INDIAN INSTITUTE OF TECHNOLOGY MADRAS

सिद्धिर्भवति कर्मजा

IIT Madras

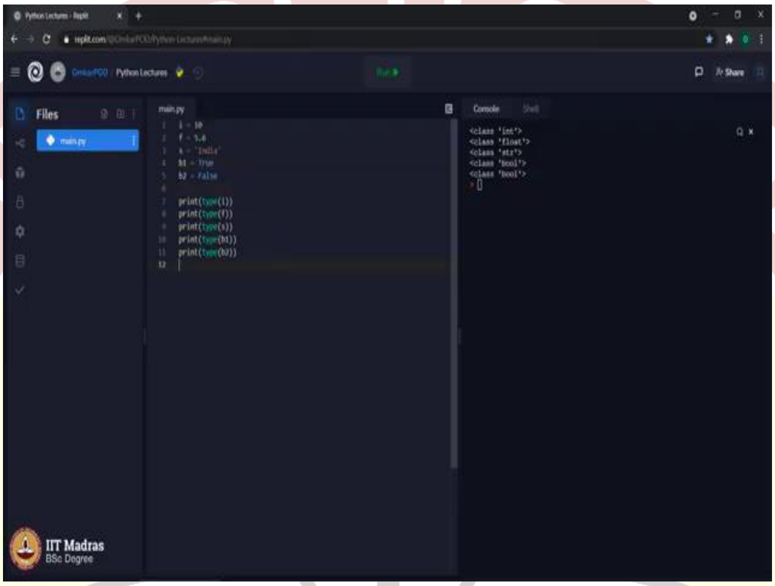
ONLINE DEGREE

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

Data Types 2

(Refer Slide Time: 0:16)



```
i = 10
f = 5.6
s = 'India'
b1 = True
b2 = False

print(type(i))
print(type(f))
print(type(s))
print(type(b1))
print(type(b2))
```

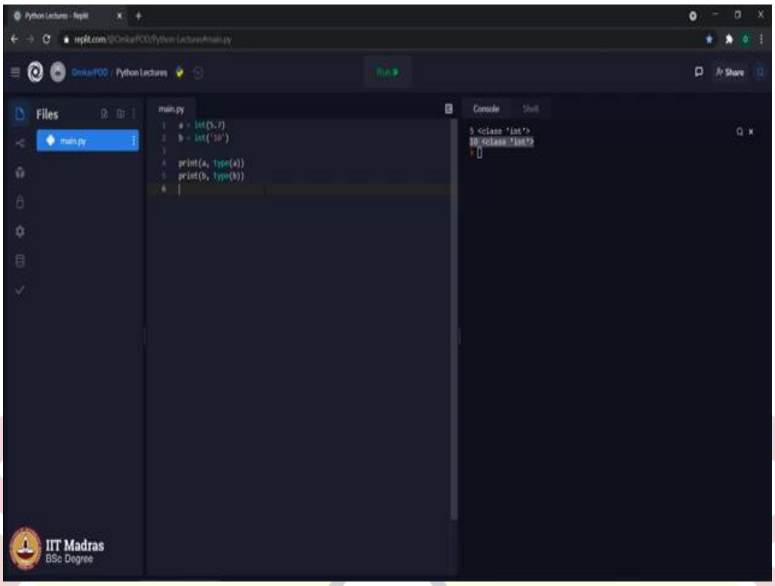
```
<class 'int'>
<class 'float'>
<class 'str'>
<class 'bool'>
<class 'bool'>
```

Hello Python students. In last lecture we saw something called as data type. Each data element has a specific data type, which represents which category of data that particular value belongs to. For example, 10 over here is of type integer, 5.6 is of float, whereas value India is of type string. A data type of any variable can be checked using a command called type. Let us print type of i, type of f, and type of s. As expected it prints int, float and string.

Now in this particular we will introduce one more data type called Boolean. Let us add one more variable B1. Now this particular data type has only 2 different values, one it is True, other is False. Let us say B1 is equal to True and B2 is equal to False. Let us print B1 and B2. Let us execute this particular program. As expected the data type for variables B1 and B2 is Boolean.

This is a different kind of data type where any variable created using this particular data type can hold only two different values, which is True or False. And notice carefully, the letter T in True and letter F in False has to be capital otherwise it will not be considered as Boolean. Now as we have studied four different data types let us try something different which is data conversion from one data type to other. First let us start with integers.

(Refer Slide Time: 2:13)



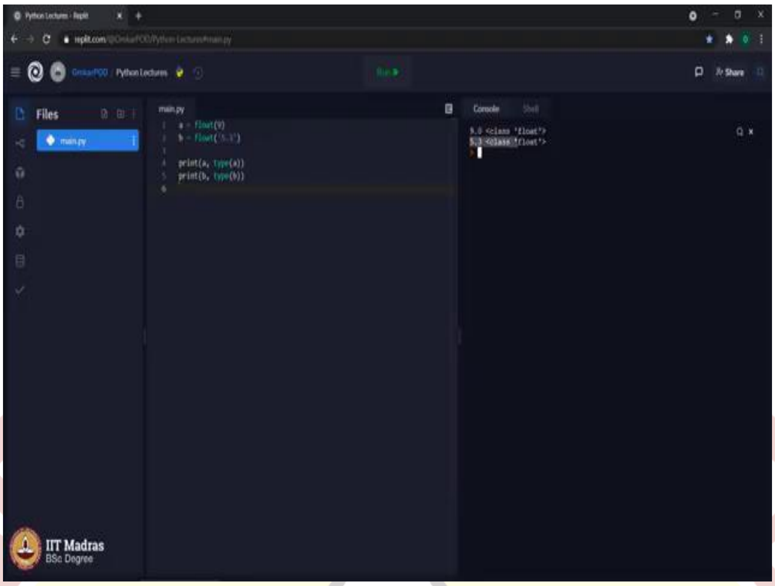
```
a = int(5.7)
b = int('10')

print(a, type(a))
print(b, type(b))
```

```
5 <class 'int'>
10 <class 'int'>
```

Let us look at this particular code, value 5.7 is of type float, but we are explicitly telling the computer to convert this 5.7 to integer and then store it in variable a. Similarly, over here, 10 is a integer but it is currently enclosed in single quotes, which means it is string and we are asking computer to convert this string into integer and then store it in variable b.

Computer will convert 5.7 into integer, string 10 into an integer and ultimately variables a and b should be integers. Let us print the values of a and b, also the type of a and type of b, and let us see how it works. As you can see value of variable a is 5 and type is integer, whereas value for variable b is 10 and type is integer. In the case of conversion from float to integer, computer will only take 5 and store it in a, whereas this 0.7 part will be ignored. This is how we can convert a float or a string into an integer.

(Refer Slide Time: 3:42)



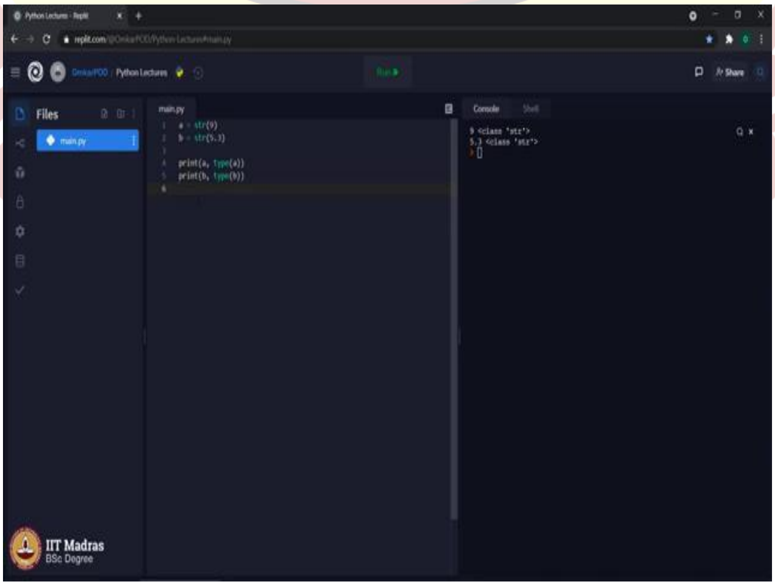
```
a = float(9)
b = float('5.3')

print(a, type(a))
print(b, type(b))
```

```
9.0 <class 'float'>
5.3 <class 'float'>
>
```

Now, equal to let us look at similar type of conversion from integer and string to float. Let us look at this particular float, where 9 is integer and 5.3 is a string, but we are telling the computer to convert both these values from integer to float and then from string to float. Let us execute the code.

As expected the original value 9 got converted to 9.0 and type if float, a string representation of 5.3 got converted to its float representation, which is again 5.3, but the type had changed to float. This is how we can convert values from integer or from string to its equivalent float representation.

(Refer Slide Time: 4:32)



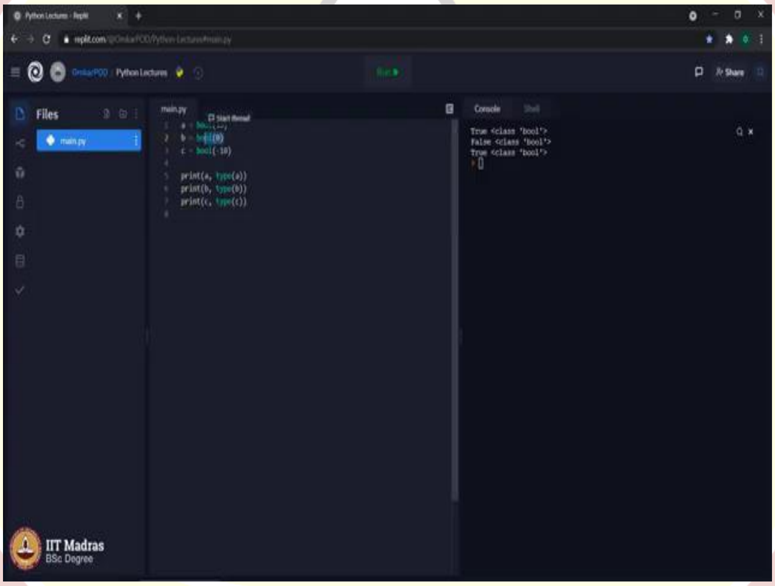
```
a = str(9)
b = str(5.3)

print(a, type(a))
print(b, type(b))
```

```
9 <class 'str'>
5.3 <class 'str'>
>
```

Now, let us see how to convert values of type integer and float to its equivalent string representation. Once again this 9 is a integer and 5.3 is a float but we are telling the computer to convert this 9 to a string, also 5.3 to a string. Let us execute the code. As expected we got the output as 9 and 5.3, but the type has been converted from integer to string and then from float to string.

This data type conversion is generally referred as type conversion or type casting as in conversion from one data type to other data type. If you have noticed so far we are trying convert values between integers, floats, and strings. But we have not discussed any type conversion related to Boolean data type. Let us see type conversion can be done from integers, float and strings to Boolean data type.

(Refer Slide Time: 5:42)



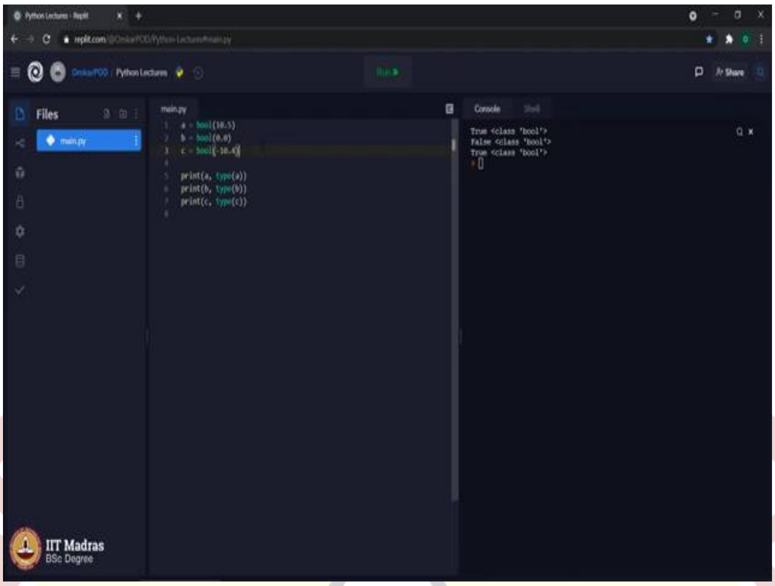
```
a = bool(10)
b = bool(0)
c = bool(-10)

print(a, type(a))
print(b, type(b))
print(c, type(c))
```

```
True <class 'bool'>
False <class 'bool'>
True <class 'bool'>
> 
```

Let us look at this particular code where we are trying to convert three different integer values to its equivalent Boolean values, Boolean of 10 and 0 minus 10. Let us first execute this code and then we will see how particular output is coming. Class or type of variables a, b, and c is Boolean but if you observe the values then you will notice there is some difference. It says True, False and True.

Value for variable b is coming out to be False, whereas values for variable a and c are True. This is happening because whenever computer converts an integer to a Boolean, all values except 0 are consider as True whereas 0 is the only value which will give us Boolean representation which is False. That is the reason 10 as well as minus 10 are getting converted to Boolean value True, whereas 0 is converted to Boolean value False.

(Refer Slide Time: 6:57)



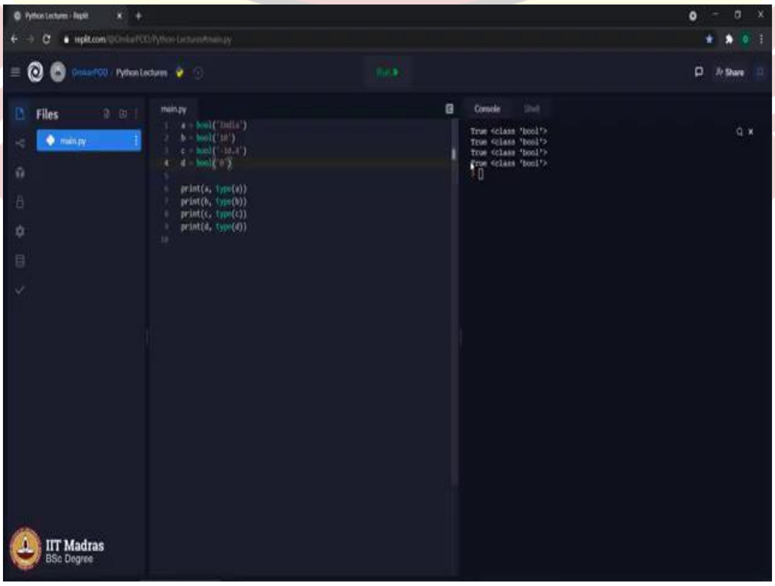
```
a = bool(10.5)
b = bool(0.0)
c = bool(-10.4)

print(a, type(a))
print(b, type(b))
print(c, type(c))
```

```
True <class 'bool'>
False <class 'bool'>
True <class 'bool'>
```

Now, let us try the similar thing with a floating point value. Now let us look at this particular code where we are converting values from float data type to Boolean data type. Let us execute, once again a data type has been converted from float to Boolean in all three cases whereas values for variables a, and c are True but for variable b it is still False, because 0.0 is nothing but 0 only.

And as we have mentioned earlier Boolean representation of 0 is always False. Other than that for all other positive as well as negative numbers irrespective of integer or float, we will get a Boolean representation as True.

(Refer Slide Time: 7:52)



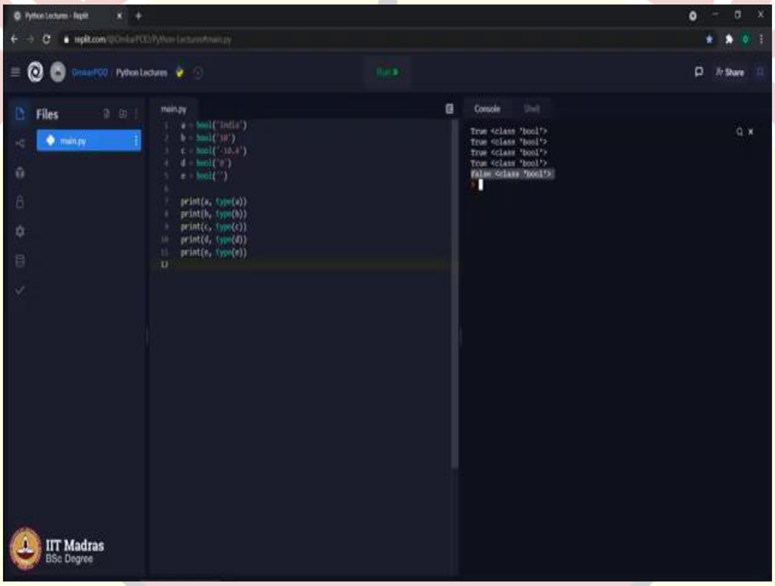
```
a = bool('India')
b = bool('10')
c = bool('-10.4')
d = bool('0')

print(a, type(a))
print(b, type(b))
print(c, type(c))
print(d, type(d))
```

```
True <class 'bool'>
True <class 'bool'>
True <class 'bool'>
True <class 'bool'>
```

Now, let us see what happens when we try to convert string to Boolean. Let us look at this particular code block. India, 10, minus 10.4 as well as 0 all are strings. We are trying to convert all these string values to its equivalent Boolean representation. Let us execute the code and see what kind of output we are getting.

All data types are Booleans and all values are True. Now even in the case of 0 it is given value True, because in this case 0 is neither an integer nor a float, here 0 is enclosed in this single quotes which makes it a string and string representation of Boolean is always True except for one condition. Let us see what that condition is.

(Refer Slide Time: 8:47)



```
a = bool('India')
b = bool('10')
c = bool('-10.4')
d = bool('0')
e = bool('')

print(a, type(a))
print(b, type(b))
print(c, type(c))
print(d, type(d))
print(e, type(e))
```

```
True <class 'bool'>
True <class 'bool'>
True <class 'bool'>
True <class 'bool'>
False <class 'bool'>
```

e is equal to Boolean of empty string, here we are simply giving single quotes and inside those single quotes we are not writing anything, which means it is an empty string and a Boolean conversion of empty string is False. As you can see over here in last line of the output, the class is Boolean but the value is False. All strings are converted to Boolean with value True except an empty string.

So, in order to summarize this particular lecture, we saw one more new data type called Boolean and also we studied type conversion within integer, float and string and at the end we converted all different types of integers, floats and strings to its equivalent Boolean representation. Thank you for watching this lecture. Happy learning!