Java Package

Package is a collection of related classes. Java uses package to group related classes, interfaces and sub-packages in any Java project.

We can assume package as a folder or a directory that is used to store similar files.

In Java, packages are used to avoid name conflicts and to control access of class, interface and enumeration etc. Using package, it becomes easier to locate the related classes and it also provides a good structure for projects with hundreds of classes and other files.

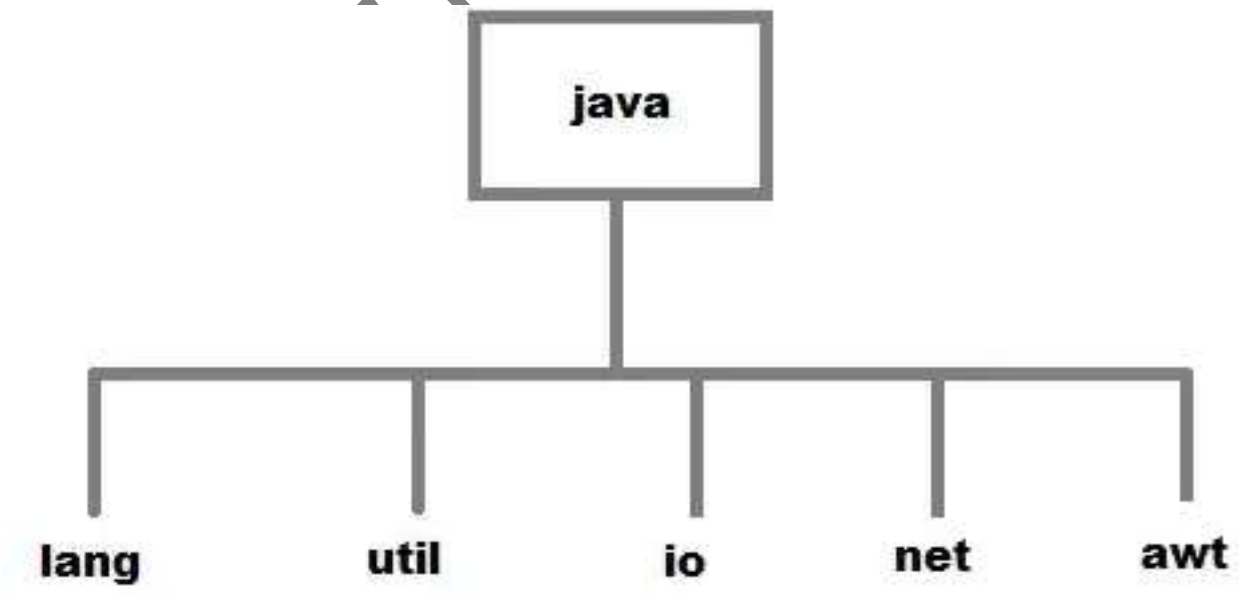
Let's understand it by a simple example, Suppose, we have some math related classes and interfaces then to collect them into a simple place, we have to create a package.

- **Types Of Java Package**

Package can be built-in and user-defined, Java provides rich set of built-in packages in form of API that stores related classes and sub-packages.

Built-in Package: math, util, lang, i/o etc are the example of built-in packages.

User-defined-package: Java package created by user to categorize their project's classes and interface are known as user-defined packages.



- **How to Create a Package?**

Creating a package in java is quite easy, simply include a package command followed by name of the package as the first statement in java source file.

```
package mypack;

public class employee

{

        String empId;

        String name;

}
```

The above statement will create a package with name mypack in the project directory.

Java uses file system directories to store packages. For example the .java file for any class you define to be part of mypack package must be stored in a directory called mypack.

- **Additional points about package:**

Package statement must be first statement in the program even before the import statement.

A package is always defined as a separate folder having the same name as the package name.

Store all the classes in that package folder.

All classes of the package which we wish to access outside the package must be declared public.

All classes of the package must be compiled before use.

- **How to import Java Package**

To import java package into a class, we need to use java import keyword which is used to access package and its classes into the java program.

Use import to access built-in and user-defined packages into your java source file so that your class can refer to a class that is in another package by directly using its name.

There are 3 different ways to refer to any class that is present in a different package:

1. without importing the package
2. importing package with specified class
3. importing package with all classes

Lets understand each one with the help of example.

Accessing package without import keyword

If you use fully qualified name to import any class into your program, then only that particular class of the package will be accessible in your program, other classes in the same package will not be accessible. For this approach, there is no need to use the import statement. But you will have to use the fully qualified name every time you are accessing the class or the interface. This is generally used when two packages have classes with same names. For example: java.util and java.sql packages contain Date class.

Example:

In this example, we are creating a class A in package pack and in another class B, we are accessing it while creating object of class A.

```
//save by A.java
package pack;
public class A{
	public void msg(){
		System.out.println("Hello");
	}
}
```

```
//save by B.java
class B {
	public static void main(String args[]) {
		pack.A obj = new pack.A();  //using fully qualified name
		obj.msg();
	}
}
```

Import the Specific Class

Package can have many classes but sometimes we want to access only specific class in our program in that case, Java allows us to specify class name along with package name. If we use import packagename.classname statement then only the class with name classname in the package will be available for use.

Example:

In this example, we created a class Demo stored into pack package and in another class Test, we are accessing Demo class by importing package name with class name.

```
//save by Demo.java
package pack;
public class Demo {
	public void msg() {
		System.out.println("Hello");
	}
}
//save by Test.java
import pack.Demo;
class Test {
	public static void main(String args[]) {
	Demo obj = new Demo();
```

```
        obj.msg();
        }
}
```

Import all classes of the package

If we use packagename.* statement, then all the classes and interfaces of this package will be accessible but the classes and interface inside the sub-packages will not be available for use.

The import keyword is used to make the classes of another package accessible to the current package.

Example :

In this example, we created a class First in learnjava package that access it in another class Second by using import keyword.

```
//save by First.java
package learnjava;
public class First{
        public void msg() {
                System.out.println("Hello");
        }
}
//save by Second.java
import learnjava.*;
class Second {
        public static void main(String args[]) {
                First obj = new First();
                obj.msg();
        }
}
```