-----------------------------------

List<Account> accList = new list <Account>();

accList = [SELECT ID, NAME FROM ACCOUNT WHERE ISDELETED = TRUE ALL ROWS];

Undelete accList;

-------------------------------------

List<Account> acclist = new List <Account>();

accList = [SELECT ID, NAME, INDUSTRY FROM ACCOUNT WHERE NAME = 'Test1'];

Delete accList;

---------------------------------------

List<Account> accList = new List <Account>();

accList = [SELECT ID, NAME, INDUSTRY FROM ACCOUNT];

for(account ac : accList ){

ac.industry = 'Wipro';

}

Update accList;

-------------------------------------------

List<Lead> acclist = new List <Lead>();

for(integer i=0; i<=10; i++){

   Lead ac = new Lead();

   ac.lastname='Test'+i;

   ac.company='Test Co.'+i;

   acclist.add(ac);

}

   insert acclist;

------------------------------------------

Lead obj = new Lead();

obj.firstname = 'Test1';

obj.lastname = 'Test2';

obj.company = 'TFT';

insert obj;

contact obj2 = new contact();

obj2.lastname = obj.lastname;

insert obj2;

----------------------------------------

public class Example1 {

   public list<Account> accs {get; set;}

   public Example1(){

       accs=[select id,name,industry from Account];

   }

}

-----------

<apex:page controller="Example1">

   <apex:pageBlock title="SOQL Results">

     <apex:pageBlockTable value="{!accs}" var="a">

         <apex:column value="{!a.name}"/>

         <apex:column value="{!a.industry}"/>

      </apex:pageBlockTable>

   </apex:pageBlock>

</apex:page>

Use both code together CLASS AND VF click preview.

Integer

|  |
| --- |
| Integer barrelNumbers = 1000; system.debug(' value of barrelNumbers variable: '+barrelNumbers); |
| value of barrelNumbers variable: 1000 |

Boolean

|  |
| --- |
| Boolean shipmentDispatched; shipmentDispatched = true; System.debug('Value of shipmentDispatched '+shipmentDispatched); |
| Value of shipmentDispatched true |

Date

|  |
| --- |
| Date ShipmentDate = date.today(); System.debug('ShipmentDate '+ShipmentDate); |
| ShipmentDate 2018-01-03 00:00:00 |

Long

|  |
| --- |
| Long companyRevenue = 21474838973344648; system.debug('companyRevenue'+companyRevenue); |
| companyRevenue21474838973344648 |

Object

|  |
| --- |
| Account objAccount = new Account (Name = 'Test Chemical'); system.debug('Account value'+objAccount); |
|  |

String

|  |
| --- |
| String companyName = 'Abc International'; System.debug('Value companyName variable'+companyName); |
|  |

Enum

|  |
| --- |
| //Declaring enum for Chemical Compounds public enum Compounds {HCL, H2SO4, NACL, HG} Compounds objC = Compounds.HCL; System.debug('objC value: '+objC); |
|  |

String, Integer, Set and Map

|  |
| --- |
| String productName = 'HCL'; Integer i = 0; Set<string> setOfProducts = new Set<string>(); Map<id, string> mapOfProductIdToName = new Map<id, string>(); |
|  |

for loop

|  |
| --- |
| for (Integer i = 0; i<10; i++) {      System.debug('This code will throw error'); } |
|  |

List

|  |
| --- |
| List<string> Products = new List<strings>(); Products.add('HCL');  system.debug('This is my added product '+Products[0]); |
|  |

Descending order list **List Sorting**

List<String> colors = new List<String>{

   'Yellow',

   'Red',

   'Green'};

colors.sort();

System.assertEquals('Green', colors.get(0));

system.debug('This is Selected color '+colors[0]);

System.assertEquals('Red', colors.get(1));

system.debug('This is Selected color '+colors[1]);

System.assertEquals('Yellow', colors.get(2));

system.debug('This is Selected color '+colors[2]);

This is Selected color Green

This is Selected color Red

This is Selected color Yellow

Contains

|  |
| --- |
| String myProductName1 = 'HCL'; String myProductName2 = 'NAHCL'; Boolean result = myProductName2.contains(myProductName1); System.debug('O/p will be true as it contains the String and Output is:'+result); |
|  |

Equals

|  |
| --- |
| String myProductName1 = 'HCL'; String myProductName2 = 'NAHCL'; Boolean result = myProductName2==myProductName1; System.debug('it contains the String and Output is:'+result); |
|  |

|  |
| --- |
| String myString1 = 'MyString'; String myString2 = 'MyString'; Boolean result = myString2.equals(myString1); System.debug('Value of Result will be true as they are same and Result is:'+result); |
|  |

equalsIgnoreCase

|  |
| --- |
| String myString1 = 'MySTRING'; String myString2 = 'MyString'; Boolean result = myString2.equalsIgnoreCase(myString1); System.debug('Value of Result will be true as they are same and Result is:'+result); |
|  |

Remove

|  |
| --- |
| String myString1 = 'This Is MyString Example'; String stringToRemove = 'MyString'; String result = myString1.remove(stringToRemove); System.debug('Value of Result will be 'This Is Example' as we have removed the MyString     and Result is :'+result); |
|  |

removeEndIgnoreCase

|  |
| --- |
| String myString1 = 'This Is MyString EXAMPLE'; String stringToRemove = 'Example'; String result = myString1.removeEndIgnoreCase(stringToRemove); System.debug('Value of Result will be 'This Is MyString' as we have removed the 'Example' and Result is :'+result); |
|  |

startsWith

|  |
| --- |
| String myString1 = 'This Is MyString EXAMPLE'; String prefix = 'This'; Boolean result = myString1.startsWith(prefix); System.debug(' This will return true as our String starts with string This and the Result is :'+result); |
| This will return true as our String starts with string This and the Result is :true |

Arrays

|  |
| --- |
| //Defining array String [] arrayOfProducts = new List<String>();  //Adding elements in Array arrayOfProducts.add('HCL'); arrayOfProducts.add('H2SO4'); arrayOfProducts.add('NACL'); arrayOfProducts.add('H2O'); arrayOfProducts.add('N2'); arrayOfProducts.add('U296');  for (Integer i = 0; i<arrayOfProducts.size(); i++) {      system.debug('Values In Array: '+arrayOfProducts[i]); } |
| Values In Array: HCL  Values In Array: H2SO4  Values In Array: NACL  Values In Array: H2O  Values In Array: N2  Values In Array: U296 |

List

|  |
| --- |
| List<string> ListOfCities = new List<string>(); System.debug('Value Of ListOfCities'+ListOfCities); |
| Value Of ListOfCities() |

|  |
| --- |
| List<string> ListOfStates = new List<string> {'NY', 'LA', 'LV'}; System.debug('Value ListOfStates'+ListOfStates); |
| Value ListOfStates(NY, LA, LV) |

|  |
| --- |
| List<List<Set<Integer>>> myNestedList = new List<List<Set<Integer>>>(); System.debug('value myNestedList'+myNestedList); |
| value myNestedList() |

Set

|  |
| --- |
| // Adds an element to the set Set<string> ProductSet = new Set<string>{'Phenol', 'Benzene', 'H2SO4'}; ProductSet.add('HCL'); System.debug('Set with New Value '+ProductSet); |
| Set with New Value {Benzene, H2SO4, HCL, Phenol} |

How to use copied value from one string to another string

Set<String> s1 = new Set<String>();

s1.add('item1');

s1.add('item2');

Set<String> s2 = new Set<String>(s1);

// The set elements in s2 are copied from s1

System.debug(s2);

{item1, item2}

Set<String> s1 = new Set<String>();

s1.add('item1');

s1.add('item2');

system.debug('this is my created value '+s1);

this is my created value {item1, item2}

List<Integer> ls = new List<Integer>();

ls.add(1);

ls.add(2);

// Create a set based on a list

Set<Integer> s1 = new Set<Integer>(ls);

// Elements are copied from the list to this set

System.debug(s1);

DEBUG|{1, 2}

List<Integer> ls = new List<Integer>();

ls.add(1);

ls.add(2);

Set<Integer> s1 = new Set<Integer>(ls);

s1.add(3);

s1.add(4);

list<integer> Iss = new list <integer>(s1);

system.debug(Iss);

DEBUG|(1, 2, 3, 4)

// Adds an element to the set  
Set<string> ProductSet = new Set<string>{'Phenol', 'Benzene', 'H2SO4'};  
ProductSet.add('HCL');  
System.debug('Set with New Value '+ProductSet);  
  
// Removes an element from set  
ProductSet.remove('HCL');  
System.debug('Set with removed value '+ProductSet);

Set with New Value {Benzene, H2SO4, HCL, Phenol}

Set with removed value {Benzene, H2SO4, Phenol}

// Adds an element to the set  
Set<string> ProductSet = new Set<string>{'Phenol', 'Benzene', 'H2SO4'};  
ProductSet.add('HCL');  
System.debug('Set with New Value '+ProductSet);  
  
// Removes an element from set  
ProductSet.remove('HCL');  
System.debug('Set with removed value '+ProductSet);  
  
// Check whether set contains the particular element or not and returns true or false  
ProductSet.contains('HCL');  
System.debug('Value of Set with all values '+ProductSet);

Set with New Value {Benzene, H2SO4, HCL, Phenol}

Set with removed value {Benzene, H2SO4, Phenol}

Value of Set with all values {Benzene, H2SO4, Phenol}

Map

// Initialize the Map  
Map<string, string> ProductCodeToProductName = new Map<string, string> {'1000'=>'HCL', '1001'=>'H2SO4'};  
  
// This statement would give as output as key value pair in Debug log  
System.debug('value of ProductCodeToProductName'+ProductCodeToProductName);

value of ProductCodeToProductName{1000=HCL, 1001=H2SO4}

try{

   merchandise\_\_c m = [SELECT name FROM merchandise\_\_c LIMIT 1];

   decimal dec=m.price\_\_c;

}

catch (exception e){

system.debug('this is a query exception block');

}

finally {

   system.debug ('i am in final block');

}

system.debug('my name is afghan');