Introduction to Python and the BeagleBone Black

IEEE Fall 2016

Taylor Skilling
Ken Afriyie

Agenda

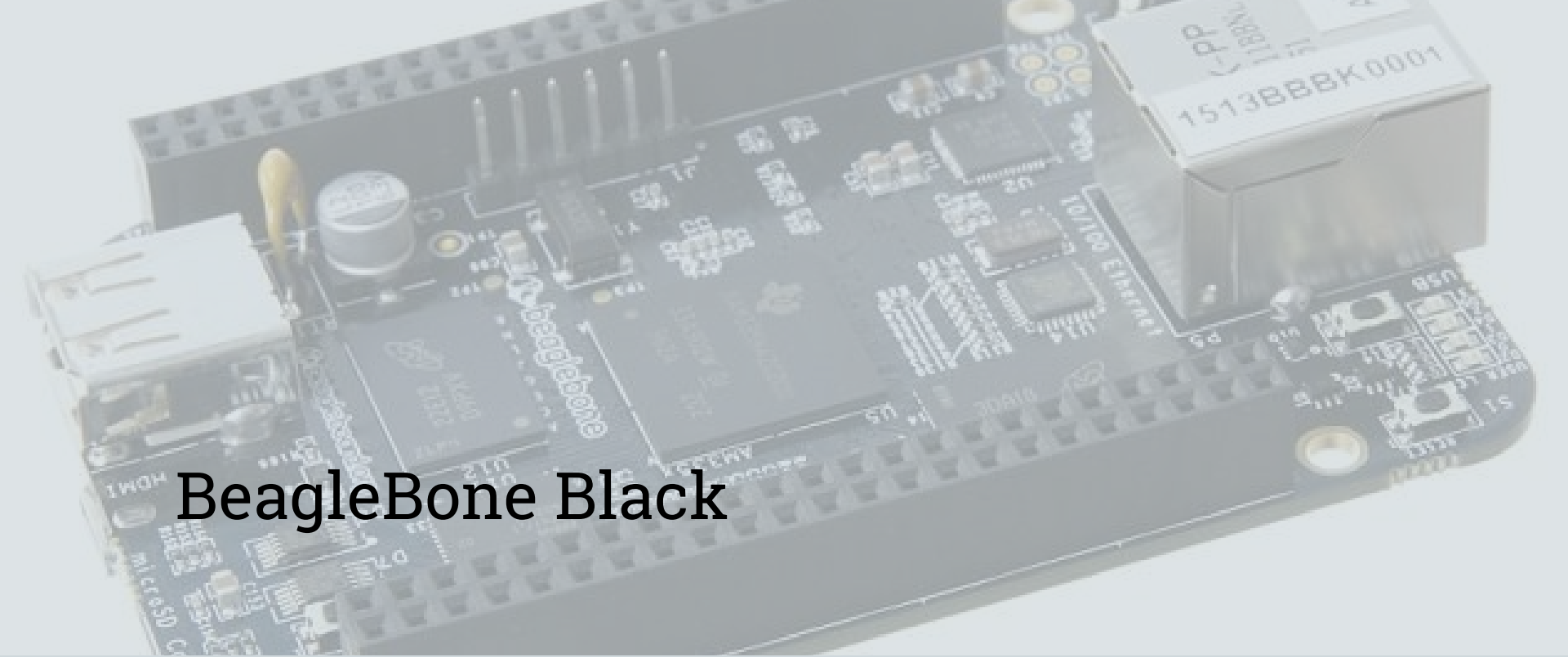
BeagleBone Black Overview

Python History and Overview

Learning Python - The Basics

BeagleBone Setup

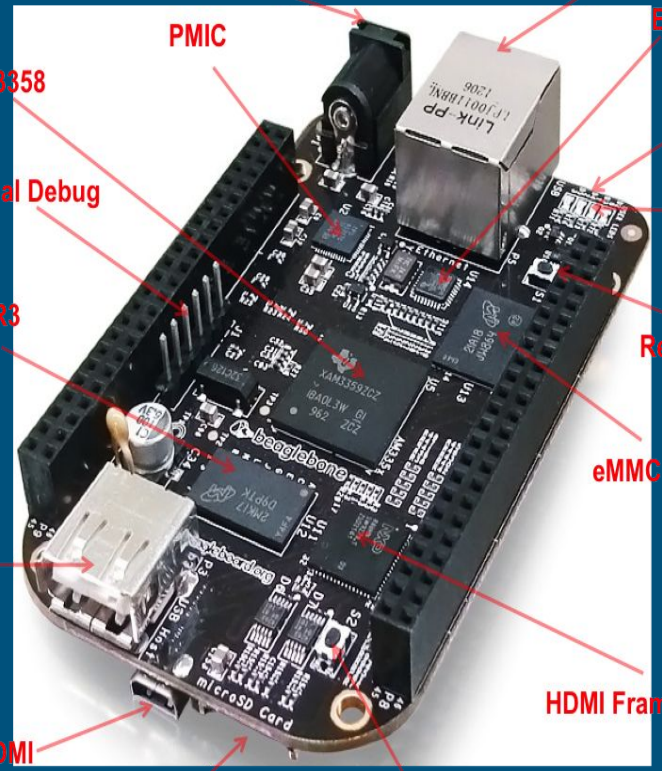
Programming Activity



BeagleBone Black

What is it?

- Beaglebone Black is a low cost, single board, open source computer
  - ARM Cortex A8 microprocessor
  - 512MB DDR3 RAM
  - 4GB 8-bit on-board flash storage
  - USB client for power and communications
  - Ethernet
  - HDMI
  - 2x 46 pin headers
  - Compatible with distros such as Debian, Ubuntu, Android, and Angstrom.



Why do people use it?

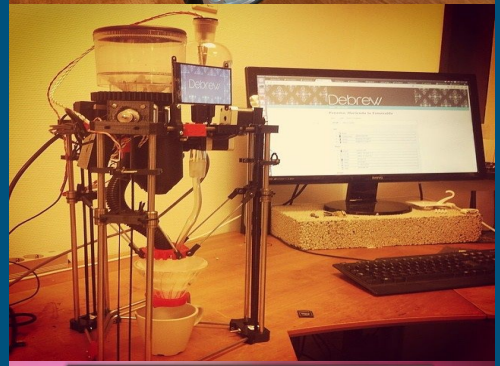
The BeagleBone is:

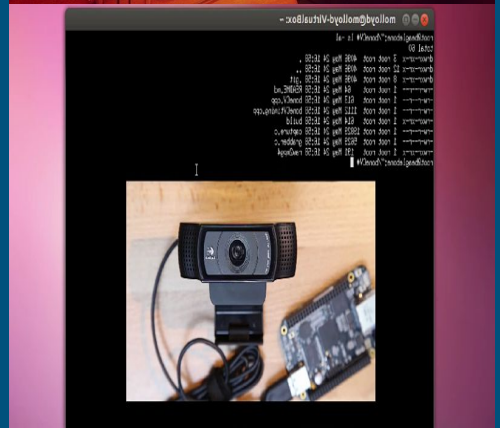
- Cheap (entire kit can be found for $50)
- Community supported
  - Large community of developers and hobbyists
- Quick to set up
  - A BeagleBone Black can boot linux in under 10 seconds
  - User just needs a USB cable to begin development
- Small
  - Size of a credit Card
- Shipped with an OS pre-installed
- Equipped with 92 GPIO pins (twice the amount of the Rasberry Pi)

What else can we do with it?

The BeagleBone Black can be used many different projects.

- The powerful single board computer can be used to control a wheeled robot
- The BeagleBone can be interfaced with stepper motors to brew coffee
- The BeagleBone Black can be used for Video Capture and Image Processing on Embedded Linux using OpenCV





python

What is it?

A programming language that is:

- High-level
- Interpreted,
- Object-oriented and Structural
- Dynamic
- Embedded Memory Management
- Concise
- Open Source

```python
import subprocess

def get_cpu_speed():
    proc = subprocess.Popen(["cat","/proc/cpuinfo"],
                            stdout=subprocess.PIPE)
    out, err = proc.communicate()

    for line in out.split("\n"):
        if "cpu MHz" in line:
            speed = float(line.split(":")[1])
            break

    return speed

if __name__ == '__main__':

    speed = get_cpu_speed()
    print("CPU speed is {0} MHz".format(speed))
```

An example Python script printing the current CPU speed

History

Created in 1991 by Guido van Rossum (Dutch)

Started as simply a “hobby” programming project

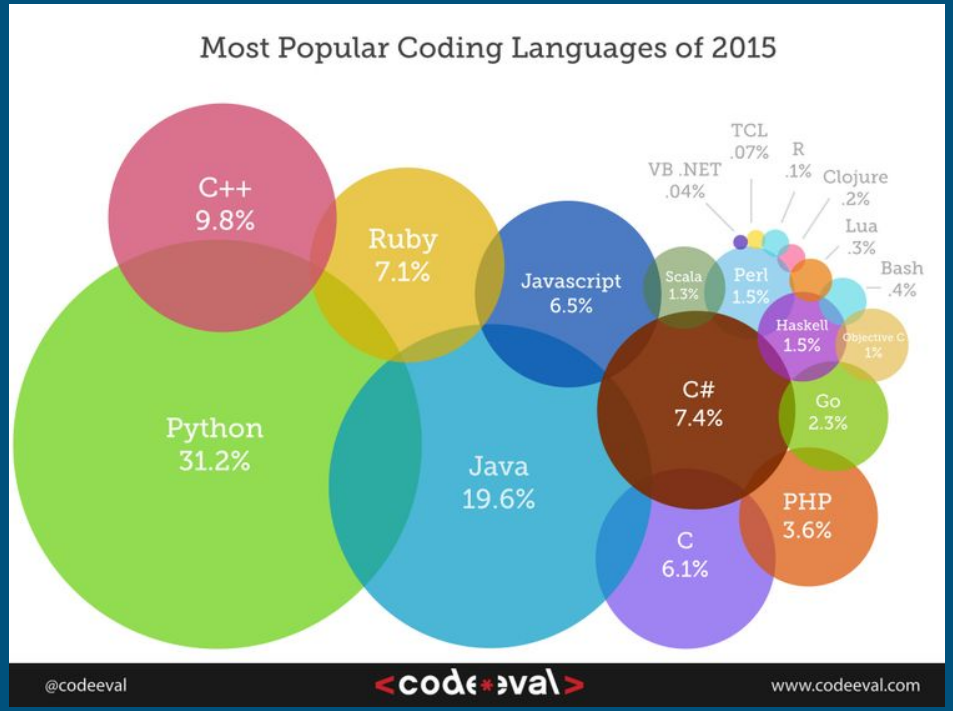
Now one of the most popular languages in the world

October 2000: Python 2.0

December 2008: Python 3.0

Why is it used?

- Easy to learn and implement
- Significantly less code than C++ or Java
- Embedded Memory Management
- Dynamic Typing
- Incredibly active community
  - Native documentation
  - Stack Overflow
  - PEP (Python Enhancement Proposal)
  - Tutorials and Guides

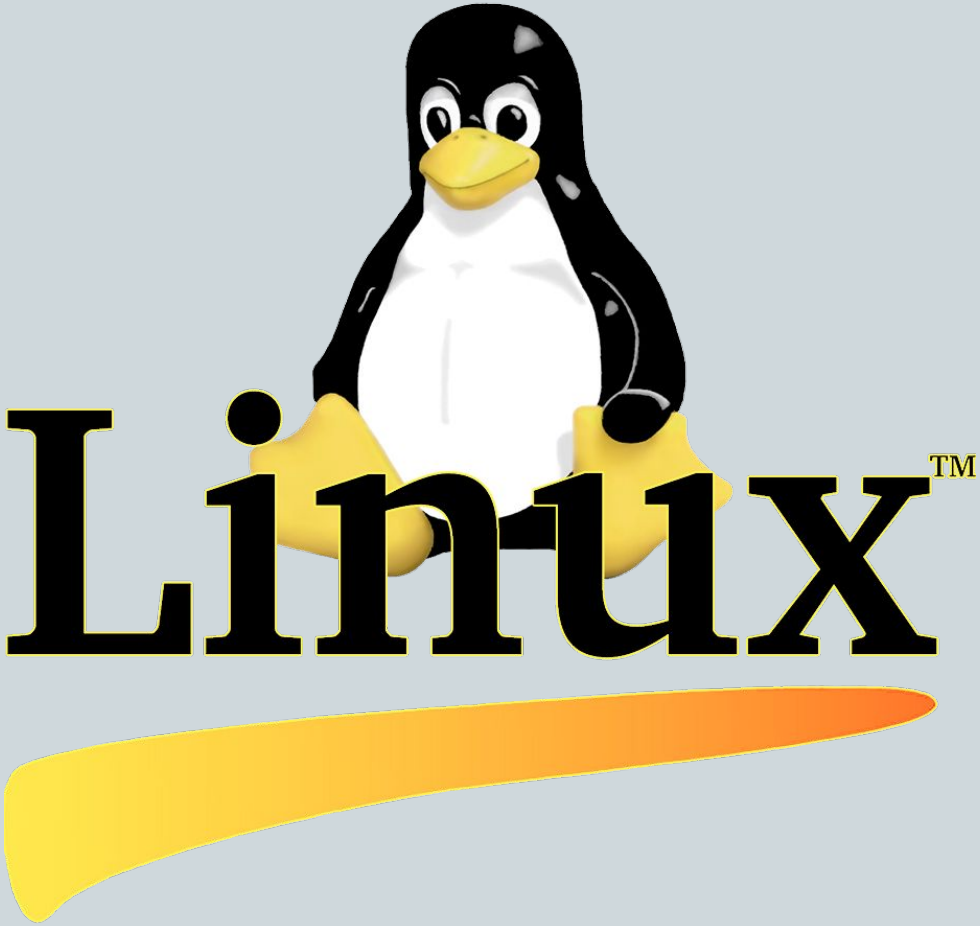


What are its applications?

- Scripting
- Connecting Code
- Computation
- Data Visualization
- Much more...

The Zen of Python

- Beautiful is better than ugly
- Explicit is better than implicit
- Simple is better than complex
- Complex is better than complicated
- Readability counts


Linux™

BASH - Navigation and Common Commands

Full Path:~$ /home/path/to/whatever

Relative Path:~$ path/to/whatever

Special Characters

/	root
~	home
..	up one
.	current

Commands

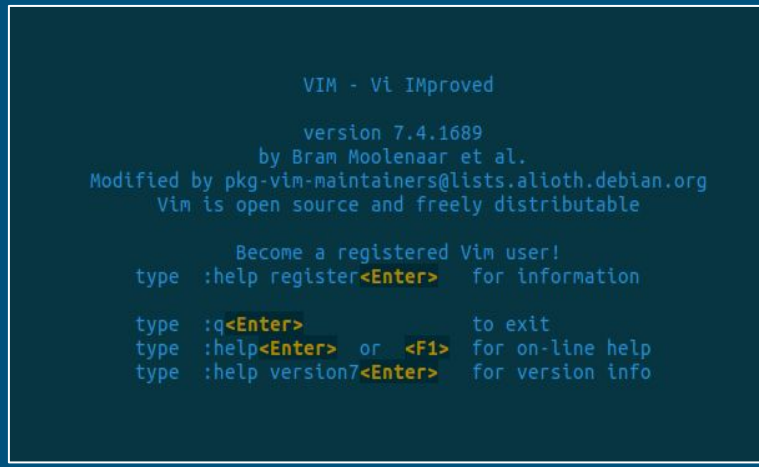
$ pwd	Print Working Directory: print the current working directory
$ cd [path]	Change Directory: change current working directory to [path]
$ ls	List: print a list of all files/directories in the cwd

```
kenny@kenny-CX62-6QD:~$ ls
AES_fullmask  CryptoTest  Downloads        Fundies1                        m5out  mytest         Public     Test
CPrograms     Desktop     Encryption       gem5-stable                     Music  openssl-1.0.2a Racket     Videos
C++Programs   Documents   examples.desktop gnome-terminal-colors-solarized myCA   Pictures       Templates
```

VIM

Vim, Vi IMproved, is a popular configurable text editor created by Bram Moolenaar.

$ vim [file] Opens file in vim]



Editing Modes

- Normal - For navigation and manipulation of text.
- Insert - For creating and editing text.
- Command - For inserting editor commands.

Vim starts in Normal mode, press “i” to enter insert mode, or “:” to enter command mode. Press “esc” during insert mode to enter normal mode, and press “esc” twice during command to enter normal mode.

VIM

Normal Mode

Commands

:	Enter Editor Mode
i	Insert text before current cursor position.
a	Append text after current cursor position
u	Undo last change
x	Delete character
dd	Delete line cursor is currently in.
yy	Yank current line (copy)
p	Paste

```
#include <stdio.h>
#include <stdlib.h>

unsigned long long counter[16][256];
unsigned long long timing[16][256];

int main(int argc, char** argv){
    FILE* f0, *f1;

    f0 = fopen("timing.bin","r");
    f1 = fopen("cipher.bin","r");
    unsigned char cipher[16];
    int i, j;
    int monitor_lines = 1;
    unsigned short time[monitor_lines];

    int threshold = 70000;
    if(argc >= 2){
        threshold = atoi(argv[1]);
    }

    for(i = 0; i < 16; i++){
        for(j = 0; j < 256; j++){
                                                    23,28-34      Top
```

VIM

Command Mode

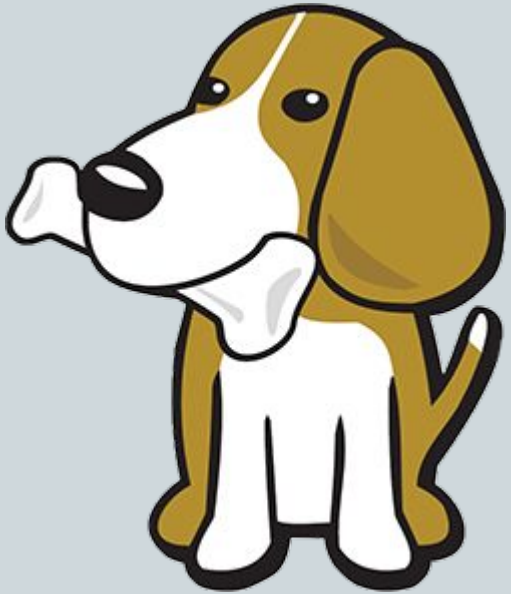
Vim starts in normal mode, enter ":" to start command mode.

```
#include <stdio.h>
#include <stdlib.h>

unsigned long long counter[16][256];
unsigned long long timing[16][256];

int main(int argc, char** argv){
    FILE* f0, *f1;

    f0 = fopen("timing.bin","r");
    f1 = fopen("cipher.bin","r");
    unsigned char cipher[16];
    int i, j;
    int monitor_lines = 1;
    unsigned short time[monitor_lines];

    int threshold = 70000;
    if(argc >= 2){
        threshold = atoi(argv[1]);
    }

    for(i = 0; i < 16; i++){
        for(j = 0; j < 256; j++){
:wq
```

Popular Editor Commands

:help	Produces main help file.
:1, $d	Deletes all the content of a file. $ means the last character in the file, d means delete.
:w	Writes file (Saves contents)
:x	Write only if there are changes and quit, analogous to :wq
:q or q!	Quit file (! overrides last change)
:wq	Writes and quits file
esc	Tap esc twice to enter Normal mode

python

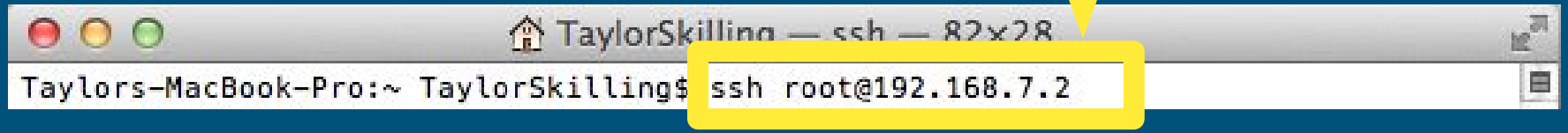


Beaglebone Black Setup

https://beagleboard.org/getting-started

Username: root

IP Address:




INTRODUCTION TO PYTHON
PYTHON

The Interpreter

```
Taylors-MacBook-Pro:~ TaylorSkilling$ python
Python 2.7.10 (v2.7.10:15c95b7d81dc, May 23 2015, 09:33:12)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> print('Hello World')
Hello World
>>>
```

Python is an *interpreted* language:

- Compiled into bytecode
- Executed in a Virtual Machine

Q: What type of language is C++?

Interpreted Languages

Pros:

- Debugging is easy!
- Small program size
- Automatic memory management

Cons:

- Longer execution time

Syntax and Conventions

Whitespace instead of curly brackets or keywords to delimit blocks of code

No semicolons or terminating characters

Python:

```
def main(argv):
    print('Hello World!\n')
```

C++:

```
void main(int argc, char *argv[])
{
    cout << "'Hello World!" << endl;
}
```

Types

Python

- *Dynamically typed*
- Values, *not variables*, are assigned to a type, no declaration necessary!
- The sequence below is totally acceptable

Java

- *Statically typed*
- Have to declare a variable with a type!
- The sequence below raises an exception...

```
myName = 'Taylor Skilling'
myName = 22
```

Data Structures - Lists

Instead of arrays, Python implements ***lists***

- **Zero-indexed**
- Many class methods including *append, pop, count, sort*
- List Comprehension
- Slicing

Declaration and Accessing Values:

```
>>> myList = range(10)
>>> myList
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>> myList = [1, 2, 3, 'testString']
>>> myList
[1, 2, 3, 'testString']
>>> myList[1]
2
```

Lists - Basic Methods

- index()
- append()
- pop()
- sort()

and many more...

```
>>> myList
[1, 2, 3, 'testString']
>>> myList.index('testString')
3
>>> myList.append('anotherString')
>>> myList
[1, 2, 3, 'testString', 'anotherString']
>>> myList.pop()
'anotherString'
>>> myList
[1, 2, 3, 'testString']
>>> newList = [13, 2, 7, 1, 24]
>>> newList
[13, 2, 7, 1, 24]
>>> newList.sort(reverse=True)
>>> newList
[24, 13, 7, 2, 1]
```

Lists - List Comprehension

Goal: Create a list that holds even number between 0 and 9

Traditional Method:

```
>>> evens = []
>>> for i in range(10):
...     if i % 2 == 0:
...             evens.append(i)
...
>>> evens
[0, 2, 4, 6, 8]
```

List Comprehension:

```
>>> evens_comp = [i for i in range(10) if i % 2 == 0]
>>> evens_comp
[0, 2, 4, 6, 8]
```

Lists - Slicing

Slicing lists is a way to easily access different parts of a list

Uses the colon (:) operator to access using the following notation

list[first index:last index:step]

```
>>> sliceList = range(10)
>>> sliceList
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>> sliceList[0:10:2]
[0, 2, 4, 6, 8]
>>> sliceList[1:10:2]
[1, 3, 5, 7, 9]
>>> sliceList[:3]
[0, 1, 2]
>>> sliceList[3:]
[3, 4, 5, 6, 7, 8, 9]
```

Data Structures - Dictionaries

Dictionaries are simply a series of Key-Value pairs, constructed with {key:value}

Declaration:

```
>>> myDict = {'first':'string value', 'second':[1,3,5]}
>>> myDict
{'second': [1, 3, 5], 'first': 'string value'}
```

Methods and Access:

```
>>> myDict.keys()
['second', 'first']
>>> myDict.values()
[[1, 3, 5], 'string value']
>>> myDict.items()
[('second', [1, 3, 5]), ('first', 'string value')]
>>> myDict.items()[0]
('second', [1, 3, 5])
>>> myDict.items()[0][0]
'second'
```

Expressions

Python	Java, C, C++
is	==
and	&&
or	||
not	!
x if c else y	c ? x : y

Indefinite Loop: For

Python

```
>>> for x in range(5):
...     print('The number is: {0}'.format(x))
...
The number is: 0
The number is: 1
The number is: 2
The number is: 3
The number is: 4
```

Java, C, C++

```
#include <iostream>
using namespace std;

int main ()
{
   // for loop execution
   for( int x = 0; x < 5; x = x + 1 )
   {
       cout << "The number is: " << x <<
endl;
   }

   return 0;
}
```

Indefinite Loop: For

Python

```
>>> x = 0
>>> while x is not 5:
...     print(x)
...     x = x + 1
...
0
1
2
3
4
```

Java, C, C++

```
#include <iostream>
using namespace std;

int main ()
{
   // Local variable declaration:
   int x = 0;

   // while loop execution
   while( x < 5 )
   {
       cout << "value of a: " << x <<
endl;
       x++;
   }

   return 0;
}
```

Functions

Like nearly everything else… Functions are simple and easy!

Basic Program:

```
def printMyString(myString):
        print(myString)

def main():
        # Declare the string to print
        stringToPrint = 'Hi IEEE!'
        # Call the main function
        printMyString(stringToPrint)

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

Invoke the Interpreter:

```
Taylors-MacBook-Pro:~ TaylorSkilling$ python func.py
Hi IEEE!
```

Libraries and Packages

Python has many libraries and packages available for download:

- NumPy
- MatPlotLib
- Libraries for sensors and breakout boards (Adafruit, SparkFun)

Frameworks:

- Django Web Framework
- PyQt UI Framework

"Python Installs Packages"

```
Taylors-MacBook-Pro:~ TaylorSkilling$ pip install matplotlib
You are using pip version 6.1.1, however version 8.1.2 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
Collecting matplotlib
  Downloading matplotlib-1.5.3.tar.gz (51.6MB)
    99% |[redacted]| 51.6MB 8.3MB/s eta 0:00:01
```

Further Reading

Classes and Objects

Exception Handling

Generator Expressions

Decorators

Doc Strings

Testing

Programming Activity

Learn to program LEDs!

Questions?

Thank You!