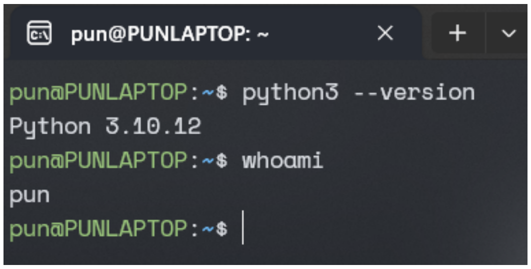
**Submitted by: Puneet Vaswani**
**E-mail: puneetvaswani0609@gmail.com**
**Ph. No.: 8619661188**

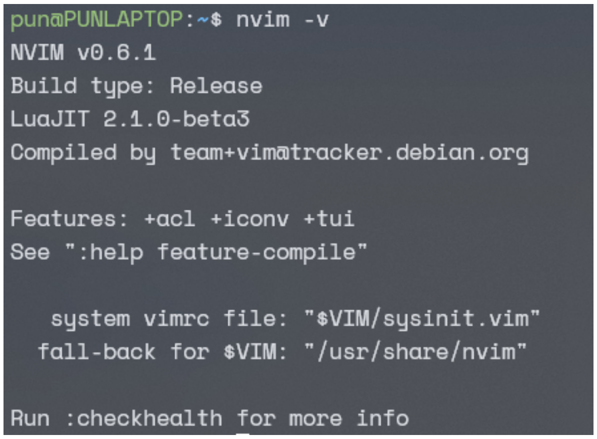
# TOPIC-1

1. Install Python 3 on your computer.



```
pun@PUNLAPTOP:~$ python3 --version
Python 3.10.12
pun@PUNLAPTOP:~$ whoami
pun
pun@PUNLAPTOP:~$
```

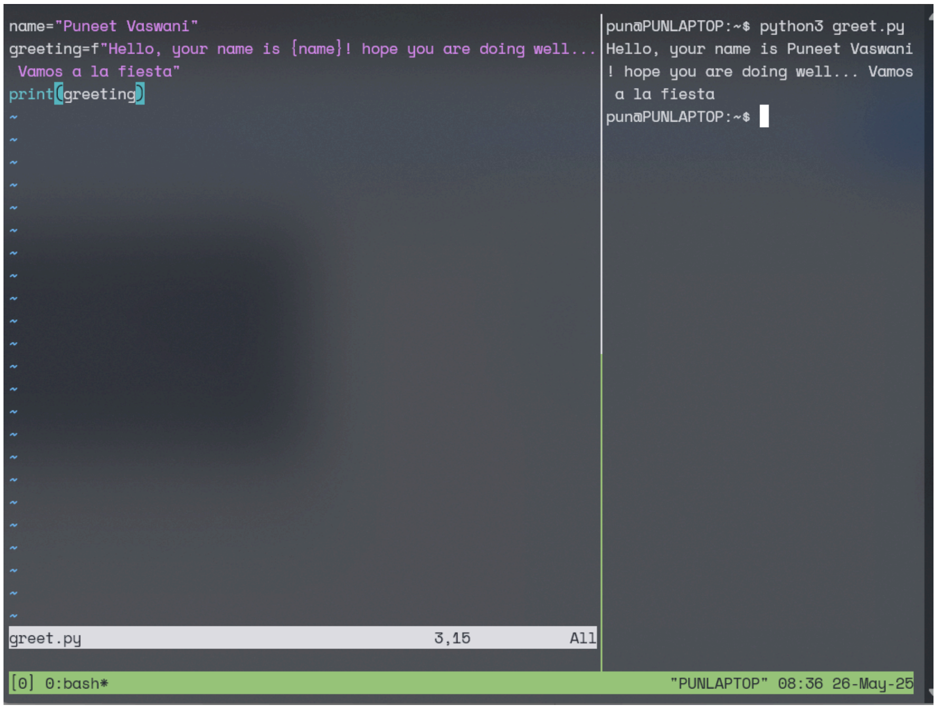
2. Install a code editor of your choice.
I am using vim as my code editor



```
pun@PUNLAPTOP:~$ nvim -v
NVIM v0.6.1
Build type: Release
LuaJIT 2.1.0-beta3
Compiled by team+vim@tracker.debian.org

Features: +acl +iconv +tui
See ":help feature-compile"

   system vimrc file: "$VIM/sysinit.vim"
  fall-back for $VIM: "/usr/share/nvim"

Run :checkhealth for more info
```

**For the following tasks i will be using *tmux* to show both the *nvim* and terminal at the same time**

3. Write a simple Python script that prints your name and a greeting.

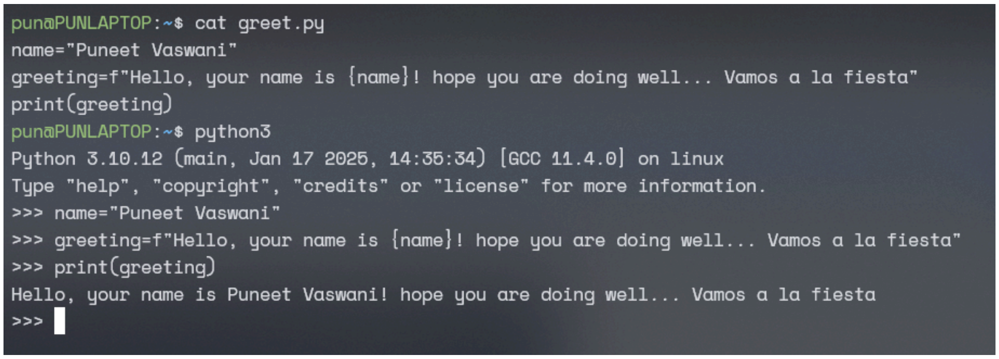


```
name="Puneet Vaswani"
greeting=f"Hello, your name is {name}! hope you are doing well...
 Vamos a la fiesta"
print(greeting)
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
greet.py                                          3,15          All
```

```
pun@PUNLAPTOP:~$ python3 greet.py
Hello, your name is Puneet Vaswani
! hope you are doing well... Vamos
 a la fiesta
pun@PUNLAPTOP:~$
```

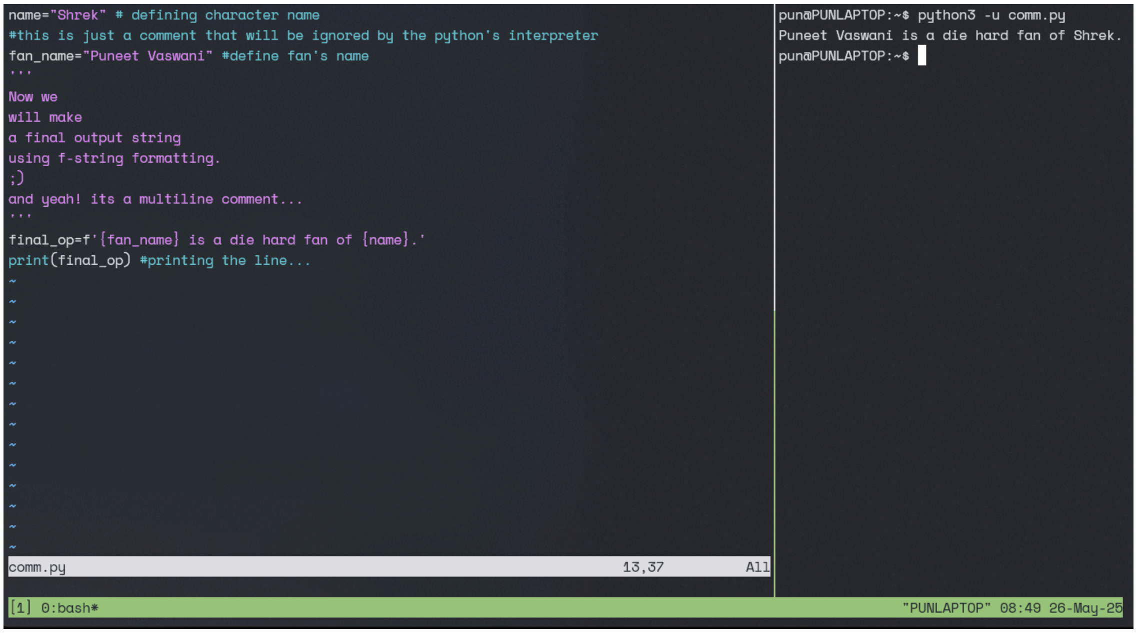
```
[0] 0:bash*                                  "PUNLAPTOP" 08:36 26-May-25
```

4. Experiment with running Python code directly in the interpreter.



```
pun@PUNLAPTOP:~$ cat greet.py
name="Puneet Vaswani"
greeting=f"Hello, your name is {name}! hope you are doing well... Vamos a la fiesta"
print(greeting)
pun@PUNLAPTOP:~$ python3
Python 3.10.12 (main, Jan 17 2025, 14:35:34) [GCC 11.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> name="Puneet Vaswani"
>>> greeting=f"Hello, your name is {name}! hope you are doing well... Vamos a la fiesta"
>>> print(greeting)
Hello, your name is Puneet Vaswani! hope you are doing well... Vamos a la fiesta
>>>
```

5. Write a script with comments explaining different parts of the code.

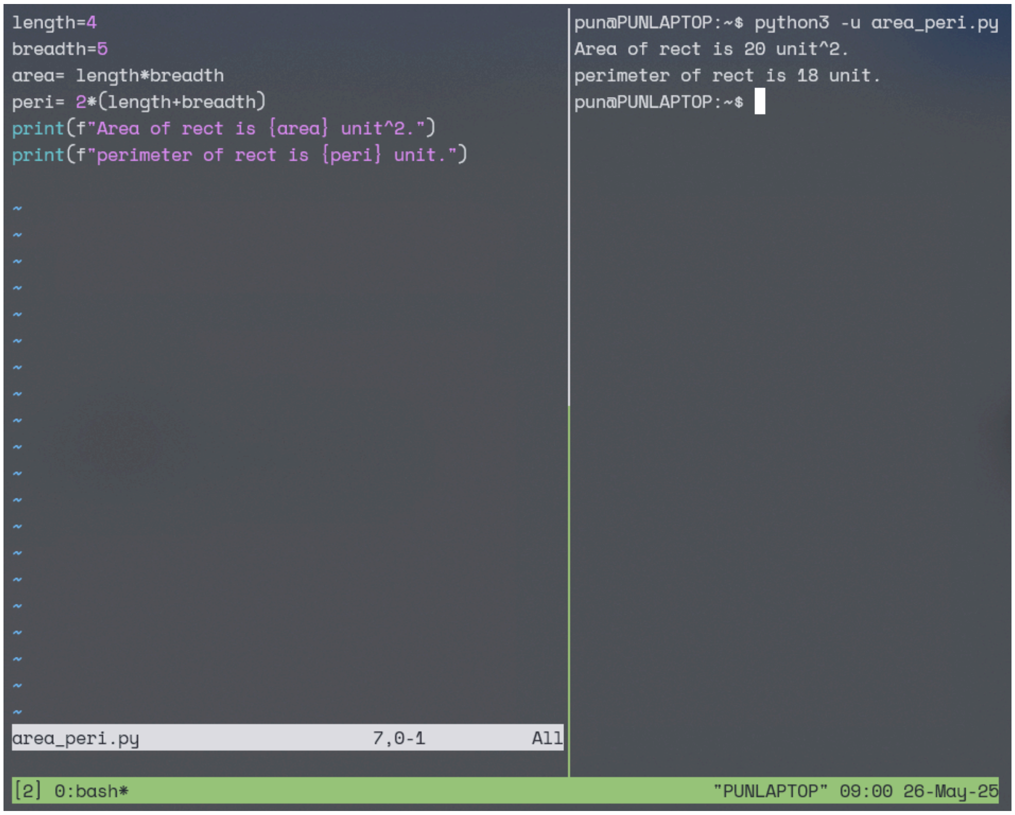


```python
name="Shrek" # defining character name
#this is just a comment that will be ignored by the python's interpreter
fan_name="Puneet Vaswani" #define fan's name
'''
Now we
will make
a final output string
using f-string formatting.
;)
and yeah! its a multiline comment...
'''
final_op=f'{fan_name} is a die hard fan of {name}.'
print(final_op) #printing the line...
```

```
pun@PUNLAPTOP:~$ python3 -u comm.py
Puneet Vaswani is a die hard fan of Shrek.
pun@PUNLAPTOP:~$
```

# TOPIC-2

1. Write a script to calculate the area and perimeter of a rectangle using variables.

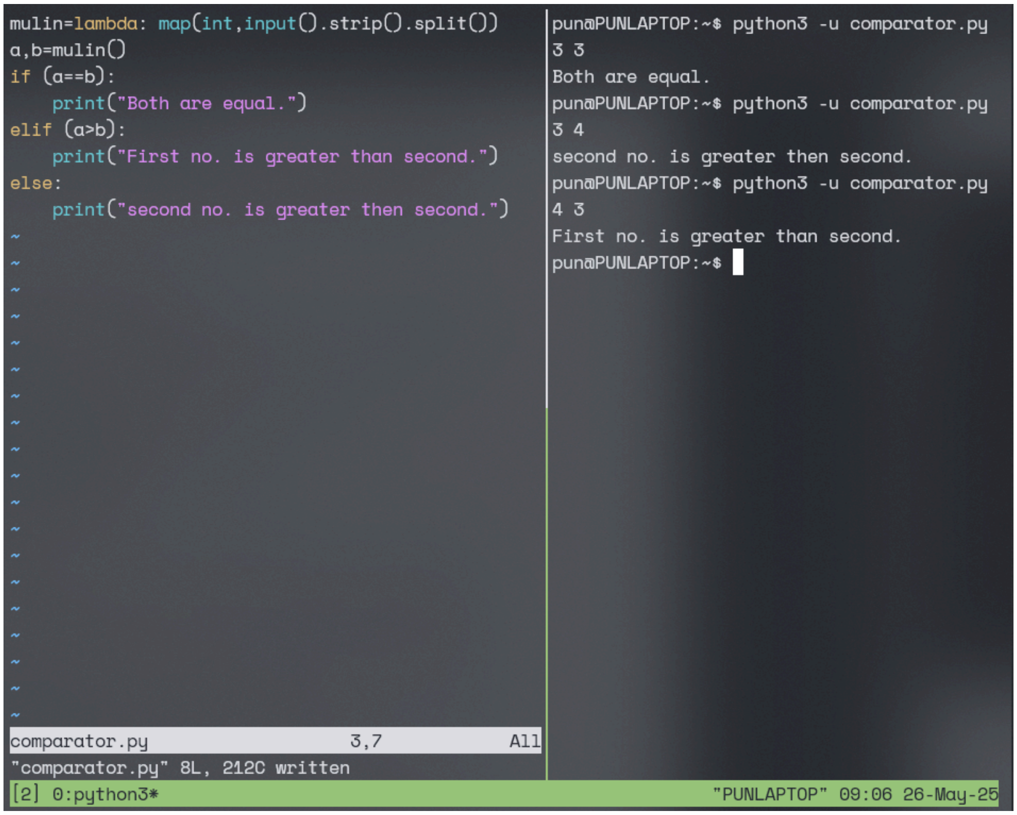


```
length=4
breadth=5
area= length*breadth
peri= 2*(length+breadth)
print(f"Area of rect is {area} unit^2.")
print(f"perimeter of rect is {peri} unit.")

~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
area_peri.py                         7,0-1          All
```

```
puna@PUNLAPTOP:~$ python3 -u area_peri.py
Area of rect is 20 unit^2.
perimeter of rect is 18 unit.
puna@PUNLAPTOP:~$
```

```
[2] 0:bash*                                                              "PUNLAPTOP" 09:00 26-May-25
```

2. Write a script that takes two numbers as input and prints whether the first number is greater than, less than, or equal to the second number.

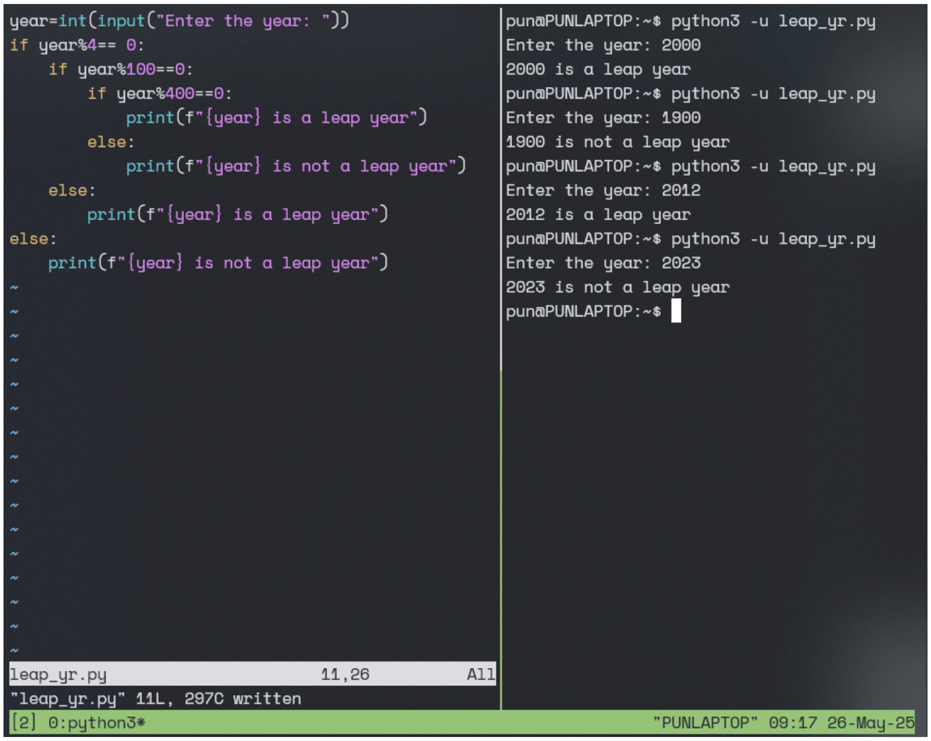


```
mulin=lambda: map(int,input().strip().split())
a,b=mulin()
if (a==b):
    print("Both are equal.")
elif (a>b):
    print("First no. is greater than second.")
else:
    print("second no. is greater then second.")
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
comparator.py                       3,7           All
"comparator.py" 8L, 212C written
```

```
puna@PUNLAPTOP:~$ python3 -u comparator.py
3 3
Both are equal.
puna@PUNLAPTOP:~$ python3 -u comparator.py
3 4
second no. is greater then second.
puna@PUNLAPTOP:~$ python3 -u comparator.py
4 3
First no. is greater than second.
puna@PUNLAPTOP:~$
```

```
[2] 0:python3*                                                           "PUNLAPTOP" 09:06 26-May-25
```

3. Write a script that checks if a given year is a leap year (divisible by 4, but not by 100 unless also divisible by 400).

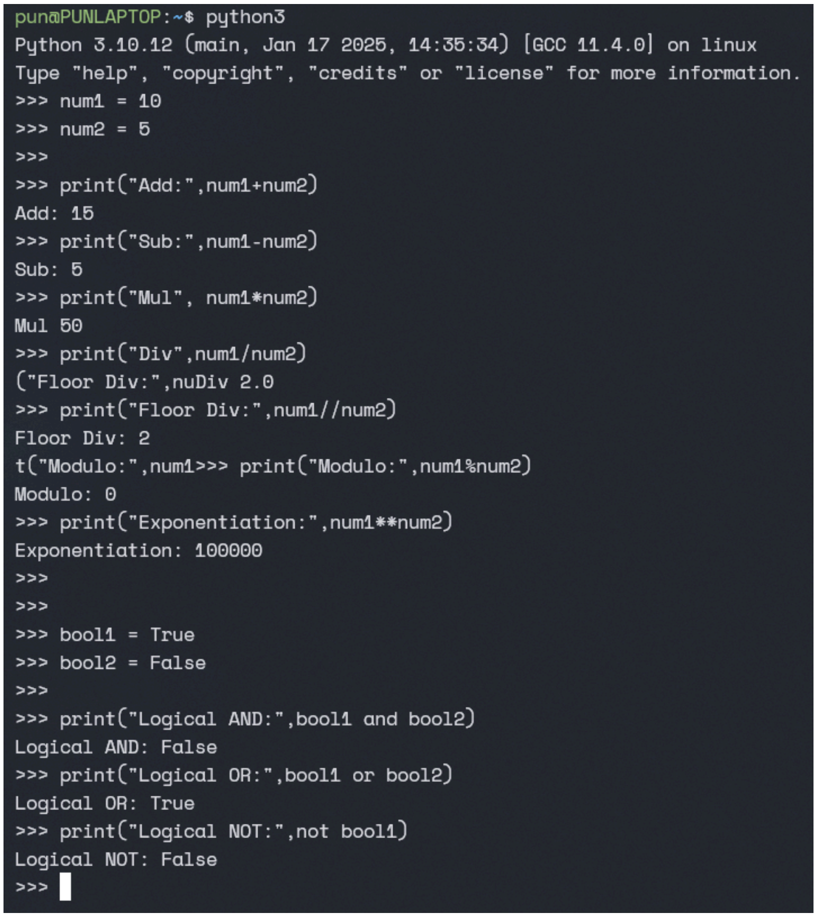


```
year=int(input("Enter the year: "))
if year%4== 0:
    if year%100==0:
        if year%400==0:
            print(f"{year} is a leap year")
        else:
            print(f"{year} is not a leap year")
    else:
        print(f"{year} is a leap year")
else:
    print(f"{year} is not a leap year")
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
leap_yr.py                                   11,26          All
"leap_yr.py" 11L, 297C written
```

```
puna@PUNLAPTOP:~$ python3 -u leap_yr.py
Enter the year: 2000
2000 is a leap year
puna@PUNLAPTOP:~$ python3 -u leap_yr.py
Enter the year: 1900
1900 is not a leap year
puna@PUNLAPTOP:~$ python3 -u leap_yr.py
Enter the year: 2012
2012 is a leap year
puna@PUNLAPTOP:~$ python3 -u leap_yr.py
Enter the year: 2023
2023 is not a leap year
puna@PUNLAPTOP:~$
```

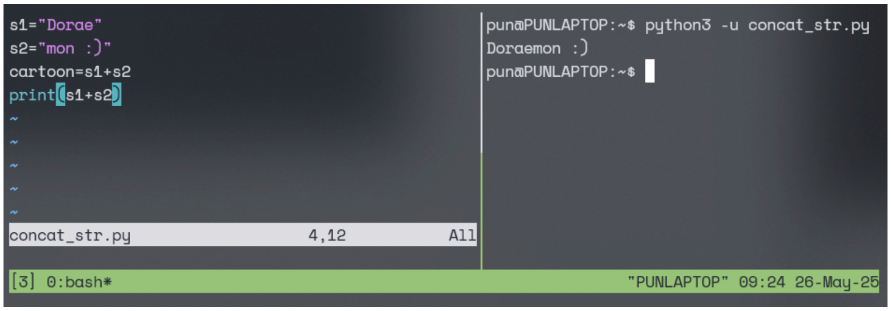
```
[2] 0:python3*                                                      "PUNLAPTOP" 09:17 26-May-25
```

4. Experiment with different arithmetic and logical operators in the interpreter.



```
puna@PUNLAPTOP:~$ python3
Python 3.10.12 (main, Jan 17 2025, 14:35:34) [GCC 11.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> num1 = 10
>>> num2 = 5
>>>
>>> print("Add:",num1+num2)
Add: 15
>>> print("Sub:",num1-num2)
Sub: 5
>>> print("Mul", num1*num2)
Mul 50
>>> print("Div",num1/num2)
("Floor Div:",nuDiv 2.0
>>> print("Floor Div:",num1//num2)
Floor Div: 2
t("Modulo:",num1>>> print("Modulo:",num1%num2)
Modulo: 0
>>> print("Exponentiation:",num1**num2)
Exponentiation: 100000
>>>
>>>
>>> bool1 = True
>>> bool2 = False
>>>
>>> print("Logical AND:",bool1 and bool2)
Logical AND: False
>>> print("Logical OR:",bool1 or bool2)
Logical OR: True
>>> print("Logical NOT:",not bool1)
Logical NOT: False
>>>
```

5. Write a script that concatenates two strings and prints the result.



```
s1="Dorae"
s2="mon :)"
cartoon=s1+s2
print(s1+s2)
~
~
~
~
~
concat_str.py                   4,12          All
```

```
pun@PUNLAPTOP:~$ python3 -u concat_str.py
Doraemon :)
pun@PUNLAPTOP:~$
```

```
[3] 0:bash*                     "PUNLAPTOP" 09:24 26-May-25
```