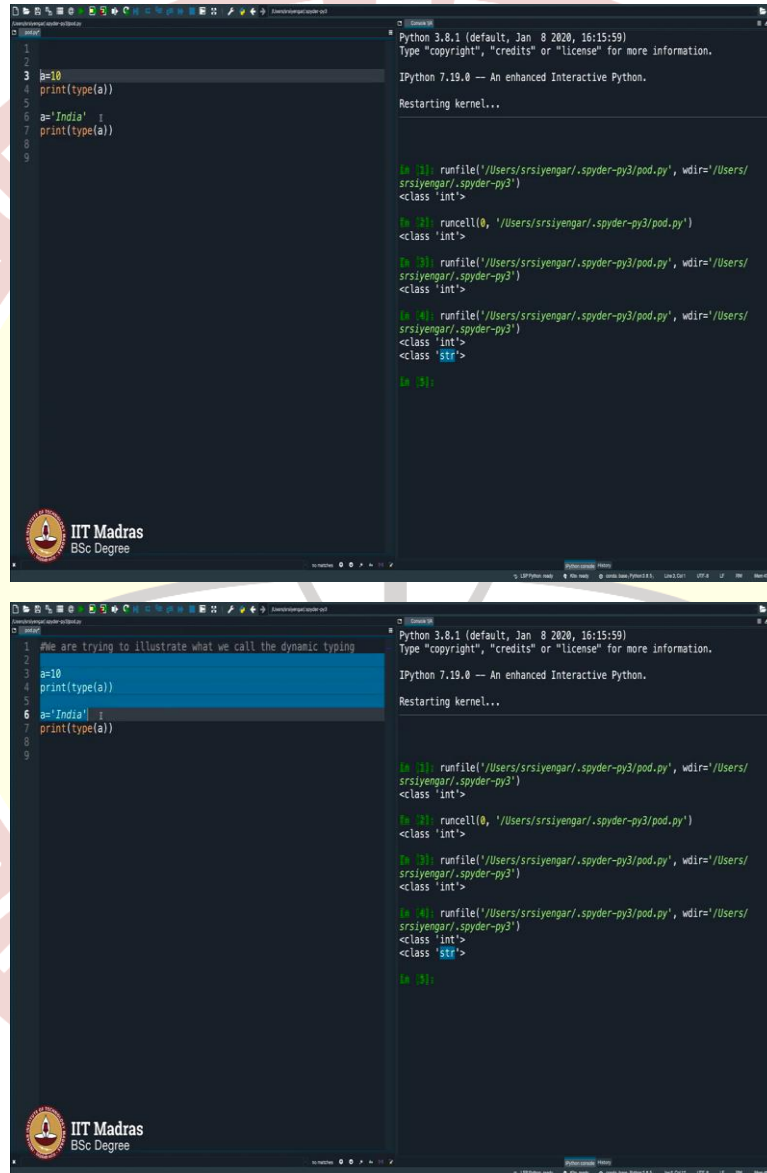




**IIT Madras**  
ONLINE DEGREE

**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 Revisited: Dynamic Typing**

(Refer Slide Time: 00:16)



The image displays two screenshots of a Jupyter Notebook interface. The left screenshot shows a code cell with the following Python code:

```
1  
2  
3 a=10  
4 print(type(a))  
5  
6 a='India'  
7 print(type(a))  
8  
9
```

The right screenshot shows the output of the code cell, which includes the Python version (3.8.1), the IPython version (7.19.0), and the output of the code execution:

```
Python 3.8.1 (default, Jan 8 2020, 16:15:59)  
Type "copyright", "credits" or "license()" for more information.  
IPython 7.19.0 -- An enhanced Interactive Python.  
Restarting kernel...  
  
In [1]: runfile('/Users/srsiyengar/.spyder-py3/pod.py', wdir='/Users/srsiyengar/.spyder-py3')  
<class 'int'>  
  
In [2]: runfile('/Users/srsiyengar/.spyder-py3/pod.py', wdir='/Users/srsiyengar/.spyder-py3')  
<class 'int'>  
  
In [3]: runfile('/Users/srsiyengar/.spyder-py3/pod.py', wdir='/Users/srsiyengar/.spyder-py3')  
<class 'int'>  
  
In [4]: runfile('/Users/srsiyengar/.spyder-py3/pod.py', wdir='/Users/srsiyengar/.spyder-py3')  
<class 'int'>  
<class 'str'>  
  
In [5]:
```

The background of the image features a large, faint watermark of the Indian Institute of Technology Madras logo.

Let us say, I declare the variable A as 10 and then print its type; if you remember, type prints, what kind of variable A is. It is saying integer. So I need to tell you people that what we are discussing right now is not very important for a beginner. But then for completeness sake, we need to tell you these things. So do not worry much in case you do not understand this really well right now, but eventually you will, I mean as we reach eighth week, ninth week, you

may want to go back and then look at what are all we discussed so far and then try to see if things make sense to you or not.

So, this is slightly advanced, but do not worry if you do not understand, but make your attempts to understand what I am saying. So I say A equals 10 and I print the type of A and now I say A equals India and then print the type of a. So what is your guess? Initially, it was int, now it became str. Remember, we were discussing about the jar in your kitchen.

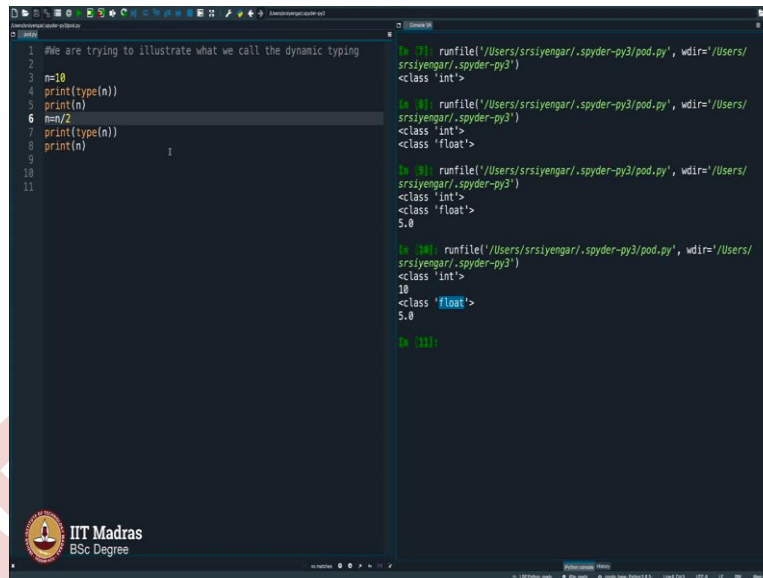
You can use the jar to store rice or maybe even water for that matter, but then we specialize in our, in the containers that we use for storage, we do not use the same container that is used to store rice, we do not use that to store milk. Milk container looks different, is made of a different material and things like that.

So similarly, a computer tries to store an integer type using a different kind of memory location and string in a different kind of memory location. Moment you say equals 10, it automatically creates, what does it do? It creates a particular data type. By data type, I mean the memory location where it, where the value of A is stored and the moment you say A equals India, of course, it loses the value of 10 that it was initially assigned to, but then it becomes of the type string.

This is this goes by the name dynamic typing, you all know what is commenting right now. I use comment; the first line will be dedicated to write what program we are writing. We are trying to illustrate what we call the dynamic typing. The word typing here does not mean the typing of the keyboard. The word typing here is the noun form of the word type. Dynamic stands for things can change.

In Python, the moment you declare an integer, it does not stay as an integer; you can change A to whatever you want. It provides flexibility for you to use a variable for something as a, as an integer and then make it a string later on. The type of a variable is dynamic; you can change it as per your wish. So let me remove this and then illustrate this with a better example.

(Refer Slide Time: 03:32)



```
1 #We are trying to illustrate what we call the dynamic typing
2
3 n=10
4 print(type(n))
5 print(n)
6 n=n/2
7 print(type(n))
8 print(n)
9
10
11
```

```
In [7]: runfile('/Users/srsiyengar/.spyder-py3/pod.py', wdir='/Users/srsiyengar/.spyder-py3')
<class 'int'>
10

In [8]: runfile('/Users/srsiyengar/.spyder-py3/pod.py', wdir='/Users/srsiyengar/.spyder-py3')
<class 'int'>
<class 'float'>
5.0

In [9]: runfile('/Users/srsiyengar/.spyder-py3/pod.py', wdir='/Users/srsiyengar/.spyder-py3')
<class 'int'>
10
<class 'float'>
5.0

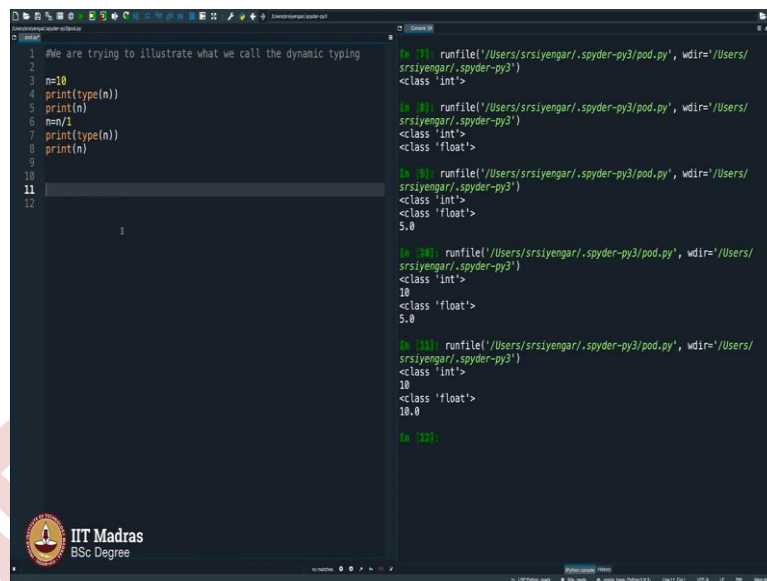
In [10]:
```

Let me take a number  $n$  equals 10 and then say  $n$  equals  $n$  by 2. Carefully observe what happens here. I print the type of  $n$  and as you would expect, it is int only. Let me clear the screen here. Clear clears the screen. So let me print. It shows you it is of type int.  $n$  is of type int is what it is saying.

Now, let me print another print statement here and display the type of  $n$ . Now after I divide by 2, let us see what it does. It says int, but the moment you divide, it becomes float. So let me try printing and now and says it is 5.0 now, but then here, if I were to print  $n$  it is, it says 10. It does not say 10.0.

But the moment you divide by 2 it, the moment division happens, it realizes that oh now the integer might become a non-integer; it can become a rational number or some real number. So, let me be careful. Let me allocate, let me deallocate that integer. Let me try to allocate something that that is easier for me to contain this object; it could be a real number. So I will use float, float stands for real numbers and in the language of computers, int is integers 1, 2, 3, 4, 5; float is 4.3, 7.61 all that comes under float. So let us see what happens?

(Refer Slide Time: 5:18)



```
1 #We are trying to illustrate what we call the dynamic typing
2
3 n=10
4 print(type(n))
5 print(n)
6 n=n/1
7 print(type(n))
8 print(n)
9
10
11
12
```

```
In [7]: runfile('/Users/srsiyengar/.spyder-py3/pod.py', wdir='/Users/
srsiyengar/.spyder-py3')
<class 'int'>
10

In [8]: runfile('/Users/srsiyengar/.spyder-py3/pod.py', wdir='/Users/
srsiyengar/.spyder-py3')
<class 'int'>
<class 'float'>
5.0

In [9]: runfile('/Users/srsiyengar/.spyder-py3/pod.py', wdir='/Users/
srsiyengar/.spyder-py3')
<class 'int'>
<class 'float'>
5.0

In [10]: runfile('/Users/srsiyengar/.spyder-py3/pod.py', wdir='/Users/
srsiyengar/.spyder-py3')
<class 'int'>
10
<class 'float'>
5.0

In [11]: runfile('/Users/srsiyengar/.spyder-py3/pod.py', wdir='/Users/
srsiyengar/.spyder-py3')
<class 'int'>
10
<class 'float'>
10.0

In [12]:
```

What does Python do even if we divide by 1, even if we divide the moment you use the division operator, Python makes the number, makes the variable float from it being int. So it changes irrespective of you assigning explicitly you did not do  $n$  equals a string, you just divided  $n$  by 1, still Python was intelligent enough to now change the type to float thinking that oh, this chap is doing some division, he may continue to do more division. Let me try making it float to be on the safer side.

This goes with the name dynamic typing, again as I repeat not very important at this stage. Just know it for completeness sake; we are trying to teach you this. But in case you do not understand you know what to do you go ahead with other weeks, confidently but and then come back and then try to look at these things, this will make more sense once you start coding extensively.