

CS506 – Programming for Computing

HOS01B Variables, Simple data types, User Input, If-else, Loops

10/03/2020 Reviewed by Kim Nguyen

3/28/2022 Reviewed by Ken Ling

7/2/2023 Reviewed by Ajay Naik

School of Technology & Computing (STC) @ City University of Seattle (CityU)



Before You Start

- Version numbers may not match with the most current version at the time of writing. If given the option to choose between stable release (long-term support) or most recent, please select the stable release rather than the beta-testing version.
- There might be subtle discrepancies along with the steps. Please use your best judgment while going through this cookbook-style tutorial to complete each step.
- For your working directory, use your course number. This tutorial may use a different course number as an example.
- All the steps and concepts in this tutorial are from the textbook, so if you encounter problems in this tutorial, please try to read and compare the textbook to solve the problem. If you still can't solve the problem, please feel free to contact your course TA.
- Avoid copy-pasting code from the book or the GitHub repository. Instead, type out the code yourself. Resort to copy-pasting only when you are stuck and find that things are not working as expected.

Learning Outcomes

Students will be able to:

- Setup the working environment for Python

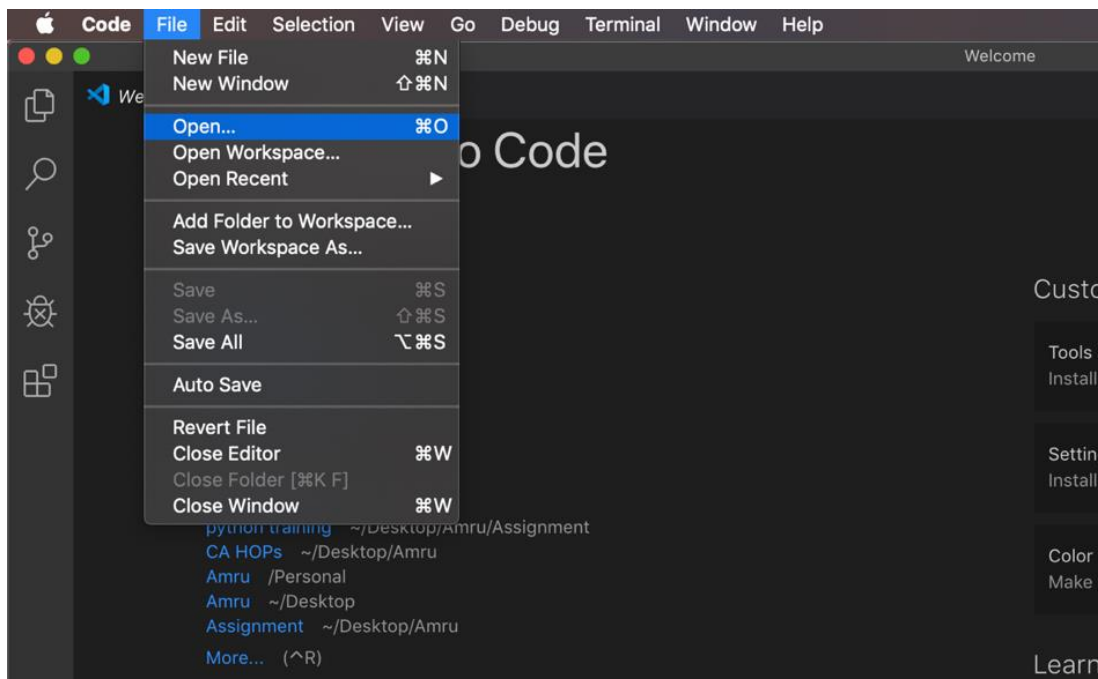
- Understand the basics python concepts
- Understand the data conversion in Python

Resources

- Python crash course: a hands-on, project-based introduction to programming: Matthes, E. (2019): [Available online link](#)

Section 1 - Preparation

- 1) In Visual Studio Code, open the private repository generated when you accepted the HOS01 assignment (If you cannot find that repository in your machine, you might have not cloned the repo, if so, please do before proceeding).



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

3) `cd Module 1`

Section 2 - Variables, Data Types and User Input

- 1) **Follow the video instructions below to code along:**

- 2) <https://youtu.be/5Y9AENc6o2Y>
- 3) 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.
- 4) More about variables: https://www.w3schools.com/python/python_variables.asp
- 5) 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.
- 6) More about data types: https://www.w3schools.com/python/python_datatypes.asp
- 7) User Input: To receive information through the keyboard, Python uses the input() function
- 8) When the input() function is called, the program flow stops until the user enters the input via the command line. To enter the data, the user needs to press the ENTER key after inputting the string.
The input() function, by default, will convert all the information it receives into a string.
- 9) More about user input: https://www.w3schools.com/python/python_user_input.asp

Section 3 - Multiple assignments

- 1) Python allows you to assign values to multiple variables in one line and you can assign the same value to multiple variables in one line. For example:

```
print("\nMultiple assignments\n")
a, b, c = 5, 3.2, "Hello"
print ("a = ",a)
print ("b = " ,b)
print ("c = ",c)

x = y = z = "Python"
print ("x = "+x)
print ("y = "+y)
print ("z = "+z)
```

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


Section 4 - String functions and Concatenation:

1) **Follow the video instructions below to code along:**

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

3) 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 object-oriented language. Another way to concatenate strings is as below example:

```
first_message = "Hi !"
second_message = "How are you ?"
full_message = f"{first_message} {second_message}"
print(full_message)
```

- i. *(This is optional, you do not have to write the code in the above screenshot)*
- ii. Output:
- iii. 

```
Hi ! How are you ?
```
- iv. 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.

Section 5 - Numbers and operators

- 1) Operators are the constructs which can manipulate the value of operands.
- 2) Consider the expression $4 + 5 = 9$. Here, 4 and 5 are called operands and + is called operator.
- 3) Python language supports the following types of operators.
 - i. Arithmetic Operators
 - ii. Comparison (Relational) Operators

- iii. Assignment Operators
- iv. Logical Operators
- v. Bitwise Operators
- vi. Membership Operators
- vii. Identity Operators
- viii. The commonly used operators are underlined above.
- ix. More about numbers and operators:

https://www.tutorialspoint.com/python/python_basic_operators.htm

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

```
print("\nFloat\n")
a=2.2
b=2
c=0.1
print("a + b =", a+b)
print("a + c =", a+c)
print("a * b = ", a * b)
print("a ** b = ", a ** b)
```

- 5) Type the following to get the output of the code in the terminal
- 6) `python3 Numbers.py`

```
Float
a + b = 4.2
a + c = 2.3000000000000003
a * b = 4.4
a ** b = 4.8400000000000001
```

- 7) 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 must represent numbers internally.

Section 6 - Python Decision making

- 1) Decision making is anticipation of conditions occurring while execution of the program and specifying actions taken according to the conditions.
- 2) Create a **HOS01_ELIF.py** file, follow the below video instruction to code along:
- 3) <https://youtu.be/4rIRGlAMEvM>
- 4) **Challenge:**

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

```
Please guess a integer between 1 and 6: 5
Congrats, you got it!
Kims-MacBook-Pro:CS506 kimnguyen$
```

5) Elif:

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

```
1 print('How old are you?')
2 age = int(input())
3
4 if age < 22:
5     print('You are too young to have a drink.')
6 elif age >= 80:
7     print('Ok, you will get a free drink.')
8 else:
9     print('Sure, enjoy your drink.')
```

6) Use the following command and give different input to test the program:

7) `python3 IfControl.py`

```
how old are you?
10
you are too young to have a drink.
```

```
how old are you?
23
Sure, enjoy your drink.
```

```
how old are you?
88
Ok, you will get a free drink.
```

8) More about If statements: https://www.w3schools.com/python/python_conditions.asp

Section 7 - Loops

- 1) 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 times.
- 2) Programming languages provide various control structures that allow for more complicated execution paths.
- 3) A loop statement allows us to execute a statement or group of statements multiple times.

Section 8 - While loop

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

2) https://youtu.be/_3665KzFIO8

- 3) The while loop runs as long as, or while, a certain condition is true.
- 4) Syntax:

- i. while expression:
- ii. statement(s)

- 5) 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.
- 6) When the condition becomes false, program control passes to the line immediately following the loop.
- 7) In Python, all the statements indented by the same number of character spaces after a programming construct are part of a single block of code. Python uses indentation as its method of grouping statements.
- 8) More about while loops: https://www.w3schools.com/python/python_while_loops.asp
- 9) **Challenge:** Fix the program so that the attempts of guessing is 3 times instead of 4.

Section 9 - For loop

- 1) “*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.
- 2) Syntax:
 - i. for iterating_var in sequence:
 - ii. statements(s)
- 3) 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.

Section 10 - Create a **ForControl.py** file and type the code as below:

```

1  import random
2
3  for i in range(1, random.randint(5, 15)):
4      print('This for loop has already run ' + str(i) + ' times.')
5

```


- i. Note: we import the random function to generate a random number from 5 to 15 (exclusive) for the end range number.
- 2) In the terminal type the following command:
- 3) `python3 ForControl.py`

```
This for loop has already run 1 times.  
This for loop has already run 2 times.  
This for loop has already run 3 times.  
This for loop has already run 4 times.  
This for loop has already run 5 times.  
This for loop has already run 6 times.
```

Push your work to GitHub

Follow the instructions on the CityU STC TA Center Github.io [Submit your work page](#).