Intro To Programming in Python: Workshop 1

Felix Loftus

Central Saint Martins - Digital Media

ual:

Structure

- What Python is
- What's possible with Python

break

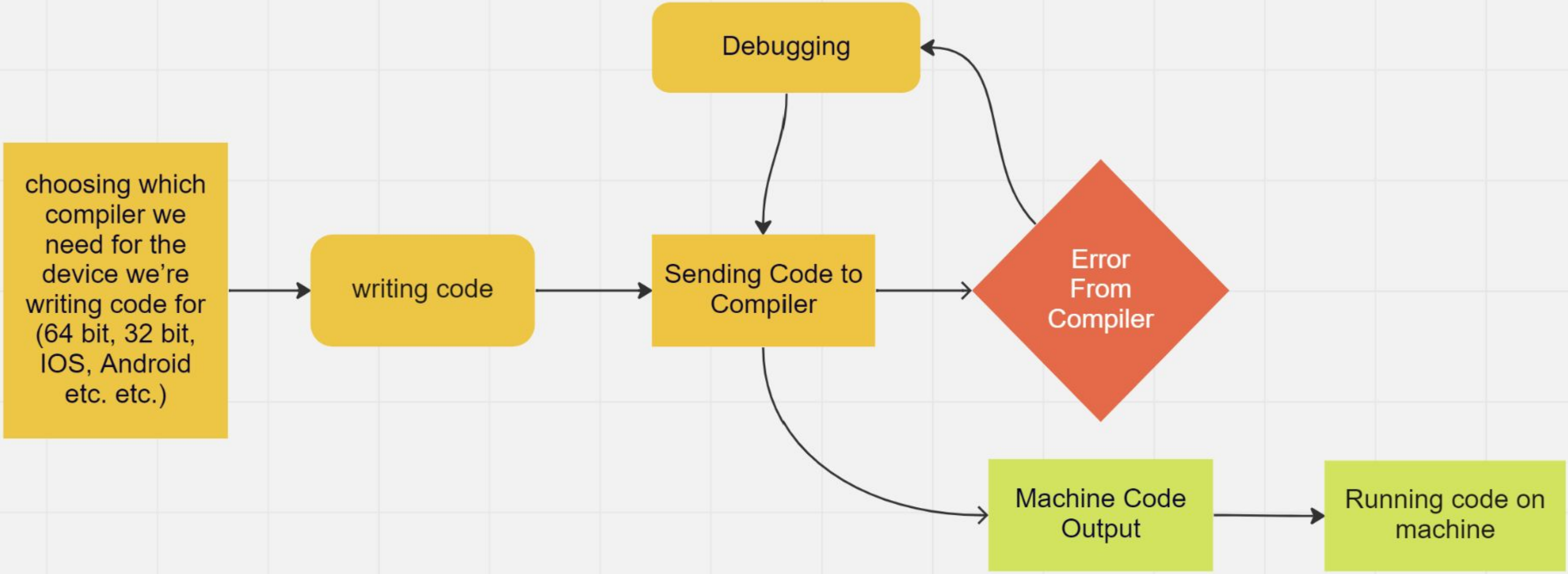
- What a good setup for coding in Python looks like
- How to run scripts and import packages in Python

What is Python?

I think you need to understand two computing ideas about python to get an understanding of what it is, why it is so useful and consequently so why it is so popular:

1. Python is an interpreted language (instead of a compiled language)
2. Python is a high level language

Interpreted vs Compiled Languages



choosing which compiler we need for the device we're writing code for (64 bit, 32 bit, IOS, Android etc. etc.)

writing code

Sending Code to Compiler

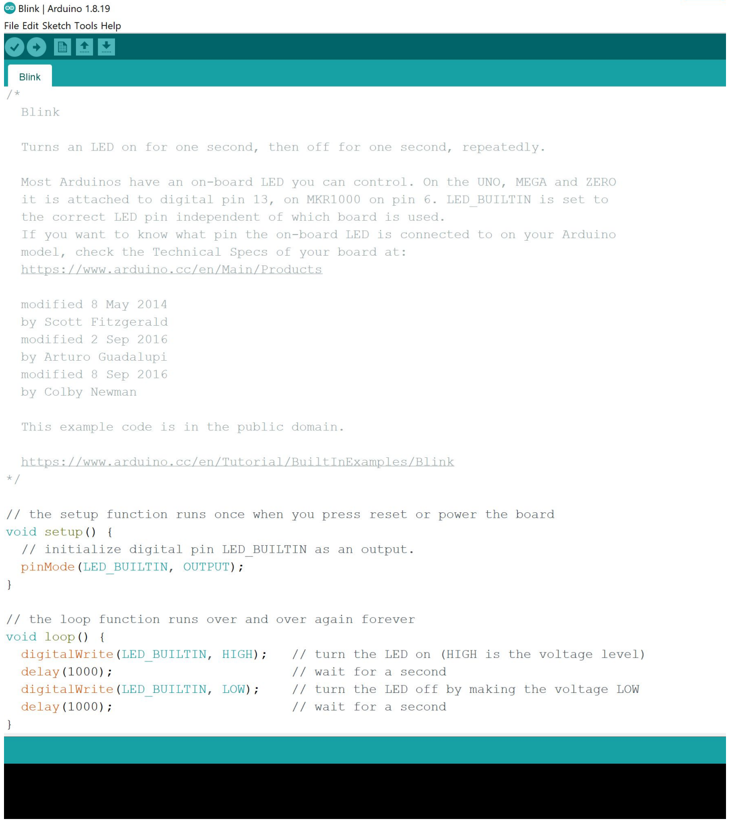
Debugging

Error From Compiler

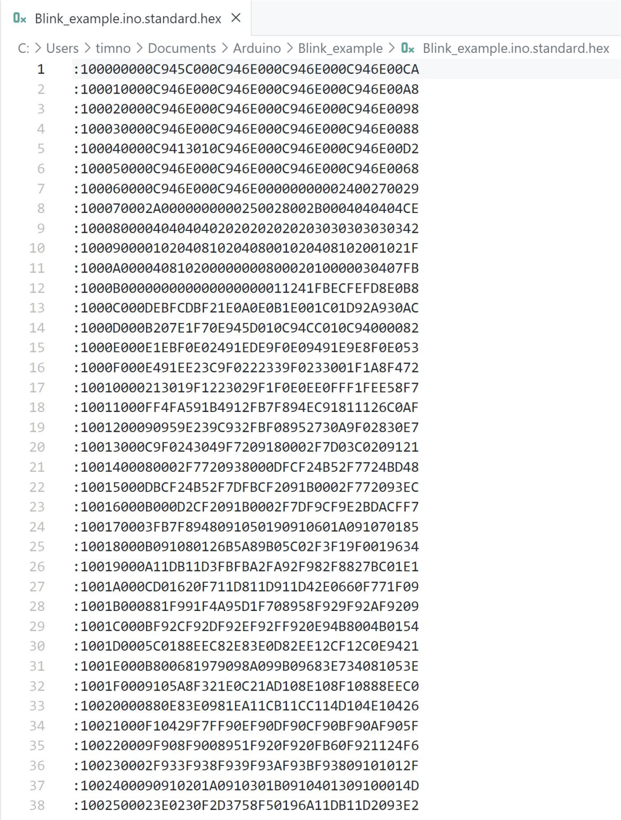
Machine Code Output

Running code on machine

Compiled Languages

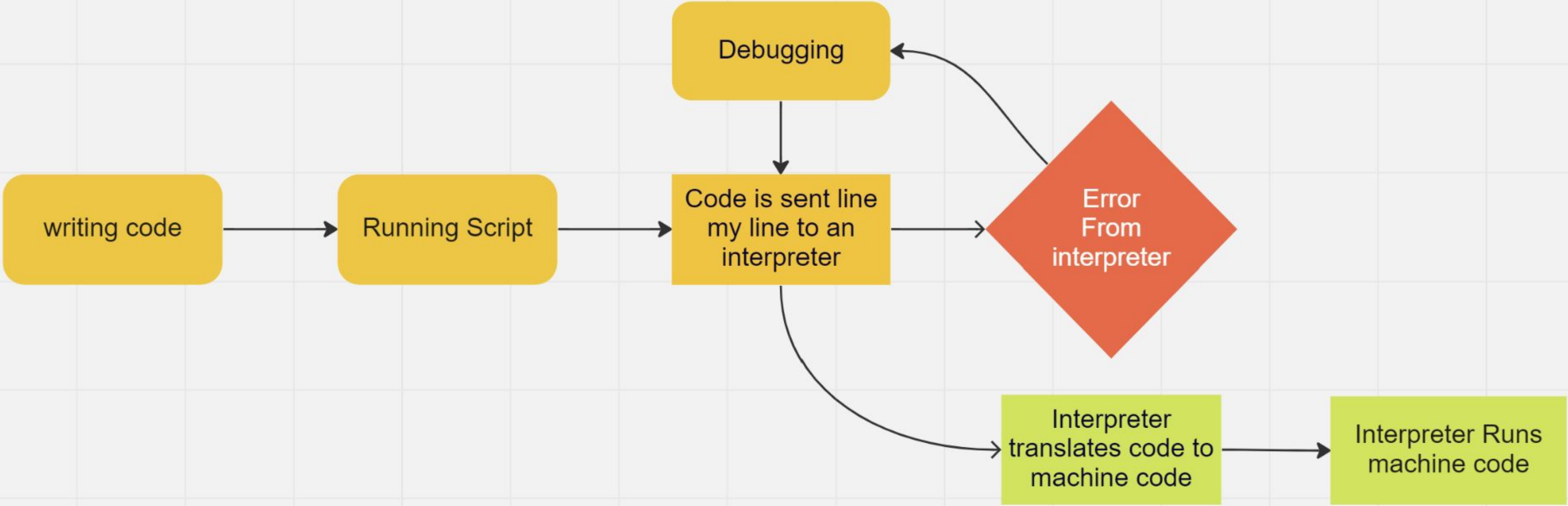


```
/*
  Blink

  Turns an LED on for one second, then off for one second, repeatedly.

  Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO
  it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is set to
  the correct LED pin independent of which board is used.
  If you want to know what pin the on-board LED is connected to on your Arduino
  model, check the Technical Specs of your board at:
  https://www.arduino.cc/en/Main/Products

  modified 8 May 2014
  by Scott Fitzgerald
  modified 2 Sep 2016
  by Arturo Guadalupi
  modified 8 Sep 2016
  by Colby Newman

  This example code is in the public domain.

  https://www.arduino.cc/en/Tutorial/BuiltInExamples/Blink
*/

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);                       // wait for a second
}
```


Blink_example.ino.standard.hex
C: > Users > timno > Documents > Arduino > Blink_example > Blink_example.ino.standard.hex

```
:100000000C945C000C946E000C946E000C946E00CA
:100010000C946E000C946E000C946E000C946E00A8
:100020000C946E000C946E000C946E000C946E0098
:100030000C946E000C946E000C946E000C946E0088
:100040000C9413010C946E000C946E000C946E00D2
:100050000C946E000C946E000C946E000C946E0068
:100060000C946E000C946E00000000002400270029
:100070002A0000000000250028002B0004040404CE
:10008000040404040202020202020303030303034 2
:10009000010204081020408001020408102001021F
:1000A00004081020000000080002010000030407FB
:1000B000000000000000000011241FBECFEFD8E0B8
:1000C000DEBFCDBF21E0A0E0B1E001C01D92A930AC
:1000D000B207E1F70E945D010C94CC010C94000082
:1000E000E1EBF0E02491EDE9F0E09491E9E8F0E053
:1000F000E491EE23C9F0222339F0233001F1A8F472
:10010000213019F1223029F1F0E0EE0FFF1FEE58F7
:10011000FF4FA591B4912FB7F894EC91811126C0AF
:1001200090959E239C932FBF08952730A9F02830E7
:10013000C9F0243049F7209180002F7D03C0209121
:1001400080002F7720938000DFCF24B52F7724BD48
:10015000DBCF24B52F7DFBCF2091B0002F772093EC
:10016000B000D2CF2091B0002F7DF9CF9E2BDACFF7
:100170003FB7F894809105019091060A091070185
:10018000B091080126B5A89B05C02F3F19F0019634
:10019000A11DB11D3FBFBA2FA92F982F8827BC01E1
:1001A000CD01620F711D811D911D42E0660F771F09
:1001B000881F991F4A95D1F708958F929F92AF9209
:1001C000BF92CF92DF92EF92FF920E94B8004B0154
:1001D0005C0188EEC82E83E0D82EE12CF12C0E9421
:1001E000B800681979098A099B09683E734081053E
:1001F0009105A8F321E0C21AD108E108F10888EEC0
:10020000880E83E0981EA11CB11CC114D104E10426
:10021000F10429F7FF90EF90DF90CF90BF90AF905F
:100220009F908F9008951F920F920FB60F921124F6
:100230002F933F938F939F93AF93BF938091010112F
:1002400090910201A0910301B09104013091000014D
:1002500023E0230F2D3758F50196A11DB11D2093E2
```

Interpreted Languages



writing code

Running Script

Code is sent line my line to an interpreter

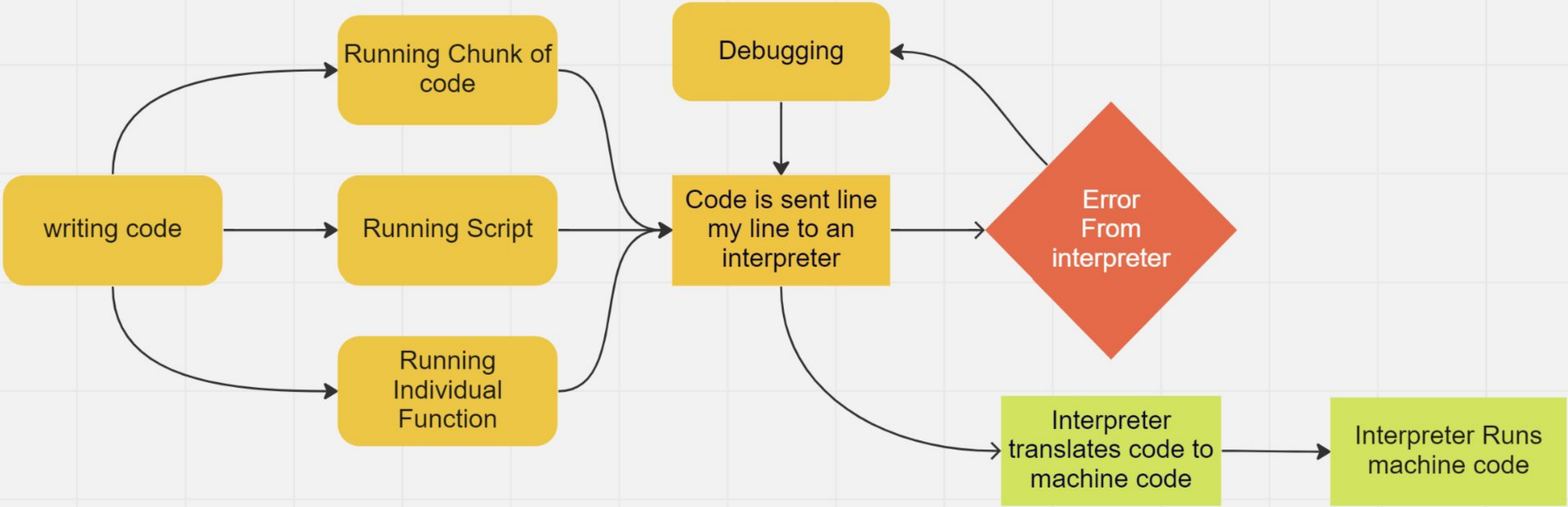
Debugging

Error From interpreter

Interpreter translates code to machine code

Interpreter Runs machine code

Interpreted Languages


writing code
Running Chunk of code
Running Script
Running Individual Function
Debugging
Code is sent line my line to an interpreter
Error From interpreter
Interpreter translates code to machine code
Interpreter Runs machine code

Interpreted Languages: Translators!

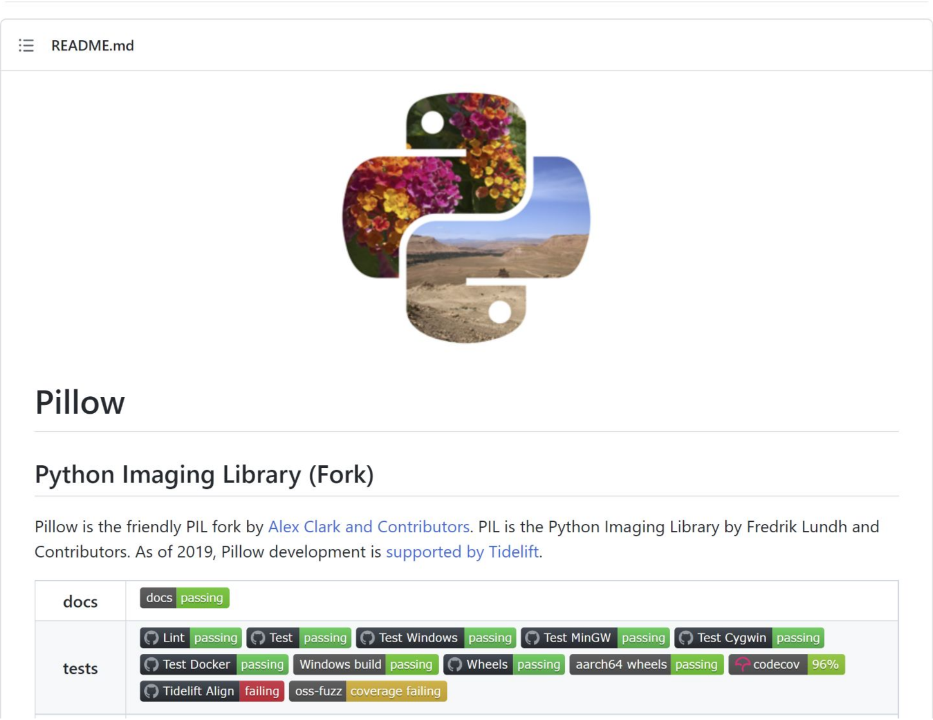
What is Python? A translator that can speak lots of languages

You can think of python as a very quick translator that can speak a whole host of languages. In python libraries, generally a large part of the code base will be from a compiled language like C or C++. Python provides an easy to write wrapper around these code written in these languages.

Potential question:

- Isn’t translation slow compared to speaking in a native language?

Interpreted Languages: Libraries



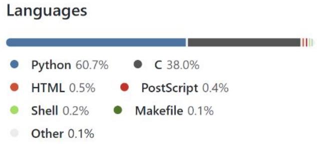
README.md

Pillow

Python Imaging Library (Fork)

Pillow is the friendly PIL fork by Alex Clark and Contributors. PIL is the Python Imaging Library by Fredrik Lundh and Contributors. As of 2019, Pillow development is supported by Tidelift.

docs	docs passing
tests	Lint passing Test passing Test Windows passing Test MinGW passing Test Cygwin passing Test Docker passing Windows build passing Wheels passing aarch64 wheels passing codecov 96% Tidelift Align failing oss-fuzz coverage failing

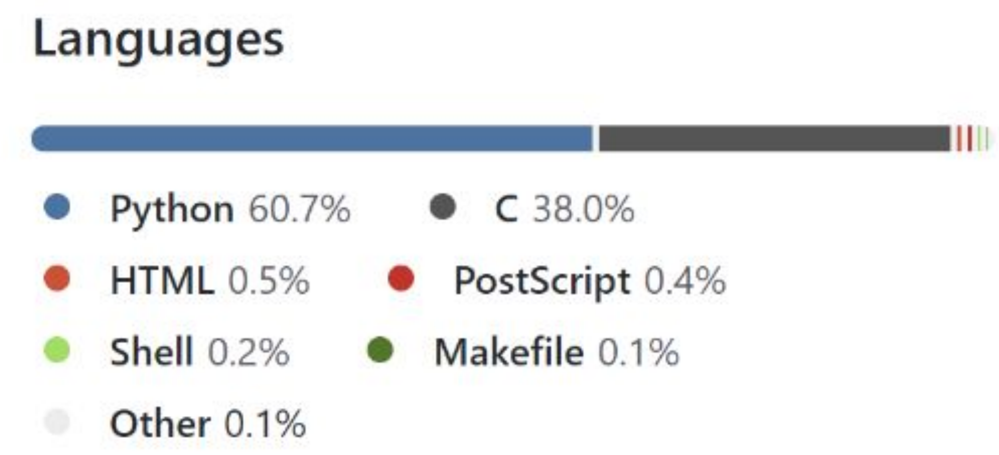


Languages

● Python 60.7%	● C 38.0%
● HTML 0.5%	● PostScript 0.4%
● Shell 0.2%	● Makefile 0.1%
● Other 0.1%	

Interpreted Languages: Libraries

Python itself



Languages

● Python 60.7% ● C 38.0%

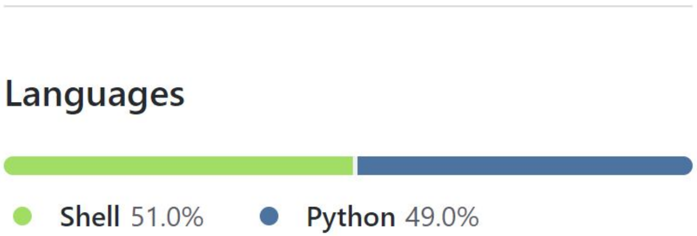
● HTML 0.5% ● PostScript 0.4%

● Shell 0.2% ● Makefile 0.1%

● Other 0.1%

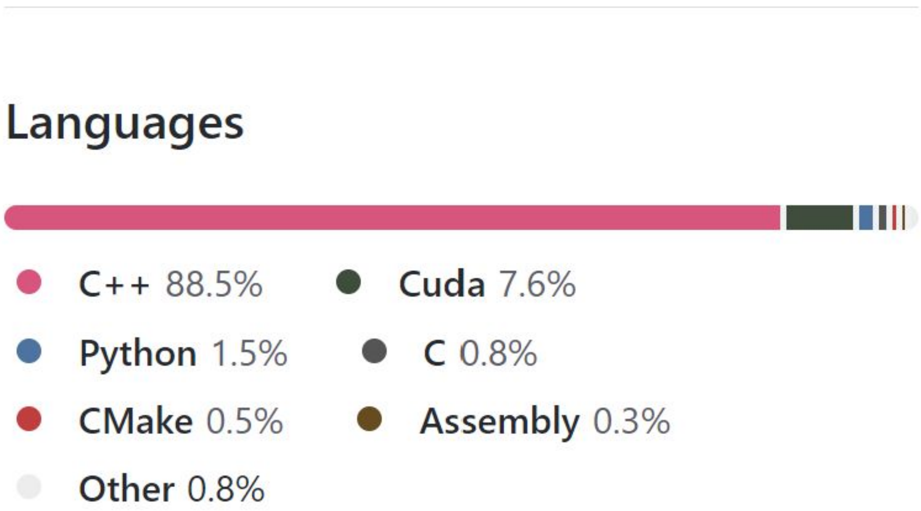
Interpreted Languages: Libraries

OpenCV (Computer Vision)



Languages

● Shell 51.0% ● Python 49.0%



Languages

● C++ 88.5% ● Cuda 7.6%

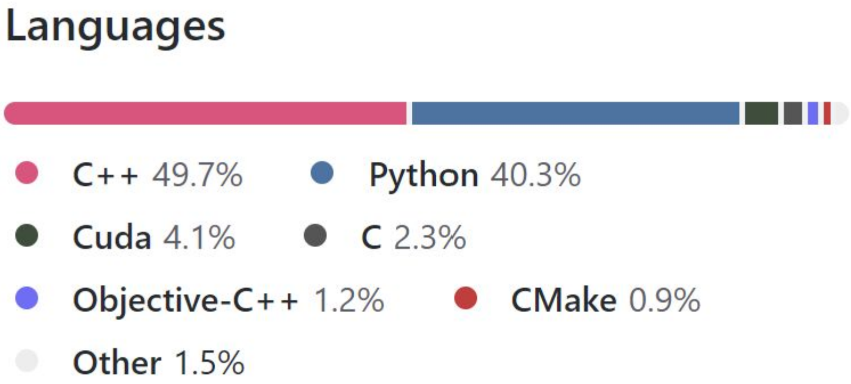
● Python 1.5% ● C 0.8%

● CMake 0.5% ● Assembly 0.3%

● Other 0.8%

Interpreted Languages: Libraries

OpenCV (Computer Vision)



Languages

● C++ 49.7% ● Python 40.3%

● Cuda 4.1% ● C 2.3%

● Objective-C++ 1.2% ● CMake 0.9%

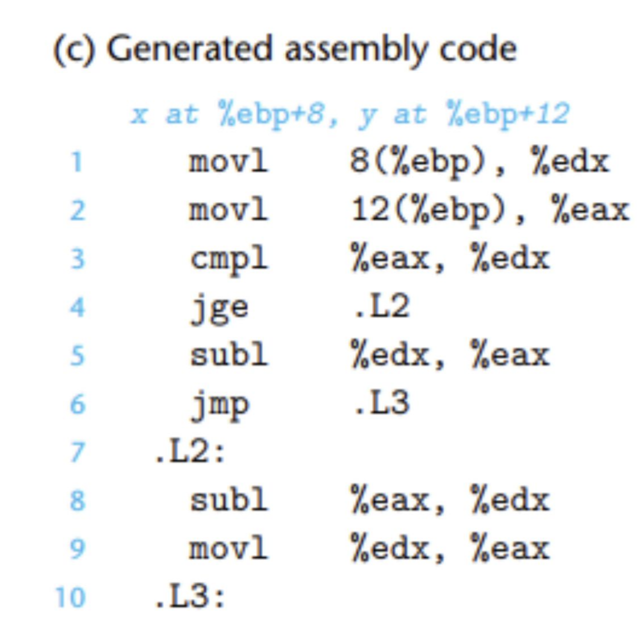
● Other 1.5%

Interpreted Languages: Libraries

Link to Resources.md

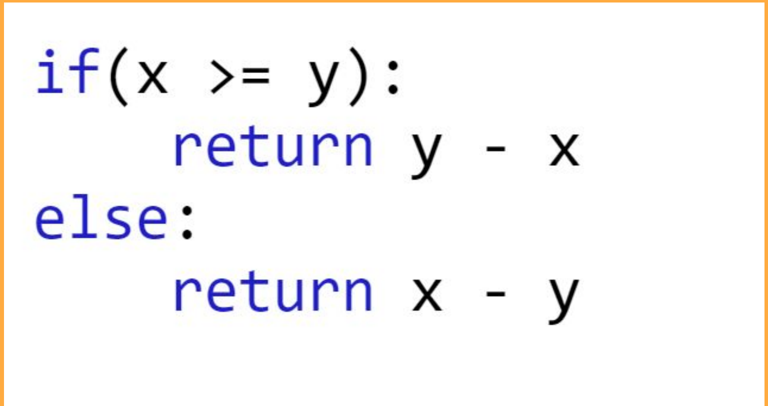
What is Python? Something that is good at pulling other machines levers

What is Python? A High Level Language



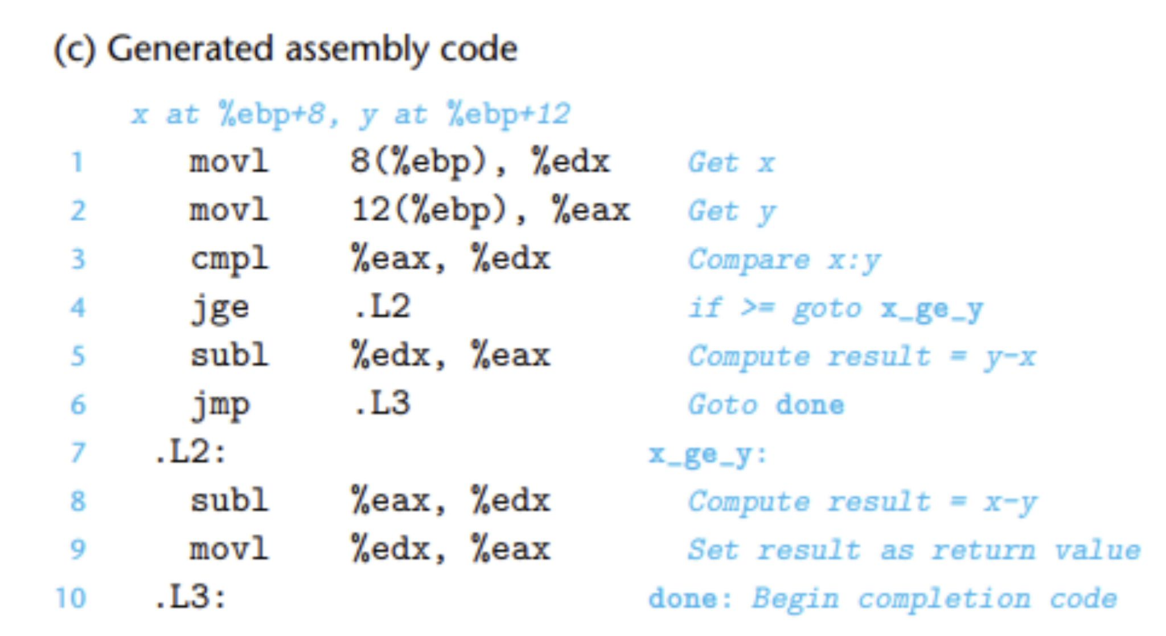
(c) Generated assembly code

```
    x at %ebp+8, y at %ebp+12
      movl    8(%ebp), %edx
      movl    12(%ebp), %eax
      cmpl    %eax, %edx
      jge     .L2
      subl    %edx, %eax
      jmp     .L3
    .L2:
      subl    %eax, %edx
      movl    %edx, %eax
    .L3:
```



```
if(x >= y):
    return y - x
else:
    return x - y
```

What is Python? A High Level Language



(c) Generated assembly code

```
   x at %ebp+8, y at %ebp+12
     movl    8(%ebp), %edx      Get x
     movl    12(%ebp), %eax     Get y
     cmpl    %eax, %edx         Compare x:y
     jge     .L2                if >= goto x_ge_y
     subl    %edx, %eax         Compute result = y-x
     jmp     .L3                Goto done
   .L2:                       x_ge_y:
     subl    %eax, %edx         Compute result = x-y
     movl    %edx, %eax         Set result as return value
   .L3:                       done: Begin completion code
```

What is Python? A High Level Language

High-level languages are furthest from machine code, and closest to human language.

They are generally easier to write and easier to learn than low-level languages.

```
 ooooooooooooooooooooooooooooooooooooo
 8                                .d88
 8  oooooooooooooooooooooooooooood8888
 8  8888888888888888888888888P"   8888    oooooooooooooooo
 8  8888888888888888888888P"      8888    8              8
 8  8888888888888888888P"         8888    8             d8
 8  8888888888888888P"            8888    8            d88
 8  8888888888888P"               8888    8           d888
 8  8888888888P"                  8888    8          d8888
 8  8888888P"                     8888    8         d88888
 8  8888P"                        8888    8        d888888
 8  8888oooooooooooooooooooooocgmm8888    8       d8888888
 8 .od88888888888888888888888888888888    8      d88888888
 8888888888888888888888888888888888888    8     d888888888
                                          8    d8888888888
    ooooooooooooooooooooooooooooooo       8   d88888888888
   d                       ...oood8b      8  d888888888888
  d              ...oood888888888888b     8 d8888888888888
 d     ...oood88888888888888888888888b    8d88888888888888
dood8888888888888888888888888888888888b
```

Benefits of Python: Easy to Use Glue, With Leverage

First some simple facts about python

- Compared to other languages it is easy to write.

- It is one of, if not the, most used coding language today. This means there is extensive documentation and a huge, active community of coders to help you with debugging.

Benefits of Python: Easy to Use Glue, With Leverage

- as an interpreted language **gives us access to a host of complex code from different languages**, and the ability to **glue together different bits of code** that might be in completely different languages.
- It also gives us the ability to pull the levers of systems created in languages understood by the interpreter. This makes python **a common choice for writing APIs** (software to allow us to communicate with other software).
- A few examples
  - Blender
    - API for creating Blender Plug Ins in Python
  - Google Maps API
  - Unreal Engine

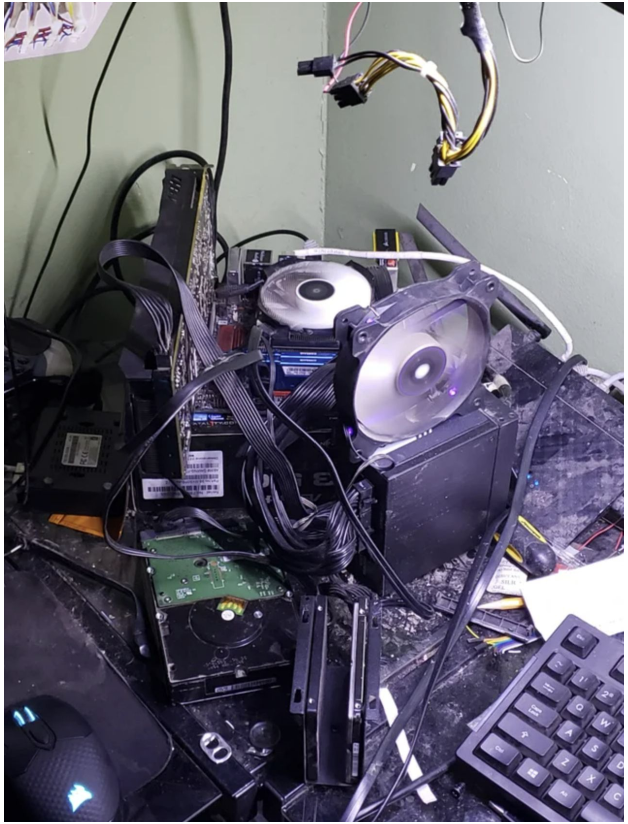
Frankenstein Coding

https://mymodernmet.com/camera-made-of-computer-parts/

Hopefully more like this

And less like this

And less like this



Break

How to get going in Python: IDEs

IDE’s are comprised primarily of three things:

- File System (e.g. finder)
- Text editor (e.g. text edit)
- Way to run code (e.g. terminal)

How to get going in Python: IDEs

Example of Python IDEs

- **Visual Studio Code**
- Pycharm
- Repl.it
- Google Colab