Selenium 4 Beginners

Rex Jones II

Conventional Naming In Java

Table of Contents

Selenium 4 Beginners

Rex Jones II

Introduction

Hello everybody, Welcome To Selenium 4 Beginners. My name is Rex Allen Jones II. In this video, I will cover the naming convention for Java packages, classes, methods, and variables. Conventional is conforming to standards that are accepted by Java. It is possible for those conventions to change from company to company and project to project. I've been on projects that have a document stating what they want and how they want things done. However, I am going to share the most used conventions for Java.

Our tutorial plan will cover the following: How To Name A Package, How To Name A Class, How To Name A Method, How To Name A Variable, followed by Questions, Answers, & Practice. The practice is always optional but I know Practice Makes Improvement.

How To Name A Package

Let's start with How To Name A Package. In our previous sessions, I did not use a conventional naming standard for the packages. As we can see, there is a default package and a package called Building Blocks which was created in our session Building Blocks For Selenium. Convention is a standard but we do not have to use that standard.

Know what, I am getting ahead of myself. Before we get into the package naming convention. Give me a moment to explain package. What is a package? A package is a group of classes with 2 purposes for each class.

Number 1 a package is used to organize classes as a group. It is beneficial to the programmer when related classes are grouped together. Number 2 a package is used to control access to the class. Classes can be defined as public to accept outside code and defined as private without an access modifier to prevent outside code.

All classes belong to a package. In Eclipse

and our screenshot, we see a default package with 3 classes: DataTypes, Print_Statements, and SystemDotSetProperty. By default, classes are placed in the default package when no package name or package statement is provided for that class.

When a package name is provided, that class is placed into a particular package. In this example, Building Blocks is the package while Search_T_SHIRTS, Search_T_SHIRTS_TestNG, and Search_T_SHIRTS_Highlight are the classes. The compiler does not allow 2 classes to have the same name in the same package. However, 2 classes can have the same name in different packages.

It is expected for packages to be written in all lowercase letters. To create a package, right click source which is src, select New, then Package. A warning or error message will show up if we start with anything other than a lowercase letter. Watch what happens when I enter a special character "Invalid package name". If I enter a dot, the message states "A package name cannot start or end with a dot". A warning shows up if we enter a capital letter "Discouraged package name. By convention, package

names usually start with a lowercase letter". Notice it did not show an error but a warning that stated it is discouraged.

Why is it discouraged? One of the reasons why It is discouraged because all lowercase letters for a package name separates packages from a class, a method, and a variable. By convention, a class, method, and variable can have uppercase as well as lowercase letters.

Now, let's dive deeper into the naming convention for a package. Many organizations use their internet domain for their package name but in reverse order. For example, I have a domain called selenium4beginners.com. Currently, there's no articles or videos on the site but it's available as a domain. As a package name in reverse order, I would write com dot then selenium4beginners.

The same is true for testng.org. If we look at class Search_T_SHIRTS_TestNG, the import statements start with org.testng. That is the reverse domain for testng.org. Go to a browser and type testng.org. Here is the website for TestNG and the book I suggested at the end of video (Part 2) Building Blocks For Selenium.

We can add related sub packages and classes under com.selenium4beginners. I am going to add 3 sub packages: java, selenium_testng, and utility. Although, there will be sub packages, a class is still allowed to be added under package and of course the sub packages.

A sub package is created by using a dot. Right click on the package, select New, then Package. At the end of the package name, add a dot, then write java. Let's add 2 more sub packages. Go to new package. Replace java with selenium_testng. One more time. Replace the existing sub package name with the new sub package name: utility. Now there are 3 sub packages under com.selenium4beginners.

It does not look like 3 sub packages due to the Package Presentation. Currently, the presentation is flat but we will change it to hierarchical. Select the drop down arrow, Package Presentation, hierarchical. Personally, I like this presentation better than flat.

We will add classes only related to Java under the java package. Classes that involve Java, Selenium WebDriver, and TestNG will be added to the selenium_testng package. Utility is a common package name which consists common functionalities. Let's consider the definition of utility. According to dictionary.com, it is the state or quality of being useful; something useful. Therefore, we can use the utility package for adding classes that we are going to reuse over and over.

For example, I prewrote a class called Highlighter which highlight web elements. It will highlight the web element before performing an action on the web element. Highlighting an element is very good for demos. I am going to show you how it works using Search_T_SHIRTS_Highlight class. First let's execute without highlighting the elements then execute highlighting the elements.

Run Search_T_SHIRTS_TestNG.

Now execute Search_T_SHIRTS_Highlight. We saw each element got highlighted before an action is performed on the element.

I am going to move the Highlighter class from one package to another package. We don't need 2 utility packages. Watch what happens, let's go to Search_T_SHIRTS_Highlight. The import statement shows utility.Highlighter: utility is the package and Highlighter is the class I am going to drag Highlighter to the other utility package. The Move dialog has a check box that states "Update references to Highlighter.java", click OK, and the import statement has been updated to show "com.selenium4beginners.utility.Highlighter". We see the Package, Sub Package, and Highlighter class.

Speaking of classes.

How To Name A Class

How To Name A Class. By convention, a class name is a noun that starts with an uppercase letter. If there are 2 or more words then each word starts with an uppercase letter. This format is called UpperCamelCase. Let's go to the java package, right click, select New, then Class. Classes can include uppercase and lowercase letters but it discouraged to start with a lowercase letter. If I type lowercase c, the message states "Type name is discouraged. By convention, Java type names usually start with an uppercase letter".

Classes can also include numbers and underscores. I have seen underscores in a class name but I am not sure if I have seen numbers in a class name. However, an error shows up if we start with a number but an error will not show up if we start with a hyphen. I have not seen anyone start a class with a hyphen although it's valid. If I type a hyphen, no warning or error shows up but when I type a number, (1) an error states "Type name is not valid".

Someone may ask or wonder, why does the warning and error messages keep saying Type name. Type name is not valid; Java type names usually start with an uppercase letter. The messages continue to show Type name because classes are Data Types.

Remember this slide from Building Blocks For Selenium, it showed 2 Data Types: Primitive which has many names such as Non-Object and Class which also has many names but I prefer Object. This slide shows Class is a Data Type. That's one of the reasons why the messages show Type name. A class definition always creates a new data type. In our example, we are going to define ConventionalNames as a class.

Next, we have methods which are written inside of a class.

How To Name A Method

Before speaking about conventional names for a method, let's discuss how to declare a method. A method is declared with 6 components: Access Modifiers, Return Type, Method Name, Parameter List, Exception List, and Method Body. We will bypass Exception List.

First, we have access modifiers which determine how a method is accessed. protected, private, and public are the modifiers. private and public are the most used modifiers. Private means the method can

only be accessed by members within the same class. Public means the method can be accessed by any class from any package.

Second is the return type. Return type dictates the type of data returned by the method.

Remember this slide, it shows the most used Data Types. If we have a return type of String then the method expects a return data type value of String. The same for the other Data Types such as int, double, boolean, and any class data type we create.

However if the method does not return a value then we write void. Void means no return value is expected from this method.

Next is the method name. By convention, method names follow a lowerCamelCase syntax where the first word is lowercase then each subsequent word is uppercase. Most of the times, method names are verbs or verb phrases. A verb phrase consists of a verb and its dependents such as an object or complements. We are going to name our method demoConventionalNames.

All methods have a parenthesis after its name. The method's name and parenthesis are called the method's signature. Parameters are included inside the parenthesis if arguments are passed to the method. A parameter is the data type and variable which we will talk about next in How To Name A Variable. Parameter list is one or more data types and variables separated by a comma. The parameter list remains empty if there are no parameters.

A Method's Body starts with an opening and closing bracket. The bracket can open on the same line after the parenthesis. However, I prefer to open my bracket on the next line. Personally, the body looks neater and more organized when it's opened on the next line. The body contains one or more statements. Code that performs an action is considered a statement. Also, variables are considered statements.

How To Name A Variable

Variables are declared with a Data Type and name. The name can be an unlimited number of characters. Nevertheless, there are some rules and conventions for naming a variable. 3 of the rules are:

1. 1 - it can contain letters, numbers, dollar sign ($), and underscore (_)
2. 2 - it can begin with a letter, dollar sign ($), or underscore (_)
3. 3 - it cannot contain a space or special character except dollar sign ($) or underscore (_)

3 of the conventions are:

1. Number 1 - Construct a descriptive name that describe the variable's purpose
2. Number 2 - Compose names utilizing the lowerCamelCase syntax. All letters are lowercase if there is only one word.
3. Number 3 – Create a name that begins with a letter

Let's go back to Eclipse. We write the Data Type then the variable name. In this case, our Data Type will be int and variable name will be numberOfYears. The goal is for variable name to be descriptive so our teammates can understand. We want to make it easier for ourselves and other people to read our code. The variable name can be written as nYr but why not add a few more letters and write numberOfYears?

Perhaps, an automation engineer wants to abbreviate the variable. That's fine too. It's better than writing nYr. We can write String custName rather than writing the entire words: customerName. If multiple variables have the same data type, feel free to write the new variable on a separate line or on the same line separated by a comma.

In closing, I failed to mention Java is a case sensitive language somewhat like passwords. An error will appear if we mistakenly write a lowercase letter instead of an uppercase letter or vice versa. For example, if we plan to use "numberOfYears" in another part of our program. The compiler will display an error if we write an uppercase N for number or lowercase y for Years.

In spite of that, these are the conventions for packages, classes, methods, and variables. We are not forced to follow the conventions. However, we are forced to make sure our names are not reserved words. A reserved word has a predefined meaning and set aside for special use. A few reserved words in our program are bold faced and written in purple: package, public, class, void, and int.

Here's a list of the 50 words reserved for Java. I highlighted the same 5 words from our program: class, int, package, public, and void.

A summary of our conventions discussion is located in this table. For packages, all of our letters are written in lowercase. Classes are usually nouns written in a CamelCase syntax. Methods use verb phrases with a lowerCamelCase syntax. Last but not least, we have variables that are descriptive and also written in a lowerCamelCase syntax.

Questions, Answers, and Practice.

Questions.

#1. How do you create a sub package?

#2. Can a class name include numbers?

#3. What is the convention for a Method?

Answers.

#1. How do you create a sub package? A sub package is created using a dot after the package name.

#2. Can a class name include numbers? Yes, a class name can include numbers but it cannot start with a number.

#3. What is the convention for a Method? The convention for a method is to follow a lowerCamelCase syntax by way of a verb phrase.

Practice

If you have time, watch this video Conventional Naming In Java at least one more time.

Create some sub packages that include a utility sub package.

Read the code from Search_T_SHIRTS_Highlight and see if you understand how to highlight web elements. If you understand how to highlight web elements then highlight each element in your Test Script VerifyDemoWebShop_Email_

Download Documents

To download the PowerPoint slides and automation code which includes highlighting the web elements for Demo Web Shop, go to https://tinyurl.com/Conventional-Naming-In-Java.

Thank You for watching Conventional Naming In Java.