

Business Programming (using Python)

Zhaohu (Jonathan) Fan

Information Technology Management

Scheller College of Business

Georgia Institute of Technology

September 14, 2023

Main topics

- Dates & Time
 - Exercises

Dates & Time

Question(s)?

- Q: How can we input a date in the **'YYYY-MM-DD' format** using Python's **datetime** library?

Quick answer

- A: We can use the `datetime.strptime()` function to parse **a string** into **a datetime object**.

Example #1

Output

```
Please enter the date in the format YYYY-MM-DD: 2023-09-14  
You entered: 2023-09-14 00:00:00
```

Example #1

- There is a **datetime** module (needs to be imported).
 - Import entire `datetime` module and give it an alias as `dt`

Code

```
# using Timedelta  
import datetime as dt  
  
# Input the date as a string in the format "YYYY-MM-DD"  
date_string = input("Please enter the date in the format YYYY-MM-DD: ")  
  
# Parse the string into a datetime object  
date_object = dt.datetime.strptime(date_string, "%Y-%m-%d")  
  
print("You entered:", date_object)
```

Example #1

- Python's built-in `input()` function to read a line of text (a string).
 - The value that the user enters will be stored as **a string** in the variable `date_string`
 - For example, if the user enters 2023-09-14, `date_string` will be `'2023-09-14'`.

Code

```
# using Timedelta
import datetime as dt

# Input the date as a string in the format "YYYY-MM-DD"
date_string = input("Please enter the date in the format YYYY-MM-DD: ")

# Parse the string into a datetime object
date_object = dt.datetime.strptime(date_string, "%Y-%m-%d")

print("You entered:", date_object)
```

Example #1

- The `datetime.strptime()` function takes two arguments:
 - The string that you want to parse ("date_string").
 - The format code that describes the expected format of the string ("%Y-%m-%d").
 - Note: The above code does not handle invalid input; you might want to include additional error checking to make it more robust.

Code

```
# using Timedelta
import datetime as dt

# Input the date as a string in the format "YYYY-MM-DD"
date_string = input("Please enter the date in the format YYYY-MM-DD: ")

# Parse the string into a datetime object
date_object = dt.datetime.strptime(date_string, "%Y-%m-%d")

print("You entered:", date_object)
```

Example #1

- The `print()` function takes two (multiple) arguments and prints them to the console separated by a space:
 - The first argument is the string `"You entered:"`
 - The second argument is `date_object`, which is a datetime object representing the date the user entered.

Code

```
# using Timedelta
import datetime as dt

# Input the date as a string in the format "YYYY-MM-DD"
date_string = input("Please enter the date in the format YYYY-MM-DD: ")

# Parse the string into a datetime object
date_object = dt.datetime.strptime(date_string, "%Y-%m-%d")

print("You entered:", date_object)
```

Example #1

Output

```
Please enter the date in the format YYYY-MM-DD: 2023-09-14  
You entered: 2023-09-14 00:00:00
```

Example #1

- Q: What if we want to **display** just the date part of the **datetime object without the time component**?

Example #1

- use the `date()` method to extract just the **date part** from the **datetime object**
 - By calling `date()` on the datetime object, you get a date object containing just the year, month, and day.

Code

```
# using Timedelta
import datetime as dt

# Input the date as a string in the format "YYYY-MM-DD"
date_string = input("Please enter the date in the format YYYY-MM-DD: ")

# Parse the string into a datetime object
date_object = dt.datetime.strptime(date_string, "%Y-%m-%d")

print("You entered:", date_object.date())
```

Example #1

Output

```
Please enter the date in the format YYYY-MM-DD: 2023-09-14  
You entered: 2023-09-14
```

Example #1

- How do I use the `.now()` function to show the current time if I am in the EST time zone?

Output

```
import datetime as dt
# install the pytz package if you haven't already:
# pip install pytz
import pytz
# Initialize the time zone
est = pytz.timezone('US/Eastern')
# Get the current time in UTC and then convert to EST
current_datetime_est = dt.datetime.now(pytz.utc).astimezone(est)
# Print the current date and time in EST
print("Current date and time (EST):", current_datetime_est.strftime("%Y-%m-%d %H:%M:%S"))
```

Example #1

Output

```
Current time (EST): 2023-09-14 16:43:24
```

Exercise: Dates & Time

- Please click on the link provided below.
 - [In-Class Exercise](#)