



Task: Control Structures

www.hyperiondev.com

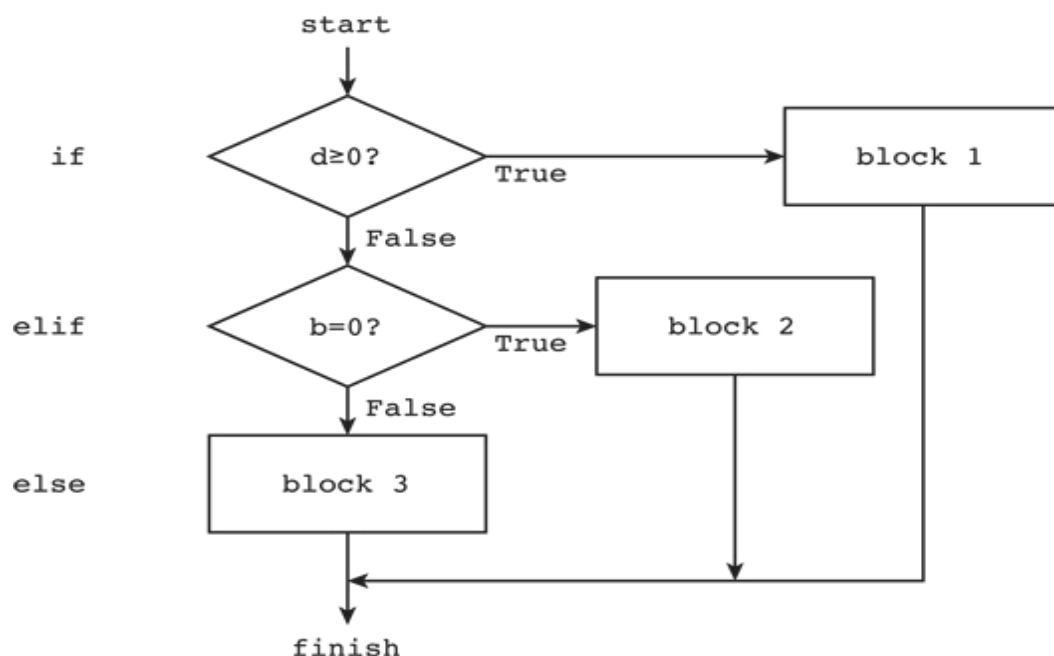
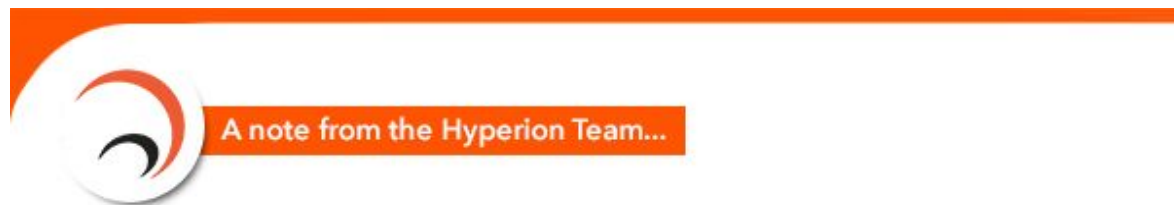
Introduction

Welcome to the Control Structures Task!

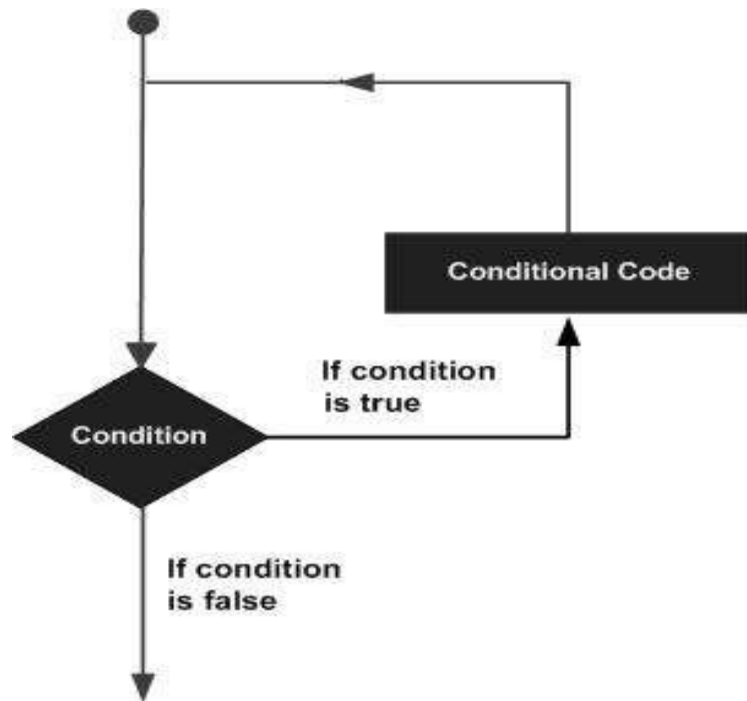
Overview:

In this Task you will learn about a program's flow control. A control structure is a block of programming code that analyzes variables and chooses which code statements to execute 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. Along with this you will also learn about type casting in Python and how to compare variables. See the instructions below.

-The Hyperion Team



When thinking about control structures it is often useful to draw a picture(similar to the one above) to represent the logic behind how these structures manage code execution. You can also use a decision tree to model the decision making process. Below is pictorial representation of how a 'loop' control structure (way of repeating code until a specific condition is met) works.



Another useful exercise to try, in the Python shell, is to display the data type of information with the `type(<information>)` function. For example:

```
>>> type('hyperion')
<type 'str'>
>>> type(123)
<type 'int'>
```

The same can be done with variables of course to test their data types. Try using variables and type casting functions like `int()`, `str()`, `float()` etc. along with the `type()` function to test out type casting.

- The Hyperion Team



Instructions

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

- **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. The instructions on how to do this are inside the file. Feel free to write and run your own example code before doing this Task to become more comfortable with Python.
- You are not required to read the **Additional Reading.pdf** in its entirety. It is purely for extra reference.

Compulsory Task

- Create a Python file called **Control.py** in this folder.
- Inside it, write a comment at the top of the program with your name. Then write code to take in a user's age using `raw_input` and store their age in an integer variable called `age`.
- Then check if the user's age is over 18. If the user is over 18, print out the message "You are old enough!" else if they are over 16 print "Almost there", otherwise print "You're just too young!". You should use one `if`, `elif` and `else` statement to do this.
- Below this, write Python code using one "for loop" and one "if statement" to print out all numbers from 0 that are less than the user's inputted age.

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 tutor for alternative instructions.

Still need help?

Just write your queries in your comments.txt file and your tutor will respond.

Task Statistics

Last update to task: 23/12/2015.

Course Content Manager: Riaz Moola.

Task Authors: Riaz Moola and Sinead Urisohn.

Task Feedback link: [Hyperion Development Feedback](#).