


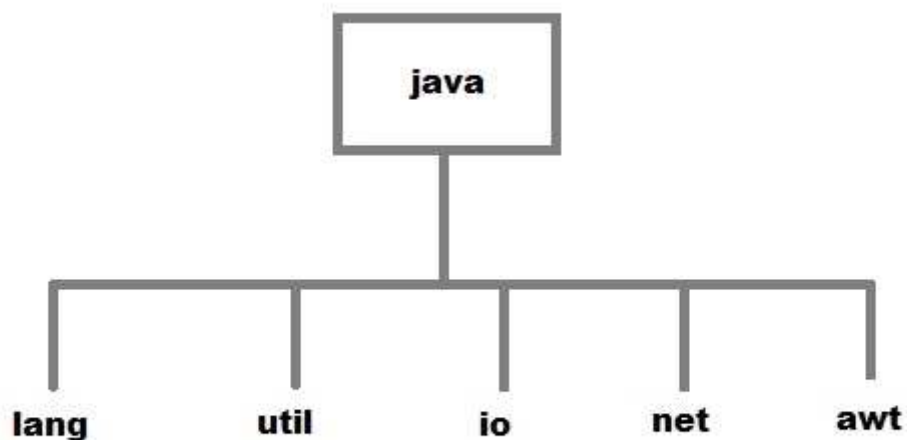
Java Package

 studytonight.com/java/package-in-java.php

Package are used in Java, in-order to avoid name conflicts and to control access of class, interface and enumeration etc. A package can be defined as a group of similar types of classes, interface, enumeration or sub-package. 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.

Types of Packages: Built-in and User defined

- **Built-in Package:** Existing Java package for example `java.lang` , `java.util` etc.
- **User-defined-package:** Java package created by user to categorize their project's classes and interface.



Creating 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  
{  
    statement;  
}
```

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:

- 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 within the package must have the package statement as its first line.
- All classes of the package must be compiled before use (So that they are error free)

Example of Java packages

```
//save as FirstProgram.java
package learnjava;
public class FirstProgram{
    public static void main(String args[]) {
        System.out.println("Welcome to package");
    }
}
```

How to compile Java programs inside packages?

This is just like compiling a normal java program. If you are not using any IDE, you need to follow the steps given below to successfully compile your packages:

```
javac -d directory javafilename
```

Example:

```
javac -d . FirstProgram.java
```

The **-d** switch specifies the destination where to put the generated class file. You can use any directory name like **d:/abc** (in case of windows) etc. If you want to keep the package within the same directory, you can use **.** (dot).

How to run Java package program?

You need to use fully qualified name e.g. **learnjava.FirstProgram** etc to run the class.