

Python, Day 1: Introduction

Andrew Bydlon

January 2, 2019

Why Python?

Python is one of many programming languages currently in use.

Others include C,C++,C#, Javascript, PHP, Rust, Go, etc.

So why should you invest your time in Python?

Ease of Use

Python is a **high level language**. This is to say that it is far abstracted from machine code and favors the programmer.

In fact, Python by design is made to be very similar to English and thus easily readable.

(C++ Code)

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

```
int main()
{
    cout << "Hello, World!";
    return 0;
}
```

(Python Code)

```
print("Hello, World!")
```

Python is portable and cross-platform

Python can be used on the following operating systems without modification:

- Microsoft Windows
- Mac OS X
- Linux
- FreeBSD
- DOS
- Android
- IOS

Even more esoteric operating systems can easily have Python implemented, since it is **open source**!

Python is interpreted

C++ is a **compiled language**; it converts code from source to machine code.

Python is an **interpreted language**; it converts elements of source code to subroutines which come precompiled into machine code.

This allows much smaller source code (less writing for programmers), but generally results in lower CPU/RAM efficiency for certain tasks.

Python has a HUGE set of libraries

Python is a very extensible language. A small set of available free-to-use packages is listed here:

`https://pypi.python.org/pypi`

This helps to mitigate some issues with efficiency. In fact many scientific computing projects (weather prediction, tensor flow, artificial intelligence, ...) are written exclusively in Python.

Python is Strongly Typed

Python uses spacing, tabs, and line breaks as part of its syntax.

This is optional in other languages (C++), but is generally done by coders to improve readability.

In addition, you declare the type of a variable (naturally) and python will NOT convert between two types of variables dynamically (e.g. an integer versus a string).

Python is Strongly Typed Cont'd

```
> > > "Python is the best! " + 1000
```

Traceback (most recent call last):

File "<input>", line 1, in <module>

TypeError: can only concatenate str (not "int") to str

Whereas other (loosely-typed) languages may convert 1000 from integer to string type, Python does NOT.

```
> > > "Python is the best! " * 1000
```

'Python is the best! Python is the best! Python is the best! Python is the best! Python is the best! Python is the best! ...'

Who uses Python?

- Google/Youtube
- Facebook
- Microsoft
- Dropbox
- Amazon
- Yahoo
- Instagram
- Reddit
- IBM
- Nvidia
- NASA
- Mozilla

This list is very much incomplete, and extends to virtually all computer oriented companies.

Common Types of Python Applications

- 1 Web applications
- 2 Android applications
- 3 GUI applications
- 4 Scientific applications
- 5 Games
- 6 System administration applications
- 7 Console applications

Installing Python3 Libraries!

The newest Python releases are available here:

<https://www.python.org/downloads/release>

Windows: Install the latest 64 bit release:

<https://www.python.org/ftp/python/3.6.4/python-3.6.4-amd64.exe>

Mac OS X: Install the latest 64 bit release:

<https://www.python.org/ftp/python/3.6.4/python-3.6.4-macosx10.6.pkg>

Linux: Generally installed by default (from repos).

Checking the configuration

In any of the above operating systems, you can check that python is installed correctly by issuing the following command from terminal:

```
Python --version  
Python 3.7.2
```

```
Python3 --version  
Python 3.7.2
```

Older versions of python often exist for compatibilities sake:

```
Python2 --version  
Python 2.7.15
```

Python IDEs (integrated development environment) are designed to make coding much simpler. Here are a few options:

- 1 PyCharm (My preference)
- 2 Eric
- 3 Stani's Python Editor
- 4 PyDev
- 5 Spyder
- 6 Visual Studio (Windows)

Exploring PyCharm

- 1 Follow the initial configuration steps.
- 2 Create a new project.
- 3 Add a new “Python File”
- 4 This will open a blank sheet in Script Mode. This allows you to write a program to be executed at any time going forward.
- 5 You can input ctrl+tab to tab through various other utilities. Some useful ones are listed on the next slide.
- 6 Execute your program w/ Run > Run ‘FileName’. You can also hotkey with Shift + F10.

This hotkey allows you to quickly navigate between several important features of the program.

- Python Console: This loads a Python “Interactive Shell”. It is incredibly useful to test segments of code on the fly before adding it to your script.
- Terminal: Opens a terminal session in the directory of your project.
- Run: Window where script runs (as step 6).
- Event Log: Useful information in case of errors.
- Structure: When multiple files are being used, this can be very useful.

Error Types and Correction

If you execute a script, Python's debugger will report errors in a pretty comprehensive manor. Here are some general facts:

- Syntax Errors: Refers to the use of python as a language. E.g. `1+"Two"`, `print("Hello, World)`, `Andrew Bydlon = "Teacher"`.
- Runtime Errors: Errors that occur while the program is running! E.g. `1/0`, Enter & double your favorite number!
- Logical Errors: Program doesn't perform its desired function, but outputs the result. An example is not accounting for order of operations.