Selenium 4 Beginners

Rex Jones II

Java's Conditional if Statement

Table of Contents

Selenium 4 Beginners

Rex Jones II

Introduction

Slide 1 – Java's Conditional if Statement

Hello Everybody, Welcome To Selenium 4 Beginners. My name is Rex Allen Jones II. We are going to discuss the if Statement which is a Java Conditional Statement. Conditional Statements are part of Java's Flow Control Structures also known as Flow Control Statements. A Flow Control Statement directs a program's flow. Regarding the if Statement, this flow executes a particular statement if the condition returns true or false.

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

Slide 2 – if Statement / 3 Types

Java provides 3 types of if Statements. First is the If Then Statement. Second is the If Then Else Statement. Third, is the If Then Else-If Statement.

Slide 3 – If Then

The syntax for If Then begins with if (condition). If is a keyword that always start the if Statement. The condition generates a boolean result which returns true or false. After the condition, is an optional opening curly bracket followed by a statement. This statement can be a block of statements that's only executed if the condition is true. A Semi-colon completes the statement. The second bracket is a closing bracket and only required if there is an opening bracket.

Slide 4 – If Then Else

Next is the syntax for If Then Else which contains an if (condition), statement, and optional brackets just like the previous Statement. However, this statement includes an else clause. After the else clause, is a statement that can be executed.

Slide 5 – If Then Else

We see an if keyword, condition, opening bracket, statement, semi-colon, and closing bracket. The same information that was already covered for If Then Statement except the second statement. Now, we get a better picture of the statements. If the condition, returns true then the first statement is executed but if the condition returns false then the second statement is executed. The else clause extends the if Statement when the condition returns false.

Slide 6 – If Then Else-If

The last if Statement is If Then Else-If. It looks like the previous Statement but there's some clauses before the else clause

We have a clause for else if followed by a condition. There's also a statement. Depending on our program, we may add more else if clauses with a condition and statement.

Slide 7 – If Then Else-If

We have an if Statement, condition, opening bracket, statements, semi-colon, and closing bracket. An else if statement extends the if statement. There are 3 descriptions that explains statement. The first

statement executes if the first condition returns true. The second and third statements execute if one of the else if conditions return true. All of the conditions are evaluated using a top down approach.

For example, a top down approach using this syntax evaluates the if condition. When the if condition returns a false result then the program flow evaluates the first else if condition. After the first else if condition is evaluated, the program flow evaluates the second else if condition. The same top down approach would continue if there were more else if conditions that returned a false result. Soon as a true condition is found, the statement associated with that true condition is executed and the remaining conditions are not evaluated.

If there are no true conditions then the else statement is automatically executed. The else clause extends the if Statement by operating like a default condition when all of the other conditions are evaluated and returned a false result.

Now, let's demo the if Statement.

Demo If Statements

If Then

Let's provide a test since the if Statement is used to make a decision. We build our test by assigning a value to a variable such as int age = 34; Start by writing if. We have the option of writing our if statement from scratch or use intellisense. You can click CTRL + SPACE to bring up intellisense. We see an if and if else statement. I am going to select if and you see the syntax is already designed for us. The condition will be our test where our program flow must make a decision. For our condition, let's go ahead and write if (age == 24).

Remember from the 9th video Java Variables and Operators, 1 equal symbol assigns a value like int age = 34 but 2 equal symbols perform our test to verify if age equals a certain value like 24. In this test, age is not equal to 24. Next, we have our statement which is also called a body: sysout What do we want our program to do? if age equals 24 then print "Age = 24".

Let's Run. We do not see an output printed to the console because the condition was not true. The statement for if (condition) only executes if the condition returns true. This statement is false. Let's copy, paste, then change 24 to 34. Let's run again. Now, we see an output because Age Is 34.

If Then Else

However, writing 2 if conditions is not efficient for our program. Let's implement an If Then Else Statement which is more efficient. Copy and Paste the code then replace if (age == 34) with an else clause. Run. We get the same result. Age Is 34

I prefer to use an opening and closing bracket although they are optional. Personally, I believe the brackets make our conditional statements look more organized. Let's remove the brackets and Run again. Age = 34.

This is how an If Then Else statement works. We write if and a condition to test our variable, if age equals 24 then print Age Is 24 but If it does not equal 24 then print Age Is 34.

There is a downside, to the If Then Else statement. It is more efficient than the previous statement: If Then but the downside is this program flow only have 2 options: if (condition) and else statement

If Then Else If

Next, is the If Then Else If statement which give our program many conditions. Let's use age as our variable again and write some conditions. int age = 65;

if (age >= 65) then sysout "You Are Eligible To Retire At " + age

else if (age >= 20) then sysout "You Are Not Eligible To Retire At " + age

else if (age >= 13 && age <= 19) then sysout "You Are A Teenager At " + age

else sysout "You Are Not Old Enough To Work At " + age

If you have not watched the Java Variables and Operators video, check it out where I explain operators: greater than or equal, the and operator, less than or equal to, and the concatenation operator. Let's run. You Are Eligible To Retire At 65.

Change 65 to 40. Run. You Are Not Eligible To Retire At 40.

Change 40 to 16. Run. You Are A Teenager At 16.

The if condition is the main test and will execute its statement if the condition returns true. The else if statements are additional conditions just in case the main test returns false. Else has a default statement just in case all of the other conditions return false.

Let's change 16 to 1, add a breakpoint, then debug.

F5 to Step Into our program, age equals 1. Do you see how our 1st condition is highlighted and getting ready to be evaluated? When I step into our program again, the 1st statement will get skipped and the next condition will get evaluated: F5 else if age is greater than 20.

F5 again – the next condition: else if age is greater than 13 and age is less than 19

F5 for the last time will finally execute the default statement: You Are Not Old Enough To Work At 1. F5

That's one of the downsides to the If Then Else If statement. All conditions are evaluated until a match is found. When a match is found then the program exits the flow. However, if a condition is not found then the default statement is executed.

That's it for all 3 if Statements: If Then, If Then Else, and If Then Else If.

Download Documents

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

Rex Jones II