Set Collection Class:

- ✓ Set is an unordered collection, which doesn't preserve the insertion order.
- ✓ Set collection will arrange the elements inside the collection in sorting order by default.
- ✓ Set collection allows to store PRIMITIVE Type elements, SObject Type, Apex Type, Collection Type and user defined type elements.
- ✓ Set collection supports the Dynamic memory allocation
- ✓ Hence the memory size can grow / shrink at runtime.
- ✓ Set collection doesn't allow the Duplicate elements. Upon adding the new elements, it will maintain the uniqueness with the help of BINARY COMPARISON.
- ✓ Set collection provides a set of readymade methods, to manage the elements inside the collection.

Syntax:

```
Set <Datatype> <objectName> = new Set<Datatype> ();
```

Example:

```
Set<integer> customer Codes = new Set<integer> ();
Set<String> CountryNames = new Set<String> ();
Set<Id> recordIds = new Set<Id> ();
Set<Account> accountsSet = new Set<Account> ();
```

Methods:

```
Set<String> countryName = New Set<String> ();
```

- **Add(<ElementName>):**

This Method will add a new element to the collection. Upon adding the element, it will arrange the elements in the sorting order by default.

```
Ex: countryNames.add('India');
    countryNames.add('usa');
    countryNames.add('Srilanka');
```

- **AddAll(<collectionName>):**

  This method is used to add a collection of elements from the source collection.

  Ex:
```
String [] countries = new String [] {'USA','INDIA', 'SRILANKA'};
countryNames.AddAll(countries);
```

- **Integer Size ():**

  It returns an integer, which indicates the number of elements exist inside the collection.

  Ex:
```
countryName.Size();
```

- **Boolean isEmpty ():**

  This method returns TRUE, when the collection isEmpty. Else it returns FALSE.

  Ex:
```
if(countryNames.isEmpty()){
System.debug('Collection is Empty.');
}
   Else {
System.debug('Collection id not Empty.');
}
```

- **Boolean Contains (<ElementName>):**

  This method is used to search for the specified element in the collection. It returns TRUE, when the element found. Else it returns FALSE.

  Ex:
```
countryNames.Countains('USA');
```

- **Remove (<ElementName>):**

  This method is used to remove the specified element from the collection.

  Ex:
```
countryNames.Remove('SRILANKA');
```

- **Clear ():**

  This method will remove all the elements from the collection.

  Ex:
```
countryNames.Clear();
```

- **Equals (<collection>):**

  This method is used to compare the two collections. It returns TRUE, when both the collections are same. Else it returns FALSE.

  Ex: countryNames.Equals (<collectionName>);

Keep Learning | Keep Practicing

Sfdc Telugu

Follow on : Twitter

https://twitter.com/sfdc_telugu

Follow on : Facebook Page

https://www.facebook.com/people/Sfdc-Telugu/100087211670093/

Support us:

https://pages.razorpay.com/Buymeacoffe