In [ ]:

```
#CALENDER MODULE WITH PYTHON
#first import the module
```

In [56]:

```
import calendar
#week header()
#use calendar.weekheader(value)
#pass a value like three to display the the days of the weeks
#with initials
print(calendar.weekheader(3))
print(calendar.weekheader(10))#pass 10 to display the whole word of the week
"""
weekheader()->returns a header containing abbreviated
weekday names

"""
print(calendar.firstweekday())#returns zero
"""
monday is stored as a zero in the week days
python starts its week from monday
"""
print(calendar.month(2020,12,w=0,l=0))

"""
calendar.month()->returns the month with the days
pass in the (year,month)

"""
print(calendar.monthcalendar(2020,12))
"""
gets data out of the calendar and stores
it inside a list of list
calendar.monthcalendar(year,month)
values outside the month are represented by zeros

"""
```

```
Sun Mon Tue Wed Thu Fri Sat
  Sunday     Monday    Tuesday   Wednesday   Thursday    Friday    Saturday
6
   December 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

[[0, 0, 1, 2, 3, 4, 5], [6, 7, 8, 9, 10, 11, 12], [13, 14, 15, 16, 17, 18, 19], [2
0, 21, 22, 23, 24, 25, 26], [27, 28, 29, 30, 31, 0, 0]]
```

O...

```
'\ngets data out of the calendar and stores \nit inside a list of list\ncalenda
r.monthcalendar(year,month)\nvalues outside the month are represented by zero
s\n\n'
```

In [39]:

```
import calendar
```

```
print(calendar.calendar(2020))
"""
calendar.calendar(year)->prints out the whole calendar
one can iterate the whole calendar to get the values

"""
print(calendar.weekday(2023,12,9))
"""
calendar.weekday(year,month,date)
returns the day of the week in integer
[0,   1,  2,  3,   4,  5,  6]
[Mon,Tue,Wed,Thur,Fri,Sat,Sun]
"""
print(calendar.isleap(2000))
"""
tells if a certain year is leap or not
"""
print(calendar.leapdays(2000,2004))
```

```
                                  2020

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

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

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

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

5
True
1
```

In [41]:

```
import calendar
print(calendar.prmonth(2020,12))
"""
works like calender.month(year,month)
"""
```

```
   December 2020
Mo Tu We Th Fr Sa Su
    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
None
```

In [50]:

```
import calendar
days=list(calendar.day_abbr)
print(days)
for d in calendar.day_abbr:
    print(d)
"""
calendar.day_abbr->returns abbreviations
used the list function to convert the abbreviations to alist

"""
month=list(calendar.month_name)
print(month)
j=0
for m in calendar.month_name:
    j+=1
    print(j,m)
"""
print out or gets month names into a list
we can also iterate through them
"""
```

```
['Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun']
Mon
Tue
Wed
Thu
Fri
Sat
Sun
['', 'January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'Sep
tember', 'October', 'November', 'December']
1
2 January
3 February
4 March
5 April
6 May
7 June
8 July
9 August
10 September
11 October
12 November
```

```
13 December
```

In [55]:

```
"""
the firts day of the week is monday and the last is sunday
this is according to pyhton
But we can change the default to our benefit for example we can
change it back to sunday by using
syntax{
calendar.setfirstweekday(weekday)
}

"""
import calendar
calendar.setfirstweekday(calendar.SUNDAY)
print(calendar.firstweekday())
"""
[0,   1,  2,  3,   4,  5,   6]
[mon,tue,wed,thur,fri,sat,sun]
our program returns 6 this shows that it has been canged to
sunday since sunday is accessed by 6 in the list
"""
```

```
6
```

In [ ]: