" ").grid(row=brow+6,column=2) bbk = Button(self,text="Prior
        Month", command="").grid(row=brow+6, column=3) bpp = Button(self, text="Print Page",
        command="").grid(row=brow+6, column=4) bnm = Button(self,text="Next Month",
        command="").grid(row=brow+6, column=5) blbl= Label(self, text=" ").grid(row=brow+7,
        columnspan=8, sticky=W+E) self.pack()
def main():
    root = Tk() app = MCalendar() root.mainloop()
if __name__ == '__main__':
    main()
```