

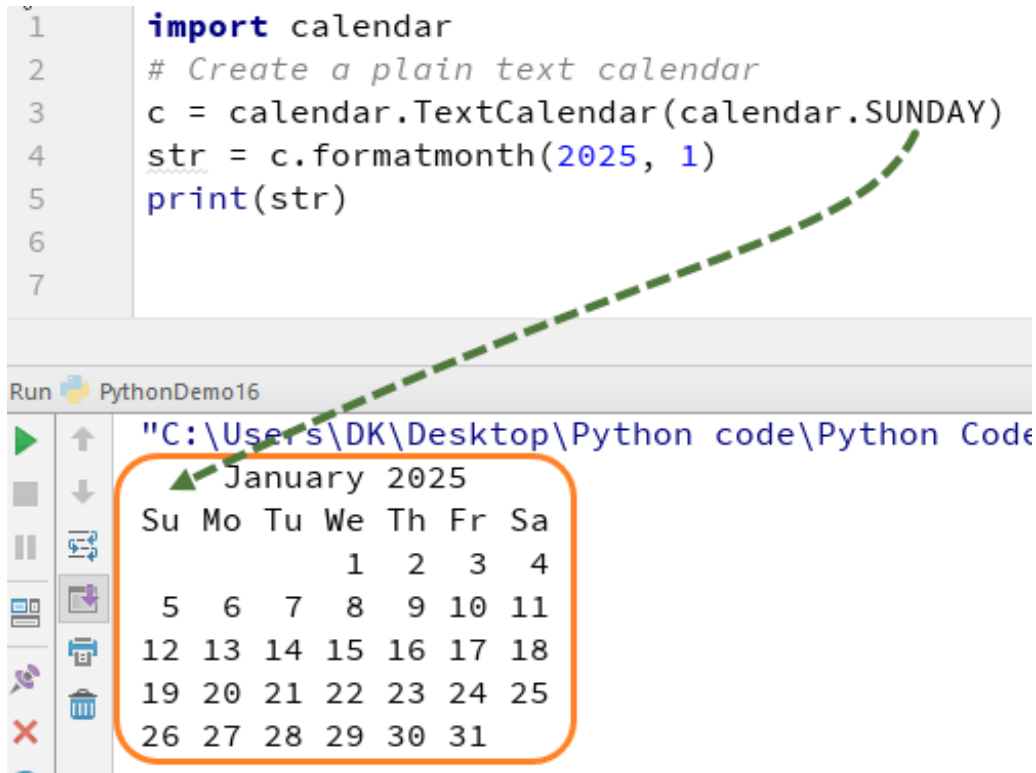
## 16.Hướng dẫn sử dụng Python CALENDAR

Mô-đun Calendar(lich) trong Python chứa lớp calendar cho phép thực hiện việc tính toán nhiều tác vụ khác nhau dựa theo ngày, tháng và năm. Trên hết, lớp TextCalendar và HTMLCalendar trong Python cho phép bạn thay đổi và sử dụng chúng tùy theo nhu cầu của mình.

Hãy xem những gì chúng ta có thể làm với mô-đun calendar.

**Bước 1)** Thực thi mã nguồn.

```
1 import calendar
2 # Create a plain text calendar
3 c = calendar.TextCalendar(calendar.SUNDAY)
4 str = c.formatmonth(2025, 1)
5 print(str)
6
7
```



The screenshot shows a Python IDE with a code editor and a console window. The code in the editor is as follows:

```
1 import calendar
2 # Create a plain text calendar
3 c = calendar.TextCalendar(calendar.SUNDAY)
4 str = c.formatmonth(2025, 1)
5 print(str)
6
7
```

The console window shows the output of the script, which is a text-based calendar for January 2025. The calendar is displayed as follows:

```
"C:\Users\DK\Desktop\Python code\Python Code
January 2025
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
```

- Dòng lệnh số 1: Chúng ta bắt đầu với "import calendar", câu lệnh này sẽ nạp toàn bộ các lớp trong mô-đun.
- Dòng lệnh số 3: Câu lệnh `c = calendar.TextCalendar(calendar.SUNDAY)` yêu cầu trình biên dịch tạo lịch ở dạng chữ. Ngày bắt đầu hiển thị tháng là chủ nhật. Python cho phép bạn định dạng lịch bằng cách thay đổi ngày bắt đầu để hiển thị tháng.
- Dòng lệnh số 4: `str = c.formatmonth(2025,1)` chúng ta đang tạo lịch cho tháng 1, năm 2025.
- Dòng lệnh số 5: In giá trị của biến str.

Bây giờ, chúng ta sẽ thay đổi giá trị từ chủ nhật (Sunday) sang thứ năm (Thursday) và kiểm tra đầu ra.

```

1 #
2 #Example file for working with calendars
3 #
4
5 import calendar
6 # Create a plain text calendar
7 c = calendar.TextCalendar(calendar.THURSDAY)
8 str = c.formatmonth(2025, 1)
9 print(str)
10

```

Run PythonDemo16

"C:\Users\DK\Desktop\Python code\Python Code\Python16\venv\Scripts\python.exe" "C:\Users\DK\Desktop\Python code\Python Code\Python16\venv\Scripts\python.exe" "C:\Users\DK\Desktop\Python code\Python Code\Python16\venv\Scripts\python.exe" "C:\Users\DK\Desktop\Python code\Python Code\Python16\venv\Scripts\python.exe"

```

January 2025
Th Fr Sa Su Mo Tu We
      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

```

Ở đầu ra, chúng ta muốn tháng 1 năm 2025 được in ra bắt đầu từ thứ năm

**Bước 2)** Bạn cũng có thể in Lịch ở định dạng HTML, tính năng này hữu ích với các lập trình viên nếu họ muốn thực hiện bất kỳ thay đổi nào trong cách hiển thị lịch.

```

1
2 import calendar
3
4 # Create an HTML formatted calendar
5 hc = calendar.HTMLCalendar(calendar.THURSDAY)
6 str = hc.formatmonth(2025, 1)
7 print(str)
8
9

```

Run PythonDemo16

"C:\Users\DK\Desktop\Python code\Python Code\Python16\venv\Scripts\python.exe" "C:\Users\DK\Desktop\Python code\Python Code\Python16\venv\Scripts\python.exe" "C:\Users\DK\Desktop\Python code\Python Code\Python16\venv\Scripts\python.exe" "C:\Users\DK\Desktop\Python code\Python Code\Python16\venv\Scripts\python.exe"

```

<table border="0" cellpadding="0" cellspacing="0" class="calendar">
<tr><th colspan="7" class="month">January 2025</th></tr>
<tr><th class="thu">Thu</th><th class="fri">Fri</th><th class="sat">Sat</th><th

```

Python cho phép bạn in ra lịch ở định dạng html và bạn có thể thay đổi dữ liệu trong tệp html nếu bạn muốn

**Bước 3)** Để duyệt qua các ngày trong tháng chúng ta sử dụng itermonthday (2025,4), nó sẽ lấy ra các ngày trong tháng đó.

```
Demo.py x PythonDemo16.py x
1 import calendar
2
3 # Create a plain text calendar
4 c = calendar.TextCalendar(calendar.THURSDAY)
5 #str = c.formatmonth(2025, 1, 0, 0)
6 #print(str)
7
8 # Create an HTML formatted calendar
9 #hc = calendar.HTMLCalendar(calendar.THURSDAY)
10 #str = hc.formatmonth(2025, 1)
11 #print(str)
12
13 # loop over the days of a month
14 # zeroes indicate that the day of the month is from the previous month
15 for i in c.itermonthdays(2025, 4):
16     print(i)
17
18
19
```

**c.itermonthdays** sẽ lấy ra danh sách những ngày trong tháng 4 với một vài số 0 ở đầu. Đây là những ngày thuộc tháng trước vì biến c được gán giá trị tháng bắt đầu từ thứ 5

Run PythonDemo16

```
0
0
0
1
2
3
```

- Khi bạn thực thi mã nguồn để lấy số ngày trong một tháng cụ thể, ở đây là "April - Tháng Tư", bạn sẽ thấy có 30 ngày trong đầu ra nhưng sẽ xuất hiện vài số 0 ở đầu hoặc ở cuối chuỗi.
- Số 0 xuất hiện trong đầu ra là do đó là ngày thuộc tháng trước.
- Những số 0 đó xuất hiện ở đầu ra là do trong đoạn mã trên, bạn muốn tháng được bắt đầu in ra từ thứ 5 (Thursday), vì thế khi bạn gọi tới hàm "c.itermonthdays", nó sẽ bắt đầu đếm từ thứ 5. Nhưng thứ 5 không phải là ngày 1 tháng 4, nó có thể là ngày 28 hoặc 29 tháng 3, vì vậy khi bạn thực thi đoạn mã trên, nó sẽ bắt đầu đếm từ ngày 28 tháng 3 và các ngày sau đó cho tới ngày 1 tháng 4. Những ngày đó sẽ được đếm là 0 và trong đầu ra, bạn sẽ thấy các số 0 đó ở những giá trị đầu. Điều này cũng đúng trong trường hợp các ngày cuối tháng.
- Vì vậy, trừ những ngày từ 1-30, tất cả những ngày thuộc tháng trước hoặc sau đều được in ra là số 0.

**Bước 4)** Bạn có thể lấy dữ liệu ngày, tháng, năm,v.v. từ hệ thống hiện tại.

```
Demo.py x Python16.py x
1 #
2 #Example file for working with calendars
3 #
4
5 import calendar
6
7 # The calendar can give info based on local suc
8 for name in calendar.month_name:
9     print(name)
10
11 #for day in calendar.day_name:
12     #print(day)
13
```

Chúng ta có thể lấy ra tên tháng trong hệ thống

Run Python16  
"C:\Users\DK\Desktop\Python code\Python Code\Python16\venv\Script  
January  
February  
March  
April  
May  
June

- Đầu ra ở đây cho thấy rằng chúng ta đã in tên các tháng từ hệ thống hiện tại. Tương tự, bạn cũng có thể thêm tên các ngày trong tuần như dưới đây.
- Kết quả đầu ra sẽ phụ thuộc vào hệ thống hiện tại, giả sử nếu hệ thống hiện tại của bạn là một số quốc gia khác thì nó sẽ trả về các giá trị theo cài đặt cục bộ của quốc gia đó. Ở đây chúng ta có các tháng nên sẽ không có gì khác biệt nhưng nếu là tuần hoặc ngày, chắc chắn nó sẽ khác.

```
Demo.py x Python16.py x
1 #
2 #Example file for working with calendars
3 #
4
5 import calendar
6
7 # The calendar can give info based on time.
8 #for name in calendar.month_name:
9 #    print(name)
10
11 for day in calendar.day_name:
12     print(day)
13
```

Câu lệnh sẽ lấy ra thứ trong tuần từ lịch thuộc hệ thống hiện tại

Run Python16  
"C:\Users\DK\Desktop\Python code\Python16\ve  
Monday  
Tuesday  
Wednesday  
Thursday  
Friday  
Saturday  
Sunday

**Bước 5)** Bạn có thể lấy danh sách ngày cụ thể trong cả năm. Ví dụ, có một ngày kiểm toán vào mỗi thứ hai đầu tiên của tuần. Bạn muốn biết ngày thứ hai đầu tiên cho mỗi tháng. Bạn có thể sử dụng mã nguồn sau.

```

1 import calendar
2
3 for month in range (1,13):
4     # It retrieves a list of weeks that represent the month
5     mycal = calendar.monthcalendar(2025, month)
6     # The first MONDAY has to be within the first two weeks
7     week1 = mycal[0]
8     week2 = mycal[1]
9
10    if week1[calendar.MONDAY] != 0:
11        auditday = week1[calendar.MONDAY]
12    else:
13        # if the first MONDAY isn't in the first week, it must be
14        auditday = week2[calendar.MONDAY]
15
16    print ("%10s %2d" % (calendar.month_name[month], auditday))
17

```

```

Run PythonDemo16
"C:\Users\DK\Desktop\Python\Scripts"
January 6
February 3
March 3
April 7
May 5
June 2
July 7
August 4
September 1
October 6
November 3
December 1

```

Khi bạn thực thi đoạn mã, bạn sẽ thu được danh sách những ngày thứ hai đầu tiên của mỗi tháng cho cả năm 2025

- mycal = calendar.monthcalendar(2025,month) sẽ tạo lịch cho các tháng.
- Đặt các biến week1, week2 là tuần đầu tiên và tuần tiếp theo của lịch
- Kiểm tra xem tuần 1 có thứ hai không, nếu đúng thì đặt nó là ngày kiểm toán.
- Nếu tuần 1 không có thứ 2, chúng ta sẽ đặt ngày kiểm toán là thứ 2 đầu tiên trong tuần 2
- Kết quả đầu ra trả về ngày thứ hai đầu tiên của từng tháng
- Đối tượng "Cal" có độ dài nhất định, dựa trên số tuần có trong tháng. Trong trường hợp của chúng ta, nó sẽ là một hoặc hai khi đó thứ hai đầu tiên trong tuần thường là tuần đầu tiên, nhưng nếu điều đó không đúng thì ta cần xét tới tuần thứ 2. Hãy xem chi tiết tại sao chúng ta cần xem xét tuần thứ 2.
- Ở đây chúng ta đang sử dụng thứ hai là một hằng số không đổi của lịch, đối tượng lịch cung cấp cho bạn các hằng số đại diện cho: Chủ Nhật, Thứ Hai, Thứ Ba, v.v. Chúng ta đã thấy các giá trị này trước đây. Vì vậy, một ngày trong tuần được đại diện bởi một số khác 0, hãy nhớ số 0 có nghĩa là những ngày thuộc về một tháng khác. Bởi vậy, trong trường hợp này, nếu nó bằng không, nó sẽ là thứ hai thuộc về tháng trước. Nhưng nếu thứ Hai đầu tiên không bằng 0, điều đó có nghĩa là ngày kiểm toán của chúng ta nằm trong tuần đầu tiên của tháng. Mặt khác, nếu đó là 0, thì thứ hai đầu tiên không nằm trong tuần đầu

tiên của tháng, nó phải nằm trong tuần thứ hai.

- Khi đó, chúng ta sẽ đặt lại biến ngày kiểm toán vào thứ hai thuộc tuần thứ hai. Vì vậy, ngày kiểm toán sẽ chỉ nằm trong ngày thuộc tuần thứ nhất hoặc thứ hai.

Dưới đây là mã nguồn hoàn chỉnh

## Ví dụ sử dụng Python 2

```
import calendar
# Create a plain text calendar
c = calendar.TextCalendar(calendar.THURSDAY)
str = c.formatmonth(2025, 1, 0, 0)
print str

# Create an HTML formatted calendar
hc = calendar.HTMLCalendar(calendar.THURSDAY)
str = hc.formatmonth(2025, 1)
print str
# loop over the days of a month
# zeroes indicate that the day of the week is in a next month or overlapping month
for i in c.itermonthdays(2025, 4):
    print i

# The calendar can give info based on local such a names of days and months (full and abbreviated forms)
for name in calendar.month_name:
    print name
for day in calendar.day_name:
    print day
# calculate days based on a rule: For instance an audit day on the second Monday of every month
# Figure out what days that would be for each month, we can use the script as shown here
for month in range(1, 13):
# It retrieves a list of weeks that represent the month
    mycal = calendar.monthcalendar(2025, month)
# The first MONDAY has to be within the first two weeks
    week1 = mycal[1]
    week2 = mycal[2]
    if week1[calendar.MONDAY] != 0:
        auditday = week1[calendar.MONDAY]
    else:
        # if the first MONDAY isn't in the first week, it must be in the second week
        auditday = week2[calendar.MONDAY]
    print "%10s %2d" % (calendar.month_name[month], auditday)
```

## Ví dụ sử dụng Python 3

```
import calendar
# Create a plain text calendar
c = calendar.TextCalendar(calendar.THURSDAY)
str = c.formatmonth(2025, 1, 0, 0)
print(str)

# Create an HTML formatted calendar
hc = calendar.HTMLCalendar(calendar.THURSDAY)
str = hc.formatmonth(2025, 1)
print(str)
# loop over the days of a month
# zeroes indicate that the day of the week is in a next month or overlapping month
for i in c.itermonthdays(2025, 4):
    print(i)

# The calendar can give info based on local such a names of days and months (full and abbreviated forms)
for name in calendar.month_name:
    print(name)
```

```
for day in calendar.day_name:
    print(day)
# calculate days based on a rule: For instance an audit day on the second Monday of every month
# Figure out what days that would be for each month, we can use the script as shown here
for month in range(1, 13):
# It retrieves a list of weeks that represent the month
    mycal = calendar.monthcalendar(2025, month)
# The first MONDAY has to be within the first two weeks
    week1 = mycal[1]
    week2 = mycal[2]
    if week1[calendar.MONDAY] != 0:
        auditday = week1[calendar.MONDAY]
    else:
        # if the first MONDAY isn't in the first week, it must be in the second week
        auditday = week2[calendar.MONDAY]
    print("%10s %2d" % (calendar.month_name[month], auditday))
```

## Tổng kết:

- Trong Python, bạn có thể định dạng lịch theo cách bạn muốn bằng cách thay đổi ngày bắt đầu in ra mỗi tháng.
- In ra lịch ở định dạng HTML
- Lấy dữ liệu từ hệ thống cục bộ, như tháng hoặc ngày trong tuần
- Lấy danh sách ngày cụ thể trong cả năm