



ARTIFICIAL INTELLIGENCE

LINUX BASICS FOR AI DEVELOPMENT

ING. CESAR SINCHIGUANO, M.S.C



```
ravi@TecMint:~$ sudo apt install git  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
git is already the newest version (1:2.43.0-1ubuntu7.2).  
0 upgraded, 0 newly installed, 0 to remove and 110 not upgraded.  
ravi@TecMint:~$ git --version  
git version 2.43.0  
ravi@TecMint:~$
```

LINUX BASICS FOR AI DEVELOPMENT



Lecture Goals:

- Understand Linux as the foundation for AI development
- Practice terminal usage, file management, permissions
- Get comfortable navigating and coding on Linux



INTRODUCTION

01

- Why Linux for AI?
 - Open-source & flexible
 - Preferred by researchers and developers
 - Native support for Python, Jupyter, Docker, and ML frameworks
- Real-world examples: Google, OpenAI, NVIDIA use Linux-based systems

"Native support for Python" in Linux means:
Linux already comes with Python installed or can run it easily without needing extra setup.



The terminal (bash shell) in Linux understands how to run Python programs directly using:
`python3 my_script.py`

Google

✓ What they do:

Google runs millions of Linux servers to power its products like:

Search Engine, Gmail, YouTube, Google Maps

Their AI research labs (e.g., Google DeepMind, Google Brain) develop models like AlphaGo and Gemini on Linux systems.

🔍 Why Linux?

Highly customizable

Scales across thousands of machines

Secure and stable

Works natively with TensorFlow (Google's AI framework)

💡 Example:

Google uses Linux in their TPU pods (Tensor Processing Units) to train large AI models. These machines run Debian or Ubuntu Linux, optimized for AI workloads.



Think of it like this:

Device What it is Analogy

💻 Computer A full system: CPU, RAM, storage, OS, etc. A car with engine, wheels, and steering

⚙️ CPU The general-purpose brain of a computer A standard car engine

🎮 GPU Fast parallel processor for graphics & AI A racecar engine (fast at visuals, also useful for AI)

🚀 TPU Specialized chip made just for AI A rocket engine for AI – super fast but only works in a straight line (AI tasks)

What is a TPU?

💡 TPU stands for Tensor Processing Unit.

It's a special type of processor developed by Google specifically for AI and machine learning tasks, especially deep learning.



✓ What they do:

Created ChatGPT, GPT-4, and DALL·E

Trains large language models (LLMs) on powerful GPU clusters running Linux.



Why Linux?

Handles massive data loads efficiently

OpenAI relies on Docker + Linux to train and test models consistently across environments.

Linux supports CUDA, which is essential for using NVIDIA GPUs.



Example:

The codebase for training ChatGPT is developed and tested on Ubuntu Linux. When they deploy APIs (like what you're using now), they're running in containerized Linux environments on cloud infrastructure.

The screenshot shows the ChatGPT interface. On the left is a sidebar with a dark background containing a list of items such as '+ New chat', 'Sentience of AI Program', 'Real World Computational Po', 'AI Development Timeline', 'How to Journal Daily', 'Peaceful Nature Walk and Ch', 'New chat', '"Funny Journaling Excuses"', 'Clear conversations', 'Light mode', 'OpenAI Discord', 'Updates & FAQ', and 'Log out'. To the right of the sidebar, the main area is titled 'ChatGPT' and contains three sections: 'Examples', 'Capabilities', and 'Limitations'. The 'Examples' section shows three examples with responses: 'Explain quantum computing in simple terms' → 'Remembers what user said earlier in the conversation' → 'May occasionally generate incorrect information'. The 'Capabilities' section shows three examples: 'Got any creative ideas for a 10 year old's birthday?' → 'Allows user to provide follow-up corrections' → 'May occasionally produce harmful instructions or biased content'. The 'Limitations' section shows two examples: 'How do I make an HTTP request in Javascript?' → 'Trained to decline inappropriate requests' → 'Limited knowledge of world and events after 2021'. At the bottom of the main area, there is a footer note: 'ChatGPT Jan 9 Version. Free Research Preview. Our goal is to make AI systems more natural and safe to interact with. Your feedback will help us improve.'



Sam Altman



NVIDIA

✓ What they do:

Builds GPUs (graphics cards) used for gaming, deep learning, and AI.

Develops AI tools like CUDA, cuDNN, TensorRT, and NVIDIA DeepStream.

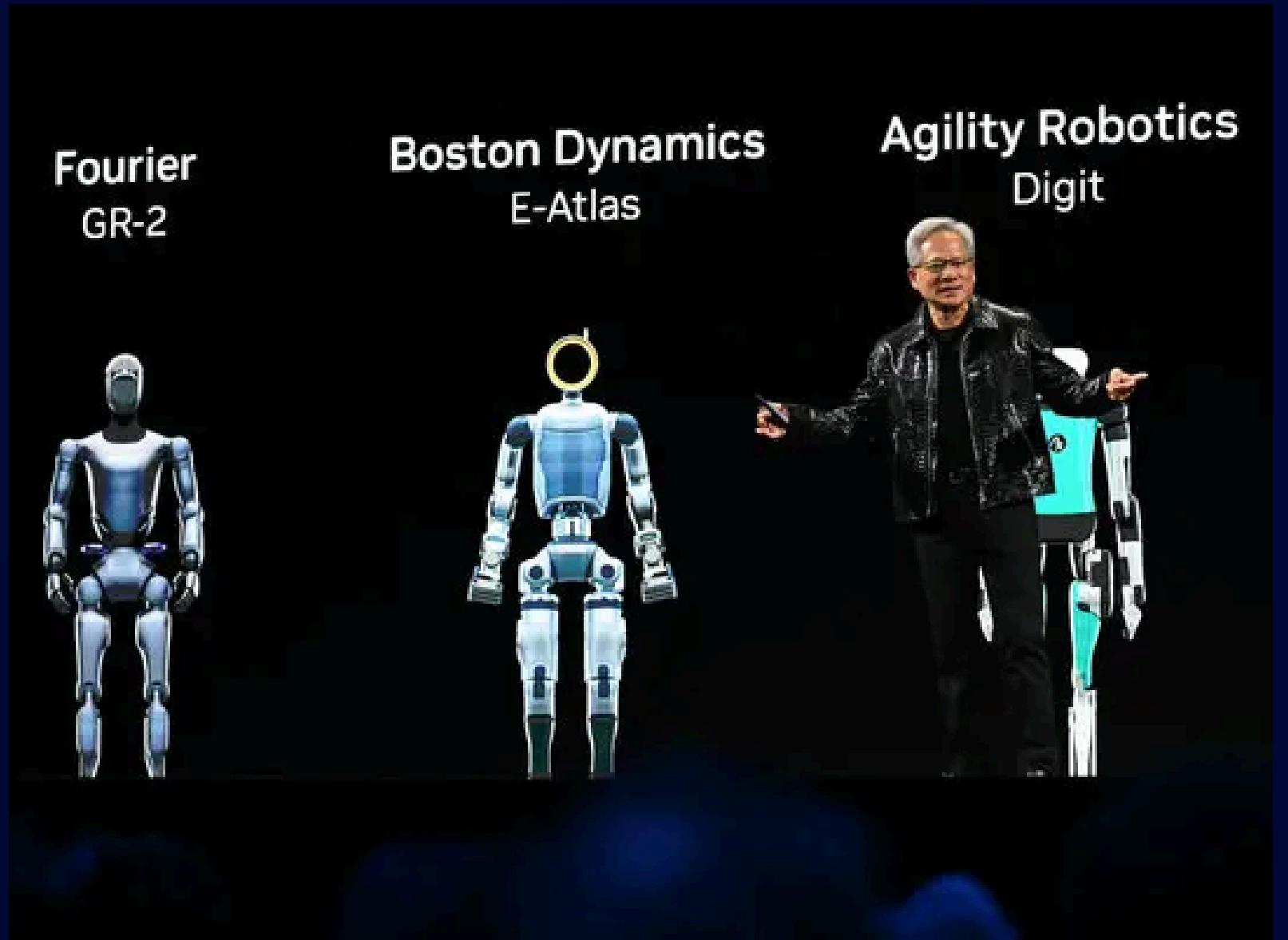
🔍 Why Linux?

All NVIDIA's AI frameworks and drivers are built for Linux first.

CUDA toolkit (for running AI models on GPUs) is optimized for Ubuntu, CentOS, and other Linux distros.

🧪 Example:

NVIDIA provides deep learning Docker containers through NGC (NVIDIA GPU Cloud). These containers are based on Linux and come pre-installed with TensorFlow, PyTorch, and more.





TERMINAL ESSENTIALS (IN TERMINATOR)

01

```
pwd      # Show current directory  
ls       # List files and folders  
cd       # Change directory  
mkdir   # Make a new directory  
touch   # Create empty files  
rm      # Remove files  
mv      # Move or rename  
cp      # Copy files  
clear   # Clear the terminal
```

02

Practice:

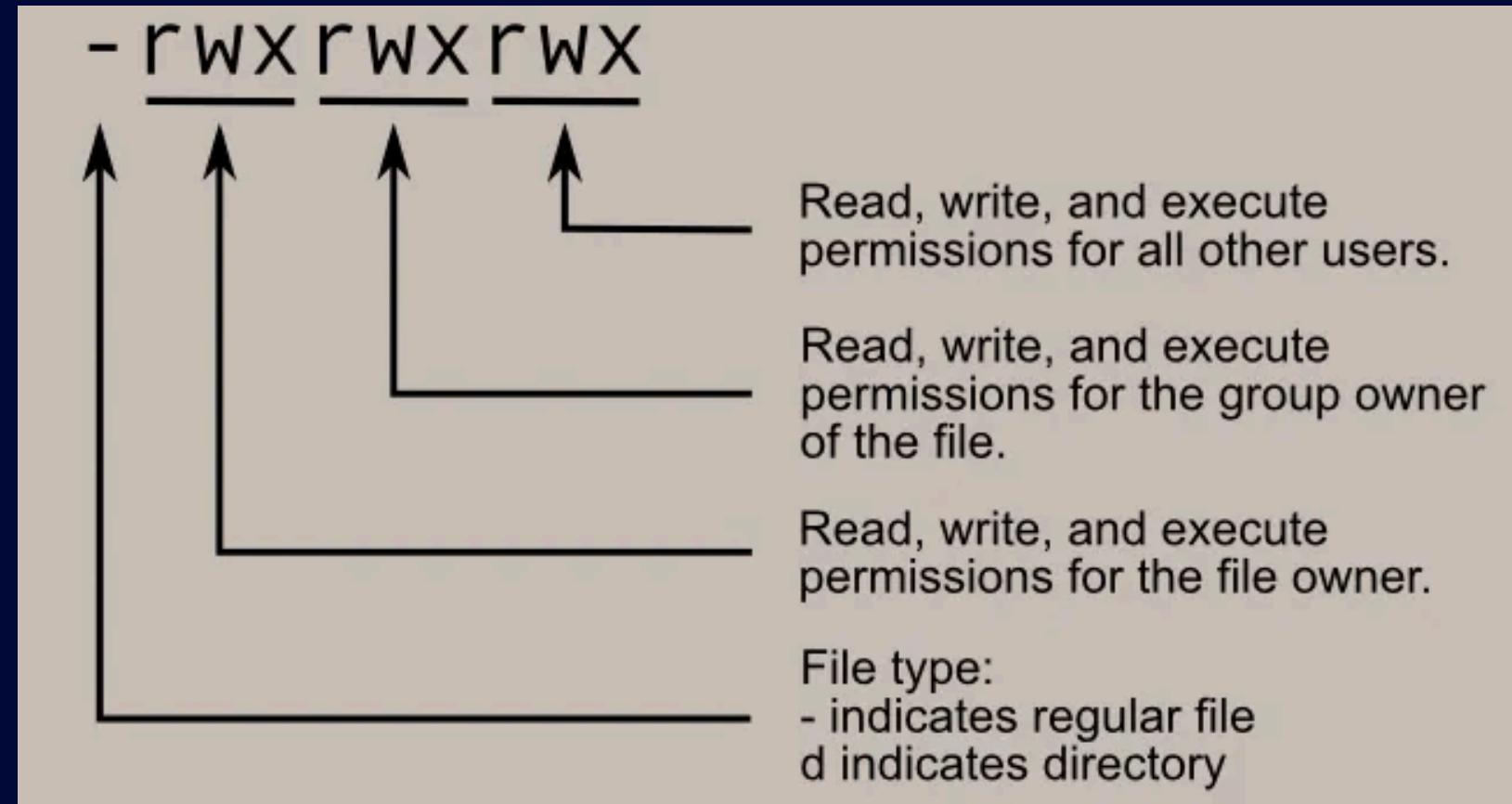
 Navigate to home folder
 Create ai_projects folder
 Inside it, create 2 folders: datasets, scripts
 Create a file readme.txt with touch

```
htop 82x42  
Tasks: 42, 73 thr; 1 running  
Load average: 0.07 0.04 0.01  
Uptime: 00:04:13  
Mem: 245M/12.4G  
Swap: 0K/4.00G  
  
PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command  
1456 root 20 0 18760 13056 6644 S 8.1 0.1 0:01.56 /usr/bin/python3 /t  
189 ltoinel 20 0 480M 63528 38316 S 7.4 0.5 0:03.04 /usr/bin/python3 /u  
590 ltoinel 20 0 5172 3488 2948 R 0.7 0.0 0:00.37 htop  
685 ltoinel 20 0 10976 6856 4528 S 0.0 0.1 0:00.19 /usr/bin/zsh  
425 ltoinel 20 0 353M 77588 28068 S 0.0 0.6 0:01.24 xfdesktop  
505 ltoinel 20 0 22660 320 0 S 0.0 0.0 0:00.02 xcape -e Super_L Co  
443 ltoinel 20 0 386M 8548 7528 S 0.0 0.1 0:00.02 /usr/libexec/gvfs-a  
171 ltoinel 20 0 312M 34612 27632 S 0.0 0.3 0:00.71 xfce4-panel  
142 ltoinel 20 0 166M 6312 5704 S 0.0 0.0 0:00.05 /usr/lib/at-spi2-co  
591 ltoinel 20 0 14028 7756 5080 S 0.0 0.1 0:01.13 /usr/bin/zsh  
177 ltoinel 20 0 312M 35944 28024 S 0.0 0.3 0:00.38 /usr/lib/x86_64-lin  
178 Terminal 20 0 2120 24064 25824 C 0 0 0.2 0:00.01 /home/h2s/.xterm-148  
  
keys: any: refresh q: quit j: jump sort: r: asc left: SWAPIN r: rev
```

```
ubuntu.com/ubuntu jammy-updates InRelease [114 kB]  
ubuntu.com/ubuntu jammy-backports InRelease [99.8 kB]  
ubuntu.com/ubuntu jammy-updates/main i386 Packages [327 kB]  
ubuntu.com/ubuntu jammy-updates/main amd64 Packages [612 kB]  
ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [427 kB]  
jd.io/eugeny/tabby/ubuntu jammy InRelease  
(75.6 kB/s)  
Done  
... Done  
.... Done  
aded. Run 'apt list --upgradable' to see them.
```

```
forum: ~ 52x11  
142.250.182.174): icmp_seq=2  
;  
142.250.182.174): icmp_seq=2  
;  
142.250.182.174): icmp_seq=2  
;  
142.250.182.174): icmp_seq=2  
;  
142.250.182.174): icmp_seq=2  
;
```

```
h2s@forum: ~ 47x11  
h2s@forum:~$ cat /etc/os-release  
PRETTY_NAME="Ubuntu 22.04 LTS"  
NAME="Ubuntu"  
VERSION_ID="22.04"  
VERSION="22.04 (Jammy Jellyfish)"  
VERSION_CODENAME=jammy  
ID=ubuntu  
ID_LIKE=debian  
HOME_URL="https://www.ubuntu.com/"  
SUPPORT_URL="https://help.ubuntu.com/"  
BUG_REPORT_URL="https://bugs.launchpad.net/  
+.../
```



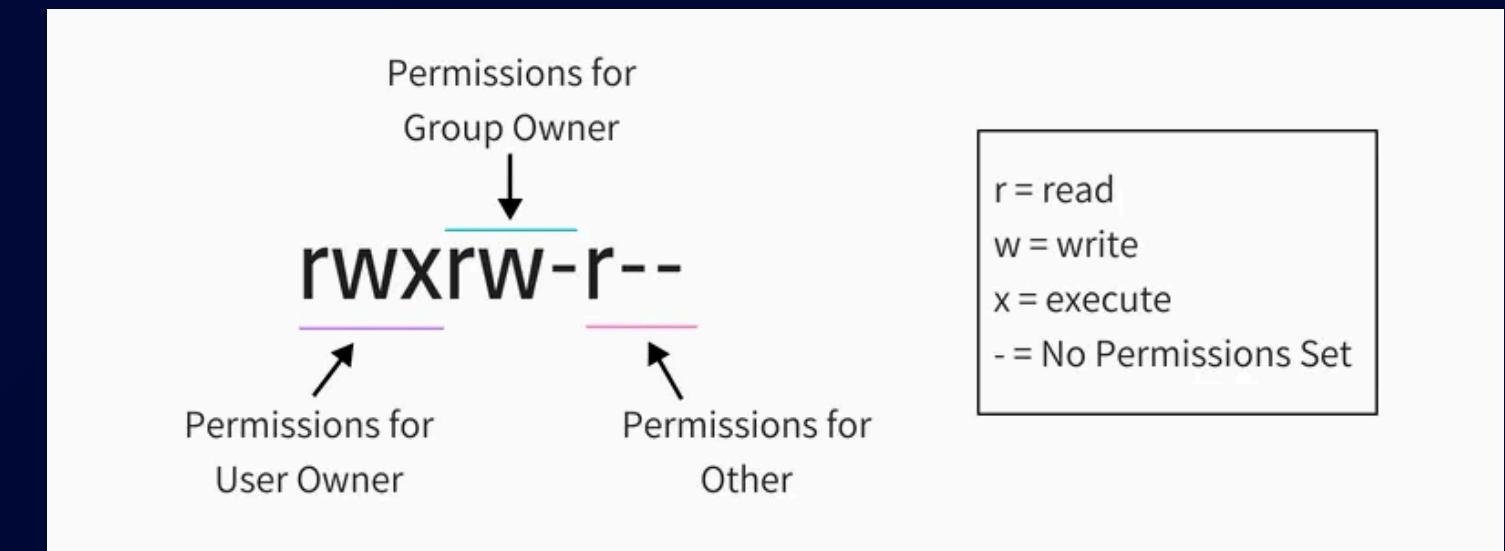
Explain permission meanings (rwx)

Practice:

Create a Python file

Make it executable

Show how to run it: ./hello.py



FILE PERMISSIONS & OWNERSHIP

01

```
ls -l      # List with permissions
chmod +x script.py # Make script executable
chown user:group # Change file owner
```

Run an Executable from the Terminal
If you're in Linux, and the Python file is executable, you can run it like this:

Step 1: Make it executable (only once)
chmod +x your_script.py

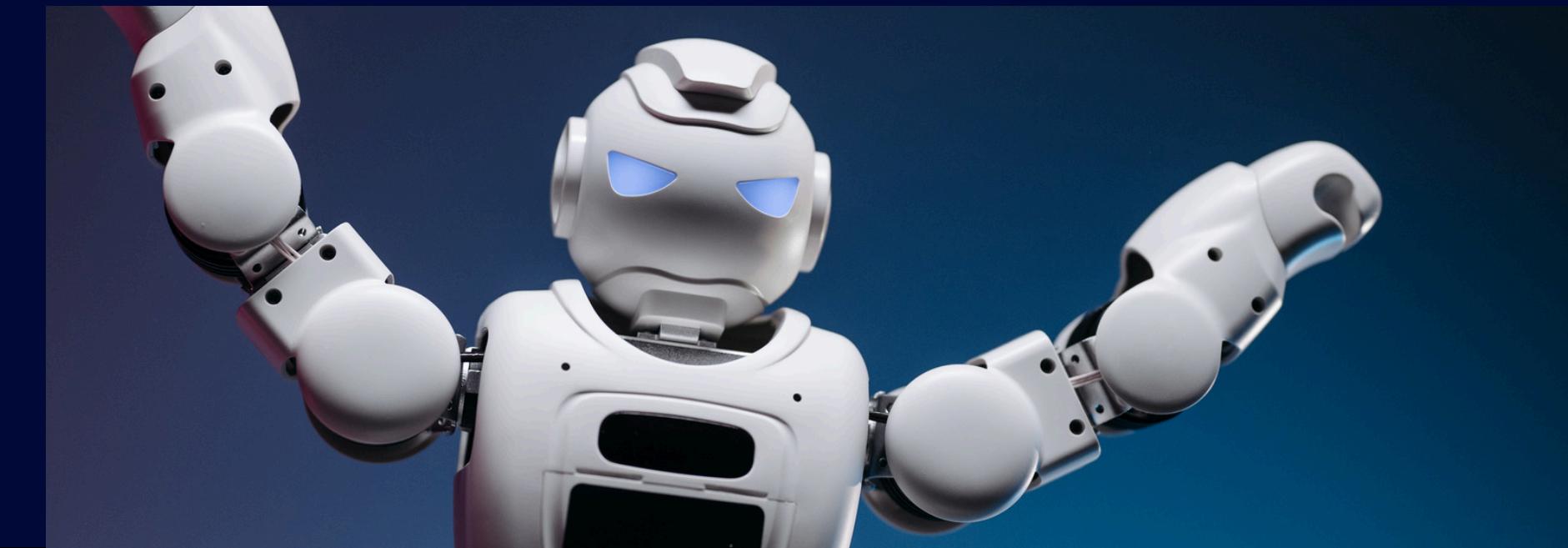
Step 2: Add a shebang line to the top of the file (very important!)

Inside your_script.py, the first line must be:
#!/usr/bin/env python3

Step 3: Run it directly
.your_script.py



SYSTEM MONITORING TOOLS



CPU	0.7%	Tasks: 53, 44 thr; 1 running									
Mem	237M/1.05G	Load average: 0.37 0.11 0.04									
Swp	42.0M/9.00G	Uptime: 05:41:34									
<hr/>											
PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU% ▾	MEM%	TIME+	Command
38228	root	20	0	28020	4064	3352	R	0.7	0.4	0:00.09	htop
1	root	20	0	258M	10920	7056	S	0.0	1.0	0:05.41	/usr/lib/system
562	root	20	0	92964	8588	7748	S	0.0	0.8	0:01.14	/usr/lib/system
599	root	20	0	116M	8980	5536	S	0.0	0.8	0:00.72	/usr/lib/system
716	rpc	20	0	67120	4616	4280	S	0.0	0.4	0:00.05	/usr/bin/rpcbin
726	root	16	-4	139M	2284	1880	S	0.0	0.2	0:00.08	/sbin/auditd
727	root	16	-4	139M	2284	1880	S	0.0	0.2	0:00.00	/sbin/auditd
728	root	16	-4	48484	2136	1832	S	0.0	0.2	0:00.03	/usr/sbin/sedis
729	root	16	-4	139M	2284	1880	S	0.0	0.2	0:00.01	/sbin/auditd
751	root	20	0	83672	5912	5048	S	0.0	0.5	0:00.43	/usr/lib/system
755	avahi	20	0	82752	3412	3188	S	0.0	0.3	0:00.23	avahi-daemon: r
756	root	20	0	207M	8508	8104	S	0.0	0.8	0:00.16	/usr/sbin/sssd
760	root	20	0	26368	3692	3168	S	0.0	0.3	0:00.05	/usr/sbin/smarts
762	root	20	0	451M	7484	6088	S	0.0	0.7	0:00.09	/usr/sbin/Modem
765	polkitd	20	0	1599M	19936	13248	S	0.0	1.8	0:03.59	/usr/lib/polkit
766	dbus	20	0	91772	7692	4940	S	0.0	0.7	0:02.89	/usr/bin/dbus-d
769	chrony	20	0	125M	2884	2460	S	0.0	0.3	0:00.22	/usr/sbin/chron
772	libstorag	20	0	18872	1908	1780	S	0.0	0.2	0:00.13	/usr/bin/lsmd -
773	root	20	0	17408	1948	1800	S	0.0	0.2	0:00.00	/usr/sbin/mcelo
775	rngd	20	0	156M	5812	5312	S	0.0	0.5	0:07.22	/sbin/rngd -f -
779	root	20	0	25244	1960	1668	S	0.0	0.2	0:00.58	/bin/bash /usr/
784	root	20	0	99M	2704	2336	S	0.0	0.2	0:00.00	/usr/sbin/gsspr
785	root	20	0	451M	7484	6088	S	0.0	0.7	0:00.00	/usr/sbin/Modem
786	root	20	0	99M	2704	2336	S	0.0	0.2	0:00.00	/usr/sbin/gsspr
787	root	20	0	99M	2704	2336	S	0.0	0.2	0:00.00	/usr/sbin/gsspr

top # See running processes

htop # (Install first)

df -h # Disk usage

free -m # RAM usage

kill PID

If it doesn't stop (force kill)

kill -9 PID

Practice:

- Monitor your system
- Find Python processes if running

In short:

top is lighter and built-in.

htop is much easier to use, especially when managing processes.



CONTACT US

 +593990296602

 www.reallygreatsite.com

 cesar.sinchiguano@gmail.com
cesar.sinchiguano@uleam.edu.ec

 Uleam - Ext El Carmen





THANK YOU FOR LISTENING

ARTIFICIAL INTELLIGENCE

