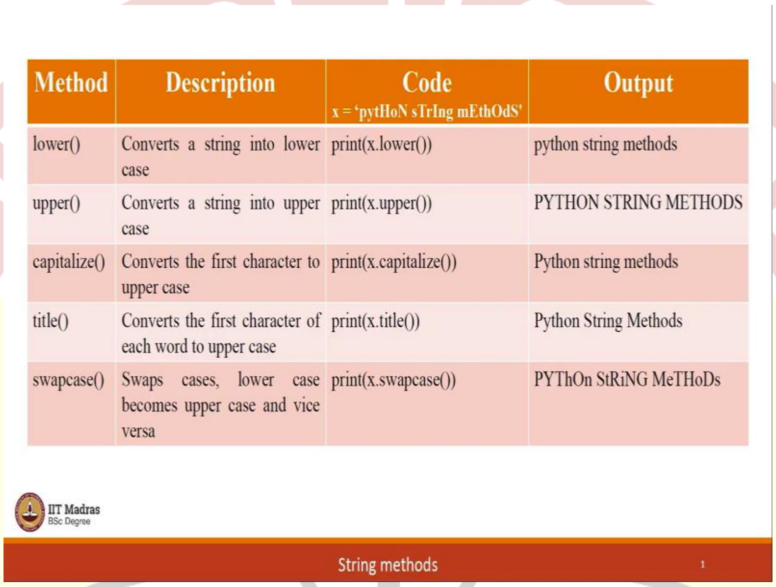
IIT Madras

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
String Methods

(Refer Slide Time: 00:16)



Method	Description	Code x = 'pytHoN sTrIng mEthOdS'	Output
lower()	Converts a string into lower case	print(x.lower())	python string methods
upper()	Converts a string into upper case	print(x.upper())	PYTHON STRING METHODS
capitalize()	Converts the first character to upper case	print(x.capitalize())	Python string methods
title()	Converts the first character of each word to upper case	print(x.title())	Python String Methods
swapcase()	Swaps cases, lower case becomes upper case and vice versa	print(x.swapcase())	PYThOn StRiNG MeTHoDs

IIT Madras
BSc Degree

String methods 1

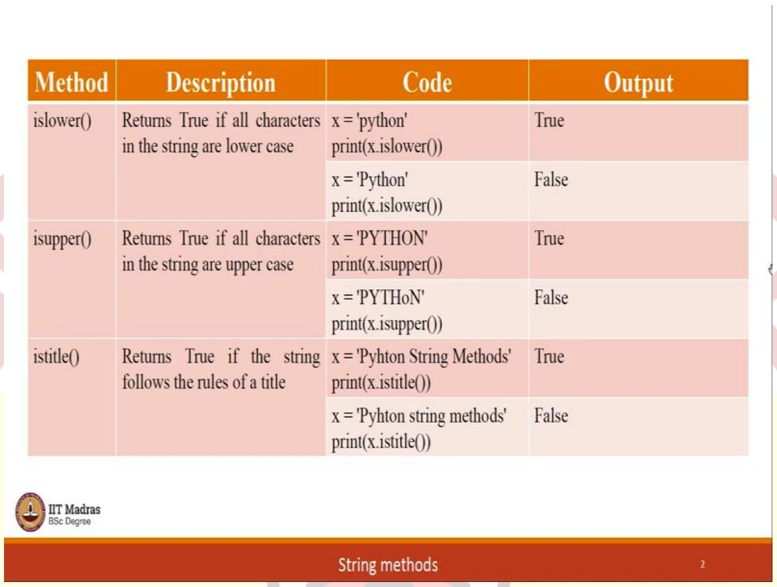
Hello Python students. In this lecture, we will see something called as string methods. We have discussed about strings multiple times in various lectures, we will extend that discussion to string methods. What are methods? Methods are nothing but functions or commands. At this point of time, you do not have to think too much about the word methods.

We will come to that in a later stage of this particular course. Right now, you can consider these methods as some kind of commands which we can execute on strings. Let us start with first set of string methods. Lower, it converts the every character in the input string into a lowercase character. Second, uppercase, this one converts every single character in the string into an uppercase character.

Next, capitalize; it converts only the first character of the string to a capital letter. Except that first letter, all remaining characters gets converted to lowercase characters. Title, it converts the first character of every single word in that particular string to an uppercase character, except that first letter of the word, all remaining letters are converted to lowercase characters.

The last swapcase, as the name suggests, it simply swaps the characters from uppercase to lowercase and from lowercase to uppercase. In the last two columns you can see a string example is given and all these string methods are executed using that specific example and its equivalent output is displayed in the last column. Let us move to next set of string methods.

(Refer Slide Time: 02:28)



Method	Description	Code	Output

islower()	Returns True if all characters in the string are lower case	x = 'python'
		print(x.islower())
			True

		x = 'Python'
		print(x.islower())
			False

isupper()	Returns True if all characters in the string are upper case	x = 'PYTHON'
		print(x.isupper())
			True

		x = 'PYTHoN'
		print(x.isupper())
			False

istitle()	Returns True if the string follows the rules of a title	x = 'Pyhton String Methods'
		print(x.istitle())
			True

		x = 'Pyhton string methods'
		print(x.istitle())
			False

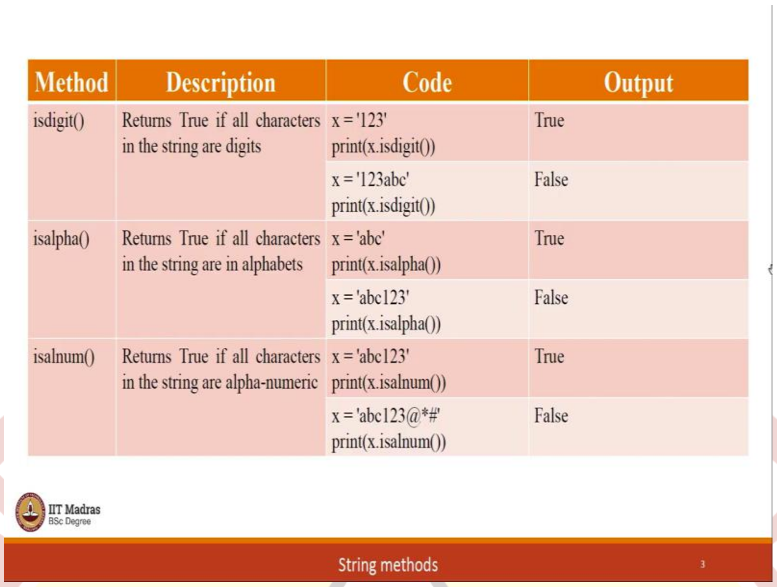
IIT Madras
BSc Degree

String methods 2

Next set of string methods are islower, isupper and iftitle. These three methods are very similar to what we saw in previous slide. With lower, upper and title methods, we were actually converting the input string to its equivalent form. But in case of these three methods, we are not actually converting any string we are simply checking whether the input string is in a lowercase form or not, whether all the characters in the input string are uppercase letters or not.

And the last method we will check whether the given input string follows all the rules of title or not. Once again, some sample codes along with their respective outputs are given. As you can notice, all the outputs are in a Boolean form as expected.

(Refer Slide Time: 03:31)



Method	Description	Code	Output

isdigit()	Returns True if all characters in the string are digits	x = '123'
		print(x.isdigit())
			True

		x = '123abc'
		print(x.isdigit())
			False

isalpha()	Returns True if all characters in the string are in alphabets	x = 'abc'
		print(x.isalpha())
			True

		x = 'abc123'
		print(x.isalpha())
			False

isalnum()	Returns True if all characters in the string are alpha-numeric	x = 'abc123'
		print(x.isalnum())
			True

		x = 'abc123@*#'
		print(x.isalnum())
			False

IIT Madras
BSc Degree

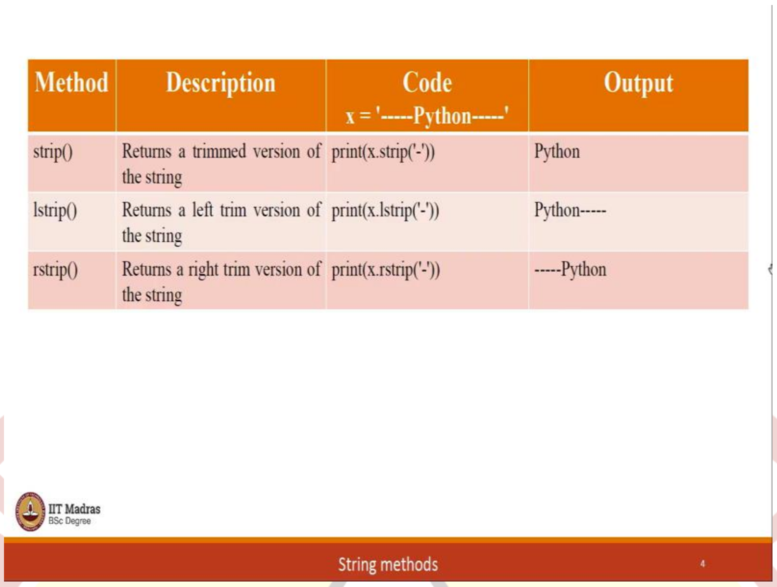
String methods 3

Next set of string methods are isdigit, isalpha, and isalnum. As the name suggests, isdigit checks whether all the characters in the string are digits or not. If that is the case, it will return true or else it will return false. isalpha checks, whether all the characters in the string are alphabets or not. Accordingly, it will return true or false.

isalnum is a combination of alphabets and numbers as in the digits. Therefore, this particular method will return true if all the characters in the string are alphanumeric characters, which means any combination of alphabets and numbers. On top of that, whenever we talk about alphabets, we consider alphabets in lowercase as well as uppercase.

Therefore, if you observe the last line in this particular table, you will observe that the output is false because the input string has some special characters like at the rate, star and hash, due to which the output we are getting is false because these three characters does not come under the set, which we call as alphanumeric characters.

(Refer Slide Time: 05:05)



Method	Description	Code x = '-----Python-----'	Output
strip()	Returns a trimmed version of the string	print(x.strip('-'))	Python
lstrip()	Returns a left trim version of the string	print(x.lstrip('-'))	Python-----
rstrip()	Returns a right trim version of the string	print(x.rstrip('-'))	-----Python

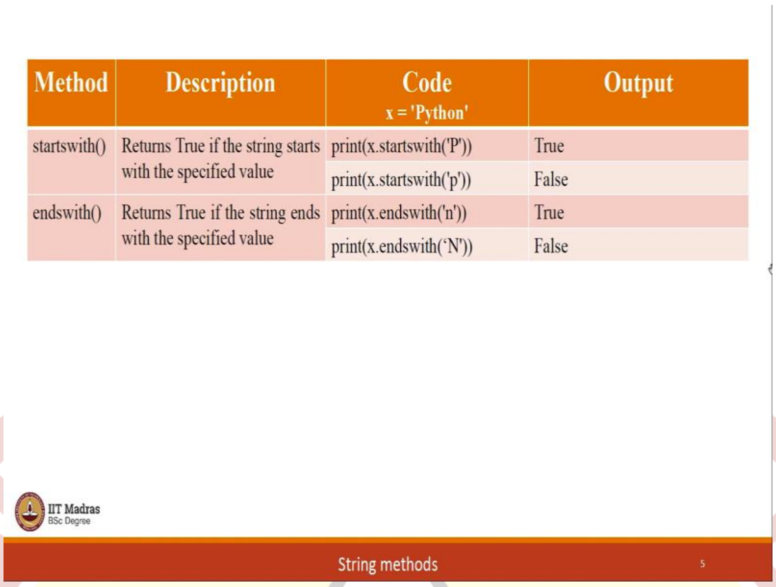
IIT Madras
BSc Degree

String methods 4

Next set of string methods are strip, lstrip and rstrip. As the description says, strip method returns a trimmed version of the string. As per the given example, it removes all those hyphens or those dash symbols, which are there on the leftmost side of the string as well as the rightmost side of the string. This particular method helps us to remove those extra spaces or extra characters, which appears on either side of the actual string.

The same strip function can be split into two different variations first is lstring, which stands for left strip, it removes a specific character from the left side of the string and the second one is rstrip stands for right strip, it removes that specific character from the right of this string. Now, I leave it up to you to find out what happens if that specific character appears somewhere in between the original string whether it will get removed or not.

(Refer Slide Time: 06:27)



Method	Description	Code x = 'Python'	Output
startswith()	Returns True if the string starts with the specified value	print(x.startswith('P'))	True
		print(x.startswith('p'))	False
endswith()	Returns True if the string ends with the specified value	print(x.endswith('n'))	True
		print(x.endswith('N'))	False

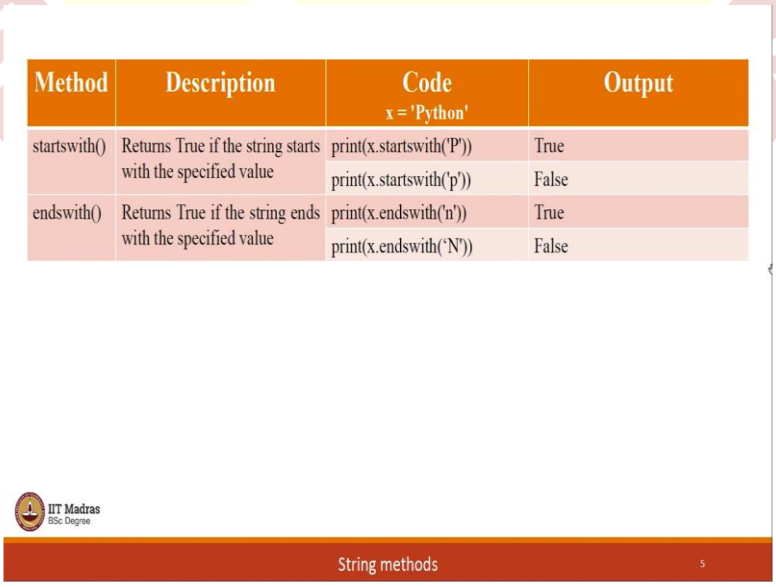
IIT Madras BSc Degree

String methods 5

Next set of string methods are startswith and endswith. As the method name such as it returns true if the string starts with the specified value or it returns true if the string ends with the specific value in case of endswith. The value for variable x is Python with capital P and if you observe the output of starts with capital P, and then starts with small p, then you will observe that this particular method is case sensitive.

As in if we give input which is a specific letter, which matches not just the letter the case as well, then only we get it as true otherwise, the output will be false even though the letter is same, but the case is different. Similarly, endswith here, if the letter n is small, then only we are getting true, if n is an uppercase letter, then we will get false.

(Refer Slide Time: 07:36)



Method	Description	Code x = 'Python'	Output
startswith()	Returns True if the string starts with the specified value	print(x.startswith('P'))	True
		print(x.startswith('p'))	False
endswith()	Returns True if the string ends with the specified value	print(x.endswith('n'))	True
		print(x.endswith('N'))	False

IIT Madras BSc Degree

String methods 5

Next string method is count. It returns the number of times a specified value occurs in a given string, which means it will count how many number of times a specific character or specific value appears in that particular string and the final count value will be given as the output. As you can see, here, the count with character t is 3, but count with character s is 1.

We are getting it one because once again, this particular string method is case sensitive, which means as we are looking for lowercase character s, it will count it will count only all those occurrences of lowercase character s. It will ignore that S which is in uppercase form.

Moving to next string method, which is index. It searches the string for a specified value and returns the position where it was found. The computer will start reading the string from left to right and whenever it gets that specific value first that particular index will be returned as output. As you can see, t appears three times in that particular string. But the first occurrence of letter t from left to right is at index 2, because we all know string index starts from 0.

Similarly, index S is 20. Once again, this particular index method is case sensitive; it is displaying the index of the last s in the particular string. And the last string method is replace, it returns a string where a specified value is replaced with another specified value. If you can see the example, we are performing an operation which says x is equal to x dot replace capital S comma small s.

It works exactly like a Find and Replace feature, which most of the common text editors support. Computer will start scanning the string; it will look for all the occurrences of letter S in uppercase for all such occurrences will be replaced by a small s. Similar thing we are doing with capital M as well in the next line. At the end, when we print the particular string x, we are getting the output where the capital S is converted to small s and capital M is converted to small m by this particular replace method.

You might have noticed, this is the first time we are conducting a lecture where actually we are not executing any piece of Python code. Instead, we are simply going through slides, we are conducting this lecture in this manner because string methods is an exercise for all the Python students. I expect you to execute all these string methods on your own with different types of input and observe how all these string methods behave with respect to the given input. So, I am leaving it up to you guys to execute all these string methods on your own with different inputs. Thank you for watching this lecture. Happy learning!