The problems of recognizing python commands

When we type Python in the command line it either says that is doesn't recognize the Python command or maybe it gives us a different version that we've installed or maybe we're trying to use a feature that's only available in a later version of Python. And also we're trying to import something that we've installed through pip but keeps giving us import error.

1. Execute Python Commands

```
$ python
Python 2.7.10
```

When we input python command to command line prompt, we can see that the version of Python is 2.7.10. To execute Python with version 3, we should input python3 command to command line.

```
$ python3
```

To check the location of python3 command, we can use which command with command which we want to know.

```
$ which python3
/usr/local/bin/python3
```

So the reason why we can only execute Python 3 with python3 command is that /usr/local/bin directory is on our PATH environment variable. We can use echo command to know which variables are in PATH environment variable.

```
$ echo PATH
```

To execute Python3 with python command, we need to append full path(/usr/local/bin/python3) to PATH environment variable.

2. Setting python command for Python 3

```
$ alias python=python3
```

To make python command for python3 permanently, we need to add aliased command to /.bashrc.

3. Debug of using different version of Python

We can solve this problem using sys.executable command.

sys.executable

A string giving the absolute path of the executable binary for the Python interpreter, on systems where this makes sense. If Python is unable to retrieve the real path to its executable, sys.executable will be an empty string or None.

```
In [2]: import sys
        sys.executable
```

```
Out[2]: '/home/jeongseokgyu/anaconda3/bin/python'
```