**IS201 Fundamentals of Computing**

**HOP01 Variables, Simple data types, User Input, If-else, Loops**

10/03/2020 Reviewed by Kim Nguyen

Center for Information Assurance (CIAE) @City University of Seattle (CityU)

**Before You Start**

* The directory path shown in screenshots may be different from yours.
* Some steps are not explained in the tutorial**.** If you are not sure what to do:
  1. Consult the resources listed below.
  2. If you cannot solve the problem after a few tries, ask a TA for help.

**Learning Outcomes**

Students will be able to:

* Setup the working environment for Python
* Understand the basics python concepts
* Understand the data conversion in Python

**Resources**

Matthes, E. (2019). [Python Crash Course: A Hands-On, Project-Based Introduction to Programming, 2nd Edition](https://login.proxy.cityu.edu/sso/skillport?context=146803). No Starch Press. (ISBN 9781593279288)

**Preparation**

1. In Visual Studio Code, open the private repository generated when you accepted the HOP01 assignment (If you cannot find that repository in your machine, you might have not cloned the repo, if so, please do before proceeding).

A screenshot of a cell phone

Description automatically generated

Open the terminal from the VSCode by hitting the control + ~ key, navigate into Module 1 folder using the following command:

>>> cd Module 1

1. **Variables, Data Types and User Input**

**Follow the below video instruction to code along:**

<https://youtu.be/5Y9AENc6o2Y>

Variable: Variables are containers for storing data values. Unlike other programming languages, Python has no command for declaring a variable. A variable is created the moment you first assign a value to it.

More about variables: <https://www.w3schools.com/python/python_variables.asp>

Data Types: Data types are the classification or categorization of data items. Data types represent a kind of value which determines what operations can be performed on that data. Numeric, non-numeric and Boolean (true/false) data are the most used data types.

More about data types: <https://www.w3schools.com/python/python_datatypes.asp>

User Input: To receive information through the keyboard, Python uses the input() function

When the input() functions is called, the program flow stops until the user enters the input via the command line. To actually enter the data, the user needs to press the ENTER key after inputting their string. The input() function, by default, will convert all the information it receives into a string.

More about user input: <https://www.w3schools.com/python/python_user_input.asp>

**Multiple assignments**

Python allows you to assign values to multiple variables in one line andyou can assign the same value to multiple variables in one line. For example:

**A screenshot of a cell phone

Description automatically generated**

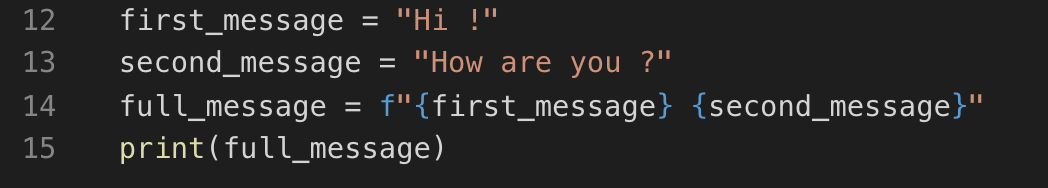
*(This is optional, you do not have to write the code in the above screenshot)*

1. **String functions and Concatenation:**

**Follow the below video instruction to code along:**

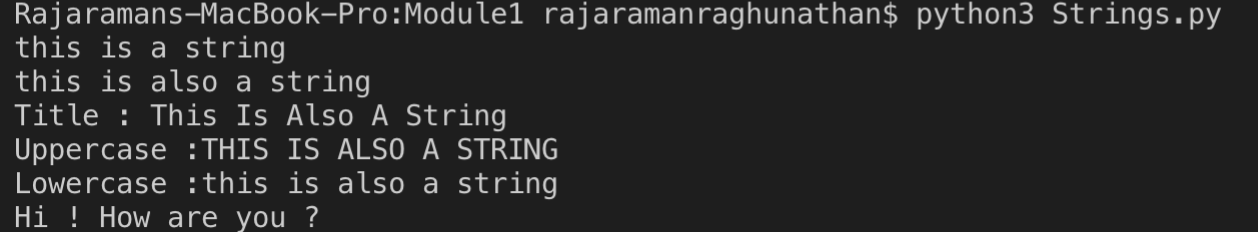
<https://youtu.be/9zW8GEogRKc>

In Python, there are a few ways to concatenate – or combine - strings. The new string that is created is referred to as a string object. Obviously, this is because everything in Python is an object – which is why Python is an objected-oriented language. Another way to concatenate strings are as below example:

****

*(This is optional, you do not have to write the code in the above screenshot)*

Output:

****

Explanation:

Also called “formatted string literals,” f-strings are string literals that have an f at the beginning and curly braces containing expressions that will be replaced with their values. The expressions are evaluated at runtime and then formatted using the \_\_format\_\_ protocol.

1. **Numbers and operators**

Operators are the constructs which can manipulate the value of operands.

Consider the expression 4 + 5 = 9. Here, 4 and 5 are called operands and + is called operator.

Python language supports the following types of operators.

* Arithmetic Operators
* Comparison (Relational) Operators
* Assignment Operators
* Logical Operators
* Bitwise Operators
* Membership Operators
* Identity Operators

The commonly used operators are underlined above.

More about numbers and operators: <https://www.tutorialspoint.com/python/python_basic_operators.htm>

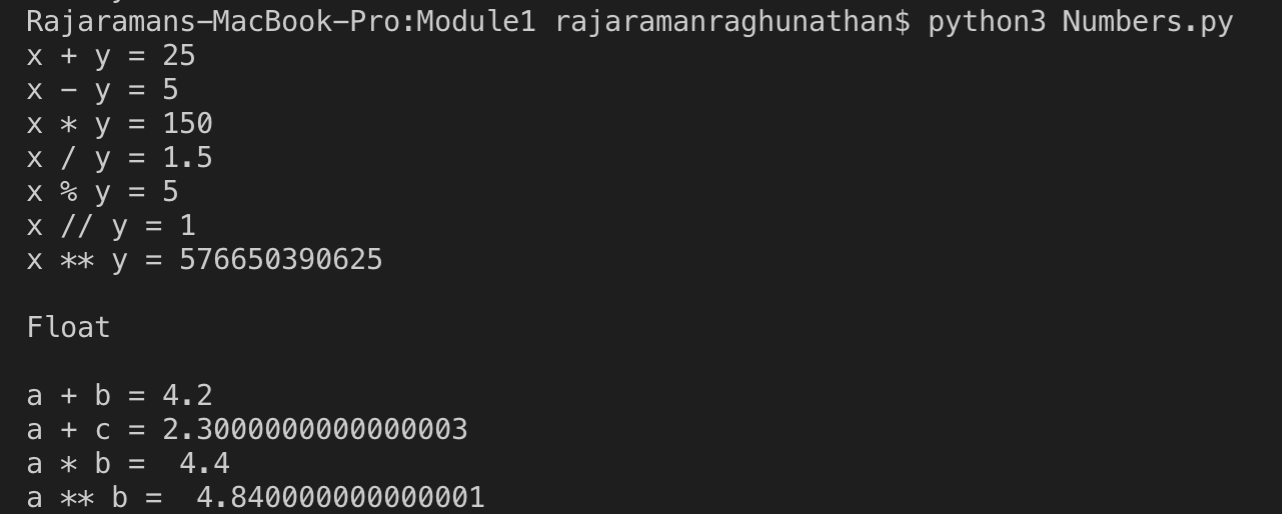
1) Under the module1, create a file **Numbers.py** and type the following code.

**A close up of a black background

Description automatically generated**

2) Type the following to get the output of the code in the terminal

>>> python3 Numbers.py

****

If you look at the answer of a+c and a\*\*b, it is displayed as an arbitrary number of decimal places. This happens in all languages and is of little concern. Python tries to find a way to represent the result as precisely as possible, which is sometimes difficult given how computers have to represent numbers internally.

1. **Python Decision making**

Decision making is anticipation of conditions occurring while execution of the program and specifying actions taken according to the conditions.

**Open the HOP01.py file, follow the below video instruction to code along:**

<https://youtu.be/4rIRG1aMEvM>

**Challenge:**

Fix the program so that the user input can be recognized as an integer. Expected result:

Text

Description automatically generated

**Elif:**

The elif keyword is pythons way of saying "if the previous conditions were not true, then try this condition". Create IfControl.py file and type the following code:

A screen shot of a social media post

Description automatically generated

1. Use the following command and give different input to test the program:

>>> python3 IfControl.py

A close up of a logo

Description automatically generated

A close up of a logo

Description automatically generated

A close up of a logo

Description automatically generated

More about If statements: <https://www.w3schools.com/python/python_conditions.asp>

**Loops**

In general, statements are executed sequentially: The first statement in a function is executed first, followed by the second, and so on. There may be a situation when you need to execute a block of code several number of times.

Programming languages provide various control structures that allow for more complicated execution paths.

A loop statement allows us to execute a statement or group of statements multiple times.

1. **While loop**

**Follow the below video instruction to code along:**

<https://youtu.be/_3665KzFlO8>

The while loop runs as long as, or while, a certain condition is true.

Syntax:

while expression:

statement(s)

Here, statement(s) may be a single statement or a block of statements. The condition may be any expression, and true is any non-zero value. The loop iterates while the condition is true.

When the condition becomes false, program control passes to the line immediately following the loop.

In Python, all the statements indented by the same number of character spaces after a programming construct are considered to be part of a single block of code. Python uses indentation as its method of grouping statements.

More about while loops: <https://www.w3schools.com/python/python_while_loops.asp>

**Challenge:** Fix the program so that the attempts of guessing is 3 times instead of 4.

1. **For loop**

*“for”* loops are traditionally used when you have a block of code which you want to repeat a fixed number of times. The Python *for* statement iterates over the members of a sequence in order, executing the block each time.

Syntax:

for iterating\_var in sequence:

statements(s)

If a sequence contains an expression list, it is evaluated first. Then, the first item in the sequence is assigned to the iterating variable iterating\_var. Next, the statements block is executed. Each item in the list is assigned to iterating\_var, and the statement(s) block is executed until the entire sequence is exhausted.

1. Create a **ForControl.py** file and type the code as below:

A picture containing drawing

Description automatically generated

Note: we import the random function to generate a random number from 5 to 15 (exclusive) for the end range number.

1. In the terminal type the following command:

>>> python3 ForControl.py

A close up of text on a black background

Description automatically generated