Week I:

Python as a Calculator:

2\*\*5 = 2^5

5//2 = 2 (integer division)

4%2 = 0 (mod)

Python and Memory:

A value *has* a memory address.

A variable *contains* a memory address.

A variable *refers* to a value.

A variable *points to* a value.

Variables:

Alt-p and Alt-n cycle between previous commands in IDLE

Rules for executing an assignment statement:

1. Evaluate the expression on the right side of the = sign to produce a value. This value has a memory address.
2. Store the memory address of the value in the variable on the left of the = sign

Rules for legal Python names

1. Names must start with a letter or \_
2. Names must contain only letters, digits, and \_

Built-in Functions:

Argument = an expression that appears between the parentheses of a function call

Pass = to provide an argument to a function

Function call = to ask Python to evaluate a function

Function Call:

*function\_name(arguments)*

Rules for executing a function call

1. Evaluate each argument one at a time, working from left to right.
2. Call the function, passing in the argument values.

Defining Functions:

def: a keyword indicating a function definition.

return: a keyword indicating the result of a function.

Parameter: a variable that appears between the parentheses of a function definition. Parameters get their values from expressions in a function call.

Return statement:

return *expression*

Rules for executing a return statement:

1. Evaluate the expression, which produces a value.
2. Produce that value as the result of the function call.

Function definition:

def *function\_name(parameters):*

*Body*

Function calls are expressions.

Week II:

String literal: a sequence of characters

str: Python’s string type—strings start and end with ‘ or “—strings are values

Type String:

Escape character: \

Escape sequence: the escape character together with the character that follows it.

Python will choose ‘ or “ depending on how the string is created and whether an escape sequence is used.

Concatenation:

Str1 + Str2 => concatenates str1 and str2

Str1 \* int1 => concatenates int1 copies of str1

Int1 \*str1 => concatenates int1 copies of str1

The \* and + operators obey order of operations

All other operators result in a TypeError

Input/Output and Str Formatting:

If a function definition reaches the end of its body without executing “return”, its return value is NONE

print(): prints a sequence of arguments for a user to read

input(): gets a string from the user—this function pauses until the user types a newling

Triple-quoted strings can span multiple lines

\n = newline

\t = tab

\\ = \

\’ = ‘

\” = “

If you write a docstring (triple-quoted explanation) under a function definition, you can use the help() command to bring up that docstring as a description

Visualizing Function Calls:

Stack frame: a region of computer memory for keeping track of information about a function being executed.

Local variable: a variable created inside a function body

When a function returns, its stack frame is removed from the call stack.

Week III:

Assignment statement and computer memory:

*variable = expression*

If a variable does not exist in the current stack frame, Python creates it.

Booleans:

Precedence of Boolean operators:

1. not
2. and
3. or

Mathematical operators have precedence over Boolean operators

Importing:

The module being imported must be in the same directory as the module that is importing it

Week IV:

String Comparison:

Capital letters are “less than” lowercase letters for string comparison

You can compare == and != with different types, but not < or >

Use “in” to check if a string contains another string

String Index:

Len() returns # of chars in string

Slices are “up to but not including” so s[0:2] only returns character at 0 and 1

If slice omits second value, it defaults to len(string)

If slice omits first value, it defaults to 0

Strings do not support modification (index assignment)

Methods:

Functions inside objects are called methods

Considering the case “for char in string” if string is “”, the for loop will never iterate since the condition “char in string” is always false

Week V:

List notation [x,y,z]

Same rules for slicing lists as strings

Aliasing: when two or more variables refer to the same object in memory

Mutable: list

Immutable: int, float, str, bool

Week VI:

Proper nested list notation: list[i][j][k]

File I-O: ‘r’ = read, ‘w’ = write (write over), ‘a’ = append (add at end)

import tkinter.filedialog

tkinter.filedialog.askopenfilename()

tkinter.filedialog.asksaveasfilename()

Week VII:

Lists are mutable

Tuples are immutable

Dictionaries are mutable

(3) = 3 (does not create a tuple)

(3,) = (3,) (does create a tuple)

“del” to remove a key in a dictionary

To add or change a value in a key, simply assign the value to that key in the dictionary