CoGrammar

Welcome to this session:
Setting up your Environment Pt 2

The session will start shortly...

Questions? Drop them in the chat.
We'll have dedicated moderators answering questions.

Cyber Security Session Housekeeping

- The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment for all - please engage accordingly. **(Fundamental British Values: Mutual Respect and Tolerance)**
- No question is daft or silly - **ask them!**
- There are **Q&A sessions** midway and at the end of the session, should you wish to ask any follow-up questions. Moderators are going to be answering questions as the session progresses as well.
- If you have any questions outside of this lecture, or that are not answered during this lecture, please do submit these for upcoming Academic Sessions. You can submit these questions here: **Questions**

Cyber Security Session Housekeeping

- For all **non-academic questions**, please submit a query: ***www.hyperiondev.com/support***
- **Report a safeguarding incident: *www.hyperiondev.com/safeguardreporting***
- We would love your feedback on lectures: Feedback on Lectures

Safeguarding & Welfare

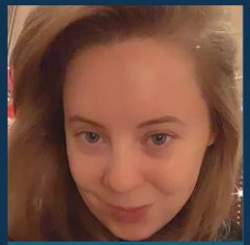
We are committed to all our students and staff feeling safe and happy; we want to make sure there is always someone you can turn to if you are worried about anything.

If you are feeling upset or unsafe, are worried about a friend, student or family member, or you feel like something isn't right, speak to our safeguarding team:

Ian Wyles
Designated Safeguarding Lead

Simone Botes

Rafiq Manan



Charlotte Witcher

Nurhaan Snyman

Ronald Munodawafa

Tevin Pitts

Scan to report a safeguarding concern

or email the Designated Safeguarding Lead:
Ian Wyles
safeguarding@hyperiondev.com

CoGrammar HyperionDev


SKILLS FOR LIFE
SKILLS BOOTCAMPS
Department for Education
CoGrammar
CyberSecurity
October 2024
CoGrammar

Learning Objectives

- Identify the differences between system-installed Python and independent distributions, and the potential issues they pose.
- Explain the purpose and function of the PATH in executing Python and other programs and how to check and configure it.
- Interpret system PATH settings and troubleshoot related installation issues.
- Apply PIP to install, manage, and troubleshoot third-party Python packages.
- Analyze the advantages of using independent Python installations over system versions for development.

Recap

- GitHub – Upload/ Download content
- VS Code – Installation and usage
- Package Managers (Chocolatey & Homebrew)
- Python Installation

Learning Objectives

- Identify the differences between system-installed Python and independent distributions, and the potential issues they pose.
- Explain the purpose and function of the PATH in executing Python and other programs, and how to check and configure it.
- Interpret system PATH settings and troubleshoot related installation issues.
- Apply PIP to install, manage, and troubleshoot third-party Python packages.
- Analyze the advantages of using independent Python installations over system versions for development.

Avoiding Apple's & Windows Distros of Python

- Old Versions
- System Dependency Risks
- Lack of Flexibility
- Install and Path Issues
- Compatibility Problems

The PATH

CoGrammar

Environment Variables - PATH

Environment variables hold values related to the current environment, like the Operating System or user sessions.

One of the most well-known is called PATH on Windows, Linux and Mac OS X. It specifies the directories in which executable programs* are located on the machine that can be started without knowing and typing the whole path to the file on the command line.

The Path

- PATH ↔ Box of Maps
- Map ↔ Points to the Location of a Program

Where to find programs you want to run

The Path - Windows

- C:\Users\YourName\Documents\file.txt
- C:\Python312\Scripts\
- C:\Users\Angela\Documents\PythonProjects\my_script.py
- C:\Python312\python.exe3
- C:\ProgramData**chocolatey**\bin\python.exe

The Path - MacOS & Linux

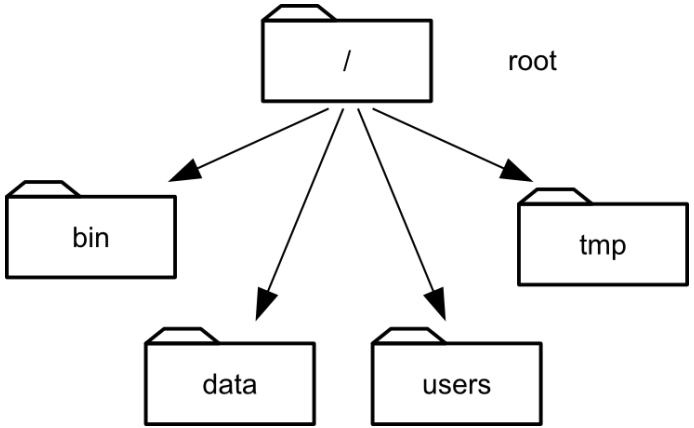
- /usr/local/bin/python3
- /opt/homebrew/bin/python3
- /Users/angela/Documents/PythonProjects/my_script.py
- /home/linuxbrew/.linuxbrew/opt/git/bin

The Path - MacOS & Linux

- /usr/local/bin/python3
- /opt/homebrew/bin/python3
- /Users/angela/Documents/PythonProjects/my_script.py
- /home/linuxbrew/.linuxbrew/opt/git/bin

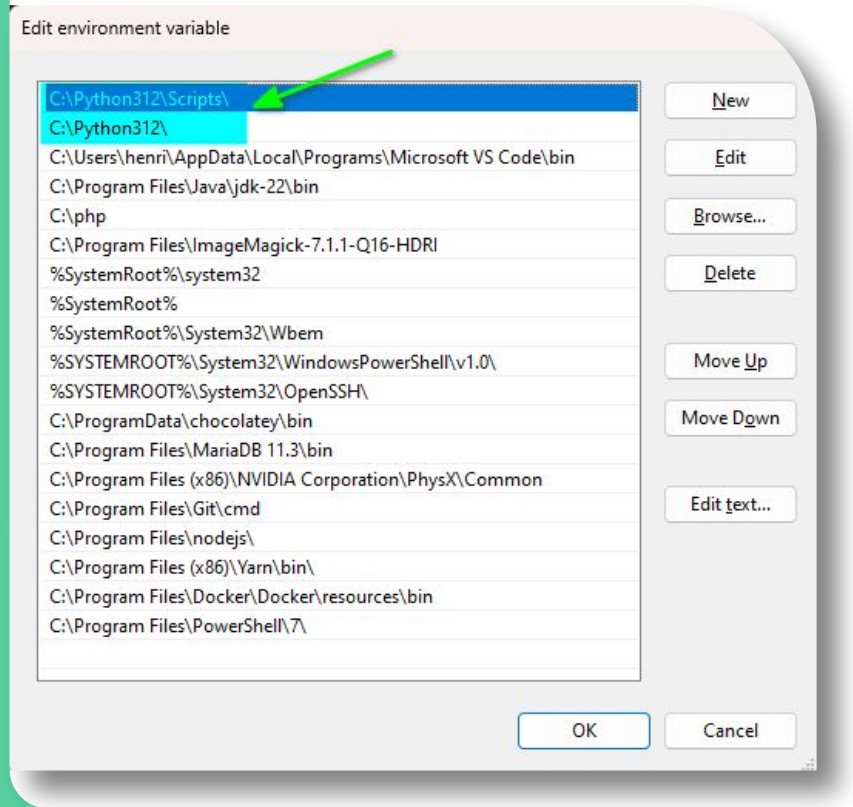
Why we use PATH

- Set Up and Access Python Correctly
- Install and Use Packages
- Use Virtual Environments
- Avoid Conflicts Between Multiple Python Versions
- Automation and Scripting
- Customizing Your Environment



Is the Python "map" in our "box"?

Is the Python "map" in our "box"?




```
which python3
```

```
where python
```

How do I check if my path is correct?

Windows

1. Open Command Prompt or PowerShell.
2. Run the following command:

```
which python3
```

3. This will show the paths to all Python executables found in your PATH. If installed correctly, you should see something like:

```
C:\Users\Angela\AppData\Local\Programs\Python\Python312\python.exe
```

4. Run:

```
python3 --version
```

macOS and Linux

1. Open a terminal.
2. Run the following command
```
which python3
```
3. This will display the full path of the Python executable that is found in your PATH environment variable. If it's correctly installed, you should see a path like **/usr/local/bin/python3 (for macOS)** or **/opt/homebrew/bin/python3 (if using Homebrew).**
4. Run:
```
python3 --version
```

What to do if PATH is incorrect?

Windows

1. Search for "**Environment Variables**".
2. In the **System Properties** window, click on **Environment Variables**.
3. Under System variables, find the Path variable, click Edit, and make sure your Python path

(e.g., C:\Python312\) is added.

macOS/Linux

1. Open your terminal and edit your shell configuration file (.bashrc, .zshrc, or .bash_profile) to add the following line:

```
export PATH="/usr/local/bin/python3:$PATH"
```

2. After editing, run:

```
source ~/.bashrc
```

or

```
source ~/.zshrc
```

Basic Commands

- `pwd` - Prints the current working directory.
- `ls` - Lists the contents of the current directory.
- `cd <directory>` - Changes directory to the specified directory.
- `python filename.py` - Executes a python script.
- `pip install package_name` - Installs Python software packages.

Exploration Starts with Uncertainty

- **How to Add Python to PATH:**

  (Real Python | Ian Currie)
- **Understanding the PATH variable:**

  (Medium | Janel Brandon, 27 March 2020)
- **What is PATH variable in Linux and how it works:**

  (howtouselinux | David Cao, 24 March 2023)

Stay Safe Series:

Mastering Online Safety One week at a Time

While the digital world can be a wonderful place to make education and learning accessible to all, it is unfortunately also a space where harmful threats like online radicalization, extremist propaganda, phishing scams, online blackmail and hackers can flourish.

As a component of this BootCamp the *Stay Safe Series* will guide you through essential measures in order to protect yourself & your community from online dangers, whether they target your privacy, personal information or even attempt to manipulate your beliefs.

Keep it Secret, Keep it Safe: Why Passwords Should Stay Private

- Use a password with 15 to 20 characters including letters and numbers.
- Do **not** share your password with other people.
- Use a password manager.
- Use multi-factor authentication.

PIP: The Package Installer for Python

PIP is a **package-management system** written in Python and is used to install and manage software package

Preferred Installer Program

P I P

PIP Installs Packages

PIP

CoGrammar

Basic Commands

```
import numpy as np      →      pip3 install numpy

import pandas as pd     →      pip3 install pandas

import requests         →      pip3 install requests
```

Conclusion

Questions and Answers

CoGrammar

Thank you for attending

CoGrammar

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education