So far in this module, we have learned about booleans and boolean expressions, and how we can write code that behaves differently depending upon the state of the program at any given time. In the last video, we learned how to write both one sided and two sided conditional statements.

Writing conditional statements makes it possible to write arbitrarily complex program logic. This is because the `if` block, or the `else` block can contain any number of different statements, including other conditionals. Consider that you want to write a program that determines if the number entered by a user is in the range between one and 100, inclusive.

Let's draw a flow chart for this. The flow chart starts as usual with start and then it goes to input. The input is the number that the user enters. Then we want to test this number to determine if it's in range. So we start by testing whether or not the number is greater than or equal to one.

The false part of this conditional is that the number is not greater than or equal to one, which means that it's too small. On the true side, we already know that the number is greater than or equal to one, so now we need to know if the number is less than or equal to 100.

If it is, that's the true side, then the number is in range. If it's not, that's the false side, then the number is too big. And then both of these conditionals will come together and go to the end. Notice that what we have is a conditional statement inside of another one.

Translating this to Python, we're going to write the code. So we're going to start the program like we start all programs by defining `main`. Inside of `main`, we need to input the number. And we do this like we have always done it with `int(input("Enter the number between one and 100"))`

Then we need to check if number is in range. Remember to do this, we had to test if the number was greater than or equal to one first. If it was, then we tested if the number was less than or equal to 100, which means that it's greater than one and less than 100.

So we would `print("in range")`. If the number is greater than one but not less than 100, we have the else, which prints "too big". Now, the `else` for the first `if` is that the number is not greater than or equal to one, and that's the else that would print too small.

```
1   def main():
2    number = int(input("Enter a number between 1 and 100"))
3    if number >= 1:
4     if number <= 100:
5      print("in range")
6     else:
7      print("too big")
8    else:
9     print("too small")
10
11  main()
12
```

Recall that we need to make sure that the indentation is correct so that Python correctly interprets the inner conditional statement to be part of the if block of the other conditional statement. As long as we are careful with indentation when we nest our conditional statements, then Python is perfectly okay with it.

Now if I run this program with the number 42, Python will first check if the number is greater than or equal to one, and it is. Then it will check if the number is less than or equal to 100. Which it is, so it will print "in range".

It then skips the inner `else` and the outer `else` too. It takes a little bit of practice to see. But oftentimes taking code and converting it back to a flow chart can help you see the flow better.

Okay, that's all for now. Thanks for watching Align Online.