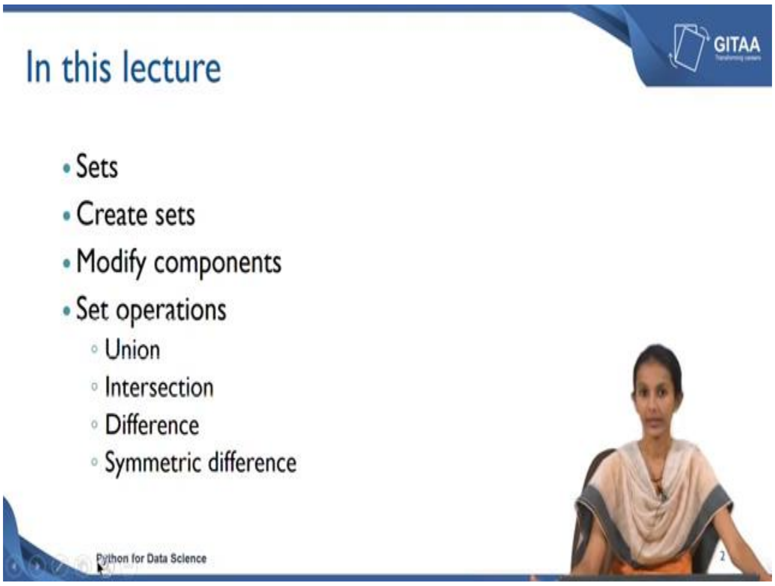
Python for Data Science
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture – 11
Sets

(Refer Slide Time: 00:16)

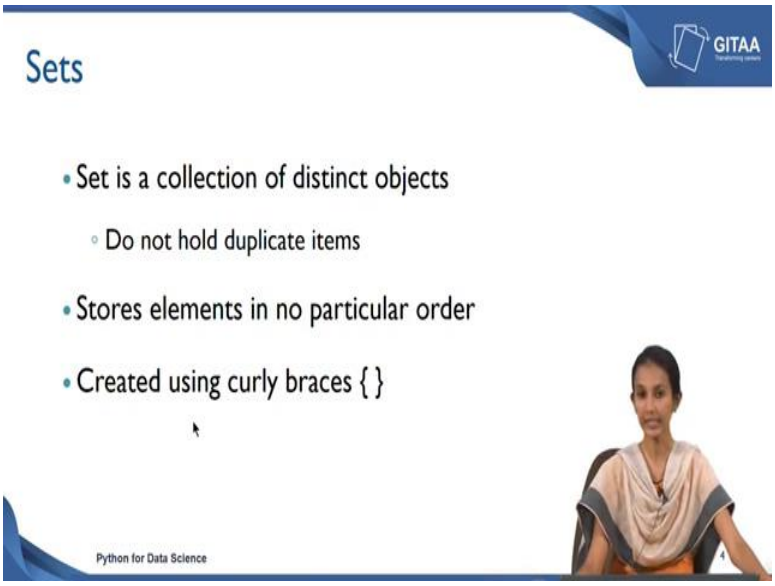


In this lecture

- Sets
- Create sets
- Modify components
- Set operations
  - Union
  - Intersection
  - Difference
  - Symmetric difference

Welcome to the lecture. So, in this lecture we will see what is mean by Sets, how to create sets, how to modify the components in the set? We will also look at the sum of the set operations which are as available in python. So, we will see Union, Intersection, Difference and Symmetric Difference.

(Refer Slide Time: 00:40)



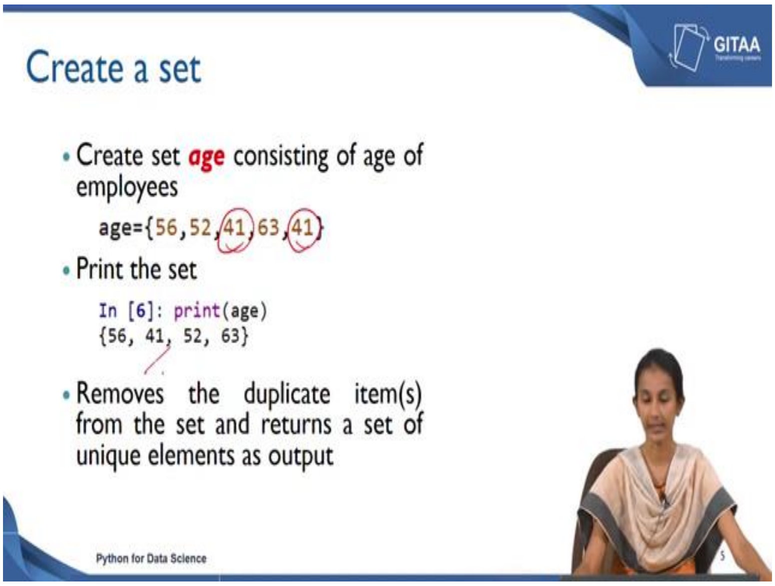
Sets

- Set is a collection of distinct objects
  - Do not hold duplicate items
- Stores elements in no particular order
- Created using curly braces { }

Python for Data Science

Let us get started. So, first we will look at what is mean by sets? Set is basically a collection of objects, distinct objects. It does not hold duplicate items. So, every element is unique in the set. Elements are not stored in any particular order in the set. We can create a set using the curly braces.

(Refer Slide Time: 01:03)



Create a set

- Create set **age** consisting of age of employees

```
age={56,52,41,63,41}
```

- Print the set

```
In [6]: print(age)
{56, 41, 52, 63}
```

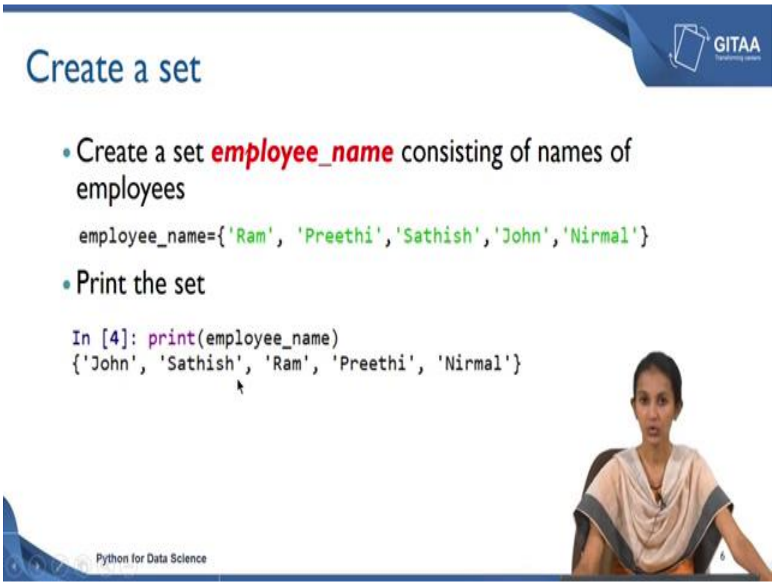
- Removes the duplicate item(s) from the set and returns a set of unique elements as output

Python for Data Science

We will see how to create a set. To create a set call age consisting of age of the employees age. Age is basically our set name. Set will be created using the curly braces. So, we will give the set of values separated by the commas. So, age which basically has five values 56, 52, 41, 63 and 41. So, there are two values which has been repeated which is basically the 41st value. So, we will print the set age.

So, when you print the set age, we already had two values right, 41, but when you print the set age now, only one value will be printed because a set does not hold duplicate items. Every element is unique in the set. So, it removes a duplicate items from the set and returns a unique elements as an output.

(Refer Slide Time: 02:01)



Create a set

- Create a set *employee_name* consisting of names of employees

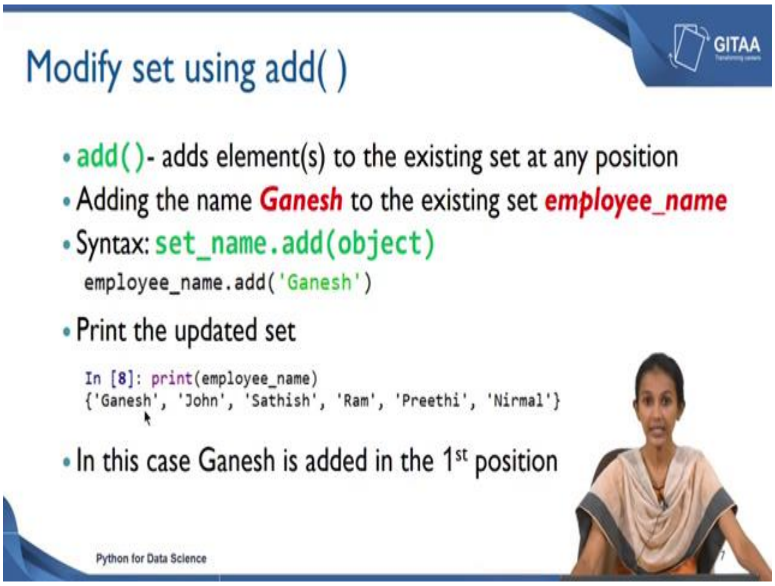
```
employee_name={'Ram', 'Preethi','Sathish','John','Nirmal'}
```

- Print the set

```
In [4]: print(employee_name)
{'John', 'Sathish', 'Ram', 'Preethi', 'Nirmal'}
```

So, we will create another set called employee_name consisting of the employee names. So, our employee_name is basically a set name consist of employees. So, Ram, Preethi, Sathish, John and Nirmal. This set as 5 values. So, let us print employee_name which is a set assigning values to the set. So, we are Ram, Preethi, Sathish, John and Nirmal. So, when you print the set, so the order will not be the same. So, it will send unordered set ie, it not be arranged in the same order. So, you will be getting a different output.

(Refer Slide Time: 02:42)



Modify set using add( )

- add()- adds element(s) to the existing set at any position
- Adding the name *Ganesh* to the existing set *employee_name*
- Syntax: set_name.add(object)

```
employee_name.add('Ganesh')
```

- Print the updated set

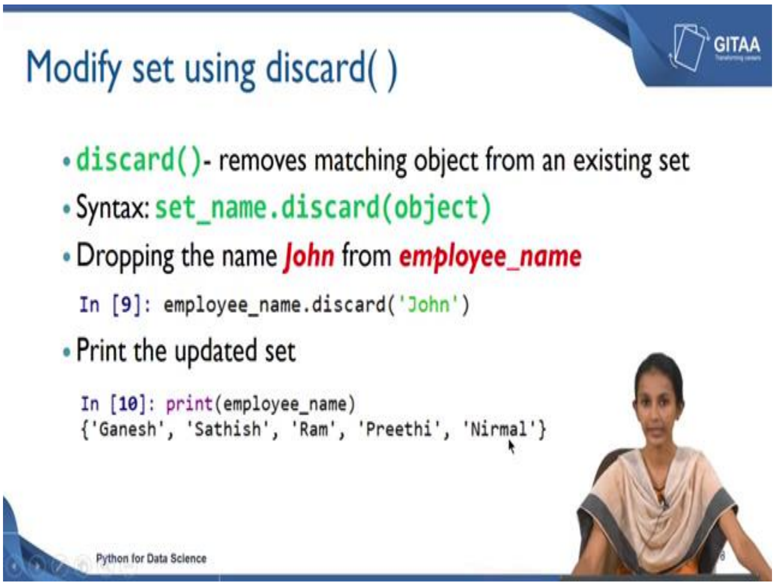
```
In [8]: print(employee_name)
{'Ganesh', 'John', 'Sathish', 'Ram', 'Preethi', 'Nirmal'}
```

- In this case Ganesh is added in the 1st position

So, let us look at how to modify a set using the built in functions available in python. So, first one is we will look at how to add elements to the set. So, the built in function available in python is add. So, it is adds element to the existing set at the any position. So, let us say if we wanted to add Ganesh to the existing set employee_names. So, we have already created and set which is employee_name. So, if we wanted to add Ganesh to the existing set; let us look at how to do that? So, first we look at the syntax. So, set name which is basically as a set.add which is a built in function available and then we have to specify the object which is basically the value.

So, our set name is employee_name.add inside the parenthesis specify the name which has to be added to the existing set so, that is Ganesh. So, let us look at the updated set by printing employee_name. So, you will have 6 values, 6 employees to the employee set so which is Ganesh. So, in this case Ganesh has been added at the first position.

(Refer Slide Time: 04:00)



Modify set using discard( )

- discard()- removes matching object from an existing set
- Syntax: set_name.discard(object)
- Dropping the name John from employee_name

```
In [9]: employee_name.discard('John')
```

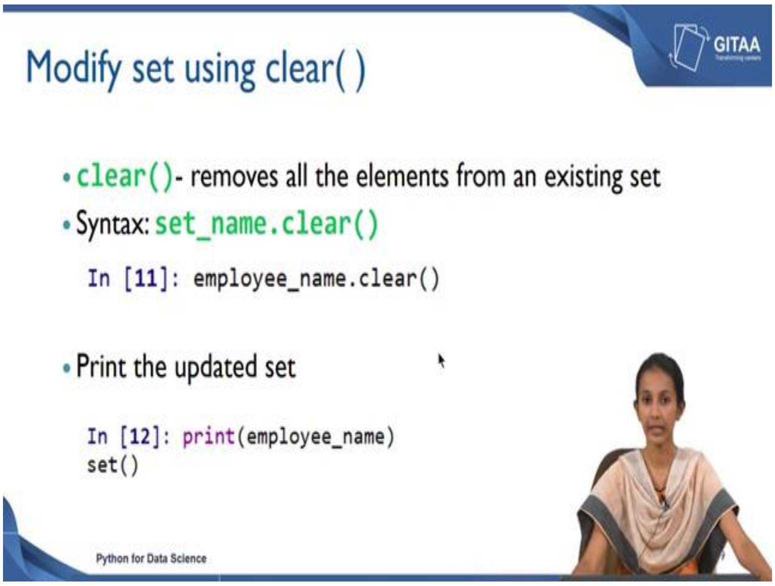
- Print the updated set

```
In [10]: print(employee_name)
{'Ganesh', 'Sathish', 'Ram', 'Preethi', 'Nirmal'}
```

Let us look at how to remove elements from this existing set. So, discard is a built in function which is available in python. So, it removes the matching object from the existing set which is employee_name. Syntax is set_name.discard and inside the parenthesis you have to specify the object which is a value. So, if you wanted to remove John from the employee_name set, the command will be employee_name.discard('John').

So, when you print the updated set we have already added 6 employees. We are going to remove the John from the existing set. So, our updated set should contains only 5 employees. So, here we will have Ganesh, Sathish, Ram, Preethi and Nirmal. So, John has been deleted from the existing set.

(Refer Slide Time: 05:02)



Modify set using clear( )

- clear()- removes all the elements from an existing set
- Syntax: set_name.clear()

```
In [11]: employee_name.clear()
```

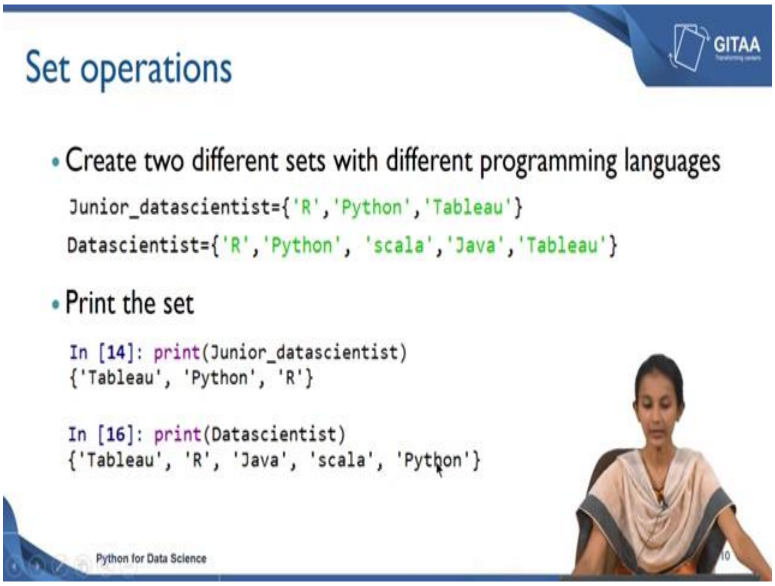
- Print the updated set

```
In [12]: print(employee_name)
set()
```

Python for Data Science

If you wanted to clear all the elements from the set, so, there is an in built command available which is it clear. So, it removes all the elements from the existing set. So, the syntax will be again set_name.clear() function. So, if you give employee_name.clear(). So, when you print the updated set, the set will be a basically a null set which does not contain any elements in the set.

(Refer Slide Time: 05:32)



Set operations

- Create two different sets with different programming languages

```
Junior_datascientist={'R','Python','Tableau'}
Datascientist={'R','Python', 'scala','Java','Tableau'}
```

- Print the set

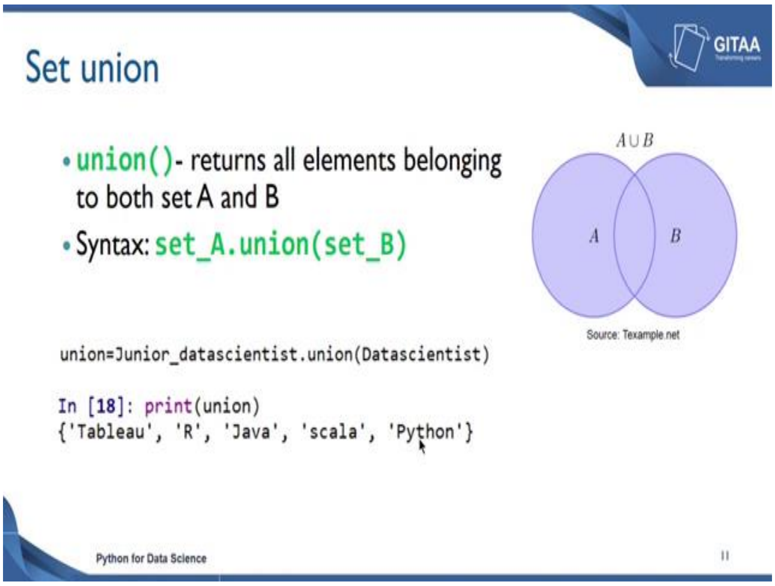
```
In [14]: print(Junior_datascientist)
{'Tableau', 'Python', 'R'}

In [16]: print(Datascientist)
{'Tableau', 'R', 'Java', 'scala', 'Python'}
```

Python for Data Science

Let us look at some of the set operations which is available. So, we will create two different sets with different programming languages. So, the first set will be a junior_datascientist. The set should be created using a curly braces separated by commas. So, the values will be separated by commas. So, our three different programming languages are given which is R, Python, Tableau. Similarly we will create another set datascientist which has programming language R, Python, scala, Java and Tableau.

So, when you print the set, so set will be printed with the values Tableau, Python and R. So, basically we given as R, Python and Tableau, but the output which has been printed is Tableau, Python and R. So, it is not ordered in the same way. So, it is unordered; when we print datascientist. Tableau, R, Java, scala and Python are the programming languages which are defined in datascientist.

(Refer Slide Time: 06:42)



Set union

- union()- returns all elements belonging to both set A and B
- Syntax: set_A.union(set_B)

Source: Texample.net

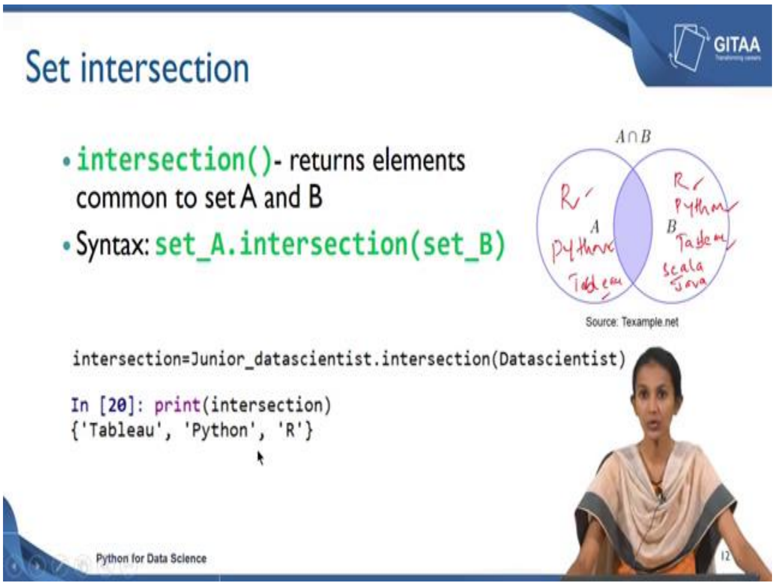
```
union=Junior_datascientist.union(Datascientist)

In [18]: print(union)
{'Tableau', 'R', 'Java', 'scala', 'Python'}
```

Python for Data Science

11

Let us start get started with first Union Operation. So, union is a built in function which is again available in python. So, it returns all the elements, it combines all the elements belonging to both set A and set B. So, this is our first set A which is junior_datascientist. So, we will consider datascientist which is our set B. Union function combined both the elements in set A and set B and returns as an output. So, the syntax is set_A which is junior_data scientist dot union set B. So, the command I am storing it in variable call union=junior_data scientist.union(datascientist), it will returns an output Tableau, R, Java, scala and Python. So, it combines all the elements in both set A and set B.

(Refer Slide Time: 07:41)



Set intersection

- intersection()- returns elements common to set A and B
- Syntax: set_A.intersection(set_B)

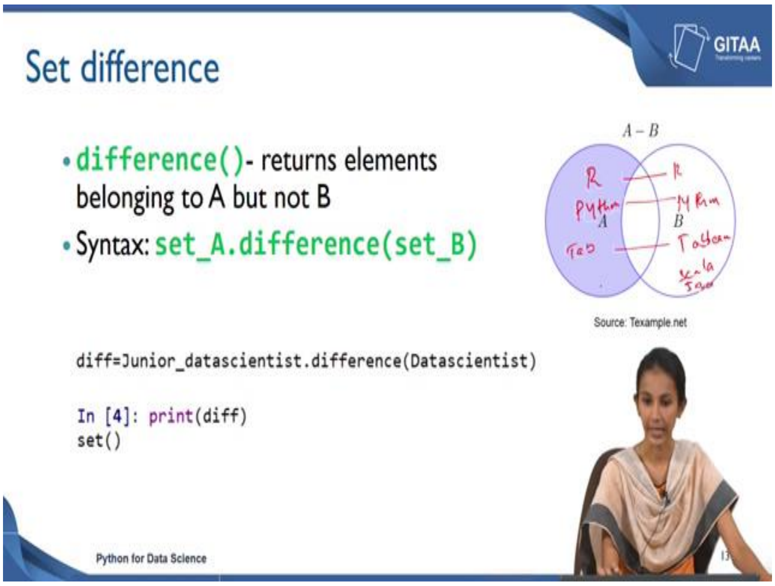
Source: Texample.net

```
intersection=Junior_datascientist.intersection(Datascientist)

In [20]: print(intersection)
{'Tableau', 'Python', 'R'}
```

Next we will look at set intersection. So, intersection is again the in built command available in python. So, it returns elements common to both the set A and set B. So, this our set A, set B. So, it returns this shaded part. So, whatever the elements which are common to both set A and set B, it returns those elements as an output. So, the syntax is set_A.intersection, inside the parenthesis you have to specify this set_B.

So, here I wanted to get the common elements from both the sets, right. So, I am storing it and variable call intersection. So, junior_data scientist.intersection(datascientist). So, in junior_data scientist we add R, Python and Tableau, right. Here we add R, Python, Tableau, Scala and Java. So, in this both the sets when you look, R is common, python is common and tableau is common. So, it returns an element common to both the sets. So, when you print intersection, it prints Tableau, Python and R. So, it is common to both the sets.

(Refer Slide Time: 09:14)



Set difference

- difference()- returns elements belonging to A but not B
- Syntax: set_A.difference(set_B)

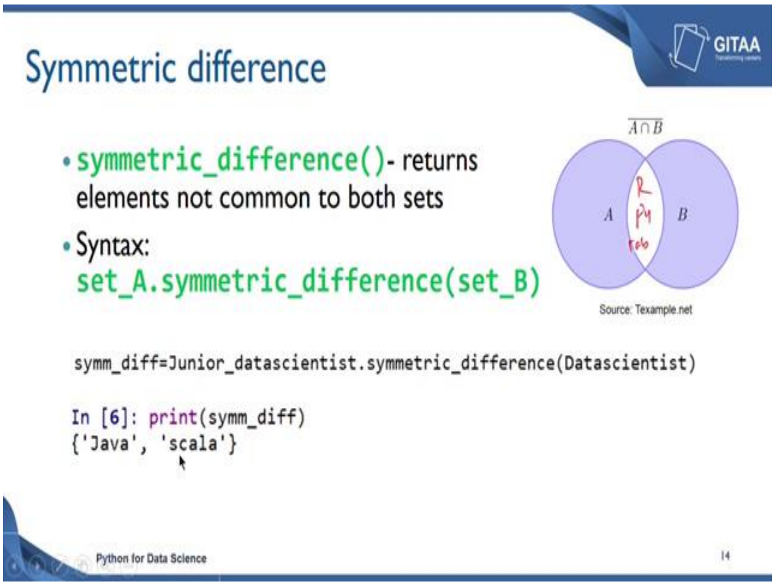
```
diff=Junior_datascientist.difference(Datascientist)

In [4]: print(diff)
set()
```

Source: Texample.net

So, next is set difference. So, difference is it returns an set that is a difference between the two sets. So, you can also do A-B or B-A. So, in this case I am taking an example for A minus B. It returns an elements belonging to A, but not B. So, the syntax is set_A.difference(set_B). So, I wanted to find the difference between the two sets. So, I am storing it and variable called diff=junior_datascientist.difference(Datascientist).

So, when you print difference, it gives a null set because we add R, Python, Tableau. Here we add R, Python, Tableau, Scala and Java. So, when you find the difference, so this get subtracted and this get subtracted and this also get subtracted. So, you do not have any more elements to get subtracted in A-B. So, there is no elements left over in A. So, that is why we are getting a null set.

(Refer Slide Time: 10:36)



Symmetric difference

- symmetric_difference()- returns elements not common to both sets
- Syntax:
  set_A.symmetric_difference(set_B)

Source: Texample.net

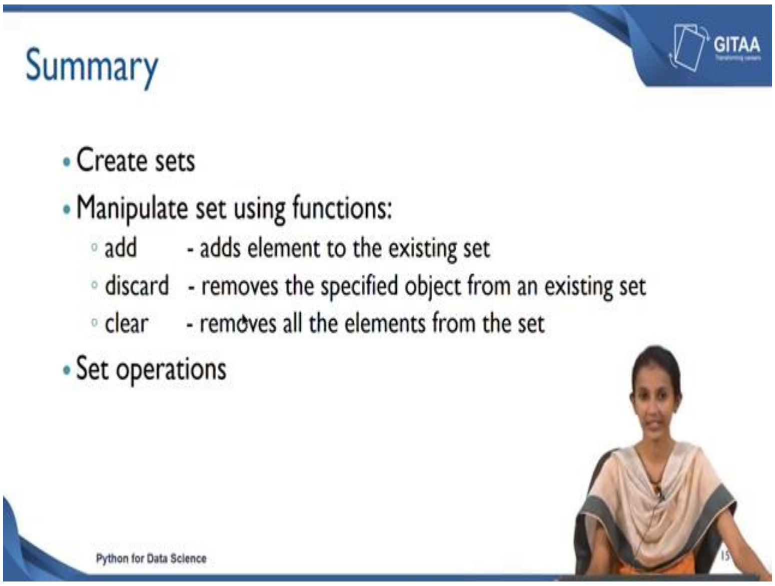
```
symm_diff=Junior_datascientist.symmetric_difference(Datascientist)

In [6]: print(symm_diff)
{'Java', 'scala'}
```

Python for Data Science 14

So, next is symmetric difference. So, it returns elements not common to both the sets. So, it is basically, 1-(A intersection B). So, we have already found at the intersections. So, which is R, Python, Tableau. So, 1-(A intersection B) is basically the symmetric difference in this case.

So, we will first look at the syntax set_A.symmetric difference set_B. So, I am storing at a variable call symmetric_diff. So, I am giving this set name which is junior_data scientist.symmetric difference and specifying the set B. So, we have already had we have already found out the intersection value which is R, Python and Tableau. So, the remaining elements are scala and Java, right. So, let us print the output of symmetric_difference. So, we will be getting the Java and scala which is basically the output for the symmetric difference.

(Refer Slide Time: 11:45)



Summary

- Create sets
- Manipulate set using functions:
  - add - adds element to the existing set
  - discard - removes the specified object from an existing set
  - clear - removes all the elements from the set
- Set operations

So, let us summarize. First we saw how to create sets. So, the sets are created using the curly braces. So, the values are specified inside the curly braces separated by the commas. We also looked at how to manipulate set using the function which is available in Python. First we saw add. So, it adds an element to the existing sets. We also saw how to remove the specified objects. So, discard is a command, clear, it remove all the elements from the set and we also saw some of the set operations, union, intersection, difference and also the symmetric difference.

Thank you.