14.3.4 Branching

Click one of the buttons to take you to that part of the video.

Branching 0:00-1:09

In this demonstration, we will be discussing branching, if-then-else statements along with case and test statements.

First let's look at a script that I've written. We'll go ahead and edit that. We'll call it branching. It's pretty straightforward script. We have the identifier letting the system know it is a script and then I've declared of variable. I've declared the variable 'TODAY', and I've just arbitrarily set it to 'Tuesday'.

Using if-then-else statements, what I did was, I wrote a few tests here to see and determine what day of the week it is. I did this for demonstration purposes, but it would be just as easy to write a routine that will extrapolate the date from the system date. But again, in this case, just for ease of demonstration. I set a variable, and we called it 'TODAY'.

The if Statement 1:10-2:07

My first statement is an if statement, and what I do is I compare the variable TODAY. That's what this dollar sign ($) means. The dollar sign indicates that I'm about to use a variable name. The variable name is 'TODAY' and I'm testing to see if 'TODAY' equals Monday. I could very easily do other tests as well. I can set other variables and test for any equality that may be. We then follow with a semicolon (;), and that's necessary as part of the scripting language, followed by the word then.

I'm testing using 'if '. This is my statement to be true or false. This is saying, "If 'TODAY' is equal to Monday (clearly it's not), then, if it were true, then we would do the next statement, 'echo Monday'." Well, it's not Monday so we can't do that.

The elif Construct 2:08-2:42

In Linux, in bash scripting, there's another command called 'elif'. What the 'elif' does, it's a combination of else and if. Basically, what it's saying here is 'elif'. In other words, I already know it's not Monday. So now I'm going to test something else. I'm going to say else, if, 'elif TODAY' is equal to Tuesday. If that's true then I go ahead and 'echo Tuesday'. So that's true. What happens is it skips all the rest of the statements.

The fi Contruct 2:43-3:45

In scripting, every if ends with a 'fi', a backwards 'if'. That's the way it's done in bash scripting. The 'if' statement checks for equality follows the 'then', if the statement is true. Otherwise, it follows the 'else'. 'else' is optional. 'elif' is optional. I can just do a test. I could have said, if 'TODAY' is equal to Monday, then 'echo Monday', 'fi', and that's all it would've done. Basically it wouldn't have done anything.

If you look through, I have the 'if', a variable, test for equality, test the string itself, 'Monday', and this is what I do if it's true, this is what I do if it's false. I just continue that. I do another test, test for equality, if it's true, I do this, if it's false, I do this. And I do that continuously throughout this test. So that's how 'if', 'then', and 'else' works.

The case Statement 3:46-5:04

To show something else, I changed a little bit. I'm using a 'case'. I've change the day, so 'TODAY' is equal to 'Sunday'. Now I'm doing a bunch of tests in one. The first thing is the word 'case', the 'case' statement defines multiple tests. The first thing I do is determine what it is I'm testing against. What is it I'm trying to find. Then the word 'in', and then I specify all of the different values that I want to test against. It's a value, followed by a closed parentheses, or parens.

Then, I have my commands. I can have one. I can have many, and they all end with the double semicolon (;;). Then, I can go on and test the next thing. This is often used when testing for a particular parameter, or particular keyword. We use cases quite a bit. Again, this last one here, I'm using the vertical pipe which in this case means 'or'. If 'TODAY' is equal to 'Saturday' or 'TODAY' is equal to 'Sunday', then I'm going to 'echo' the 'Weekend'.

Run the Script 5:05-5:46

Let's see how this runs. I'll go ahead and exit out, and I will go ahead and run branching. The first test shows us 'Tuesday'. The second shows us 'Weekend'. One more time we'll look at the script, and you'll see that at the beginning, 'TODAY' is defined as 'Tuesday'. After the if, else-if statements are are finished, I then change the value to 'TODAY' is 'Sunday'. If 'TODAY' is 'Saturday' or 'TODAY' is 'Sunday', then 'echo Weekend', and that's exactly what we saw.

The test Statement 5:47-6:45

There is third type of value, and that's called test. The test command is good for a single test, a single examination or validation, to determine if something is either true or false. Often it's used to determine whether a file or a directory exists, as one example. What's different between this and the 'if' statement and/or the 'case' statement, is we can really only test one thing at a time to make this make sense. What I've seen it used for is to test to see if a file exists. Let's do that. The keyword is 'test', then I do my equality. I test for the existence. I can check for existence, nonexistence, directory name, filename, zero-byte file. There's many, many tests that I can do, For this example, I'm just seeing if a file exists.

Double Ampersand and Double Vertical Bar 6:46-7:51

As we saw before, I created the 'branching' script. We'll just see if that branching script exists. Obviously, we know it does. The next part is a double ampersand (&&), and what the double ampersand (&&) means is the same thing is that then, in an if-then-else statement. The test is going to determine whether or not something is true. This keyword test is the command itself. What I'm testing is to see if the file 'branching' exists, so that's my test, '&& echo "File exists"'.

We need the else component. Well, what if the file does not exist. What if this test here is false. Then, over here we use a double vertical bar (||). That's the same thing as an else in the if-then-else statement. Anything following this means that the test was false.

Review and Run 7:52-9:02

The keyword is test, that's the command. I'm testing to see if the file 'branching' exists. If it does, if the result is true, then I'm echoing the 'File Exists'. If the result is false, then I'm echoing, 'No file'. Press Enter, and sure enough the file does exist.

Well, let's just test something just to show you how this works. So 'test', I'm going to check for the existence of 'testout'. We know that doesn't exist. Again, I'll do the exact same thing, 'echo "File exists"', else, 'echo "No file"'. This time, pressing enter, of course, since 'testout' does not exist, 'No file' is what is displayed. So again, command, This is what were testing for, either true or false. If true, do this, if false, do this.

Summary 9:03-9:13

In this demonstration, we showed you the if statement, and how that works with the 'if-then-else-elif', and compared that with the 'case' statement and showed you how to use the 'test' command.