Selenium 4 Beginners

Rex Jones II

Java's Conditional Switch Statement Transcript

Table of Contents

Selenium 4 Beginners
Rex Jones II

Introduction

Slide 1 – Java's Conditional switch Statement

Hello Everybody, Welcome To Selenium 4 Beginners. My name is Rex Allen Jones II. We are going to discuss the switch Statement which is a Java Conditional Statement. I mentioned during video 15 Java's Conditional if Statement that Conditional Statements are part of Java's Flow Control Structures. You may hear some people say Flow Control Structures and some people say Flow Control Statements.

Either way it controls a program's flow. A switch statement controls a program's flow with a multiway branch. That branch enables a program to select one of several possibilities.

Slide 2 – Switch Statement

The syntax for switch Statement begins with keyword switch and a variable name. After the required opening bracket, there's a case value followed by a statement or block of statements then the break keyword. Keywords are also called reserved words. A switch Statement has multiple case values, multiple statements, and multiple break keywords. The blue dots indicate there can be many more case values. After the case values, the switch Statement has an optional default statement. For a lot of situations, the switch Statement is more efficient than the if Statement.

Slide 3 – Switch Statement

Starting from the top, we see variable name equals a value. For switch statements, Java does not allow the value to be a double or boolean primitive Data Type. The value must be an enumeration, String or primitive types byte, short, char, and int. Switch checks the variable's value. All we add is the variable's name within the parenthesis. After the variable's name, we have the opening bracket followed by the case values.

The case values verify if there is a possible match for the value. If there is a match, then our program executes the associated statement. The semi-colon completes the statement or block of statements. Next, is the break keyword. The break keyword is optional but I have never seen a switch Statement without a break keyword. All of the statements at and after the matching case value executes if break is not available.

For example, if case value 1 has a match and there are no break keywords, then case value 1, case value 2, case value 3, and the default statement will all execute. Therefore, the break keyword exits out of the program flow. Default is another optional keyword that should always be included in our switch Statement. The statement for default executes if there is no case value match. Then we have the closing bracket.

Now, let's dive into our demo of Switch statements.

Demo switch Statements

Let's imagine we have a track race. The contestants receive a medal if they come in 1st, 2nd, or 3rd. For our switch statement, we initialize the variable with a value: int medal = 3 which is 3rd place. Next, we write switch. Just like the if statement, we can choose to write our switch statement from scratch or use intellisense. I'm going to use intellisense by clicking CTRL + SPACE then select switch. We change key to medal. So far, we have switch and our variable which is medal. The purpose of switch is to test our variable by finding the value. Right now, we hard-coded our value to be 3. However, our application would send a variety of values. So, let's write those other values.

case 1: This means if the value of medal = 1 then it will execute sysout "1 = Gold Medal"

break will exit out of the program flow after printing 1 equals Gold Medal

We can add a case for as many options as we want. To save time, let's copy and paste then change case 1 to case 2. 2 = Silver Medal

case 2: if the value is 2 then our program prints "2 = Silver Medal"

Copy and Paste again
then change case 2 to case 3. 3 = Bronze Medal

case 3: if the value is 3 then our program prints "3 = Bronze Medal"

The last thing we have is default which catches invalid choices. If medal does not equal 1, 2, or 3 then our program flow executes

sysout "No Medal". We don't need case and a value for default because the program knows if the other values are false then execute the default statement.

Let's Run. 3 = Bronze Medal. Change 3 to 2.

Do you remember what happened with the if statements? Each condition was evaluated until a match was found. Look what happens with the switch statement. Place a break point at switch (medal) and Debug. We see medal has a value of 2 then Step Into and our program flow goes directly to the statement for case 2. It bypassed case 1. Switch does not evaluate every case. Step Into again and here's the output. 2 = Silver Medal. Now, we are at break which will exit the flow after execution. F5

Break is optional but let me show you what happens when we remove all breaks.

Change 2 to 1. Let's debug

Medal has a value of 1. F5 and execution goes directly to the statement for case 1. F5 again and 1 = Gold Medal. However, there is no break for case 1 so execution goes to the next statement for case 2. The same process continues until we execute the default statement. F5 and we see 2= Silver Medal. F5 and we see 3 = Bronze Medal. F5 and No Medal prints to the console. Let me add the breaks back. Break, Break, Break, and Break. That's it for the Switch Statement.

Download Documents

You can download the transcript, presentation and code at https://tinyurl.com/Java-Switch-Statement