

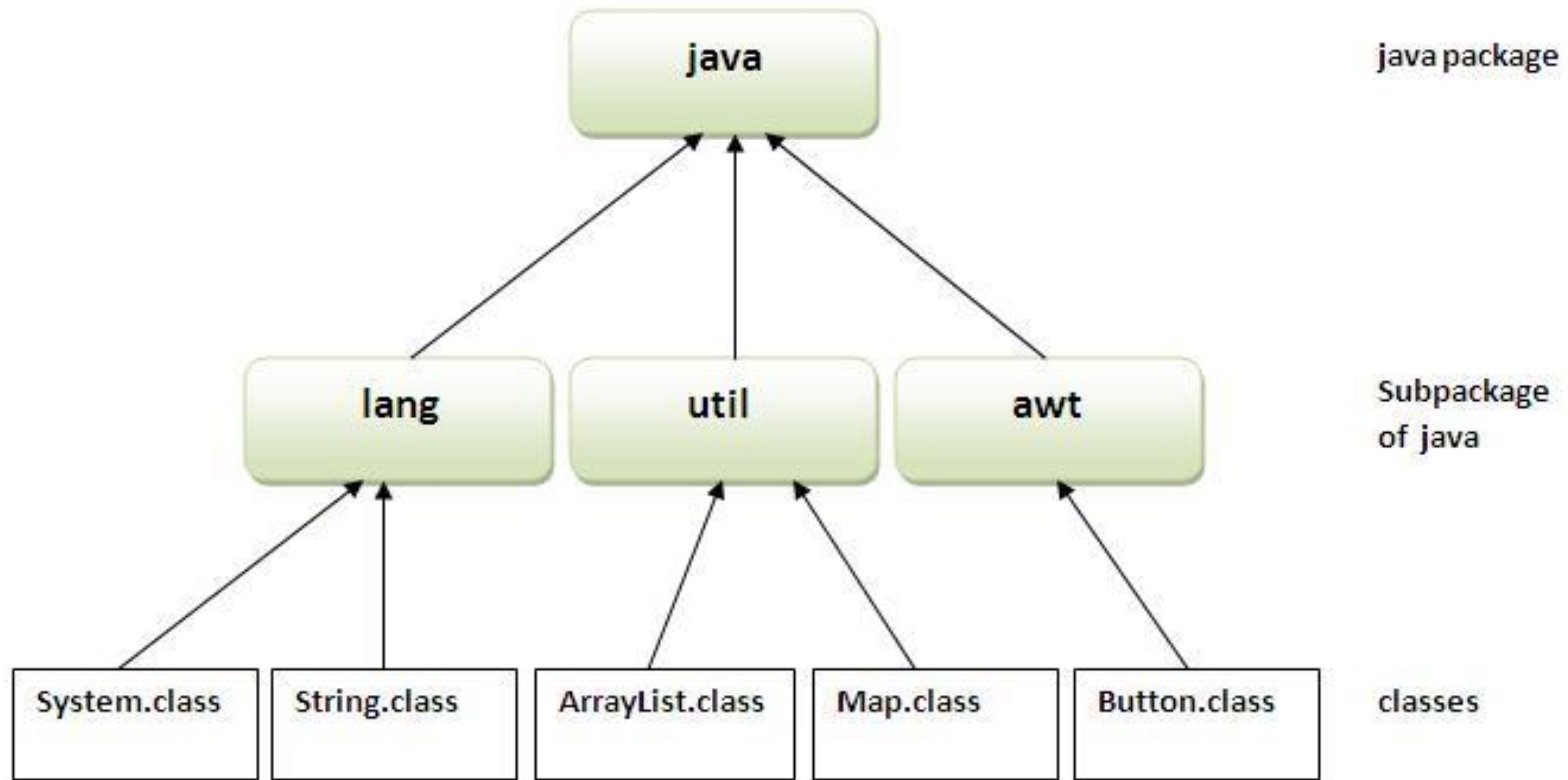
Package in Java

- A package is a group of similar types of classes, interfaces and sub-packages.
- Package can be categorized in two form, **built-in package** and **user-defined package**.
- There are many built-in packages such as java, lang, awt, javax, swing, net, io, util, sql etc.
- Java API is having nearly 5000 pre defined packages.
- Package statement will be the first statement in a java program.
- We can access the members of one class from another class of same package.
- 'import' statement is used to connect classes in java application of different packages.

Some Important Packages

Package	Description
java.lang	Lang stands for 'language', this got primary classes and interfaces essential for developing a basic java program
java.util	Util stands for 'utility', This package contains useful classes and interfaces like Stack, LinkedList, Hashtable, etc ... These classes are called collections.
java.io	Io stands for 'input and output'. This package contains streams.
java.awt	awt stands for 'abstract window toolkit'. This package helps to develop GUI.
javax.swing	This package helps to develop GUI like java.awt. The 'x' in javax represents that it is an extended package.
java.net	net stands for 'network'. Client-Server programming can be done by using this package.
java.applet	Applets are programs which came from a server into a client and get executed on the client machine on a network.
java.text	This package has two important classes, DateFormat and NumberFormat.
java.sql	Sql stands for 'structured query language'. This package helps to connect to databases.

Structure of a package



Simple example of package

- The **package keyword** is used to create a package.

//save as Simple.java

```
package mypack;  
public class Simple  
{  
    public static void main(String args[]){  
        System.out.println("Welcome to package");  
    }  
}
```

How to access package from another package?

- There are three ways to access the package from outside the package.

✓ `import packageName.*;`

✓ `import packageName.classname;`

✓ fully qualified Classname

Using *packagename.**

- If you use *packagename.** then all the classes and interfaces of this package will be accessible but not subpackages.
- The `import` keyword is used to make the classes and interface of another package accessible to the current package.

//save by A.java

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

//save by B.java

```
package mypack;  
import pack.*;  
class B {  
    public static void main(String args[]){  
        A obj = new A();  
        obj.msg();  
    }  
}
```

Output:Hello

Using *packagename.classname*

- If you import *packagename.classname* then only declared class of this package will be accessible.

//save by A.java

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

//save by B.java

```
package mypack;  
import pack.A;
```

```
class B{  
    public static void main(String args[]){  
        A obj = new A();  
        obj.msg();  
    }  
}
```

Output:Hello

Using fully qualified name

- If you use fully qualified name then only declared class of this package will be accessible.
- Now there is no need to import.

//save by A.java

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

//save by B.java

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

Output:Hello

Understanding Access Modifiers

- Access modifiers determine whether other classes can use a particular field or invoke a particular method.
- There are '2' levels of access modifiers.
 1. At Class level: public & default
 2. At member level: public, private, protected and default

At Class level:

- If a class is declared as public then that is visible to all the classes everywhere.
- If a class is declared as default (package-private) then that is visible to only within its own package.

At Member level:

- At the member level, we can also use the public or default (package-private) just as with class level, and with the same meaning.

Modifier	Class	Package	Sub-Class	World
public	YES	YES	YES	YES
protected	YES	YES	YES	
default	YES	YES		
private	YES			

PUBLIC:

- If a method, variable or constructor is declared as public then we can access them from anywhere.
- When we are accessing the public member its class also should be public otherwise will be getting compile time error.

DEFAULT:

- If a method, variable or constructor is declared as default then we can access them from current package only. So it is also called "PACKAGE -PRIVATE"

PRIVATE:

- If a method, variable or constructor is declared as private then we can access them in the current class only.
- Private is the most restricted access modifier.
- If a constructor is declared as private we can't create a object for that class in other classes.

PROTECTED:

- If a method, variable or constructor is declared as protected then we can access them within the current package.
- We can use PROTECTED members outside the package only in child class, and we can access them by using **child class reference only** not from parent class reference.

Kishan