

## (Transcript) Java Packages



### Introduction

What is a package? A package operates like a folder with files related to each other. In this case, the related files are classes. Every package maintains 2 purposes. First, it organizes classes as 1 unit and second, it controls access to members of a class. A class defined within a package must be accessed through their package name. Access to members of a class such as variables and methods are carried out using access modifiers.

In this session, we are going to look at Package Convention and Package Creation.

### View Class Organization

In Eclipse, we have src which is the source folder that contains source files for our project. It has 5 packages. A package can have sub packages. Under com.selenium4beginners, I have Java, selenium\_testng, and utility as the sub packages. Here's a set of related packages and 3 classes under java. There's a lot of classes under selenium\_testng.

## Package Convention

When it comes to the Package Convention, if we do not include a package name then the class goes to default package. For example, right click src > New > Class. The Package field is blank but we see default in parenthesis. This is not recommended. Look what happens when I write Class1 for class name. A warning shows “The use of the default package is discouraged”. Click Finish and we see Class1 in the default package. Notice how line 1 is blank. I’m going to drag Class1 to the java. Now, line 1 shows a package statement

## Create Package

To create a package, we right click src > New > Package. By convention, packages must begin with a lowercase letter. If I write capital D for DemoPackage. The warning shows “Discouraged package name. By convention, package names usually start with a lowercase letter”. Click Finish. It’s not wrong but not recommended to start with an uppercase letter. Delete DemoPackage.

## Create Sub Package

Do you see how com.selenium4beginners is different than the other packages? Some projects use a reverse website domain as their package name. If I had a domain called selenium4beginners.com then in reverse my package would com.selenium4beginners. The dot creates a sub package. Let’s create a sub package under java: right click java, new, package, add a dot after java then demopackag1 and click Finish. Drag Class1 again but this time to demopackag1.

Dialog shows Update references to Class1.java. Hit OK and I realize I misspelled packag. So I am going to right click packag1 and Refactor, Rename and I’m going to add the e to packag1 and click OK again. Now, we see the package has been updated. In the statement and Package Project Explorer. The package statement updated to demopackage1.

That’s it for packages and organizing our classes. Next, I’m going to show you how to access the control features using all 4 access modifiers and I’ll see you in the next video.