

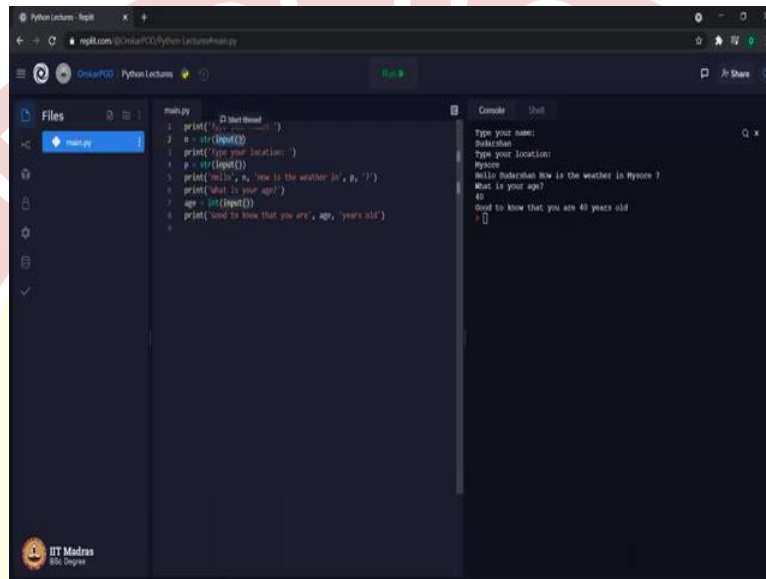


# IIT Madras

ONLINE DEGREE

**Programming in Python**  
**Professor Sudarshan Iyengar**  
**Department of Computer Science and Engineering**  
**Indian Institute of Technology Ropar**  
**Omkar Joshi**  
**Course Instructor**  
**Indian Institute of Technology Madras Online Degree Program**  
**Variables and Literals**

(Refer Slide Time: 0:16)



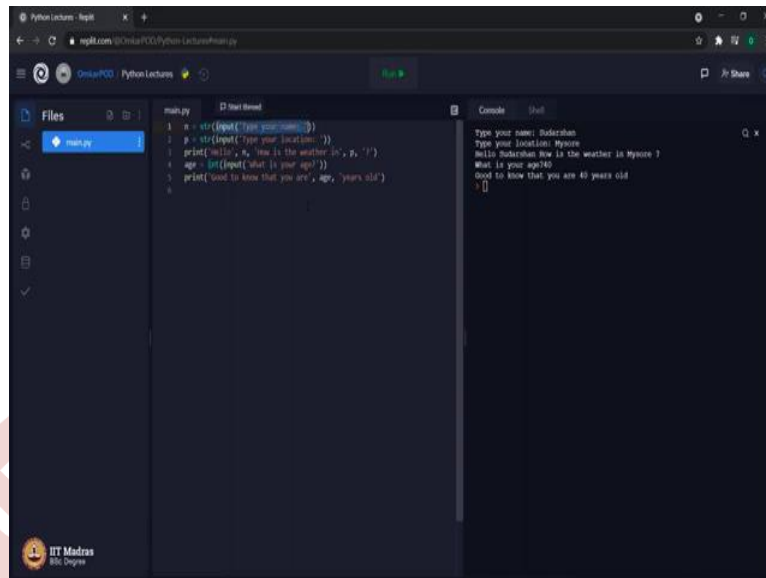
```
main.py (Python 3.8.1)
1 print('Hello World!')
2 n = input()
3 print('Type your location: ')
4 p = str(input())
5 print('Hello, ' + n, 'how is the weather in ' + p, '?')
6 print('What is your age?')
7 age = int(input())
8 print('Good to know that you are', age, 'years old')
9
```

```
Console (Python 3.8.1)
Type your name:
Sudarshan
Type your location:
Mysore
Hello Sudarshan how is the weather in Mysore ?
What is your age?
40
Good to know that you are 40 years old
```

Hello Python students. In this lecture we will see variables and literals. In order to demonstrate these concepts, we will use a Python code which is familiar to you. This is a similar code which we saw in the previous lecture. Type your name, Sudarshan, type your location, Mysore, then we will print message, ‘Hello Sudarshan, how is the weather in Mysore?’ Then the next question is what is your age. Let us say 40. Good to know that you are 40 years old.

Before moving to variables and literals let us see is it necessary to have print and input as two separate commands or can we merge them together? To answer that question first we have to understand what both these commands are doing. A print command is only printing a message, whereas this particular input command is taking the input from the user and storing it in n. This printing a message and taking a input can be combined together.

(Refer Slide Time: 1:33)

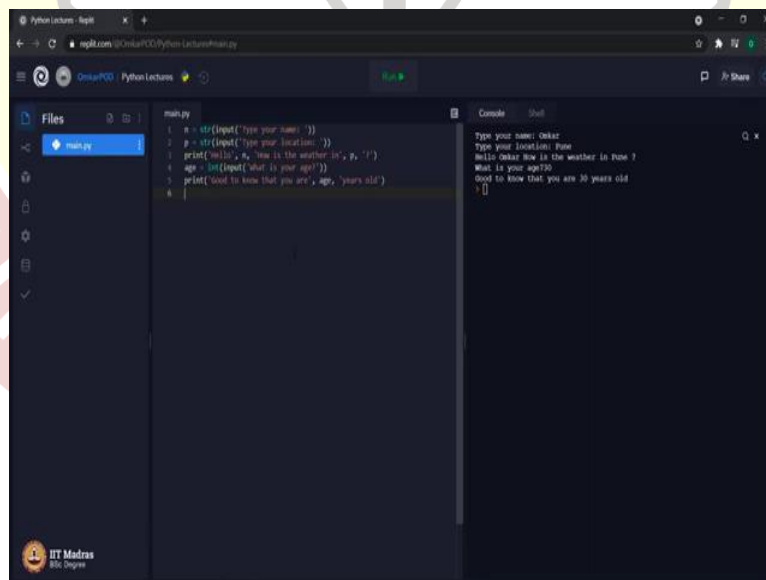


```
main.py | Shell | Console | Shell
1 n = str(input("Type your name: "))
2 p = str(input("Type your location: "))
3 print('Hello', n, 'How is the weather in', p, '?')
4 age = int(input("What is your age? "))
5 print('Good to know that you are', age, 'years old')
6

Type your name: Sudarshan
Type your location: Mysore
Hello Sudarshan How is the weather in Mysore ?
What is your age? 40
Good to know that you are 40 years old
```

The same message we can place inside the brackets of input command and we can remove this print. Let us execute, type your name, Sudarshan, type your location, Mysore, 'Hello, Sudarshan, how is the weather in Mysore. What is your age? You say 40. Good to know that you are 40 years old. Still we are getting the same output. This is how we can merge print and input together where we are displaying the message along with the input statement itself.

(Refer Slide Time: 2:29)



```
main.py | Shell | Console | Shell
1 n = str(input("Type your name: "))
2 p = str(input("Type your location: "))
3 print('Hello', n, 'How is the weather in', p, '?')
4 age = int(input("What is your age? "))
5 print('Good to know that you are', age, 'years old')
6

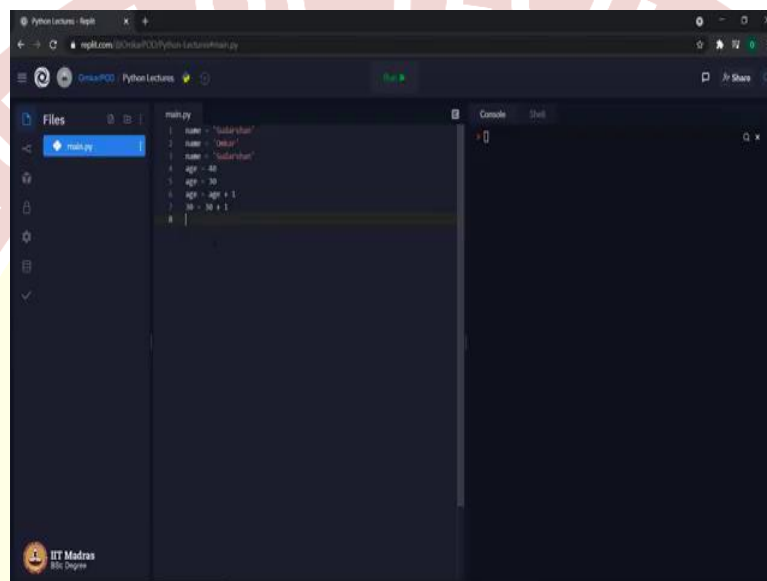
Type your name: Omkar
Type your location: Pune
Hello Omkar How is the weather in Pune ?
What is your age? 30
Good to know that you are 30 years old
```

Now the next question is – is it necessary to have the name as Sudarshan, location as Mysore and age as 40 or can we change the values. Let us try it. What is your name? Omkar. Type your location, Pune. Hello, Omkar, how is the weather in Pune? That works. Let us see whether next statement works. What is your age? Let us say 30. Good to know that you are

30 years old. Which means the same program executes perfectly with same messages even though the values are changed.

Now that brings the next question – How that happens? There is something called as variable, for example, this n or this p or this age, all these three are variables; which means their values can be changed, they are nothing but a container where we can store different values. Let us try to understand it with a different example.

(Refer Slide Time: 3:48)



The screenshot shows a Python IDE with a file named 'main.py'. The code in the file is as follows:

```
1 name = "Sudarshan"
2 name = "Omkar"
3 name = "Sudarshan"
4 age = 40
5 age = 30
6 age = age + 1
7 30 = 30 + 1
8
```

The IDE interface includes a 'Files' panel on the left, a 'Console' panel on the right, and a 'Run' button at the top right. The background features a large, semi-transparent watermark of the IIT Madras logo.

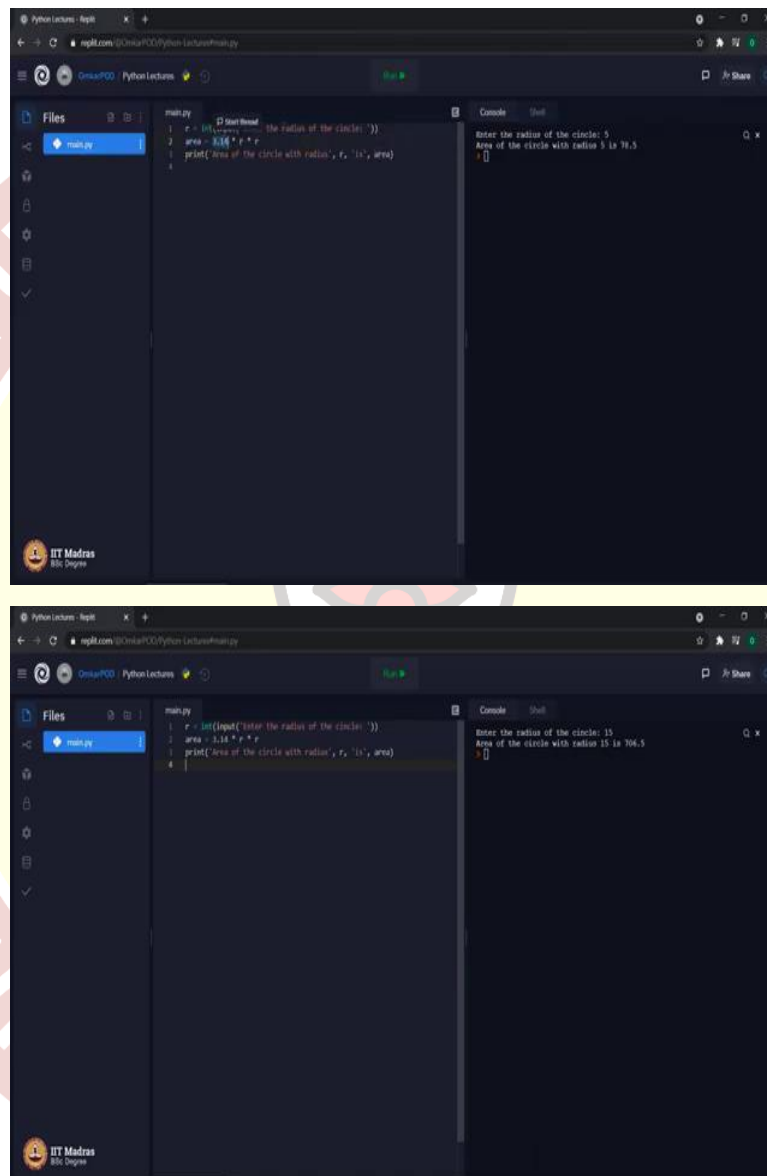
Let us say name is a variable. Now as per first program the name was Sudarshan, but now as I have mentioned name is just a variable, we can change the value which is stored in variable name from Sudarshan to Omkar. The name time we can also change it back to Sudarshan, which means variable we have used over here can store a different value as we go on. Similarly, age can be 40, it can be changed to 30 and so on.

So, in this case the name or the age are referred as variables whereas these names Sudarshan, Omkar or the values 40, 30 are referred as literals. Literals are the actual values which are stored inside a variable. Variable can store different literal values and they can be modified as per the requirement. The same age can be modified like this, age is equal to age plus 1. This particular thing is possible but if we try to execute something like this, then it will not work.

Because as per this particular line number 6, what we are saying, take the latest value of age which is 30, add 1 into it and store the updated values in variable age. Which means at this particular point the value is 30, then we did 30 plus 1, and now the updated value is 31. But if you come here we cannot say 30 is equal to 30 plus 1.

Hence, you can conclude the literals can be used only on the right hand side of the equal to sign, whereas variable can be used on either sides of the symbol equal to. This particular difference between variable and literal leads us to a new question – when to use variables and when to use literals?

(Refer Slide Time: 6:39)



The image displays two screenshots of a Python IDE (replit.com) showing a program to calculate the area of a circle. The code is as follows:

```
1 r = int(input("Enter the radius of the circle: "))
2 area = 3.14 * r * r
3 print("Area of the circle with radius", r, "is", area)
```

**Top Screenshot:** The input is 5. The output is: Enter the radius of the circle: 5, Area of the circle with radius 5 is 78.5.

**Bottom Screenshot:** The input is 15. The output is: Enter the radius of the circle: 15, Area of the circle with radius 15 is 706.5.

Let us write one Python program which will demonstrate the use of variables and literals. You observe and tell me what happens in this particular code. Let us see what we have done over here, r is equal to int of input of enter the radius of the circle, area is equal to 3.14 multiplied by r multiplied by r. Print area of the circle with radius r is area. First let us execute this particular program, then we will try to answer our previous question.

Enter the radius of the circle, let us say 5, area of the circle with radius 5 is 78.5, we are getting the output but still the question is when to use variable and when to use literal. The answer of this question is in this second line over here. The answer is we use  $r$  as a variable, area as a variable but the value of  $\pi$  as literal, because value of  $\pi$  will never change, whereas radius of a circle might change, and with respect to that area will also change, which means we can execute this program again.

And instead of 5 now we can enter 15, which will give us a new area value which is 706.5, irrespective of the radius of the area the value of  $\pi$  is always 3.14. We will use variables only when there is a possibility that the value which is being stored may change and when we are sure that the value is not going to change we will use literal.

Similarly, in previous program we used name as a variable, age as variable because those two values may change from person to person, but over here value for  $\pi$  will always remain same. Thank you for watching this lecture. Happy learning!

