

**November 1, 2018**

**Notebook:** Computers and Programming I

**Created:** 11/1/2018 3:03 PM

**Updated:** 11/13/2018 1:06 PM

**Author:** Anonymous

---

- Functions
  - A group of statements within a program that performs a specific task
  - Usually one task of a large programs
  - Can be executed in order to perform overall program task
  - Divide and conquer approach.
- Modularized Program:
  - Program wherein each task within the program is in its own function
- def function ( )
  - statement
  - statement
  - statement
- Benefits of Modularizing A Program With Functions
  - Simplier code -
    - a program's code tends to be simpler and easier to understand when it is broken down into functions
  - Code Reuse
    - write the code once and call it multiple times
  - Better Testing and Debugging
    - Can test and debug each function individual
  - Faster Development
- Void Functions
  - Simply executes the statements it contains and then terminates
- Value Returning Function
  - Executes the statements it contains, and then it returns a value back to the statement it called.
    - The input, int and float functions are examples of value returning functions
- Functions Are Given Names:
  - Rules
    - Cannot use keywords as a function name
    - No spaces
    - First character must be a letter or underscores
    - All characters must be a letter, number or underscore
    - Uppercase and lowercase letters are distinct
  - Function name should be descriptive of the task carried out by the function
    - often includes a verb
  - Function definition:
    - specifies what function does
      - `def function_name ( ) <== Function Header`
      - statement
      - statement
  - Funtion Header
    - Include keyword defand function name follwed by parentheses and colon
  - Block
    - set of statements that belong together as a group
      - the statements included in a function. Notice in the general format that all of the statements in the block.
  - Call A Function To Execute It
    - When a function is called:
      - Interpreter jumps to the function and executes statements in the block
      - Interpreter jumps

```
1 def message ( ) :
```

```

2 print ("Hello! My name is Saied.")
3 print ("I am taking CS 101 this semester.")
4 message ()

```

- Main Function

- called when the program starts
  - calls other functions when they are needed
  - Defines the mainline logic of the program, which is the overall logic of the program

```

1 def main ():
2     print ("I have a message for you:")
3     message ()
4
5 def message ():
6     print ("I am Saied")
7     print ("I am taking CS 101 this semester.")
8
9 main ()

```

- Each block must be indented

- Lines in block must begin with the same number of spaces
  - Use tabs or spaces to indent lines in a block, but not both as this can confuse Python Interpreter
  - Blank lines that appear in a block are ignored

- Designing A Program To Use Functions

- In a flowchart, function call shown as rectangle with vertical bars at each side.
  - Function name written in the symbol
  - Typically separate flow chart for each function in the program
    - End terminal symbol usually reads return

- Top Down Design:

- Technique for breaking algorithm into functions
  - The overall task that the the program is to perform is broken down into a series of subtasks.
  - Each of the subtasks