**IS201 Fundamentals of Computing**

**HOS05 Functions & Debugging**

03/22/2024 Developed by Scott Zhou

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

 A white cat in a circle with a blue and black logo

Description automatically generated

**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**

* Learn ways to pass information to functions.
* How do you write certain functions whose primary job is to display information? How do you design other functions to process data and return a value or set of values?
* Learn to store functions in separate files called modules to help organize your main program files.

**Resources**

* Python crash course: a hands-on, project-based introduction to programming: Matthes, E. (2019): [Available online link](https://cityu.alma.exlibrisgroup.com/discovery/openurl?institution=01CITYUNIV_INST&rfr_id=info:sid%2Fsummon&rft_dat=ie%3D5152833400004251,language%3DEN&svc_dat=CTO&u.ignore_date_coverage=true&vid=01CITYUNIV_INST:Services)

**Python Functions**

A function is a block of organized, reusable code that performs a single, related action. Functions provide better modularity for your application and a high degree of code reuse.

A function can be written as a set of statements that take inputs, perform specific computations, and produce output.

**Syntax:**

def functionname( parameters ):

code

return [expression]

**Note:** The statement returns [expression] exits a function, optionally passing back an expression to the caller. A return statement with no arguments is the same as return None. It is not mandatory to use a return statement.

A function can process some data and then return a value or set of values. The value the function returns is called a return value. The return statement takes a value from inside a function and sends it back to the line that called the function. Return values allow you to move much of your program's grunt work into functions, which can simplify the body of your program.

A function doesn’t always have to return something, sometime, a function just prints something out, it is the same as return None. Example as below (You don’t have to write the below code):

**Passing Arguments**

Because a function definition can have multiple parameters, a function call may need multiple arguments. You can pass arguments to your functions in several ways.

Positional arguments: which need to be in the same order the parameters were written

Keyword arguments: where each argument consists of a variable name and a value

Arbitrary Number of Arguments: Sometimes you will need to know ahead of time how many arguments a function needs to accept. In that case arbitrary can be used.

##### The order matters in positional arguments. The parameters specified in the function definition must be in the same order as how the function is called. There won’t be any error showing when running the code, but the result might not be as you expected it to be.

**Note**: title() method is a built-in Python string method that returns a new string with the first character of each word capitalized, and all other characters in lowercase.

**Arbitrary number of arguments**

Sometimes you'll want to accept an arbitrary number of arguments, but you won't know ahead of time what kind of information will be passed to the function. In this case, you can write functions that accept as many key-value pairs as the calling statement provides.

**Create a Project**

Follow HOS1 to set up the project in Codespaces, or use any other code editor you prefer, such as Visual Studio, Visual Studio Code, Sublime, Vim, etc…

We will create a Python project to cover all the learning outcomes; the project simulates a basic contact book application that allows users to add, remove, search for, and display contacts.

1. Create a Python file named contact\_book.py.
2. Create an add contact function to store name as the key, phone as the value in contacts, and return contacts after the function gets called.

A close-up of text

Description automatically generated

1. Create a remove function based on the name in contacts.

***A screenshot of a computer code

Description automatically generated***

1. Create a search function to retrieve contact information by name and a list function to show all the contacts.

***A close-up of text

Description automatically generated***

1. Create a display.py file under the same directory, this file only need to contain display message function and display contact function.

*A screen shot of a computer code

Description automatically generated*

1. Create a main.py file under the same directory for the main application logic. Then, import all the functions we created from other files.

**

1. Create a main function to initialize a dictionary to test all the functions.

*A screenshot of a computer code

Description automatically generated*

1. Add Python idiom to check if the script is being run as the main module, which is commonly used at the end of a Python Script.

A close-up of a number

Description automatically generated

1. Confirm all the files have the correct naming convention.

A screenshot of a computer

Description automatically generated

1. Execute the program from the hos05.py file.

A close up of a phone

Description automatically generated

**Submit your Work to Brightspace**

Please upload your .py file to the HOS05 assignment on Brightspace.