



TASK

Boolean Data Type and Logical Operators

[Visit our website](#)

Introduction

WELCOME TO THE BOOLEAN DATA TYPE AND LOGICAL OPERATORS TASK!

Booleans are a very interesting data type as, unlike many of the other data types, they can only store two values, namely true and false. Learning about the Boolean data type is very important as it enables you to use and have a better understanding of operators.

An operator is a symbol that tells the compiler or interpreter to perform specific operations, whether it be mathematical, relational or logical, and produce a final result.

This task will introduce you to the different types of operators and show you how to use them.



Get in touch

Connect for support

Remember that with our courses, you're not alone! You can contact your mentor to get support on any aspect of your course.

The best way to get help is to login to www.hyperiondev.com/portal to start a chat with your mentor. You can also schedule a call or get support via email.

Your mentor is happy to offer you support that is tailored to your individual career or education needs. Do not hesitate to ask a question or for additional support!



WHAT ARE BOOLEANS?

A Boolean data type can only be one of two values, namely TRUE or FALSE. One byte is reserved for the Boolean data type.

You use Booleans when checking if one of two outcomes is true. For example: Is a student in a class present or absent? Is the car insured or not? Is the password correct or not?

Once the information is stored in a variable, it is easy to use loops and if statements to check an extensive sample of items, and base your calculations on the result of a Boolean value.

ASSIGNING BOOLEAN VARIABLES

Assigning a Boolean variable is very simple. You declare the variable name and then choose its starting value (initialise the Boolean). This value can then be changed as the program runs.

For example:

```
password = False  
password = True
```

BOOLEANS IN CONTROL STATEMENTS

Control statements allow you to use Booleans to their full potential. As of now, we only know how to declare a Boolean variable as either true or false, but how would this benefit us? How would we use it? This is where the if statement comes into play. Let's look at a simple decision we might make in our everyday lives.

When you are about to leave your house, do you always take an umbrella? No, you would only take an umbrella if it is raining outside. This is a very rudimentary example of decision making where there are only two outcomes.

We can apply these basic principles to create more complex programs.

```
umbrella = "Leave me at home."  
rain = False
```

```
if rain == True:
    umbrella = "Bring Me With"
```

Notice the double equal sign (==) above? You may be wondering what that is because so far, we have only encountered the single equal (=) sign to assign values to a variable. Let's find out more about them in the section below.

LOGICAL OPERATORS

Operators are symbols that tell the computer which mathematical calculations to perform or which comparisons to make.

Comparison Operators

As a programmer, it's important to not forget the basic logical commands. We use comparison operators to compare values or variables in programming. These operators work well with if statements and loops to control what goes on in our programs.

The four basic comparative operators are:

- greater than >
- less than <
- equal to ==
- not equal to !=

We can combine the 'greater than', 'less than' and 'not' operator with the 'equals' operator and form three new operations.

- greater than or equals to >=
- less than or equals to <=
- not equals to !=

Comparing Strings

```
my_name = "Tom"
if my_name == "Tom":
    print("I was looking for you")
```

Comparing Numbers

```
num1 = 10
num2 = 20

if num1 >= num2:          # The symbol for 'greater than or equal to' is >=
    print("It's not possible that 10 is bigger than or equal to 20.")

elif num1 <= num2:         # The symbol for 'less than or equal to' is <=
    print("10 is less than or equal to 20.")

elif num1 != num2:        # The symbol for 'not equal to' is !=
    print("This is also true since 10 isn't equal to 20, but the elif statement before comes
first and is true so Python will execute that!")

elif num1==num2 :         # The symbol for 'equal to' is ==
    print("Will never execute this print statement...")
```

The program will check the first part of the if statement (is num1 bigger than or equal to num2?).

If it is not, then it goes into the first 'elif' statement and checks if num1 is less than or equal to num2.

If it is not, then it goes into the next 'elif' statement, and so on.

'And' and 'Or' Operators

What if the program needs to test two or more conditions?

Let's take this real-life situation. When buying items at a store two criteria need to be met. The item needs to be in stock, and you need enough money to pay for the item. This is an example of a conjunction operation where both conditions need to be true for the whole statement to be true. This is called an 'and' operation.

If you were to receive a good mark at school, it's because you either very bright or studied hard. In this instance, either one of the options can be true, or even both can be true, but at least one needs to be true. This is a disjunction operation where at least one of the conditions needs to be met for the whole statement to be true. This is also called the 'or' operation.

Example of an AND condition:

```
team1_score = 3
team2_score = 2
game = "Over"
```

```
if (team1_score > team2_score) and (game == "Over"):
    print("congratulations you have won the match!")
```

Example of an OR condition:

```
speed = int(input("How many kilometres per hour are you travelling at?"))
belt = input("Are you wearing a safety belt?")

if (speed > 80) or (belt != "Yes"):
    print("Sorry Sir but I have to give you a traffic fine")
```

ARITHMETIC OPERATORS

We have covered the basic arithmetic operators in the lesson on Numerical Datatypes, but for a revision, here are the arithmetic operators used in Python:

- + (Addition) - Adds values on either side of the operator
- - (Subtraction) - Subtracts the value on the right of the operator from the value on the left
- * (Multiplication) - Multiplies values on either side of the operator
- / (Division) - Divides the value on the left of the operator by the value on the right
- % (Modulus) - Divides the value on the left of the operator by the value on the right and returns the remainder
- ** (Exponent) - Performs exponential calculation

Instructions

Before you get started, we strongly suggest you start using Notepad++ or IDLE to open all text files (.txt) and python files (.py). Do not use the normal Windows Notepad or Mac TextEdit as it will be much harder to read.

First, read *example.py* in this task folder by opening it with your IDLE or Notepad++ (right click the file and select 'Edit with Notepad++').

Compulsory Task 1

Follow these steps:

One of the most important patterns in computers and on the internet is your password. For a password to be classified as "Strong" the password needs to be structured in a certain way.

Password Strength is determined by:

- The length of the password (at least 6 characters) (**have_length**)
 - Needs to contain uppercase letters (**up_case**)
 - Needs to contain lowercase letters (**low_case**)
 - Needs to contain numbers (**have_num**)
-
- Declare Boolean variables for each one of these characteristics.
 - You will find the name of the variable next to the condition above. They must all be initialised as false.
 - Then ask the user a series of yes or no questions for each variable, and change the Boolean variable to True based on their answer.
 - Once 3 of the characteristics are met (3 of the variables == True) then display a message saying this is a suitable password.

Completed the task(s)?

Ask your mentor to review your work!

[Review work](#)

Things to look out for:

1. Make sure that you have installed and setup all programs correctly. You have setup **Dropbox** correctly if you are reading this, but **Python or Notepad++** may not be installed correctly.
2. If you are not using Windows, please ask your mentor for alternative instructions.



Rate us

Share your thoughts

HyperionDev strives to provide internationally-excellent course content that helps you achieve your learning outcomes.

Think that the content of this task, or this course as a whole, can be improved or think we've done a good job?

[Click here](#) to share your thoughts anonymously.

