Functions and Modules

Dr. Baldassano
chrisb@princeton.edu
Yu's Elite Education

Last week recap

- The print function
- Variables
  - Assignment operator =
  - Variable types
- The input function

What is a function?

- A function is like a mini-program: it takes some information, and performs some action
- Variables passed into a function are called *arguments*
- Functions in python are called like:

```
functionname(argument1, argument2)
```

Two types of functions

- **Void** function: simply performs some action
  - print('This is a void function')
- **Value-returning** function: performs some processing, then "returns" a value
  - x = input('This function returns a string: ')
  - y = type(x)

Multiple function arguments

- Many functions take more than 1 argument
- Order matters!
- Some arguments may be optional

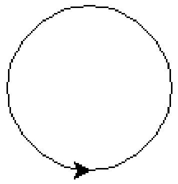
Example function: turtle circle

circle(*radius*, *extent=None*, *steps=None*)

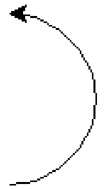
- First argument sets the circle radius
- Second argument sets extent of circle (degrees to draw) – this is optional, defaults to whole circle
- Third argument sets number of “sides” in circle – this is optional, defaults to a large number

Example function: turtle.circle

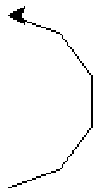
```
# Draw circle of radius 50
circle(50)
```



```
#  Draw semicircle of radius 50
circle(50, 180)
```



```
#  Draw semicircle of radius 50 with 20 sides
circle(50, 180, 20)
```

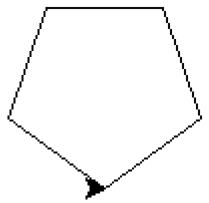


Named arguments

- Can override order of arguments by naming them

circle(*radius*, *extent=None*, *steps=None*)

circle(steps=5, radius=50)



Class functions

- Sometimes a variable has functions attached to it, which we call using the period operator

```
alex = turtle.Turtle()
alex.circle(50)
```

- This happens when a variable points to a type of data called a “class”
- We will cover this in more detail later
- For now, just think of the variable as part of the function name

Where do functions come from?

- Built in to python (print, input, int)
- Imported from modules
- Written yourself

Modules

- Modules are collections of functions
- Many “standard” modules were automatically installed when you downloaded python
- Other modules can be installed by downloading them using the python “pip” program

Installing a module: pafy

- Open the command prompt (win+r, type cmd)
- cd C:\Python34\Scripts
- pip.exe install pafy

- That’s it! Almost every popular package can be installed this way

Using a module

- Even if a module is installed, python doesn't automatically know about it
- We need to *import* module we want to use (either from the standard library or installed with pip)
- Simplest way:

```
import pafy
```

- Then we call functions as pafy.function()

Example: pafy

```
import pafy
video = pafy.new(“dQw4w9WgXcQ”)
print(video)
```

Example: math

```
import math
math.exp(3)
math.cos(3.14159)
```

Example: censusname

```
import censusname
censusname.generate()
```

Example: pyowm

```
import pyowm

owm = pyowm.OWM('3d58b22a95a92c1f69f37c372844ecea')

report = owm.weather_at_place('Princeton, NJ')
weather = report.get_weather()

print(weather)
print(weather.get_wind())
print(weather.get_temperature('fahrenheit'))
```

Example: twython

```
import twython

twitter = twython.Twython([my secret keys])
obama = twitter.get_user_timeline(screen_name="BarackObama")
print(obama[0]['text'])

twitter.update_status(status="Tweeting from python!")
```

Other ways to import modules

from math import *

- This imports all math functions, and we can use them without writing “math.” first
- Downside: can get confusing if you import many modules

from math import cos

- Imports only a single function (which we can call without writing “math.” first)
- Good if you only need a single function from a module

Writing your own functions

- Why write your own functions? Why not just write a single program?
- 1) Might want to perform some set of statements multiple times with different arguments

  doSomethingComplicated(‘A’)

  doSomethingComplicated(‘B’)
- 2) Makes your program easier to understand and easier to collaborate on with others

Function syntax

```
def functionName(arguments):
    statement
    statement
    return variable     # if a value-returning function
```

- Note that all function statements must be indented with either a tab or spaces (not both!)
  - IDLE will automatically indent lines for you
- Let’s try some examples…

Defining vs. calling function

- Can *define* a function (with def command) only once
- This just specifies what a function does, but doesn’t actually execute any instructions

- We can then *call* a function with actual values for the arguments as many times as we want
- This is when the function is actually executed

Local variables

- What happens if we try to create a variable in a function, then try to use it outside the function?
- What happens if we try change the value of a variable passed in as an argument?
- What happens if we try to access a variable from the main program in a function?

Local variables

- Variables inside a function (including its names for the arguments) are *local* to the function and can't be used outside it
- Similarly, variables in the main program are *local* to the main program and can't be used within functions
- The part of a program where a variable lives is called its *scope*

Local variables

- Local variables are actually one of the best things about functions
- When you call a function, it is guaranteed not to mess with your variables
- If a function happens to have a variable with the same name as something you’re using, there’s no conflict
- [There is an exception to this involving classes, which we’ll cover later]

Global variables

- It is possible to create variables that are seen by all functions
- Almost always a terrible idea – can be very hard to keep track of who’s changing the variable
- One exception: constant values that all functions need to read (but not write)

Assignment

- Write two functions that compute the area and perimeter of a rectangle given its two side lengths
- Prompt the user to input the side lengths, then print the area and perimeter
- Extra credit: install a new module and use a function from it
  - See PyPI
  - Some possibilities: NameThatColor, Cheetah, FridayThe13th