

1 Carlsbad Library Beginning Python with Raspberry PI Class

- Disclaimer
- What is Python? Why Python?
- Turtle Graphics
- IDLE and Python Shell
- Hello Turtle
- Goto the Races
- Common Problems and Situations
- Resources
- The Zen of Python

2 Disclaimer

Python is a pretty complicated programming language actually, so we will be just scratching the surface on what you can do. The goal is to give you a bit of background on what python is and why you might want to learn some more about it, give you a taste of what python programming is actually like, and give you some resources if you are interested in more.

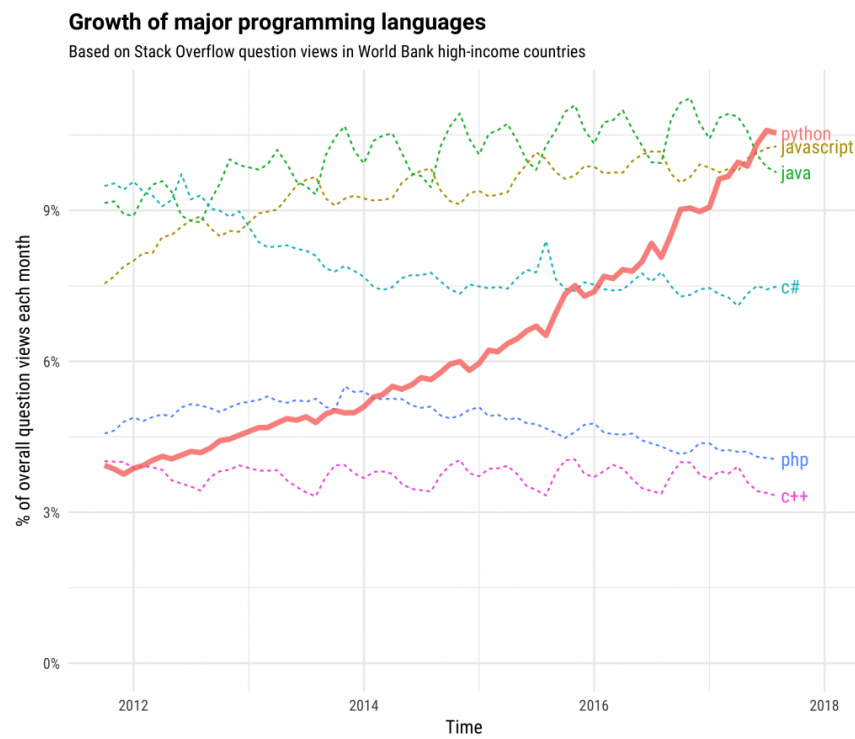
3 What is Python?

3.1 A Interpreted Programming language first released in 1991

- Name comes from "Monty Python's Flying Circus"
- What is Python's relationship to other
- Even though python is over 20 years old, it's still a relative newcomer when compared to programming languages like C and Basic.

Language	Year
Fortran	1954
Lisp	1958
Basic	1964
Pascal	1970
C	1972
C++	1980
Perl	1987
Python	1991
Java, Javascript	1995
Scratch	2002
Go	2009

3.2 Recently becoming more popular.



4 Why Python

4.1 Open Source, Supported on many different platforms and frameworks

4.2 Many high quality libraries exist for a wide variety

4.3 Used in the teaching of many technical subjects

4.4 A few different frameworks and programs using python

- TensorFlow - Very popular machine learning framework
- OpenCV - computer vision library
- ROS - Robotics operating System
- Blender - Computer Graphics

4.5 A few different projects I've done in python for myself or classes I took that used python

- Robot Control
- Hydrophone data format conversion
- bluetooth to wifi gateway for temperature sensor
- financial data manipulation and reporting
- Converting audio files to text notes for import to another program
- Machine Learning Class
- Computer Vision Class

5 Turtle Graphics

5.1 Turtle Graphics is a old (late 60's) part of the LOGO programming language that was created for education

5.2 Cartesian (x,y) coordinate system based drawing. The basic commands are

5.2.1 Move and Draw

1. forward, back, right, left

5.2.2 Pen control

1. penup, pendown

5.2.3 Appearance

1. color, shape

5.3 Full reference is on python.org website

6 IDLE and Python Shell

There are a couple of different ways you can create and run a python program. Many online courses will let you compose and run your python program from your web browser (like trinket.io). Some systems like Linux and Raspberry Pi usually come with python already installed. Generally you just enter "python" on the command line to get a python shell. And there are some frameworks that let you run python inside a html document, like jupyter notebooks. We are going to start with the IDLE (Integrated Development and Learning Environment), which includes a python shell and a text editor. *

7 Hello Turtle

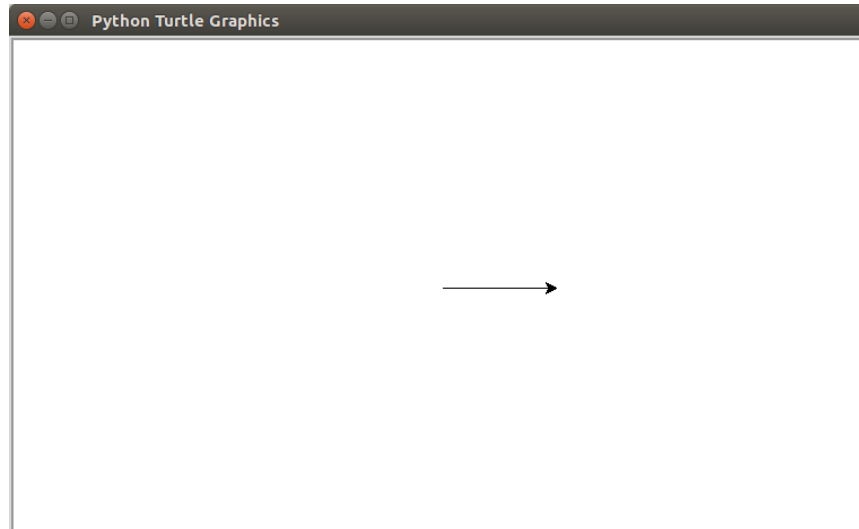
7.1 From the Raspberry Pi Programming menu select "Python 2"

7.2 In your shell type the following

```
from turtle import *  
*forward(100)*
```

```
Python 3.5.2 (default, Nov 23 2017, 16:37:01) [GCC 5.4.0 20160609] on  
linux Type "copyright", "credits" or "license()" for more information. »>  
from turtle import * »> forward(100) »>
```

7.3 You should see a new window like this



8 Hello Explanation

8.1 `from turtle import *`

This line tells python you want to use the turtle graphic library in your program. the `*` means make all the functions available.

8.2 `forward(100)`

This tell the turtle graphic program to move the pen 100 coordinates in whatever direction it is pointing.

9 Another python piece

9.1 Variables and Objects

`"="` assigns a value to a variable. To make a variable you can just assign it. In you shell type `>> a = 1` if you type `"a"` then a new line you see the value `>> a 1`

Most thing in pythons are objects and you can do thing to this. For instance a `"string"` is list of characters.

assign phase `"hello"` `>> phrase = "hello"` make a new variable caps which is the phrase capitalized `>> caps = phrase.capitalize()` display the value of

phrase »> caps 'Hello' typing # and then something does nothing (it's a comment) »> a # nothing happens after the hash mark 1

10 Get some turtles moving

```
# Type "File->" New to create new program
# Lets create a turtle with some more parts
# To run you program "Run-> Run Module" or F5

from turtle import *
from random import randint

ada = Turtle()
ada.color('red')
ada.shape('turtle')
ada.penup()
ada.goto(-160,100)
ada.pendown()

# Notice now I do ada.forward()
ada.forward(10)
ada.forward(randint(1,20)) # Move forward a random amount from one to 20
ada.forward(randint(1,20)) # Do it again
```

11 One more Python piece

In the above example we did several forward statements. If you want to do something many times in python (and most programming languages) you use whats called a "for" loop. Python syntax is a little different from most programming language. There are now curly braces like } or { to mark the end beginning. You simply indent, but the amount matters. You can use spaces or tabs, but just you tabs, it will make you life easier. For example try this in the shell.

```
>>>for x in range(10):
>>>print(x)

0
1
```

```
2
3
4
5
6
7
8
9
```

12 Turtle Race program

```
# Ok now we are ready to add a second turtle and create a race!
# Lets add a random number
from turtle import *
from random import randint # for random numbers

ada = Turtle()
ada.color('red')
ada.shape('turtle')
ada.penup()
ada.goto(-160,100)
ada.pendown()
bob = Turtle()
bob.color('blue')
bob.shape('turtle')
bob.penup()
bob.goto(-160, 70) # make bob below ada
bob.pendown()

for turn in range(100):
    ada.forward(randint(1,5))
    bob.forward(randint(1,5))
```

13 Common Problems and Situations

13.1 Python 2 vs Python 3

Which one to install? First check what you have. Many systems (including raspberry pi) have python 2 by default. When you start the shell the version info is at top. If your system doesn't have python I recommend installing python 3. Many examples require it anyway. If you run into a problem, like a python program you want to use is only python 2, then you will have to learn to use "virtual environments" or some other way. To be honest if your machine is fast enough and drive is big enough you might consider just using virtualbox to make a new machine.

13.2 PIP

Python Package management system. Use this to install new packages, i.e. you get an error when you have

```
from coollibrary import neatostuff
```

From your command line type `pip install coollibrary`. on linux you might need to do `sudo pip install coollibrary`

13.3 Virtual Environments

Virtual Environments is a tool to allow to have multiple python setups on the same machine (i.e version 2 and version 3, but also this library installed, but not that library, etc). It's a tricky tool and I only use it when I have to, i.e. I'm going through some tutorial and it's part of the steps. Use pip to install it.

```
pip install virtualenv
```

Full gory details are here <http://docs.python-guide.org/en/latest/dev/virtualenvs/>

14 Python Resources

too many to list! Coursera, Udacity, Udemy all have online courses in python as well as edx and mit open courseware.

14.1 Getting Python <https://www.python.org/>

14.2 This Class Material

<https://curriculum.raspberrypi.org/> <https://codeclubprojects.org/en-GB/python/turtle-race/>

14.3 Books at the Carlsbad Library

This one is the first returned by Amazon if you search on "learn python". I have not read it, but O'Reilly programming books tend to be good, and it's got good reviews on amazon. Title: Learning Python Author: Lutz, Mark,

14.4 Stack Overflow - Programming Information Exchange

You don't really need to bookmark this - just google "how do I do XYZ in python" and it almost always comes up at the first site on the list.

14.5 Introduction to Interactive Programming in Python (parts 1 and 2, Coursera)

There is a fee for this course, but you can audit for free. I took it a long time ago (for free) so I don't know if you can still get the self checking programs for free. The lecture material and example programs were excellent. <https://www.coursera.org/learn/interactive-python-1>

14.6 Carlsbad Library Lynda.com portal has several python classes

<http://cbcl.idm.oclc.org/login?url=http://iplogin.lynda.com>

14.7 You Tube Python Videos

I have a programmer friend who like sentdex, I've watched a few. He has many different python videos. These tend to be short, say 10 minute long pieces. Good for busy people who want to do a bit at a time. <https://www.youtube.com/user/sentdex>

15 The Zen of Python

Just for fun (and some python words of wisdom) in your shell type this.

```
»> import this
```

16 More Exercises

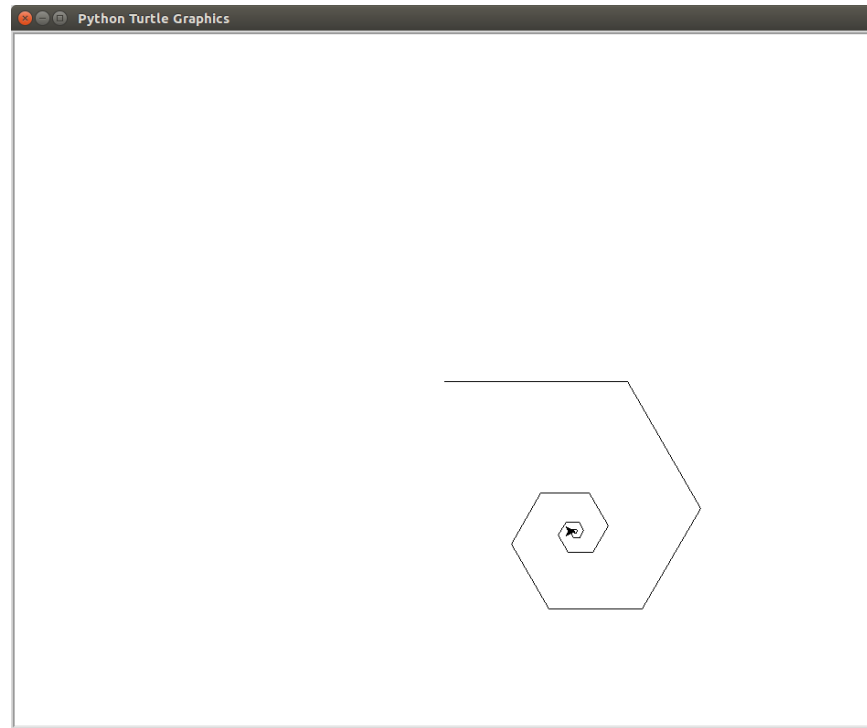
There are some more exercises on the codeclub page. <https://codeclubprojects.org/en-GB/python/turtle-race/> I haven't worked out solutions, but can help if anyone wants to try

17 More Turtle Graphics

Examples from *Measure, Topology, and Fractal Geometry* by Gerald Edgar
qThese are written in Logo, but I ported them to python.

18 Spiral

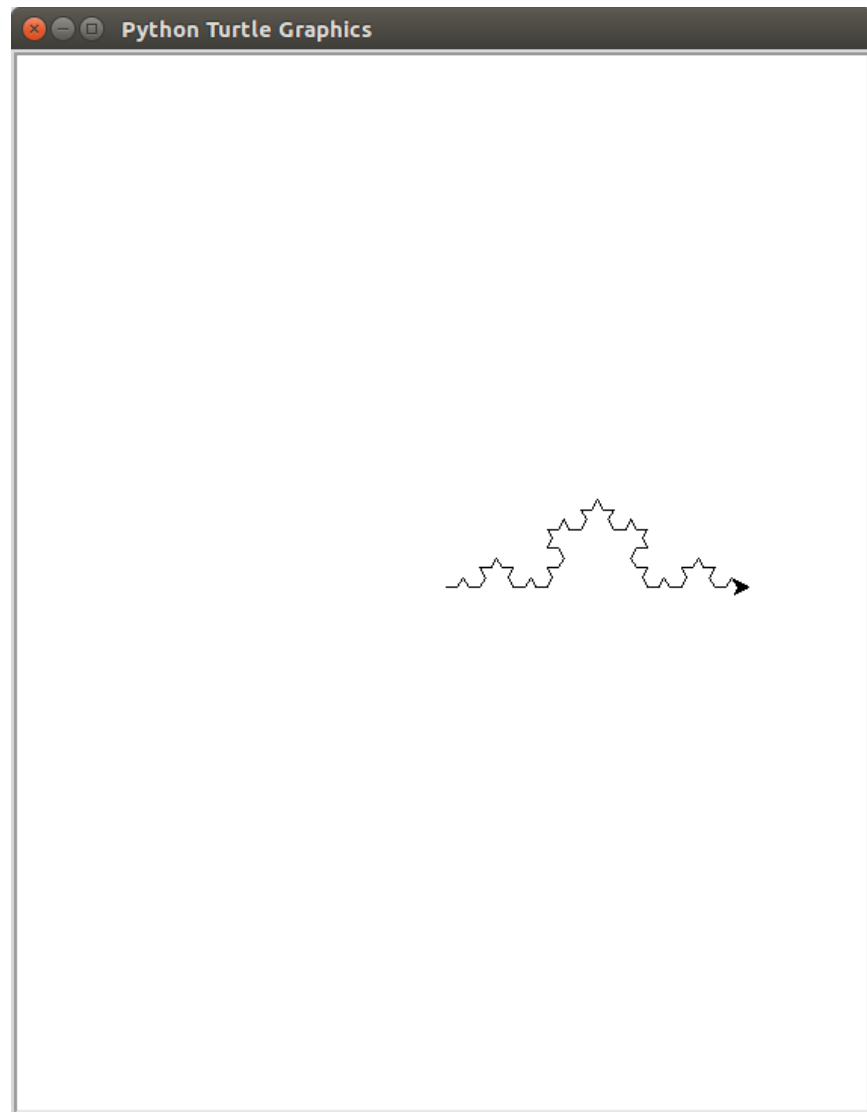
```
from turtle import *
def spiral(size):
    if (size > 1):
        forward(size)
        right(60)
        spiral(size * 0.8)
    speed("fastest")
    spiral(60)
```



19 Koch

```
from turtle import *
def Koch(depth, size):
    if(depth == 0):
        forward(size)
    else:
        Koch(depth-1, size/3)
        left(60)
        Koch(depth-1, size/3)
        right(120)
        Koch(depth-1, size/3)
        left(60)
        Koch(depth-1, size/3)

Koch(3, 200)
```



20 Leaf

```
from turtle import *
from math import sqrt
sqrt2 = sqrt(2)
def leaf(depth, size):
    if depth < 1:
```

```
forward(2*size)
back(2*size)
    else:
forward(size)
leaf(depth-2, size/2)
back(size)
left(45)
leaf(depth-1, size/sqrt2)
right(90)
leaf(depth-1, size/sqrt2)
left(45)
left(90)
leaf(11,200)
```

