

## 15. Python DateTime

Trong Python, các lớp **date**, **time** và **datetime** cung cấp các hàm xử lý ngày, thời gian và khoảng thời gian. Date và datetime là đối tượng trong Python, vì vậy khi bạn thao tác với chúng, thực tế là bạn đang thao tác với các đối tượng chứ không phải chuỗi hoặc mốc thời gian. Bất cứ khi nào bạn cần thao tác với ngày hoặc thời gian, bạn cần nạp hàm datetime.

Các lớp datetime trong Python được phân thành 5 nhóm chính.

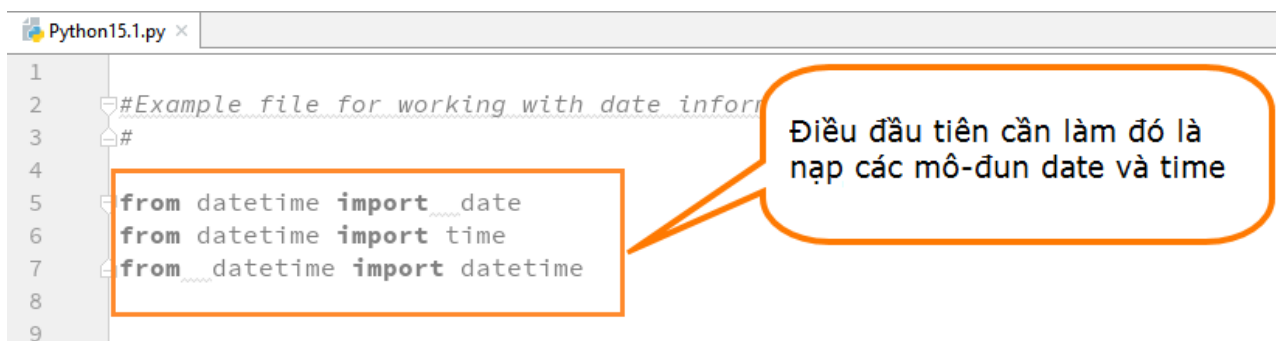
- date – Chỉ chứa thao tác với ngày (Tháng, ngày, năm)
- time – Chỉ thao tác với thời gian (Giờ, phút, giây, micro giây)
- datetime – Kết hợp giữa ngày và thời gian (Tháng, ngày, năm, giờ, giây, micro giây)
- timedelta – Một khoảng thời gian được sử dụng để thao tác với ngày
- tzinfo – Một lớp trừu tượng để thao tác với các múi giờ

Trong bài này, chúng ta sẽ tìm hiểu về:

- Cách sử dụng lớp Date & DateTime
- In ngày bằng date.today ()
- Ngày và giờ hiện tại trong Python: now() today()
- Định dạng ngày và giờ cho đầu ra với Strftime()
- Cách sử dụng Đối tượng Timedelta

### Cách sử dụng lớp Date & DateTime

**Bước 1)** Trước khi chạy mã nguồn với datetime, điều quan trọng cần làm là nạp mô-đun date và time như hình dưới đây:



```
1
2 #Example file for working with date inform
3 #
4
5 from datetime import date
6 from datetime import time
7 from datetime import datetime
8
9
```

Các câu lệnh nạp (import) trên sẽ cung cấp các hàm được định nghĩa sẵn trong thư viện Python, mà qua đó bạn có thể thao tác với ngày tháng và thời gian mà không cần viết thêm mã nguồn.

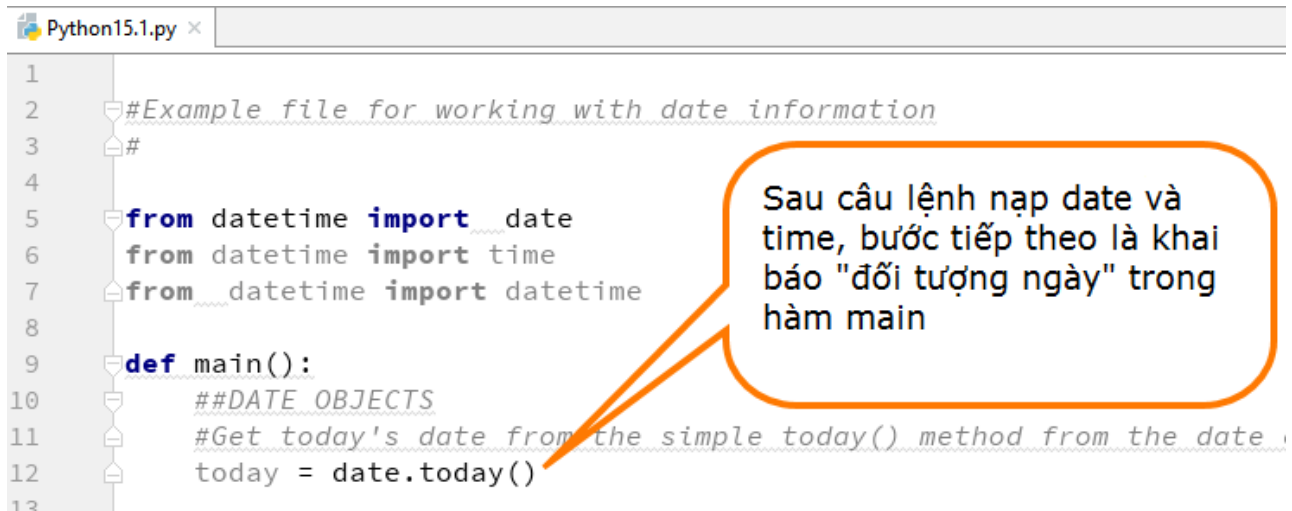
Trước khi thực hiện mã nguồn với datetime, hãy xem dòng lệnh dưới đây:

```
from datetime import date
```

Câu lệnh này thông báo cho trình thông dịch Python rằng hãy nạp lớp date từ mô-đun datetime.

Chúng ta sẽ không cần viết thêm mã nguồn cho chức năng ngày tháng mà chỉ cần nạp chúng trước khi sử dụng.

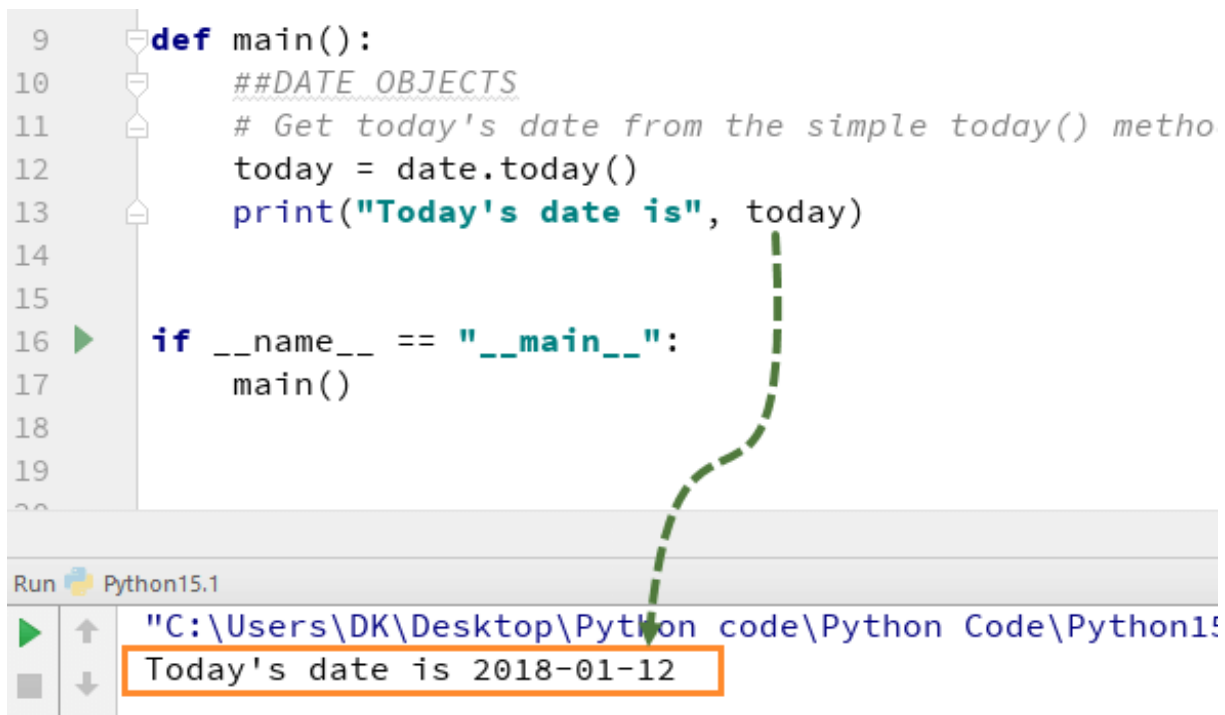
**Bước 2)** Tiếp theo chúng ta sẽ tạo một thực thể cho đối tượng này.



```
1
2 #Example file for working with date information
3 #
4
5 from datetime import date
6 from datetime import time
7 from datetime import datetime
8
9 def main():
10     ##DATE OBJECTS
11     #Get today's date from the simple today() method from the date
12     today = date.today()
13
```

Sau câu lệnh nạp date và time, bước tiếp theo là khai báo "đối tượng ngày" trong hàm main

**Bước 3)** Tiếp theo, chúng ta sẽ in ra ngày tháng và chạy đoạn mã.



```
9 def main():
10     ##DATE OBJECTS
11     # Get today's date from the simple today() metho
12     today = date.today()
13     print("Today's date is", today)
14
15
16 if __name__ == "__main__":
17     main()
18
19
20
```

Run Python15.1

"C:\Users\DK\Desktop\Python code\Python Code\Python15  
Today's date is 2018-01-12

Đầu ra thu được đúng như mong đợi.

## In ngày bằng date.today()

Hàm date.today có một số đặc tính đi kèm. Chúng ta có thể in ngày / tháng / năm riêng lẻ và rất nhiều thứ khác.

Xét ví dụ sau:

```
9 def main():
10     ##DATE OBJECTS
11     # Get today's date from the simple today() method from the date class
12     today = date.today()
13     print("Today's date is", today)
14
15
16     # print out the date's individual component
17     today = date.today()
18     print("Date Components:", today.day, today.month, today.year)
19
20     if __name__ == "__main__":
21         main()
22
```

Run Python15.1  
"C:\Users\DK\Desktop\Python code\Python  
Date Components: 12 1 2018  
Today's date is 2018-01-12

Python cho phép thao tác với các đối tượng ngày tháng riêng lẻ. Trong trường hợp này, chúng ta liệt kê ngày, tháng và năm một cách độc lập. Kết quả in ra đúng như chúng ta mong muốn.

## Số Weekday của hôm nay

Hàm `date.today()` cũng cung cấp cho bạn số weekday (ngày thứ mấy trong tuần). Dưới đây là bảng Weekday, nó bắt đầu từ thứ hai là số 0 tới chủ nhật là số 6

Ngày	Số Weekday
Thứ hai	0
Thứ ba	1
Thứ tư	2
Thứ năm	3
Thứ sáu	4
Thứ bảy	5
Chủ nhật	6

Số weekday rất hữu dụng cho các mảng sử dụng chỉ số phụ thuộc vào các ngày trong tuần.

```
Python15.1.py x
17
18 #retrive today's weekday (0=Monday, 6=Sunday)
19 print("Today's Weekday#", today.weekday())
20
21
22 if __name__ == "__main__":
23     main()
24
25
26
```

Trong trường hợp này, today.weekday() sẽ trả về số weekday là 4.

```
Run Python15.1
"C:\Users\DK\Desktop\Python code\Python Code\Python15\venv\Scripts\python.exe"
Date Components: 12 1 2018
Today's Weekday#: 4
Today's date is 2018-01-12
```

## Ngày và giờ hiện tại của Python: now() today()

**Bước 1)** Giống như đối tượng Date, chúng ta cũng có thể sử dụng **đối tượng Datetime** trong Python. Nó đưa ra ngày cùng với thời gian tính bằng giờ, phút, giây và mili giây.

```
4 from datetime import date
5 from datetime import time
6 from datetime import datetime
7
8 def main():
9     ##DATE OBJECTS
10    #Get today's date from the datetime class
11    today = datetime.now()
12    print("The current date and time is", today)
13
14
15 if __name__=="__main__":
16    main()
17
```

Giống đối tượng date, chúng ta có "đối tượng datetime" trong Python. Nó cho phép chúng ta sử dụng hàm thời gian kèm theo ngày. Ở đây, chúng ta in ra ngày với giờ ở đầu ra

```
Run Python15.1
"C:\Users\DK\Desktop\Python code\Python Code\Python15\venv\Scripts\python.exe"
The current date and time is 2018-01-12 11:59:35.976715
```

Khi chúng ta thực thi mã nguồn với datetime, nó sẽ in ra ngày và giờ hiện tại.

**Bước 2)** Với "đối tượng Datetime", bạn cũng có thể gọi tới lớp thời gian.

Giả sử chúng ta chỉ muốn in ra thời gian hiện tại mà không có ngày.

```
t = datetime.time(datetime.now())
```

- Chúng ta đã nạp lớp thời gian time trước đó. Chúng ta sẽ gán cho nó giá trị thời gian hiện tại với datetime.now().

- Chúng ta đang gán giá trị thời gian hiện tại cho biến t.

Và kết thu được chỉ có thời gian. Cùng chạy chương trình.

```

11 #Get the current time
12 t = datetime.time(datetime.now())
13 print("The current time is",t)
14
15
16 if __name__=="__main__":
17     main()
18
19

```

Run Python15.1

```

"C:\Users\DK\Desktop\Python code\Python Code\Python15\venv\Script
The current time is 12:15:09.438851

```

Bạn cũng có thể lấy ra giờ hiện tại thông qua đối tượng datetime. Chương trình sẽ in ra kết quả là thời gian hiện tại

Bạn có thể thấy chúng ta có được cả ngày và giờ. Ở dòng tiếp theo, chúng ta chỉ lấy ra giá trị thời gian.

**Bước 3)** Chúng ta sẽ áp dụng chỉ số weekday cho mảng danh sách weekday để tìm ra hôm nay là thứ mấy.

- Toán tử weekday(wd) được gán một số từ (0-6) tùy thuộc vào ngày hiện tại trong tuần là thứ mấy. Ở đây chúng ta khai báo một mảng cho các ngày trong tuần (Mon, Tue, Wed... Sun).
- Sử dụng giá trị chỉ số đó để biết ngày hiện tại là thứ mấy. Trong trường hợp này, giá trị thu được là #2, tức là ngày thứ tư, vì vậy kết quả đầu ra là "Which is a Wednesday".

```

12 #weekday returns 0 (monday) through 6 (sunday)
13 wd = date.weekday(today)
14 #Days start at 0 for monday
15 days= ["monday", "tuesday", "wednesday", "thursday", "friday", "saturday", "sunday"]
16 print("Today is day number %d" % wd)
17 print("which is a " + days[wd])
18
19 if __name__ == "__main__":
20     main()
21
22

```

Run Python15.1

```

"C:\Users\DK\Desktop\Python code\Python Code\Python15\Scripts\python.exe" "C:/P
Today is day number 4
which is a friday

```

Chúng ta dùng chỉ số để xác định hôm nay là ngày nào và dựa trên kết quả đó, chúng ta sẽ biết được hôm nay là thứ mấy

**Đây là đoạn mã hoàn chỉnh để lấy ra ngày và giờ hiện tại sử dụng hàm now trong mô-đun datetime.**

```

from datetime import date
from datetime import time
from datetime import datetime
def main():

```

```

##DATETIME OBJECTS
#Get today's date from datetime class
today=datetime.now()
#print (today)
# Get the current time
#t = datetime.time(datetime.now())
#print "The current time is", t
#weekday returns 0 (monday) through 6 (sunday)
wd=date.weekday(today)
#Days start at 0 for monday
days= ["monday", "tuesday", "wednesday", "thursday", "friday", "saturday", "sunday"]
print("Today is day number %d" % wd)
print("which is a " + days[wd])

if __name__ == "__main__":
    main()

```

## Định dạng ngày và giờ cho đầu ra với strftime()

Giờ chúng ta đã học được cách sử dụng đối tượng datetime và date trong Python. Chúng ta sẽ tiến thêm một bước và tìm hiểu cách sử dụng hàm định dạng để định dạng thời gian và ngày.

**Bước 1)** Đầu tiên chúng ta sẽ tìm hiểu một cách định dạng năm đơn giản. Cùng xem ví dụ dưới đây:

```

Python15.1.py x
1 #
2 #Example file for formatting time and date output
3 #
4 from datetime import datetime
5 def main():
6     #Times and dates can be formatted using a set of predefined string
7     #Control codes
8     now= datetime.now() #get the current date and time
9
10    ##### Date Formatting #####
11    #%%y%%Y - Year, %a/%A- weekday, %b/%B- month, %d - day of month
12    print (now.strftime("%Y")) #full year with century
13
14    if __name__ == "__main__":
15        main()
16
17
18
19
20

```

Hàm "strftime" được sử dụng để định dạng các tham số khác nhau của giờ, ngày, tháng và năm. Ở đây chúng ta in ra năm kèm theo thế kỷ, tức là "2018" và sẽ thay đổi nó thành chỉ còn "18" trong bước tiếp theo

Run Python15.1  
 C:\Users\DK\Desktop\Python  
 2018

- Chúng ta sử dụng "hàm strftime" để định dạng.
- Hàm này sử dụng các **mã điều khiển** khác nhau để trả về kết quả.
- Mỗi mã điều khiển giống với các tham số khác nhau như năm, tháng, thứ trong tuần và ngày [(%y /%Y - Năm), (%a /%A- thứ trong tuần), (%b /%B- tháng), (%d - ngày của tháng)].

- Trong trường hợp của chúng ta, đó là ("% Y") đại diện cho năm, chương trình sẽ in ra năm cùng với thế kỷ (ví dụ 2018).

**Bước 2)** Bây giờ nếu bạn thay thế ("% Y") bằng chữ thường, nghĩa là ("% y") và thực thi đoạn mã, đầu ra sẽ chỉ hiển thị (18) chứ không phải (2018). Thế kỷ của năm sẽ không được hiển thị như trong ảnh bên dưới.

```

10 ##### Date Formatting #####
11
12 # %y%Y - Year, %a/%A- weekday, %b/%B- month, %d - day of month
13 print (now.strftime("%y")) #full year with century
14
15 if __name__ == "__main__":
16     main()
17
18
19
20

```

Khi chúng ta thay thế "%Y" bằng ký tự thường "%y", kết quả in ra sẽ thay đổi từ "2018" thành chỉ còn năm là "18"

Run Python 15.1  
 "C:\Users\DK\Desktop\Python\venv\Scripts\pyth  
 18

**Bước 3)** Hàm Strf có thể khai báo riêng ngày, thứ, tháng và năm. Ngoài ra chỉ với thay đổi nhỏ trong mã điều khiển trong hàm strftime bạn có thể định dạng lại kiểu văn bản hiển thị.

```

1 #
2 #Example file for formatting time and date output
3 #
4 from datetime import datetime
5
6 def main():
7     #Times and dates can be formatted using a set of predefined string
8     #Control codes
9     now= datetime.now() #get the current date and time
10    print (now.strftime("%a,%d %B,%y"))
11
12
13 if __name__ == "__main__":
14     main()
15
16
17

```

Thứ

Tháng

Năm

Ngày

Với hàm strftime, bạn có thể khai báo ngày, thứ, tháng và năm một cách riêng biệt

Run Python 15.1  
 "C:\Users\DK\Desktop\Python code\Python Code\Python15\  
 Fri,12 January,18

Bên trong hàm strftime nếu bạn thay thế (%a) bằng chữ A, nghĩa là (%A), đầu ra sẽ in ra là "Friday" thay vì chỉ viết tắt "Fri".



```

7 #Control codes
8 now= datetime.now() #get the current date and time
9 print (now.strftime("%A %d %B,%y"))
10
11
12
13
14 if __name__ == "__main__":
15     main()
16
17
18

```

Khi thay thế chữ a thường trong (%a) bằng (%A), chương trình sẽ in ra "Friday" thay vì chỉ viết tắt là "Fri"

Run Python15.1  
 "C:\Users\DK\Desktop\Python code\Python Code\Python15\venv"  
 Friday 12 January,18

**Bước 4)** Với sự trợ giúp của chức năng "Strftime", chúng ta cũng có thể truy xuất thời gian, ngày hoặc cả hai của hệ thống.

1. %C- cho biết ngày và giờ địa phương
2. %x- cho biết ngày địa phương
3. %X- cho biết giờ địa phương

```

8 #Control codes
9 now= datetime.now() #get the current date and time
10 # %c - local date and time, %x-local's date, %X- local's time
11 print(now.strftime("%c"))
12 print(now.strftime("%x"))
13 print(now.strftime("%X"))
14
15
16
17
18 if __name__ == "__main__":
19     main()
20
21

```

Chúng ta có thể sử dụng hàm strftime để lấy ra ngày giờ địa phương, ngày địa phương hoặc giờ địa phương của hệ thống đang chạy

Run Python15.1  
 "C:\Users\DK\Desktop\Python code\Python Code\Python15\venv"  
 Fri Jan 12 13:08:33 2018  
 01/12/18  
 13:08:33

Bạn có thể thấy chương trình in ra kết quả đúng như mong muốn.

**Bước 5)** "Hàm strftime" cho phép bạn trả về thời gian dưới định dạng 24 giờ hoặc 12 giờ.



```

8      #Control codes
9      now= datetime.now() #get the current
10     # %c - local date and time, %x-local
11     # print now.strftime("%c")
12     # print now.strftime("%x")
13     # print now.strftime("%X")
14
15
16     ##### Time Formatting #####
17
18     # %I/%H - 12/24 Hour, %M - minute, %S - second, %p - local's AM/PM
19     print(now.strftime("%I:%M:%S %p")) 1 12-Hour:Minute:Second:AM
20     print(now.strftime("%H:%M")) 2 24-Hour:Minute
21
22     if __name__ == "__main__":
23         main()
24

```

Với hàm strftime, bạn có thể làm được rất nhiều thứ như in ra thời gian ở bất kỳ định dạng nào (24 giờ hoặc 12 giờ)

Run Python15.1  
 "C:\Users\DK\Desktop\Python code\Python Code\Python15\venv\Scripts\python  
 01:19:18 PM 1  
 13:19 2

Chỉ bằng cách xác định mã điều khiển như %I/%H cho giờ, %M cho phút, %S cho giây, ta có thể trả về thời gian dưới các định dạng khác nhau.

**Định dạng 12h** được khai báo như sau: [print now.strftime("%I:%M:%S %P) ]

**Định dạng 24h** được khai báo như sau: [print now.strftime("%H:%M")]

**Đây là mã hoàn chỉnh để chuyển đổi datetime thành đối tượng dạng chuỗi.**

```

#
#Example file for formatting time and date output
#
from datetime import datetime
def main():
    #Times and dates can be formatted using a set of predefined string
    #Control codes
    now= datetime.now() #get the current date and time
    # %c - local date and time, %x-local's date, %X- local's time
    print(now.strftime("%c"))
    print(now.strftime("%x"))
    print(now.strftime("%X"))
    ##### Time Formatting #####
    # %I/%H - 12/24 Hour, %M - minute, %S - second, %p - local's AM/PM
    print(now.strftime("%I:%M:%S %p")) # 12-Hour:Minute:Second:AM
    print(now.strftime("%H:%M")) # 24-Hour:Minute

if __name__ == "__main__":
    main()

```

## Cách sử dụng Đối tượng Timedelta

Với các đối tượng timedelta, bạn có thể ước tính thời gian cho tương lai và quá khứ. Nói cách khác, đó là một khoảng thời gian để dự đoán bất kỳ ngày, thứ hoặc thời gian nào.

**Hãy nhớ rằng hàm này không phải để in ra thời gian hay ngày tháng, mà là để TÍNH TOÁN trong tương lai hoặc quá khứ.** Cùng xem ví dụ sau để hiểu rõ hơn.

**Bước 1)** Để chạy đối tượng Timedelta, bạn cần khai báo câu lệnh nạp trước đó.

```
1 #
2 # Example file for working with timedelta objects
3 #
4 from datetime import date
5 from datetime import time
6 from datetime import datetime
7 from datetime import timedelta
8
9 # construct a basic timedelta and print it
10 print_(timedelta(days=365, hours=8, minutes=15))
11
12
13
14
15
16
17
18
```

1) Viết câu lệnh nạp bao gồm cả timedelta  
2) Câu lệnh in ra kết quả sử dụng timedelta

Run Python15.1  
"C:\Users\DK\Desktop\Python code\Python Code\Python15\venv\Scripts\python.exe  
365 days, 8:15:00

- Viết câu lệnh nạp cho timedelta.
- Giờ hãy viết đoạn mã để in ra đối tượng từ timedelta như hình ảnh.
- Chạy đoạn mã nguồn trên. Timedelta biểu diễn một khoảng 365 ngày, 8 giờ và 15 phút giống như những gì được in ra.

Bạn vẫn thấy thắc mắc? Bước tiếp theo sẽ giúp bạn hiểu rõ hơn.

**Bước 2)** Cùng lấy ra ngày và giờ của ngày hôm nay để kiểm tra xem câu lệnh nạp có chạy đúng hay không. Khi đoạn mã được thực thi, nó sẽ in ra ngày hôm nay, điều đó có nghĩa là câu lệnh nạp đã chạy đúng.

```
9 # construct a basic timedelta and print it
10 print_(timedelta(days=365, hours=8, minutes=15))
11 # print today's date
12 print_("today is: " + str(datetime.now()))
13
14
15
16
17
18
```

Run Python15.1  
"C:\Users\DK\Desktop\Python code\Python Code\Pytl  
365 days, 8:15:00  
today is: 2018-01-12 14:40:05.569077

**Bước 3)** Chúng ta sẽ xem cách tìm ra ngày này 1 năm sau thông qua đối tượng timedelta. Khi chúng ta chạy đoạn mã này, kết quả sẽ trả về như mong muốn.

```
Python15.1.py x
8
9 # construct a basic timedelta and print it
10 print_(timedelta(days=365, hours=8, minutes=15))
11
12 # print today's date
13 print_("today is: " + str(datetime.now()))
14 # print today's date one year from now
15
16 print_("one year from now it will be:" + str(datetime.now() + timedelta(days=365)))
17
```

timedelta thêm cộng thêm 365 ngày vào ngày hiện tại

```
Run Python15.1
"C:\Users\DK\Desktop\Python code\Python Code\Python15\venv\Scripts\python.exe" "C:/Pyt
365 days, 8:15:00
today is: 2018-01-12 14:43:18.407611
one year from now it will be:2019-01-12 14:43:18.407611
```

**Bước 4)** Một ví dụ khác về cách sử dụng timedelta để tính ngày trong tương lai từ ngày và giờ hiện tại

```
14
15 # print today's date one year from now
16 print_("one year from now it will be:" + str(datetime.now() + timedelta(days=365)))
17
18 # create a timedelta that uses more than one argument
19 print_("in one week and 4 days it will be " + str(datetime.now() + timedelta(weeks=1, days=4)))
20
21
```

Đoạn mã này sử dụng timedelta để in ra ngày sau 1 tuần 4 ngày nữa tính từ hôm nay

```
Run Python15.1
"C:\Users\DK\Desktop\Python code\Python Code\Python15\venv\Scripts\python.exe" "C:/Pyt
365 days, 8:15:00
today is: 2018-01-12 14:48:25.567629
one year from now it will be:2019-01-12 14:48:25.567629
in one week and 4 days it will be 2018-01-23 14:48:25.567629
```

**Bước 5)** Cùng xem một ví dụ phức tạp hơn. Chúng ta cùng tìm xem Tết đã trôi qua được bao nhiêu ngày. Đây là cách làm.

- Sử dụng today = date.today () chúng ta sẽ có được ngày hôm nay.
- Chúng ta biết rằng Tết luôn luôn diễn ra vào ngày 1/1, nhưng năm thì có thể khác. Sử dụng nyd = date(today.year,1,1) chúng ta sẽ lưu ngày Tết vào biến nyd.
- if nyd < today: kiểm tra xem ngày hiện tại có lớn hơn ngày Tết hay không. Nếu có thì thực hiện vòng lặp while.
- ((today-nyd) .days) trả về số ngày đã qua giữa ngày hiện tại và năm mới.

```
15
16 today = date.today() # get today's date
17 nyd = date(today.year, 1, 1) # get New Year Day for the same year
18 # use date comparison to see if New Year Day has already gone for this year
19 # if it has, use the replace() function to get the date for next year
20
21 if nyd < today:
22     print("New Year day is already went by %d days ago" % ((today - nyd).days))
```

Run Python15.1  
"C:\Users\DK\Desktop\Python code\Python Code\  
365 days, 8:15:00  
today is: 2018-01-12 14:56:35.604281  
New Year day is already went by 11 days ago

Với hàm timedelta, bạn có thể tính toán bao nhiêu ngày còn lại hay đã qua cho bất kỳ sự kiện nào. Ở đây chúng ta tính số ngày đã qua kể từ ngày Tết.

Chương trình in ra "New Year Day already went by 11 days ago."

Đây là đoạn mã hoàn chỉnh:

```
#
# Example file for working with timedelta objects
#
from datetime import date
from datetime import time
from datetime import datetime
from datetime import timedelta

# construct a basic timedelta and print it
print (timedelta(days=365, hours=8, minutes=15))
# print today's date
print ("today is: " + str(datetime.now()))
# print today's date one year from now
print ("one year from now it will be:" + str(datetime.now() + timedelta(days=365)))
# create a timedelta that uses more than one argument
# print (in one week and 4 days it will be " + str(datetime.now() + timedelta(weeks=1, days=4)))
# How many days until New Year's Day?
today = date.today() # get today's date
nyd = date(today.year, 1, 1) # get New Year Day for the same year
# use date comparison to see if New Year Day has already gone for this year
# if it has, use the replace() function to get the date for next year
if nyd < today:
    print ("New Year day is already went by %d days ago" % ((today - nyd).days))
```

## Ví dụ sử dụng Python 2

```
from datetime import date
from datetime import time
from datetime import datetime
def main():
    ##DATETIME OBJECTS
    #Get today's date from datetime class
    today=datetime.now()
    #print today
    # Get the current time
    #t = datetime.time(datetime.now())
    #print "The current time is", t
    #weekday returns 0 (monday) through 6 (sunday)
    wd = date.weekday(today)
    #Days start at 0 for monday
```

```
days= ["monday","tuesday","wednesday","thursday","friday","saturday","sunday"]
print "Today is day number %d" % wd
print "which is a " + days[wd]
```

```
if __name__ == "__main__":
    main()
```

```
#
#Example file for formatting time and date output
#
from datetime import datetime
def main():
    #Times and dates can be formatted using a set of predefined string
    #Control codes
    now= datetime.now() #get the current date and time
    #%c - local date and time, %x-local's date, %X- local's time
    print now.strftime("%c")
    print now.strftime("%x")
    print now.strftime("%X")
##### Time Formatting #####
    %I/%H - 12/24 Hour, %M - minute, %S - second, %p - local's AM/PM
    print now.strftime("%I:%M:%S %p") # 12-Hour:Minute:Second:AM
    print now.strftime("%H:%M") # 24-Hour:Minute

if __name__ == "__main__":
    main()
```

```
#
# Example file for working with timedelta objects
#
from datetime import date
from datetime import time
from datetime import datetime
from datetime import timedelta

# construct a basic timedelta and print it
print timedelta(days=365, hours=8, minutes=15)
# print today's date
print "today is: " + str(datetime.now())
# print today's date one year from now
print "one year from now it will be:" + str(datetime.now() + timedelta(days=365))
# create a timedelta that uses more than one argument
# print "in one week and 4 days it will be " + str(datetime.now() + timedelta(weeks=1, days=4))
# How many days until New Year's Day?
today = date.today() # get today's date
nyd = date(today.year, 1, 1) # get New Year Day for the same year
# use date comparison to see if New Year Day has already gone for this year
# if it has, use the replace() function to get the date for next year
if nyd < today:
    print "New Year day is already went by %d days ago" % ((today - nyd).days)
```

## Tổng kết

Để thao tác với ngày và giờ theo cách đơn giản hoặc phức tạp, mô-đun datetime cung cấp các lớp hoặc danh mục khác nhau bao gồm:

- date – Chỉ chứa thao tác với ngày (Tháng, ngày, năm)
- time – Chỉ thao tác với thời gian (Giờ, phút, giây, micro giây)
- datetime – Kết hợp giữa ngày và thời gian (Tháng, ngày, năm, giờ, giây, micro giây)
- timedelta – Một khoảng thời gian được sử dụng để thao tác với ngày

- tzinfo – Một lớp trừu tượng để thao tác với các múi giờ

## **Sử dụng đối tượng datetime**

- Nạp đối tượng datetime trước khi thực thi mã nguồn là bắt buộc
- Sử dụng hàm date.today để in ra giá trị ngày/tháng/năm riêng biệt cũng như trả về chỉ số của ngày.
- Sử dụng đối tượng date.time để lấy ra thời gian theo giờ, phút, giây hoặc mili giây

## **Định dạng thời gian in ra với hàm “strftime”**

- Sử dụng "hàm thời gian strftime" để thay đổi định dạng của năm
- In riêng ngày, ngày, tháng và năm,
- Trả về thời gian dưới định dạng 12 giờ hoặc 24 giờ

## **Đối tượng Timedelta**

- Với các đối tượng timedelta, bạn có thể ước tính thời gian cho cả tương lai và quá khứ
- Tính số ngày còn lại cho tới một ngày đặc biệt (sinh nhật) kể từ thời điểm hiện tại
- Tính tổng số ngày đã qua tính từ một ngày đặc biệt (sinh nhật) kể từ thời điểm hiện tại