

APEX - VARIABLES

Apex Variables

Java and Apex are similar in many manners. Variable declaration in Java and Apex is also quite same. Below are some examples to show how to declare local variables.

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

Note that all the variables are assigned with the value null.

Declaring Variables

You could declare the variables in Apex like String and Integer as follows:

```
String strName = 'My String';//String variable declaration
Integer myInteger = 1;//Integer variable declaration
Boolean mtBoolean = true;//Boolean variable declaration
```

Apex variables are Case-Insensitive

This means that below code will throw an error since the variable 'i' has been declared two times and both will be treated as same.

```
Integer m = 100;
for (Integer i = 0; i<10; i++) {
    integer m=1; //This statement will throw an error as m is being declared again
    System.debug('This code will throw error');
}
```

Scope of Variables

An Apex variable is valid from the point where it is declared in code. So it is not allowed to redefine the same variable again and in code block. Also, if you declare any variable in a method then that variable scope will be limited to that particular method only. However, class variables can be accessed through out the class.

Example:

```
//Declare variable Products
List<string> Products = new List<strings>();
Products.add('HCL');

//You cannot declare this variable in this code clock or sub code block again
//If you do so then it will throw the error as the previous variable in scope
//Below statement will throw error if declared in same code block
List<string> Products = new List<strings>();
```