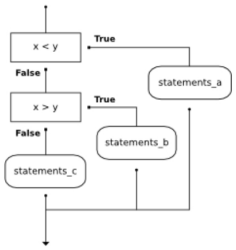
# 8.9. Chained conditionals

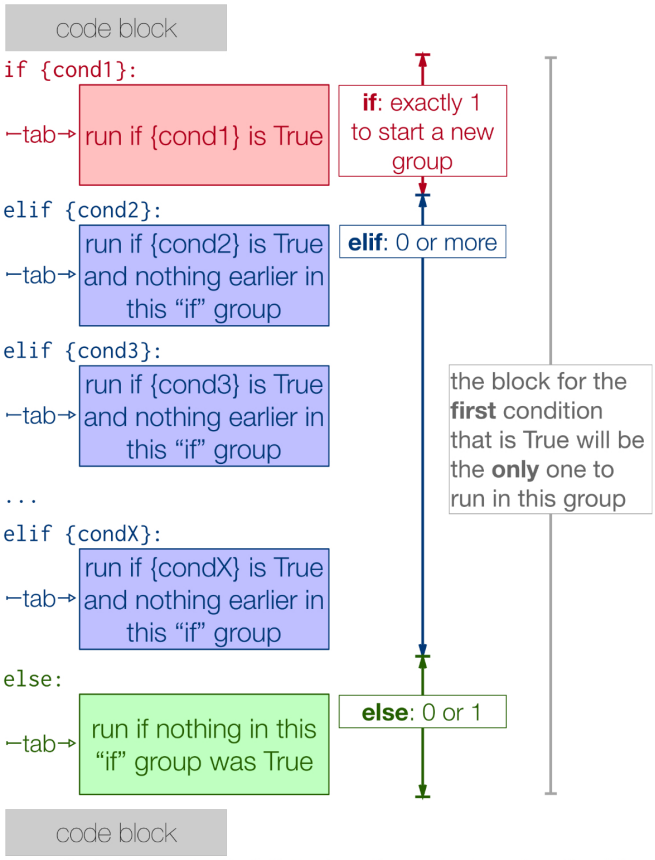
Python provides an alternative way to write nested selection such as the one shown in the previous section. This is sometimes referred to as a **chained conditional**.

```
if x < y:
    print("x is less than y")
elif x > y:
    print("x is greater than y")
else:
    print("x and y must be equal")
```

The flow of control can be drawn in a different orientation but the resulting pattern is identical to the one shown above.




`elif` is an abbreviation of `else if`. Again, exactly one branch will be executed. There is no limit of the number of `elif` statements but only a single (and optional) final `else` statement is allowed and it must be the last branch in the statement.




Each condition is checked in order. If the first is false, the next is checked, and so on. If one of them is true, the corresponding branch executes, and the statement ends. Even if more than one condition is true, only the first true branch executes.

Here is the same program using `elif`.

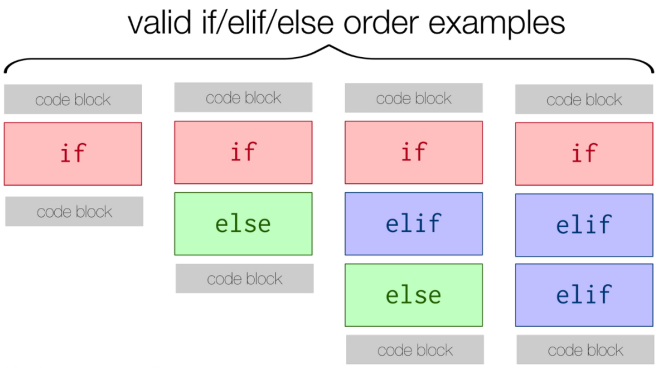
Save & Run 5/9/2021, 11:43:33 AM - 2 of 2 Show in CodeLens

```
x = 10
y = 10

if x < y:
    print("x is less than y")
elif x > y:
    print("x is greater than y")
else:
    print("x and y must be equal")

```

```
x and y must be equal
```

Activity: 1 -- ActiveCode (ac7_9_1)

The following image highlights different kinds of valid conditionals that can be used. Though there are other versions of conditionals that Python can understand (imagine an if statement with twenty elif statements), those other versions must follow the same order as seen below.




**Check your understanding**

condition-9-1: Which of I, II, and III below gives the same result as the following nested if?

```
# nested if-else statement
x = -10
if x < 0:
    print("The negative number ",  x, " is not valid here.")
else:
    if x > 0:
        print(x, " is a positive number")
    else:
        print(x, " is 0")
```

```
I.

if x < 0:
    print("The negative number ",  x, " is not valid here.")
else (x > 0):
    print(x, " is a positive number")
else:
    print(x, " is 0")
```

```
II.

if x < 0:
    print("The negative number ",  x, " is not valid here.")
elif (x > 0):
    print(x, " is a positive number")
else:
    print(x, " is 0")
```

```
III.

if x < 0:
    print("The negative number ",  x, " is not valid here.")
if (x > 0):
    print(x, " is a positive number")
else:
    print(x, " is 0")
```

- A. I only
- B. II only
- C. III only
- D. II and III
- E. I, II, and III

Check me | Compare me

✔ Yes, II will give the same result.

Activity: 2 -- Multiple Choice (question7_9_1)

condition-9-2: What will the following code print if x = 3, y = 5, and z = 2?

```
if x < y and x < z:
    print("a")
elif y < x and y < z:
    print("b")
else:
    print("c")
```

- A. a
- B. b
- C. c

Check me | Compare me

✔ Since the first two Boolean expressions are false the else will be executed.

Activity: 3 -- Multiple Choice (question7_9_2)

Create one conditional to find whether "false" is in string `str1`. If so, assign variable `output` the string "False. You aren't you?". Check to see if "true" is in string `str1` and if it is then assign "True! You are you!" to the variable `output`. If neither are in `str1`, assign "Neither true nor false!" to `output`.

Save & Run | 5/13/2021, 5:47:43 PM - 4 of 4 | Show in CodeLens

```
str1 = "Today you are you! That is truer than true! There is no one alive who is yo
if "false" in str1:
    output = "False. You aren't you?"
elif "true" in str1:
    output = "True! You are you!"
else:
    output = "Neither true nor false!"

```

Activity: 4 -- ActiveCode (ac7_9_2)

| Result | Actual Value | Expected Value | Notes | |
|---|---|---|---|---|
| Pass | 'True!... you!' | 'True!... you!' | Testing that output has the correct value, given the str1 provided. | Expand Differences |
| Pass | 'else' | 'str1 ...se!"\n' | Testing output (Don't worry about actual and expected values). | Expand Differences |
| Pass | 'elif' | 'str1 ...se!"\n' | Testing output (Don't worry about actual and expected values). | Expand Differences |

You passed: 100.0% of the tests

Create an empty list called `resps`. Using the list `percent_rain`, for each percent, if it is above 90, add the string 'Bring an umbrella.' to `resps`, otherwise if it is above 80, add the string 'Good for the flowers?' to `resps`, otherwise if it is above 50, add the string 'Watch out for clouds!' to `resps`, otherwise, add the string 'Nice day!' to `resps`. Note: if you're sure you've got the problem right but it doesn't pass, then check that you've matched up the strings exactly.

Save & Run | 5/13/2021, 5:49:39 PM - 5 of 5 | Show in CodeLens

```
percent_rain = [94.3, 45, 100, 78, 16, 5.3, 79, 86]
resps = []
for i in percent_rain:
    if i > 90:
        resps.append("Bring an umbrella.")
    elif i > 80:
        resps.append("Good for the flowers?")
    elif i > 50:
        resps.append("Watch out for clouds!")
    else:
        resps.append("Nice day!")

```

Activity: 5 -- ActiveCode (ac7_9_4)

| Result | Actual Value | Expected Value | Notes | |
|---|---|---|---|---|
| Pass | ['Bri...rs?'] | ['Bri...rs?'] | Testing the value of resps | Expand Differences |

You passed: 100.0% of the tests

We have created conditionals for you to use. Do not change the provided conditional statements. Find an integer value for `x` that will cause `output` to hold the values `True` and `None`. (Drawing diagrams or flow charts for yourself may help!)

Save & Run | 5/13/2021, 5:49:59 PM - 5 of 5 | Show in CodeLens

```
x = 65
output = []

if x > 63:
    output.append(True)
```

```
elif x > 55:
    output.append(False)
else:
    output.append("Neither")

if x > 67:
    output.append(True)
else:
    output.append(None)
```

Activity: 6 -- ActiveCode (ac7_9_5)

| Result | Actual Value | Expected Value | Notes |
|---|---|---|---|
| Pass | [True, None] | [True, None] | Testing that value of output is correct. |
| Pass | True | True | Testing that value of x is reasonable for this problem |

You passed: 100.0% of the tests

You have attempted 7 of 6 activities on this page

8.8. Nested conditionals">

ted conditionals">

8.10. The Accumulator Pattern with Conditionals">

✔ Completed. Well Done!



| Back to top