Python Programming

Tutorial 2 – Variables

UEE60411- Advanced Diploma in Computer Systems Engineering.

UEENEED111A - Develop, Implement and Test Object Oriented Code

For further information:
Leanne Matheson
ICT – Creative Business Enterprise
Ph: (03) 5225 0616
Email: lmatheson1@gordontafe.edu.au

Section 4 – Variables

Variable Types

Variables are containers that hold data which can be altered as the program executes.
Data may be in many forms. (words, single characters, whole or counting numbers [both positive and negative], decimal numbers [+ve and –ve], or simply true/false)
A variable will be declared as of a particular **data type**, depending upon the form of data that a variable will be used to hold.
Python has many **data types** of **variables**, but initially, we will only use **int**, **float** and **String**.

int Used to hold whole or counting numbers.
float Used to hold numbers which contain decimal points, including money.
String Used to hold text.

Text may be manipulated, but arithmetic cannot be performed on Strings
(Note: a **String** is not strictly a data type, it is a class, but we will initially treat it as one.)
All variables should to be declared before they can be used.
Example:

variable	Data type
`month`	`string`
`age`	`integer`
`bankBalance`	`Float`

Could be:

Variables are usually declared at the beginning of the block of code in which they will be used. Variables can also be initialised when they are declared. (ie: given a starting value)
Arithmetic can be performed on **integers** and **floats**, using normal mathematical rules and symbols.
Brackets can be used to change the normal order of operations.
Example:

```
age = 84
yearBorn = 0
yearBorn = 2016 – age
```

Variable naming conventions:

As a general rule:

- all variables should be given meaningful names, preferably nouns.
- variables should be made up of alphanumerics only
- Start with a letter which is **lower case**
- Where a variable name should consist of two or more words, capitalise the first character of the second and subsequent words.
- There should be **no spaces** in variable names

Examples:

`monthName, ageInYears, bankBalance, middleInitital, hasPaid`

2. Strings

A String is a class that contains text, and has text manipulation methods associated with it.
A String type variable to hold text must be declared before it can be used. Strings are written inside double quotes – “Happy Christmas”
Example:

String greeting;
```
greeting = “Good Evening”
```
or
```
String greeting = “Good Evening”;
```

1. Concatenation of Strings

Concatenation is the joining together of two or more Strings to form one larger String.

Use the **,** symbol to concatenate two strings.

Example:
```
address1 = “47 Masons Lane,”
address2 = “Moe”
print(address1, address2);
```
displays
```
47 Masons Lane, Moe
```

New Lines in Strings

The new line symbol is the **\n** – an escape character, signified by the **** preceding the **n**

Example:
```
Address1 = “47 Masons Lane, \nMoe”
address2 = “”
print (address1)
```
displays
```
47 Masons Lane
Moe
```

Example

Examine the sample file called **Sample01.py**

```
#   Sample01.py

        message1 = “ ”
        message2 = “Today is very cold and wet.”

        message1 = “Today is the first day of winter.”
        print(message1)
        print(message2)
```

2. Exercises

Open and examine the sample file called **Sample02.py**

Note the use of **\n** escape sequence to give a new line.

1. Modify **Sample02.py** to display your name on one line, and your address and suburb/town on two more lines.

Save this file as **Concat1.py**

2. Create a py file called **Concat2.py** that will display the names of the days of the week, one per line, under the heading "Days".

Under a second heading "Seasons", display the seasons of the year.

3. Create a py file called **Concat3.py**

Declare and initialise four Strings – friend1 to friend4

Display friend1 and friend 3 on one line, and the other two on the line below, using two `print ( ) 's`.

4. Create a py file called **Concat4.py** with three string variables: firstName, middleName and lastName. Initialise all three variables.

Using concatenation, display the names as follows (including the spaces and commas):

Robert William McPherson

Robert McPherson

McPherson, Robert William

Robert

William

McPherson

Exercises – String Variables

5. Open and examine the sample file called **Sample03.py** Note the following points:- Variables may be initialized on declaration or after being declared.

The message is created, then displayed as a separate variable.

6. Write a python program called **Var05.py**

This program should declares two String variables called season1 and season2. Initialise them with "Winter" and "Spring".

Display, using the variables, "Winter comes before Spring" on one line.

7. Modify **Var05.py** so the output is displayed on <u>three</u> lines:

Winter

comes before

Spring

Save this as **Var06.py.**

8. Modify **Var06.py**

Declare another String variable called **message**.

Use concatenation to assign the sentence about the seasons to message, then display message. Save this as **Var07.py.**

9. Create a python program called **Var8.py** that has three String variables – name, address and phone. Initialise these variables with your details.Create a fourth String variable called details. Use the name, address and phone variables to assign a string to details that places

your name, address and phone on three separate lines. Display details using `print()`.

Exercises – Numeric Variables

All displays are converted to Strings.

Examine the file called **Sample04.py** Note the following points:- line 15 – the calculation line 17 – putting the sum together to display meaningfully

10. Write a python program called **Var11.py** that declares three integer variables: length, width and area.

Initialize length with 57 and width with 19, and calculate the area of a house block.
Display the area only. (1083)

11. Modify **Var11.py** so the program displays

LENGTH nnn x WIDTH nnn = AREA nnnn
Save this as **Var12.py**

12. Modify **Var12.py** so the output displays over three lines

LENGTH = nnn metres
WIDTH = nnn metres
AREA = nnnnn sq metres

Save this as **Var13.py.**

13. Write a python program called **Var14.py.** It should declare three variables – unit price, number of items purchased, and total price. (create appropriate variable names)
Initialize the number of items and price, make the calculation, assigning the answer to the variable holding total price.
Display the total price only.

14. Modify **Var14.py** so the output displays

```
Unit Price $nn.nn
Number bought nn
Total Price $nnn.nn
```

Save this as **Var15.py.**

15. Write a python program called **Var16.py**.

Your program should declare and initialize variables to hold a name, a wage rate and the number of hours worked. It should calculate the weekly wages earned by multiplying the rate * the hours worked.
The output should appear as follows:

```
Wages for
Fred
Flintstone:
Hourly Rate:  $10
Hours Worked: 34
Weekly Wage: $340
```