Chapter 3 - Conditionals

Conditionals

Lab - Conditionals

Lab - If Statement

Tutorial Lab 1: If Statements

The if statement allows for your program to make a decision about what it should do. It asks a simple question, is this condition true? If yes, then the computer will execute certain commands.

```
x = 5

if x < 10:
    print("Less than 10")
```

Code Visualizer

An if statement is comprised of the keyword `if`, followed by a boolean expression, and a `:`. Any code that should run if the boolean expression is true must be indented. In Python, programmers indent four spaces.

If the boolean expression is false, the indented code is skipped, and the program continues as normal.

```
x = 20

if x < 10:
    print("Less than 10")

print("And the program continues...")
```

Code Visualizer

Lab - If Else Statement

Tutorial Lab 2: If Else Statements

The if else statement gives your program the ability to ask a question, perform actions if the answer is true, and perform another set of actions if the answer is false.

```
x = 10

if x > 50:
    print(str(x) + " is greater than 50")
else:
    print(str(x) + " is less than 50")
```

Code Visualizer

The if part of the if else statement is written as before. The `else` keyword is **not** indented; it should be aligned with the `if` keyword. `else` is followed by a `:`. You do not use a boolean expression with `else`. All code that should run when the boolean expression is false must be indented four spaces.

Be careful when expression your boolean expression in terms of "less than or greater than". This does not take into account when the values being compared are equal. Consider the code from above, but with `x` having the value of `50`.

```
x = 50

if x > 50:
    print(str(x) + " is greater than 50")
else:
    print(str(x) + " is less than 50")
```

Code Visualizer

The output of the program does not make sense. 50 is not less than 50. Sometimes using <= and >= need to be used. Be sure to think through all of the possible outcomes, and make sure your code can function properly in all of those scenarios.

Lab - Compound Conditional

Tutorial Lab 3: Compound Conditionals

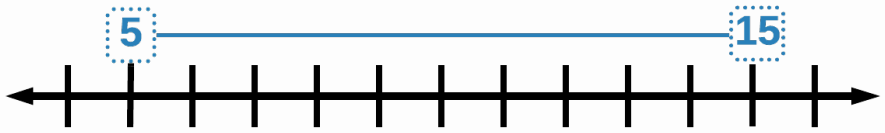
Sometimes you need to have two more things to be true before specific code can run. These are called compound conditionals because there are multiple boolean expressions in the conditional statement. The boolean expressions need to be linked together with either the `and` or `or` statements.

```
x = 10

if x > 5 and x < 15:
    print(str(x) + " is between 5 and 15")
else:
    print(str(x) + " is not between 5 and 15")
```

Code Visualizer

The code above will be true for any number that is greater than 5 and less than 15. If `x` has the value of `5` or `15`, then the boolean expression will be false.



Compound Conditional And

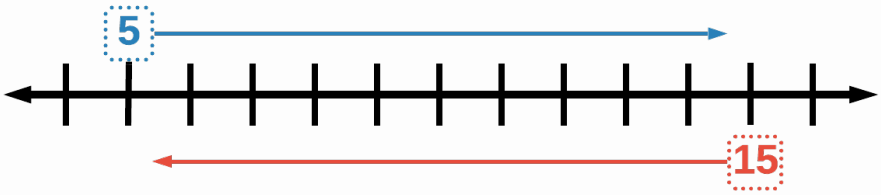
If you take the same code and change `and` to `or`, you get a very different program.

```
x = 10

if x > 5 or x < 15:
    print(str(x) + " is between 5 and 15")
else:
    print(str(x) + " is not between 5 and 15")
```

Code Visualizer

It does not matter what value you give to `x`, the boolean expression will always be true. The boolean expression is true even if `x` is `5` or `15`.



Compound Conditional Or

Lab - If Elif Else Statement

Tutorial Lab 4: If Elif Else Statements

If Elif Else statements allow you to write a series of questions about the value of a variable. This gives you more precision about the decisions you can make. Here are the rules for using if elif else statements:

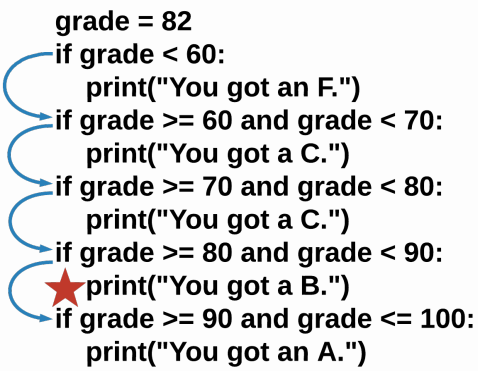
- Start with the traditional if statement
- Use the `elif` keyword followed by a conditional and a `:`
- Use as many `elif` statements as necessary
- End with the `else` keyword

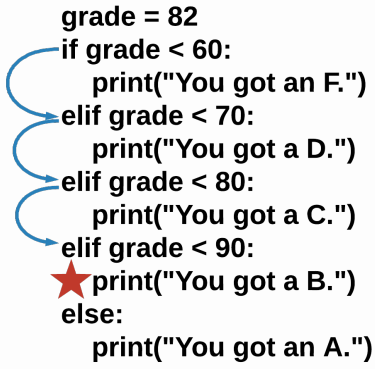
```
michelin_stars = 3

if michelin_stars == 1:
    print("This is a very good restaurant")
elif michelin_stars == 2:
    print("This is an excellent restaurant")
else:
    print("This restaurant is among the best in the world")
```

Code Visualizer

The if elif else statements can also be faster than a series of if statements. As soon as a boolean expression is true, Python stops computing the remaining boolean expressions. Python will execute all of the if statements, even if one of them is true.


grade = 82
if grade < 60:
print("You got an F.")
if grade >= 60 and grade < 70:
print("You got a C.")
if grade >= 70 and grade < 80:
print("You got a C.")
if grade >= 80 and grade < 90:
print("You got a B.")
if grade >= 90 and grade <= 100:
print("You got an A.")


grade = 82
if grade < 60:
print("You got an F.")
elif grade < 70:
print("You got a D.")
elif grade < 80:
print("You got a C.")
elif grade < 90:
print("You got a B.")
else:
print("You got an A.")

Efficiency

Lab Challenge - Month of the Year

Conditionals Challenge

Write a program that determines the month of the year based on the value of a variable called `month`. The variable will be a number from 1 to 12 (`1` is January, `2` is February, etc.). Use a `print` statement to write the month to the screen.

Important, you will need to declare the variable `month` as you write and test your code. However, **do not** submit your code to be graded with the variable declaration. The auto-grader will declare the variable for you.

Code Visualizer