



Getting Data in Python

stdin & stdout

By Armita Razavi

In This Video

💡 sys.argv

💡 stdin

💡 stdout

💡 Command Line Data Pipe

What is sys?

sys module provides access to some variables used by the interpreter

sys.argv

The list of **command line arguments** passed to a Python script.

argv[0] is the script name.

argv[1] is the first argument after the Python script

argv[2] is the second argument after the Python script

...

sys.argv

```
python myfile.py "[0-9]" "text"
```

argv[0]

argv[1]

argv[2]

```
sys.argv = ['myfile.py', '[0-9]', 'text']
```

stdin and stdout

If you run your Python scripts at the command line, you can **pipe data** through them using **sys.stdin** and **sys.stdout**

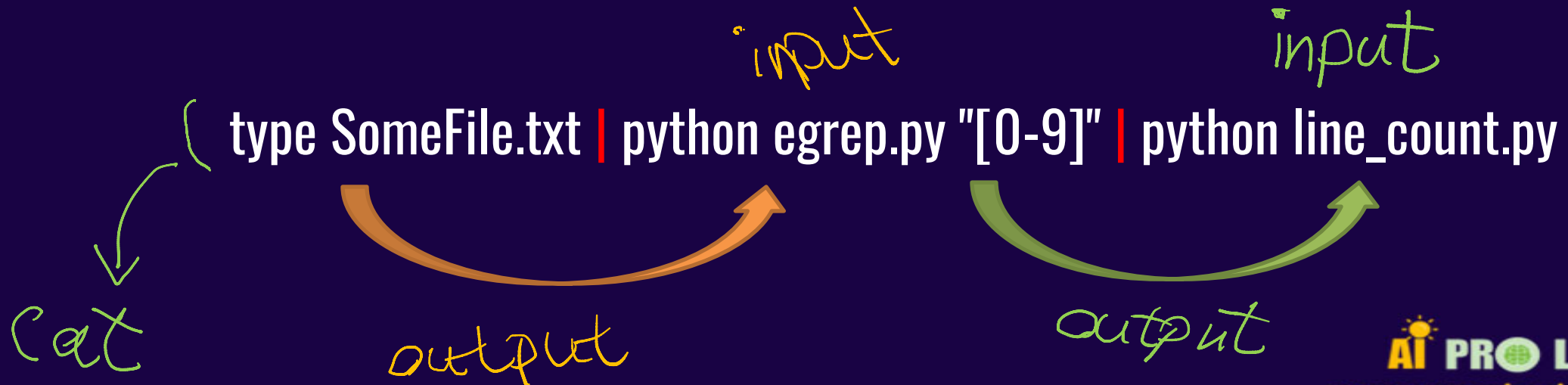
stdin: is used for all interactive input (including calls to `input()`)

stdout: is used for the output of `print()`

Command Line Data Pipe

The **|** is the **pipe character**, which means “use the output of the left command as the input of the right command.”

You can build **data-processing pipelines** this way.



Warm Up Practice

Write a Python script to give **your name** and **your lastname** as two arguments in command line and print the **full name**

Using the **pipe character** in command line, passes the **SomeFile.txt** and output only lines contain “)” character. Also count the number of lines having “)”