Python, Day 6: Modules

Andrew Bydlon

January 16, 2019

Review: Functions

Recall from last time, we learned how to produce functions in python for repeated use:

Syntax (Functions)

```
def FunctionName(var1, var2, var3,. . . ):
    command1
    command2
    ReturnStatement
```

Here the *ReturnStatement* can be anything from nothing, to a print statement, to a return statement.

Solution from last time

Example (Gathering a list)

```
def getlist():
    List=[]
    while True:
        user = input("Enter a value (or q to exit): ")
        if List==[] and user=="q":
            continue
        elif user=="q":
            break
        List += [user]
    return(List)
⋮
```

Example (Trimming a list)

```
⋮
def listtrimmer(List):
    NewList = []
    while List != []:
        Char = List[0]
        NewList += [Char]
        while List.count(Char) != 0:
            List.remove(Char)
    return(NewList)

def listreverser(List):
    List.reverse()
    return(List)

print(listreverser(listtrimmer(getlist())))
```

Modules!

A **module** is a file storing several functions. They are simply .py files like any ordinary program.

You can bring in files with the import command. We have seen this done before with the **math** module.

Example (math)

```
> > > import math
> > > math.pi
3.141592653589793
> > > math.ceil(123.4684561623)
124
> > > help(math)
```

Provides a list of the documentation for the math module

Other modules: datetime

The datetime module can provide useful information from the users local clock.

Example (datetime)

```
> > > import datetime
> > > print(datetime.date.now())
2019-01-16
> > > print(datetime.datetime.now().strftime("%y-%m-%d-%H-%M"))
# To print year-month-day-hour-minute
19-01-16-11-10
> > > datetime.date.today().strftime("%B")
‘January’
> > > datetime.date.today().strftime("%A")
‘Sunday’
```

Other modules: random (number generator)

Example (random)

```
> > > import random
> > > random.random() # Returns a random float between 0 and 1.
0.7035274803062691
> > > random.random()*100
47.69716175684814
> > > random.randint(0,5)# Returns a random integer in 0,1,. . . ,5.
3
> > > random.randint(0,5)
5
> > > MyList = [’blue’, ’purple’, ’green’, ’red’, ’yellow’, ’orange’]
> > > random.choice(MyList)
’purple’
> > > random.shuffle(MyList)
> > > MyList
[’green’, ’red’, ’orange’, ’blue’, ’yellow’, ’purple’]
```

Importing a specific function

Sometimes, you need quick access to a specific function or variable from a module. This can be done via the from command:

Syntax

from MyModule import var1, var2, . . . , varN

Example (Random Circle)

```
> > > from random import random, randint
> > > from math import pi
> > > Radius = random()
> > > print("A circle of radius", Radius, "has circumference",
2*pi*Radius)
A circle of radius 0.12034681845494088 has circumference
0.7561613614818936
```

Note: We MUST drop the math. or random. prefix when using from.

Making your own modules

Inside of a project, you can store multiple files (in the same location). You can import a previous file, or a specific function or variable, using the same command:

Example (AndrewFunctionFile.py)

```
def listtrimmer(List):
    NewList = []
    while List != []:
        Char = List[0]
        NewList += [Char]
        while List.count(Char) != 0:
            List.remove(Char)
    return(NewList)
⋮
```

Making your own modules

Example (AndrewRunFile.py)

```
import AndrewFunctionFile

MyList = AndrewFunctionFile.getlist()
MyList = AndrewFunctionFile.listtrimmer(MyList)
MyList = AndrewFunctionFile.listreverse(MyList)

print(MyList)
```

One other small note: you can import everything (to save a few keystrokes) using the following syntax:

```
from MyModule import *
```

Assignment 10

Let's make a card dealing program! For this project, I would like you to create the following:

1. A main file where you take the users input of how many cards to deal, and prints the result of the roll.
2. A module of your creation with functions to perform the above tasks:
   - Check if a card is already is a list.
   - Create a list of certain number of cards (a hand)

You can import the desired random functions in either file. You may want to pull your suit from the list ["Club","Spade","Heart","Diamond"] and pull the size from the integers between 2 and 14 (Ace is 14, King is 13, etc).

It may be helpful to declare cards as a 2-element list.

Upload your 2 .py files!