Hyperiondev

TASK

Beginner Control Structures - else Statements

Visit our website

Introduction

Welcome to The Beginner Control Structures if - else Task!

In this task, you will learn about a program's flow control. A control structure is a block of code that analyzes 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 to certain conditions.

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!

If Statements:

You are already familiar with if statements. In Python, an if statement looks like this:

```
num = 10

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

As you can see, the if statement is pretty limited.

else Statements:

If statements are one of the most important concepts in programming, but on their own they are a bit limited. The else statement represents an alternative path for the flow of logic if the condition of the if statement turns out to be False.

Imagine if you were hungry and you sent your friend to the shop to buy 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. Instead of us having a many 'if' statements to test each scenario, we can add an 'else' statement to give us a single alternative.

In Python, the general if-else syntax is:

```
if condition :
   indented Statements
else:
   indented Statements
```

If the condition turns out to be False, the statements in the indented block following the if statement is skipped and the statements in the indented block following the else statement are executed.

Take a look at the following example.

```
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 false (num ends up being greater than or equal to 12), the else statement will be executed.

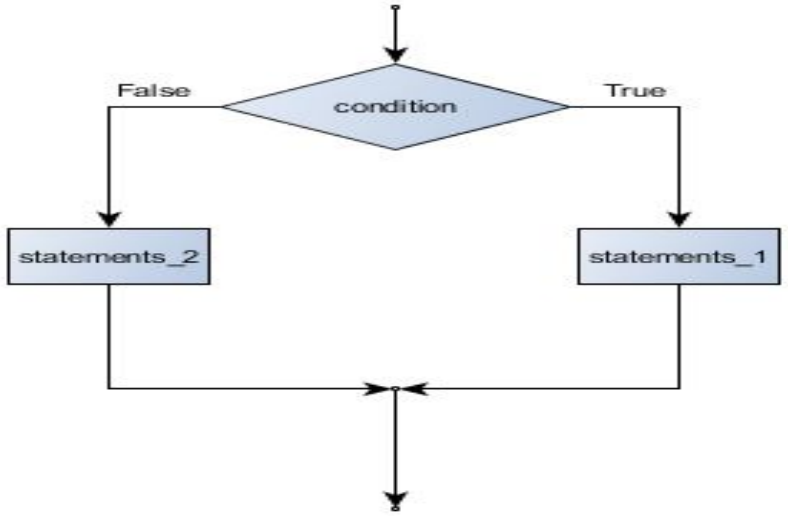
Another example of using an else statement with an if statement can be found below. The value that the variable hour holds, determines what string is assigned to greeting.

```
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, do one thing and if the condition is false, 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:

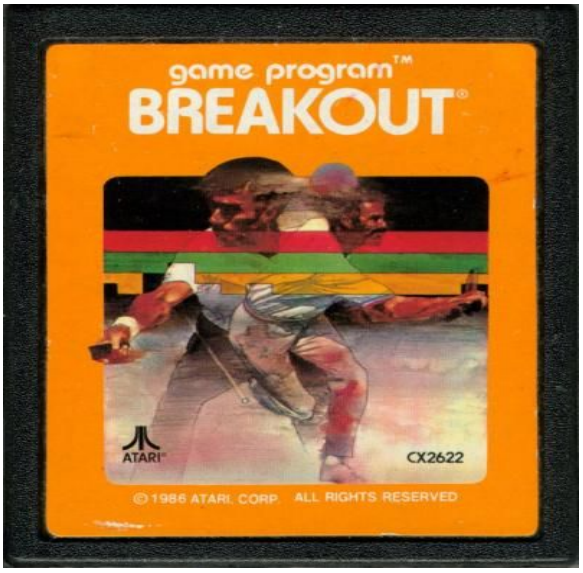


It is mainly used when you want one thing to happen when a condition is True, and something else to happen when it is False.

A note from our coding mentor

Sabir

Sorry for the interruption, but did you know that many of the people who shaped our digital world started 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)

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 IDLE (Right-click the file and select 'Edit with 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.

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.

Optional Bonus Task

Follow these steps:

- Create a Python file called “Optional_task.py” in this folder.
- Design a program for a department store to calculate the monthly wage for two different types of employees.
- Employees can either be a salesperson or a manager.
- Salespeople earn an 8% commission on their gross sales and a fixed salary of $2000.00 per month. Managers earn an hourly rate of $40.00.
- Determine if the user is a salesperson or a manager.
- Then, depending on their answer, calculate the monthly wage for the employee.
- If the user is a salesperson, ask for their gross sales for the month.
- If the user is a manager, ask for the number of hours worked for the month.
- Display the total monthly wage for the employee.

Thing(s) 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

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.