Task: Beginner Control Structures - else Statements

www.hyperiondev.com

Introduction

Welcome to The Beginner Control Structures - else Statement Task!

In this task you will learn about a program's flow control. A control structure is a block of code that analyses variables and chooses a direction in which to go based on given parameters. In essence, it is a decision-making process in computing that determines how a computer responds when given certain conditions and/or parameters.

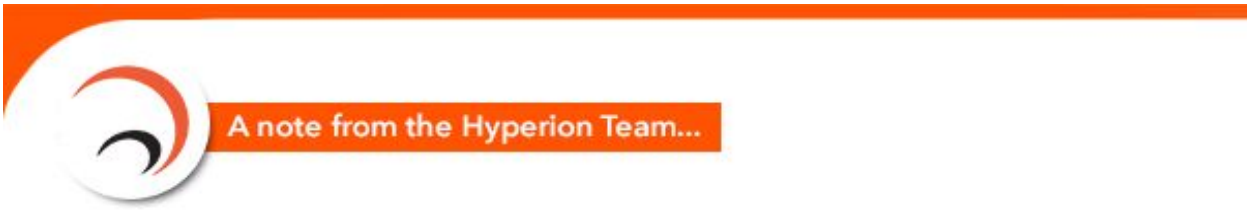
Connect with your mentor

CONNECT

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/support** 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!



Have you ever wondered what's possible as a software developer? Do you see it just as programming for PC? In fact, that's quite far from the truth. Software development spans devices, platforms and form factors well beyond PC.

Here you can find one of the most popular articles on the Hyperion Hub about the 10 different types of software development.

-The Hyperion Team

else Statements:

If statements are one of the most important concepts in programming, but on their own they are a bit limited.

Imagine if you were hungry and you sent your friend to the shop to buy a chocolate. When they get to the shop they find no chocolates and just leave because you told them of no alternatives. They would have to keep coming back for instructions unless you provide them with an alternative. So instead of us having a million 'if' statements to test each scenario we can add an 'else' statement to give us a single alternative.

Take a look at the following example.

```
num = 10

if (num < 12):
        print("the variable num is lower than 12")
```

We are now going to expand on it with an else statement.

```
num = 10

if (num < 12):
        print("the variable num is lower than 12")
else:
        print("the variable num is greater than 12")
```

Now instead of nothing happening if the condition of the if statement is not met (num ends up being greater than or equal to 12), the else statement will be executed.

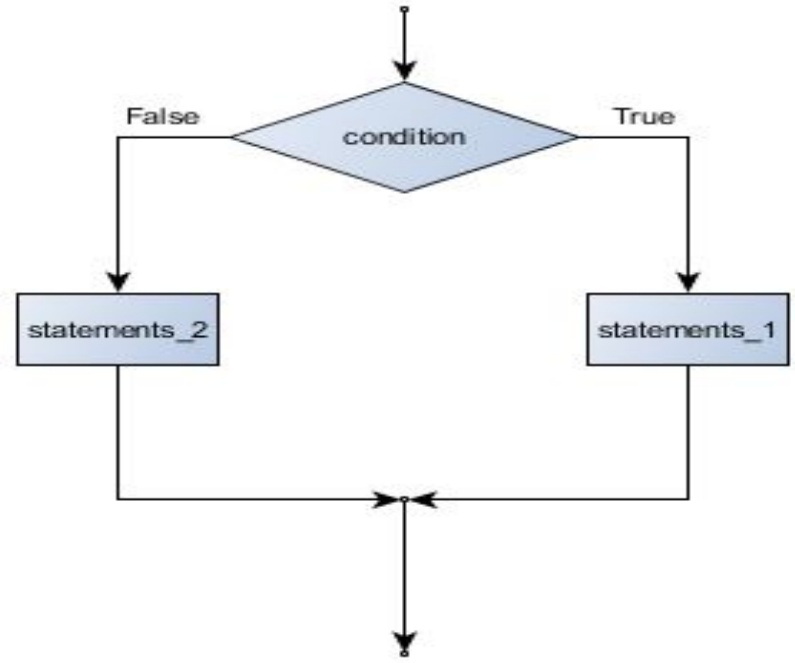
Another example of using an else statement with an if statement can be found below. Do you see that value the variable hour has, determines what string is assigned to greeting?

```
if hour < 18:
     greeting = "Good Morning";
else:
     greeting = "Good evening";
```

We are faced with decisions like this on an everyday basis. For instance, if it is cold outside you would likely wear a jacket, however, if it not cold you might not find a jacket necessary. This is a type of branching. If one condition is true, you do one thing and if the condition if false you do something else. This type of branching decision making can be implemented in Python programming using 'if else' statements.

The Structure of If Else Statements

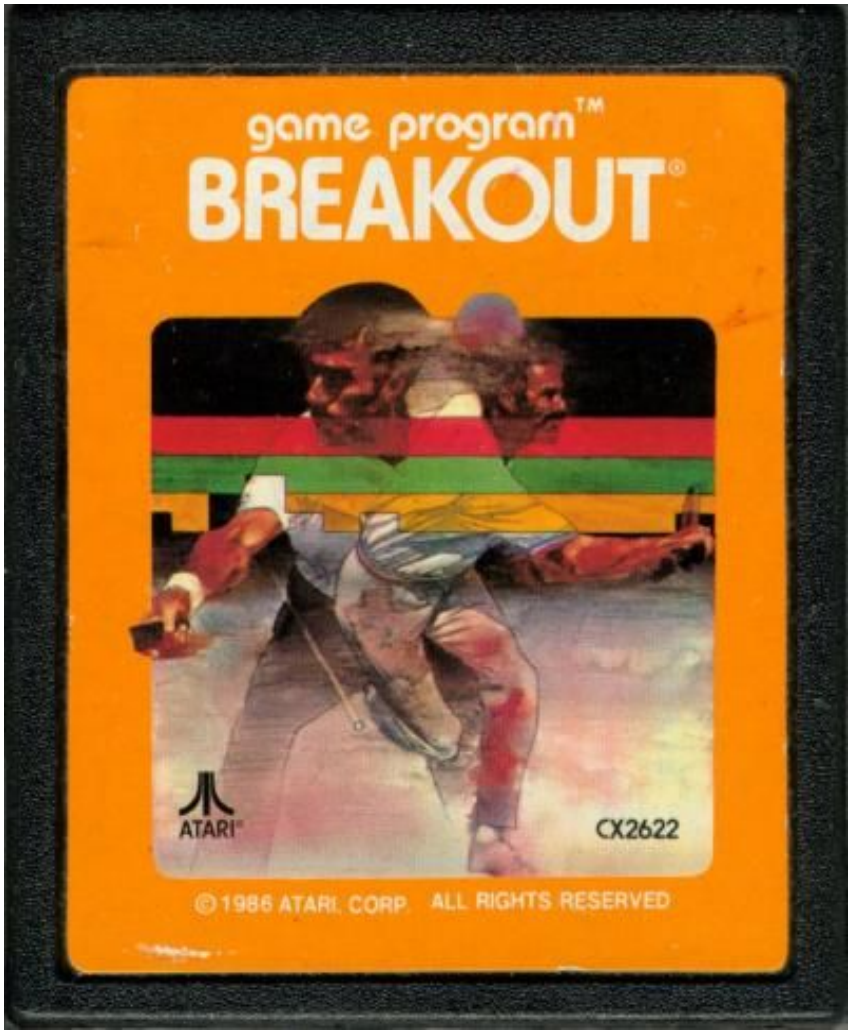
The basic structure of an **If Else** statement can be represented by this diagram:



It is mainly used in the case that you want one thing to happen when a condition is True, and something else to happen when it is False.

A note from Masood…

Sorry for the interruption, but did you know that many of the people who shaped our digital world started out by coding games for fun. For example, Steve Jobs and Steve Wozniak, the co-founders of Apple, began their coding careers as teenagers when they created the arcade game, Breakout.



Breakout (Arcade Game)

- **Masood Gool**

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 as it will be much harder to read.

First read example.py, open it using Notepad++ (Right click the file and select 'Edit with Notepad++') or IDLE.

- example.py should help you understand some simple Python. Every task will have example code to help you get started. Make sure you read all of example.py and try your best to understand.

- You may run example.py to see the output. Feel free to write and run your own example code before doing the Task to become more comfortable with Python.

- You are not required to read the entirety of Additional Reading.pdf, it is purely for extra reference.

Compulsory Task

Follow these steps:

- Create a Python file called “Courier.py” in this folder.
- You need to design a program for a courier company to calculate the cost of sending a parcel.
- Ask the user to enter the price of the package they would like to purchase.
- Ask the user to enter the total distance of the delivery in km’s.
- Now add on the delivery costs to get the final cost of the product.
- There are four categories to factor in when determining a parcel’s final cost each with two options based on their delivery preferences. (Use an **if else** statement based on the choice they make)
- Air R0.36 per km or freight R0.25 per km
- Full insurance R50.00 or limited insurance R25.00
- Gift R15.00 or no gift R0.00
- Priority R100.00 or standard delivery R20.00
- Work out the total cost of the package based on the selection in each category.

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.

Give your thoughts..

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