COL1000
Introduction to Programming

Priyanka Golia

Most (if not all) of the content is borrowed from Prof. Subodh Kumar's slides




*COL*1000 so far!!

Time for a breather!

Time for a breather!

Never have I ever:

1. Performed in front of 100+ people !
2. Copied homework from a friend/ let a friend copy my homework.
3. Written a computer program.
4. Wrote an essay.

Who Are We Talking To?

Writing Essay: Human Audience

- **Purpose** : To convey ideas, emotions, stories, and information to other humans.
- **Interpretation** : Relies on shared cultural context and human experience.
- **Feedback** : Emotional responses, questions, discussions, and engagement.

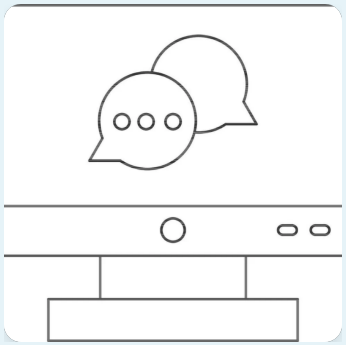
Who Are We Talking To?

Writing Essay: Human Audience

- **Purpose** : To convey ideas, emotions, stories, and information to other humans.
- **Interpretation** : Relies on shared cultural context and human experience.
- **Feedback** : Emotional responses, questions, discussions, and engagement.

Writing Programs: Computer Interpreter

- **Purpose** : To give commands, automate tasks, and solve computational problems.
- **Interpretation** : Strictly follows predefined syntax and logical rules.
- **Feedback** : Binary execution (success/error), specific error messages, and output data.



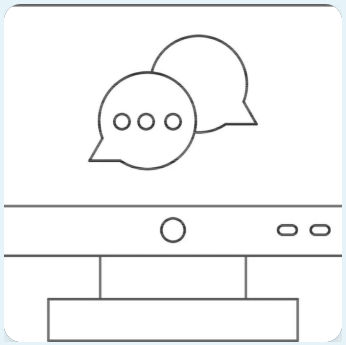
Who Are We Talking To?

Writing Essay: Human Audience

- **Purpose** : To convey ideas, emotions, stories, and information to other humans.
- **Interpretation** : Relies on shared cultural context and human experience.
- **Feedback** : Emotional responses, questions, discussions, and engagement.

Writing Programs: Computer Interpreter

- **Purpose** : To give commands, automate tasks, and solve computational problems.
- **Interpretation** : Strictly follows predefined syntax and logical rules.
- **Feedback** : Binary execution (success/error), specific error message, and output data.



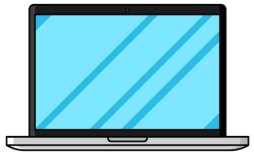
Programming!

To give commands, automate tasks, and solve computational problems.

Make it do a task! — just program it!

Programming!

To give commands, automate tasks, and solve computational problems.



Make it do a task! — just program it!

Programming!

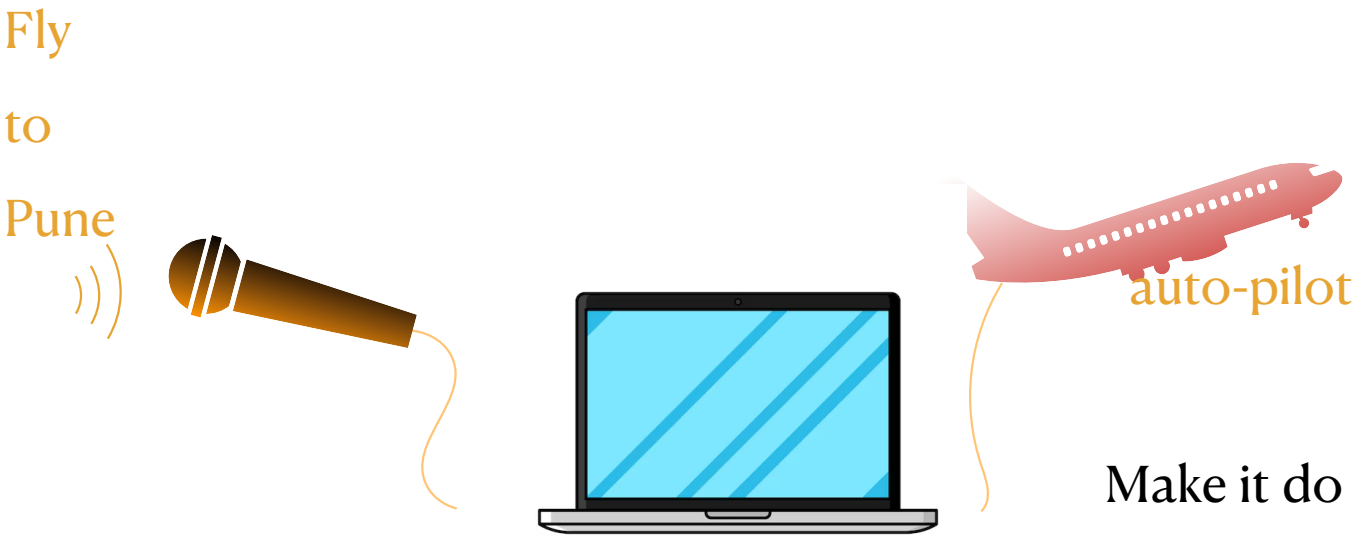
To give commands, automate tasks, and solve computational problems.

Make it do a task! — just program it!

1. Decide the operations you want your calculator to perform: +, -, *, %, log, etc.
2. Write a description of the task.
   Addition should take two inputs and return their sum.
3. Tell this description to the computer (i.e., program it)
4. Run the program and get the output.
5. Check output to verify that it behaves as expected.

Programming!

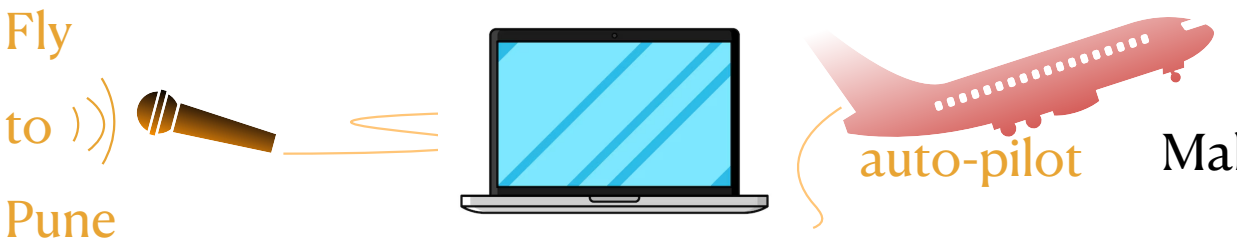
To give commands, automate tasks, and solve computational problems.



Make it do a task! — just program it!

Programming!

To give commands, automate tasks, and solve computational problems.

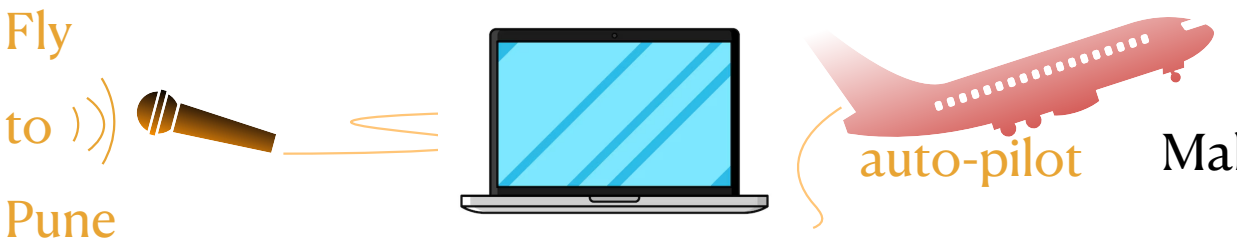


Make it do a task! — just program it!

1. What actions does the “system” support
2. Design task description in terms of those actions

Programming!

To give commands, automate tasks, and solve computational problems.



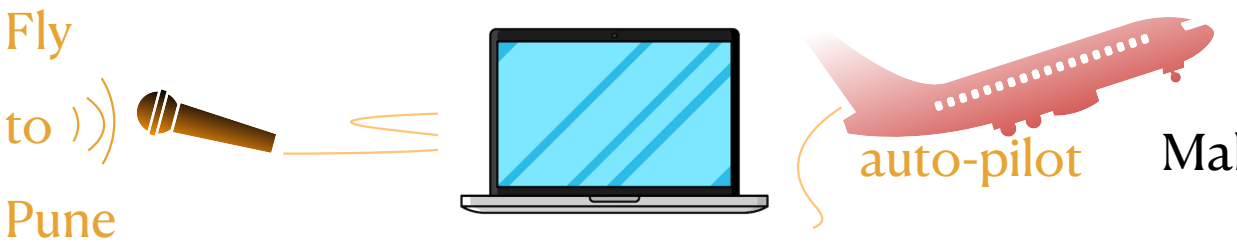
Make it do a task! — just program it!

1. What actions does the “system” support
2. Design task description in terms of those actions

Is it possible ?

Programming!

To give commands, automate tasks, and solve computational problems.



Make it do a task! — just program it!

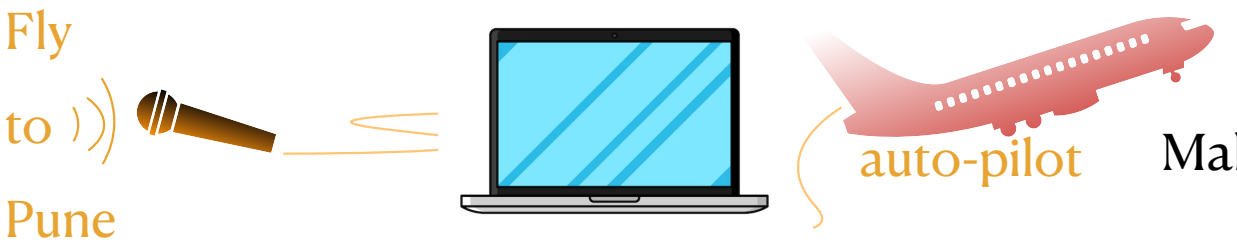
1. What actions does the “system” support
2. Design task description in terms of those actions

Is it possible ?

3. Communiate the description

Programming!

To give commands, automate tasks, and solve computational problems.



Make it do a task! — just program it!

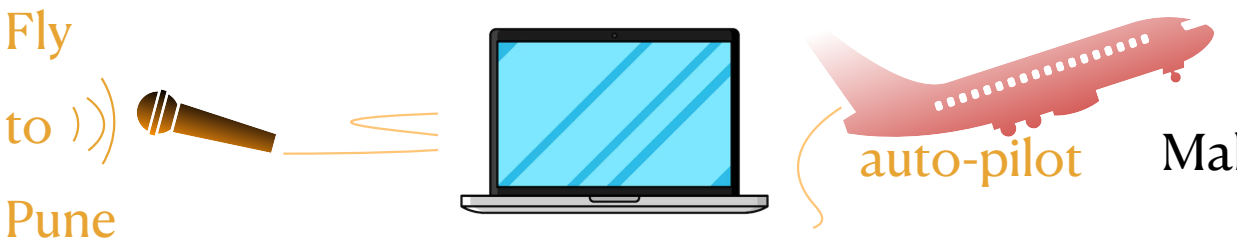
1. What actions does the "system" support
2. Design task description in terms of those actions

Is it possible ?

3. Communiate the description Program

Programming!

To give commands, automate tasks, and solve computational problems.

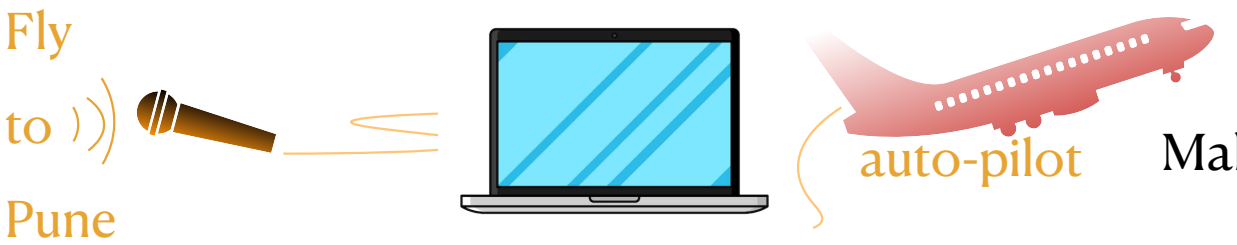


Make it do a task! — just program it!

1. What actions does the "system" support
2. Design task description in terms of those actions — Is it possible ?
3. Communiate the description Program — And, receive the results

Programming!

To give commands, automate tasks, and solve computational problems.



Make it do a task! — just program it!

1. What actions does the “system” support
2. Design task description in terms of those actions

Is it possible ?

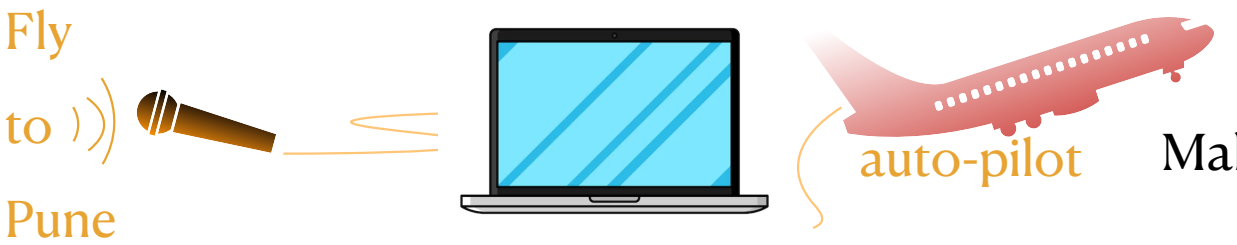
3. Communiate the description Program

And, receive the results

4. Check results

Programming!

To give commands, automate tasks, and solve computational problems.



Make it do a task! — just program it!

1. What actions does the "system" support
2. Design task description in terms of those actions

Is it possible ?

3. Communiate the description Program

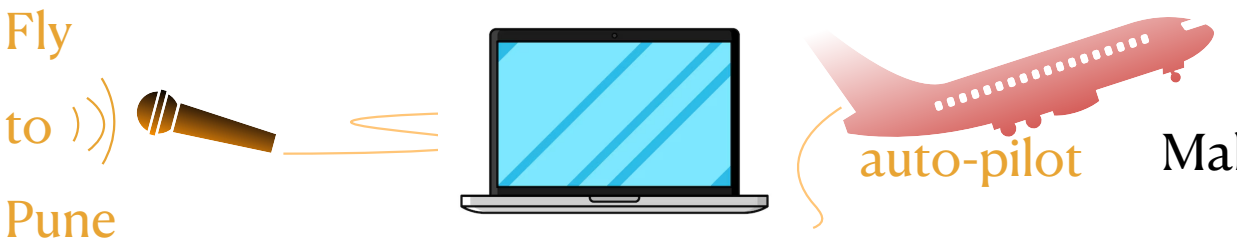
And, receive the results

4. Check results

Should be able to interpret and verify

Programming!

To give commands, automate tasks, and solve computational problems.



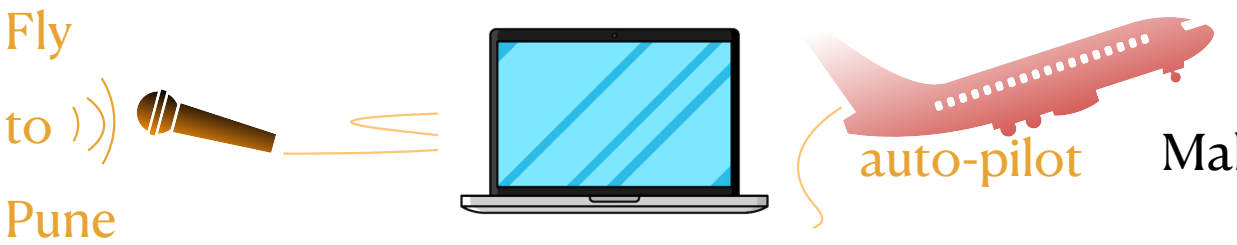
Make it do a task! — just program it!

Comprises many smaller programs:

Communicate: Buttons, Dials, Switches, Display panelss

Programming!

To give commands, automate tasks, and solve computational problems.



Make it do a task! — just program it!

Comprises many smaller programs:

Communicate: Buttons, Dials, Switches, Display panelss

Most do only one specific task

Programming!

To give commands, automate tasks, and solve computational problems.

Programming!

To give commands, automate tasks, and solve computational problems.

Programming!

To give commands, automate tasks, and solve computational problems.

Input: Communicate to device

Programming!

To give commands, automate tasks, and solve computational problems.

Output: Device communicates back

Input: Communicate to device

Program ➜ Device

Data ➜ Program

Programming!

To give commands, automate tasks, and solve computational problems.

Output: Device communicates back

Input: Communicate to device

Program ➜ Device

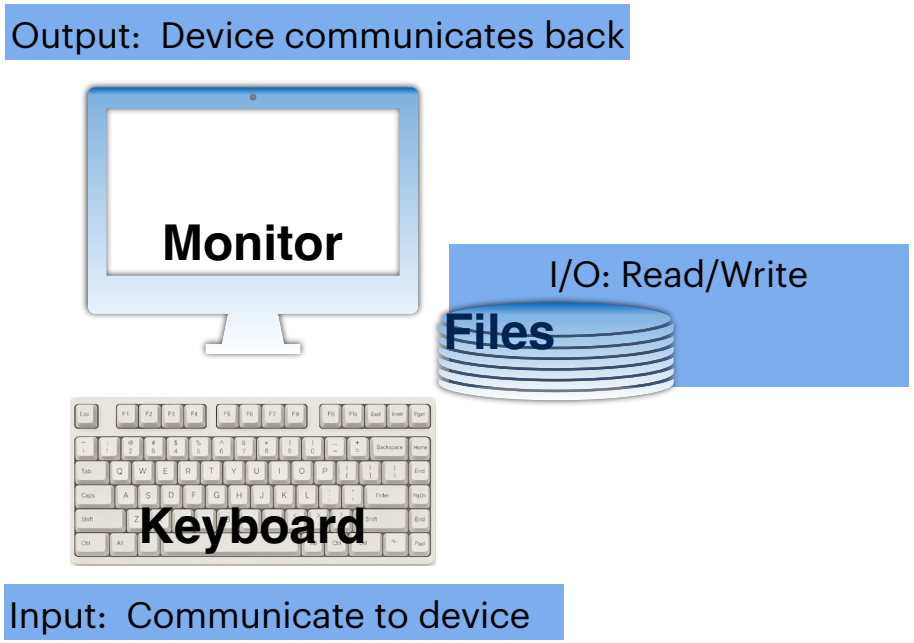
Data ➜ Program

☞Program/Data can be in storage (Files)

(In addition to being typed on the keyboard)

Programming!

To give commands, automate tasks, and solve computational problems.



Program ➜ Device

Data ➜ Program

☞Program/Data can be in storage (Files)

(In addition to being typed on the keyboard)

Elements of Programming

Elements of Programming

Elements of Programming

- Not programmable: Accepts one key at a time,
  Each key result in a fixed action

Elements of Programming

- Not programmable: Accepts one key at a time,
  Each key result in a fixed action

- An important aspect of programming is to remember what has happened so far — (including prior input, prior actions, prior output)

Elements of Programming

- Not programmable: Accepts one key at a time,
  Each key result in a fixed action
- An important aspect of programming is to remember what has happened so far — (including prior input, prior actions, prior output)

  The next action depend on
  the current state and any provided input
- The result may be committed to memory

Elements of Programming

- Not programmable: Accepts one key at a time,
  Each key result in a fixed action
- An important aspect of programming is to remember what has happened so far — (including prior input, prior actions, prior output)

  The next action depend on
  the current state and any provided input
- The result may be committed to memory

Programs have memory so they may remember

Elements of Programming

- Not programmable: Accepts one key at a time,
  Each key result in a fixed action
- An important aspect of programming is to remember what has happened so far — (including prior input, prior actions, prior output)

  The next action depend on
  the current state and any provided input
- The result may be committed to memory

Programs have memory so they may remember
and actions which are taken on parts of memory

Programming!

To give commands, automate tasks, and solve computational problems.

"The System"

1. Notices the command being typed
2. Repeats the command (on the monitor)
3. Interprets and executes the command
5. Provides the answer on monitor when commanded

Programming!

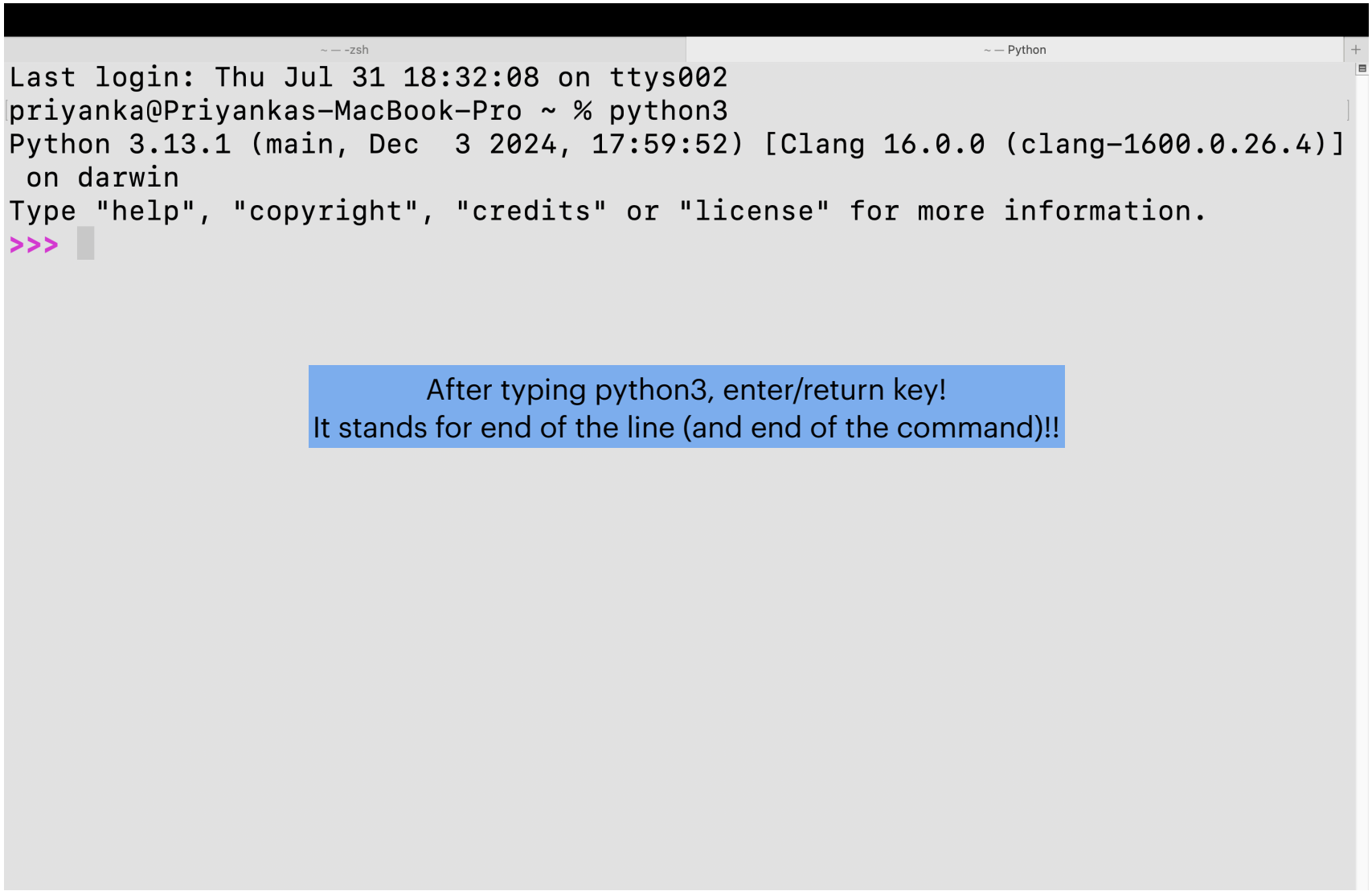
To give commands, automate tasks, and solve computational problems.

"The System"

1. Notices the command being typed
2. Repeats the command (on the monitor)
3. Interprets and executes the command
5. Provides the answer on monitor when commanded

First tell the system to start taking commands in Python language


~ — -zsh
~ — Python
Last login: Thu Jul 31 18:32:08 on ttys002
priyanka@Priyankas-MacBook-Pro ~ % python3
Python 3.13.1 (main, Dec 3 2024, 17:59:52) [Clang 16.0.0 (clang-1600.0.26.4)]
 on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>


~ — -zsh
~ — Python
Last login: Thu Jul 31 18:32:08 on ttys002
priyanka@Priyankas-MacBook-Pro ~ % python3
Python 3.13.1 (main, Dec 3 2024, 17:59:52) [Clang 16.0.0 (clang-1600.0.26.4)]
 on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
After typing python3, enter/return key!
It stands for end of the line (and end of the command)!!

Hello Python!

- We will (mainly) use Python on the cloud (Moodle)
  - ➡ You will not get Python prompt — unlike Google colab, Jupyter notebook
  - ➡ Instead you will type Python code in IDE ➜ Save  in a remote file ➜ Run it 
- Indentation of lines is very important in Python
  - ➡ Where code begins on each line

Hello Python! print

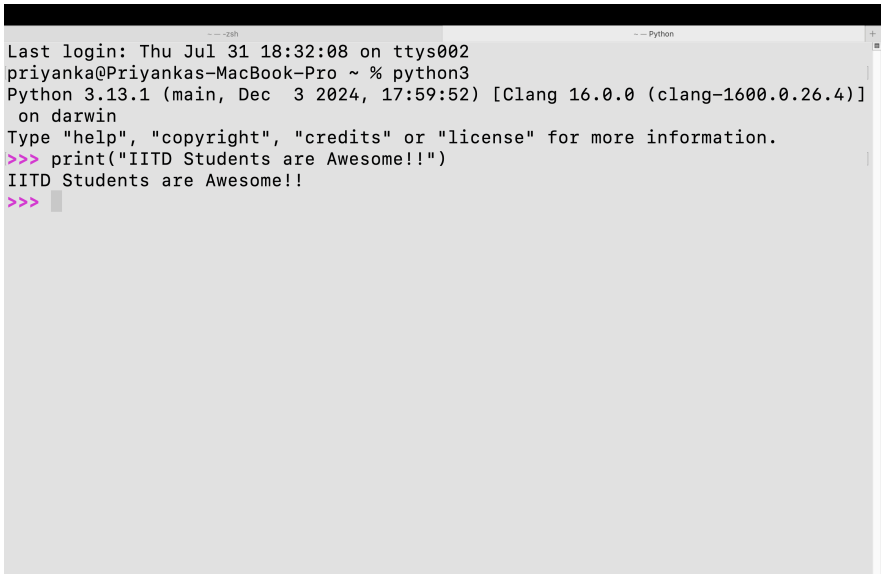
!

Let's start with the most basic Python command -

The print function displays text or values to the console:

```
>>> print("IITD students are awesome!")
```

```
IITD students are awesome!
```



It's that simple! Just type your message in quotes inside the print function.

Hello Python! print

!

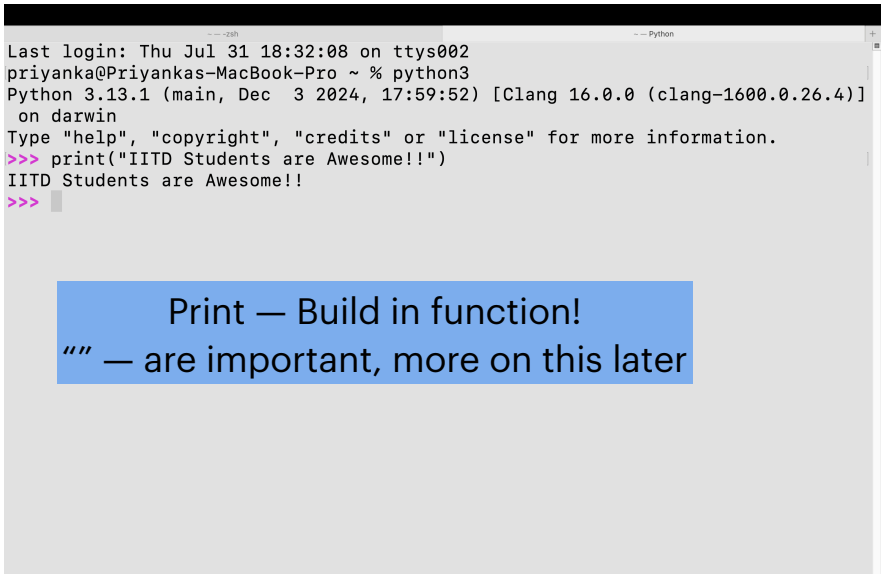
Let's start with the most basic Python command -

The print function displays text or values to the console:

```
>>> print("IITD students are awesome!")
```

```
IITD students are awesome!
```

It's that simple! Just type your message in quotes inside the print function.



Hello Python!

```
priyanka@Priyankas-MacBook-Pro ~ % python3
Python 3.13.1 (main, Dec  3 2024, 17:59:52) [Clang
16.0.0 (clang-1600.0.26.4)] on darwin
Type "help", "copyright", "credits" or "license" fo
r more information.
>>> input("name")
namepriyanka
'priyanka'
>>> print("hello", name)
Traceback (most recent call last):
  File "<python-input-1>", line 1, in <module>
    print("hello", name)
                   ^^^^
NameError: name 'name' is not defined
```

To allow program to read the keyboard input

input("some prompt")

Hello Python!

```
priyanka@Priyankas-MacBook-Pro ~ % python3
Python 3.13.1 (main, Dec  3 2024, 17:59:52) [Clang
16.0.0 (clang-1600.0.26.4)] on darwin
Type "help", "copyright", "credits" or "license" fo
r more information.
>>> input("name")
namepriyanka
'priyanka'
>>> print("hello", name)
Traceback (most recent call last):
  File "<python-input-1>", line 1, in <module>
    print("hello", name)
                   ^^^^
NameError: name 'name' is not defined
```

To allow program to read the keyboard input

input("some prompt")

We took the input, but didn't ask program to memorize it