

## CSE 110

Home

W1

W2

W3

W4

W5

W6

W7

## Learning Activity (1 of 2): If Statements

### Prepare

You can now write fairly involved programs using many different variables of different types. In this lesson, you will learn one of the most important concepts of program, and that is the ability for a program to make decisions, or for it to do different things depending on a users response.

For example, in the assignment from last week, it would have been nice if you could ask the user if they wanted to pay with cash or a credit card, or ask if they had any coupons, and if so, ask what the discount was, and then to have the program behave appropriately depending on how the user responded.

You see this in almost every program you use. If I click this button, it should take me to that page. If I type my phone number incorrectly, it should ask me again. If I have more lives remaining, the game should restart, otherwise, it should inform me that the game is over.

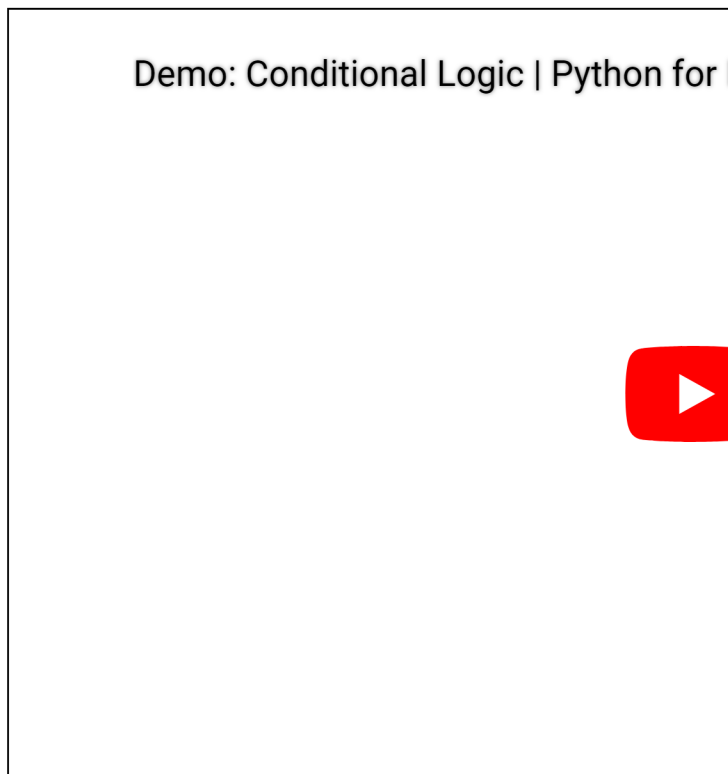
In all of these cases, the program is following the structure of "if ... then ..." and this is what you'll be learning in this lesson.

### Preparation Material

Watch the following videos:

- Direct Link: [Conditional Logic](#) (5 mins)

- Direct Link: [Demo: Conditional Logic](#) (5 mins)



- [Handling Multiple Conditions](#) (6 mins)

- [Demo: Multiple Conditions](#) (8 mins)

Demo: Multiple Conditions | Python for Beginners [22 of 44]



## The Basic Framework

As shown in these videos, the basic framework for "if statements," or conditional statements is this:

```
if condition:
    then-stuff-goes-here
```

If there are things that should happen if the condition is *not* true, then we can list other things in the "else" clause like this:

```
if condition:
    then-stuff-goes-here
else:
    the-otherwise-stuff-goes-here
```

There are a few things in the syntax that are really important. The first is the colon :, it says, there is a block of code that is coming that applies to either the "if" part or the "else" part.

The next thing that is important is the indentation of the block. It technically doesn't matter how many spaces or tabs you use, as long as it is consistent. It is very common to either use two or four spaces, but again the important thing is to be consistent.

In Visual Studio Code, by default when you press the "tab" key, it will indent your code four spaces, which is what you should use.

## The Condition

There are many options for the condition, but usually it has the form of **x == y** or **x > y** or something similar.

Don't forget that when you want to see if two items are equal **you must use two equals symbols ==** not just one. In most programming languages, including Python, one equals sign = is used to assign a value to a variable, whereas two equal signs == are used to check if two variables are equal.

## Comparing Strings

When comparing two strings, remember that case matters (upper case versus lower case). Because of this, it is common to convert strings that come from the user to lowercase before comparing them against a value you expect.

```
color = input("What is your favorite color? ")
```

```
# This if statement will only match "blue" not "Blue" or "BLUE"
if color == "blue":
    print("I like blue too.")

# This if statement will match the word blue, regardless of the capital
if color.lower() == "blue":
```

## Activity Instructions

Practice writing programs that compare strings and numbers.

### Comparing Numbers

Write a program that asks the user for two integers.

Then, write three separate **if/else** statements as follows:

- If the first number is greater than the second, print "The first number is greater". Otherwise, print "The first number is not greater".
- If the two numbers are equal print "The numbers are equal". Otherwise, print "The numbers are not equal".
- If the second number is greater than the first, print "The second number is greater". Otherwise, print "The second number is not greater".

### Comparing Strings

Have the program ask the user for their favorite animal. Then write an **if** statement as follows:

- Check to see if the user's favorite animal is the same as *your* favorite animal (meaning you, the programmer). If it is, print "That's my favorite animal too!" If it's not, print "That one is not my favorite." Verify that it works regardless of the capitalization.

#### Hint from Instructor:

When you write the program, you'll "hard code" your favorite animal into the program as the string to check against.

Hard coding just means that you write it directly into the program, sometimes called a *literal* value. Because this value doesn't come from a user or a file or anything, it will always be used, every time the program is run.

Here is an example of the program when it runs:

```
What is the first number? 4
What is the second number? 3
The first number is greater
The numbers are not equal
The second number is not greater

What is your favorite animal? giraffe
That one is not my favorite.
```

Another example:

```
What is the first number? 1009
What is the second number? 5250
The first number is not greater
The numbers are not equal
The second number is greater

What is your favorite animal? bear
That's my favorite animal too!
```

And finally, note that in this example, the animal matches, even though the case is different:

```
What is the first number? 42
What is the second number? 42
The first number is not greater
The numbers are equal
The second number is not greater

What is your favorite animal? BEAR
That's my favorite animal too!
```

## Sample Solution

When your program is finished, please view a sample solution of this program to compare your approach to that one.

You should work to complete this checkpoint program first, without looking at the sample solution. However, if you have worked on it for at least an hour and are still

having problems, you may feel free to use the sample solution to help you finish your program.

- [Sample solution](#)

## Testing Procedure

Verify that your program works correctly by following each step in this testing procedure:

1. Test the first part of your program with pairs of numbers where the first is greater and also where the second is greater. Verify that all three values that are printed are correct.
2. Test it with two numbers that are equal. Verify that all three values that are printed are correct.
3. Test the second part of your program with an animal that is different. Verify that the correct message is shown.
4. Test it with an animal that is the same. Verify that the correct message is shown.
5. Test it with an animal that is the same, but using different capitalization. Verify that it still matches, even with different capitalization.

## Submission

When you have completed all of the learning activities for this week, you will return to Canvas and submit the associated quiz there.

## Up Next

- Learning Activity (2 of 2): [Complex Conditions](#)

Other Links:

- Return to: [Week Overview](#) | [Course Home](#)