Task: Variables – Storing data in programs

www.hyperiondev.com

Introduction

Welcome to the Variables Task!

Now that you have completed your first Python program, this task will introduce you to the concept of variables and more complex programming problems which can be solved through the use of them. Variables are a computer programming term that is used to refer to the storage locations for data in a program. Each variable has a name which can be used to refer to some stored information known as a value. By completing this task you will gain an understanding of variables and how to declare and assign values to them, as well as the different types of variables and how to convert between types.

Connect with your mentor

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!

Now that you are a bit familiar with python and creating basic programs we would like to show you some stuff to help you on your journey to becoming a seasoned programmer. Creating excellent content requires good tools and equipment.

This applies equally well to programming. There are some great tools and resources to become aware of as soon as possible, if you have not already, to make the coding process just that much more convenient. Here is a link to the Hyperion Hub where you will find 5 essential utilities & resources for programmers.

-The Hyperion Team

What are Variables?

How are computer programs able to solve such complex problems? They don't just magically come up with the answers. There is a lot of background code that is carried out behind the scenes. This code that is carried out is made up of certain calculations and instructions. To be able to perform these calculations and instructions we need a place to store values in the computer's memory. This is where variables come in. A variable is a way to store information. It can be thought of as a type of "container" that holds information.

Variables in programming work the same as variables in maths. We use them in calculations to hold values that can be changed. In maths variables have names of letters like x and y; in programming variables can have any name you desire. It is best to name them something useful to the program or calculation you are working on, for example numLearners, which could contains the number of learners in a class or totalAmount, which could store the total value of a calculation.

In maths, variables only deal with numbers but in programming, we have many different types of variables and each variable type deals with a specific set of information.

Types of Variables

There are 5 major variable types that we will learn about. These are **Strings, Integers, Float, Char** and **Boolean**.

String: A string consists of a combination of characters. It can be used to store the surname, name, or address of a person for example.

Integer: An integer is a whole number or number without a decimal or fractional part. For example, it can be used to store the number of items you would like to purchase or the number of students in a class.

Float: We make use of a Float data type when working with numbers that contain decimals. For example, it can be used to store measurements or monetary amounts.

Char: Short for Character, Char is a single letter, number, punctuation mark or any other special character. It can be used for storing the gender of a pupil for example.

Boolean: Can only store one of two values, namely TRUE or FALSE.

The situation you are faced with will determine which variable you need to make use of. For example when dealing with money or mathematical calculations you would likely use **integers** or **floats.** When dealing with sentences or displaying instructions to the user we would make use of either **strings** or **chars**. When dealing with decisions that have only two possible outcomes we would use **booleans,** as the scenario would only either be True or False.

A note from Masood…

Hey there, have you heard about Alan Turing?

Alan Turing (1912 – 1954) was a British mathematician, logician and cryptographer. He is considered by many to be the father of modern computer science. He designed and built some of the earliest electronic, programmable, digital computers.

Father of modern day computing

During the Second World War, he was recruited by the military to head a classified mission at Bletchley Park. This mission was to crack the Nazi's Enigma machine code which was used to send secret military messages. Many historians believe that breaking the Enigma code was key to bringing the end of the war in Europe. Turing published a paper in 1936 that is now recognised as the foundation of computer science.

- **Masood Gool**, Online Trainer

Let's answer some of the most common questions when it comes to variables:

Q: Do variables store specific kinds of values?
A: Their content is specific. The content that a variable stores is the data and the type of data is intuitively called the datatype.

Q: Do we have to declare the datatype of a variable?
A: No, the datatype can be neglected when creating a variable. This is known as 'weak-typing'.

For example:
```
a_string_variable =  "ABC"
an_int_variable = 2
```

Q: How does Python know the difference between variables?
A: Python detects the type of a variable by reading how data is assigned to the variable.

Strings are detected by quotation marks " ".
Integers are detected by the lack of quotation marks and presence of digits or another accepted number format but no decimal.
Floats are detected by the presence of decimal point numbers.

So if you enter numbers then Python will automatically know you are using Integers or Floats. If you enter a sentence Python will automatically know it is storing a string.

Take heed that types can be converted from one to another, and you need to take care when setting a string with numerical information.

For example:

```python
numberStr = "10"
print numberStr
```

Watch out! Since you defined 10 within quotation marks, Python knows this is a string. It's not an integer even though we understand 10 is a number. Numbers can also be displayed as a string if you put them between quotation marks. Now, because 10 is declared as a string we will be unable to do any type of arithmetic calculations with it.

Casting

Putting the 10 in quotation marks will automatically convert it into a string but there is a more formal way to change between variable types. This is known as **casting** or **type conversion**.

Casting in Python is pretty simple to do. All you need to know is which data type you want it to convert to and then use the corresponding function.
- **str()**   - converts variable to a string
- **int()**   - converts variable to a integer
- **float()** - converts variable to a float

```python
number = 10
numberStr = str(number)
print str(number) + numberStr
```

This example converts numberStr into a string so that it can be easily printed without the need of quotation marks.

You can also convert the variable type entered via a raw_input. By default anything entered into a raw_input is a string and to convert it to a different data type we would simply need to use the desired function.

```python
noDays = int(raw_input("How many days did you work this month?"))
payPerDay = float(raw_input("How much is your pay per day?"))
salary = noDays * payPerDay
print "My salary for the month is R" + str(salary)
```

When writing programs, you'll have to decide what data types or variables you will need in order do what you want your program to do. In most cases, it will be pretty obvious – want to store someone's name? Use a String. Their age? Use an integer.

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 1

Follow these steps:

- Create a new Python file in this folder called "Variables.py"
- Inside it, write Python code declaring and assigning values for each variable type
- Print each variable on separate lines.

Compulsory Task 2

Follow these steps:

- Create a new Python file in this folder called "details.py"
- Use a Raw Input to get the following information from the user.
  - Name
  - Age
  - House number
  - Street name
- Print out a single sentence containing all the details of the user.
- For example:
  - This is John Smith he is 28 years old and lives at house number 42 on Hamilton Street.

Compulsory Task 3

Follow these steps:

- Create a new Python file in this folder called "conversion.py"
- Declare the following variables:
  - num1 = 99.23
  - num2 = 23
  - num3 = 150
  - string1 = "100"
- Convert the as follows:
  - num1 into a integer
  - num2 into a float
  - num3 into a String
  - string1 into an integer
- Print out all the variables on separate lines

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.

RATE