

Python Notes:

Print Statement:

A print statement is the easiest way to get your Python program to communicate with you.

-There are two different Python versions. Both Python 2 and Python 3 are used throughout the globe.

The most significant difference between the two is how you write a *print* statement. In Python 3, *print* has parentheses.

-Python code interpretation executes everything line by line from the top.

-Python is a case sensitive language. When defining variables, always use lower case characters.

When using boolean keywords (true/false) use capitals or python won't recognize it.

-In Python, indentation is very important. Indentation is used to indicate a block of code.

-Using “#” on a line of code initiates the comment function (meaning whatever you write will not affect what is outputted in the code).

To verify the type of any object in Python, use the “type()” function:

Function List:

-print: writes a message

-input: stores information entered by user. Output always interpreted as a string

-int: converts string into an integer

-float: converts string into a float

-bool: converts a string to a boolean (true or false) value

-type:

-len: returns the number of items in a list (LENgth)

-round: rounds a float to the nearest integer

-abs: “absolute”. Returns the positive representation of a value (even if that value is negative)

Variables:

-Used to temporarily store data in a computer's memory.

-The variable's “identifier” is the name of the variable (ex. Price)

-Use underscores to separate multiple words in our variable's name.

-Variables do not need to be declared with any particular type and can even change type after they have been set.

-A variable can have a short name (like x and y) or a more descriptive name (age, carname, total_volume). Rules for Python variables:

- A variable name must start with a letter or the underscore character
- A variable name cannot start with a number
- A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and _)
- Variable names are case-sensitive (age, Age and AGE are three different variables)

-The Python “print” statement is often used to output variables

-To combine both text and a variable, Python uses the “+” character. This can also be used to add a variable to another variable.

Python Numbers:

-There are three numeric types in Python:

- int: a whole number, positive or negative, without decimals, of unlimited length
- float: "floating point number" is a number, positive or negative, containing one or more decimals. A Float can also be a scientific numbers with an "e" to indicate the power of 10.

- complex: Complex numbers are written with a "j" as the imaginary part

Python Casting (Converting the type of variable):

There may be times when you want to specify a type on to a variable. This can be done with casting (essentially changing the type of variable) ex. Converting an integer into a string.

-Casting in python is done using constructor functions:

- int: constructs an integer number from an integer literal, a float literal (by rounding *DOWN* to the previous whole number), or a string literal (providing the string represents a whole number) ex. Converting 2.8 to an integer would = 2.0
- float: constructs a float number from an integer literal, a float literal or a string literal (providing the string represents a float or an integer) ex. Converting 3 to a float would = 3.0
- str(): constructs a string from a wide variety of data types, including strings, integer literals and float literals ex. Converting an integer or a float into a string would input the number as text

Strings:

-A sequence of characters

-Interpreted in between double quotes ("") or single quotes (')

-Can also use triple quotes (single or double is fine) for interpreting more than one line of string characters

“Indexes” refer to the order of the characters in a string. Ex. “0”, would be the first letter of any given string. “Negative indexes” to refer to the the order of the string characters starting from the end of the string

-Use square brackets to get a character at a given index of a string. ex. [0] would retrieve the first character of a string

-Using colon in between two index numbers will retrieve all the numbers in between the 2 index values (but not including the 2nd index value)

String Literals:

-String literals in python are surrounded by either single quotation marks, or double quotation marks

-Python does not have a character data type, a single character is simply a string with a length of 1

-Square brackets can be used to access elements of the string

-“[0]” will retrieve the first character of a string

-“[0:2]” will retrieve the first three characters of a string

Formatted Strings:

-Prefixed with an “f”

-Use curly brackets “{ }” to dynamically insert values into your strings

String Methods:

-Specific functions for strings:

-Adding a dot (.) operator to a variable brings up all functions that are specific to strings

Python Collection Data Types:

There are four collection data types in the Python programming language:

- List: is a collection which is **ordered** and **changeable**. **Allows duplicate** members. Written with [square brackets]
- Tuple: is a collection which is **ordered** and **unchangeable**. **Allows duplicate** members. Can

not add or remove items. Written with (round brackets)

- Set: is a collection which is **unordered** (will appear in a random order) and **unindexed** (can not access items in a set by referring to an index). **No duplicate** members. Written in {curly} brackets.

- Dictionary: is a collection which is **unordered**, **changeable** and **indexed**. **No duplicate** members. Written in {curly brackets}, and has keys and values. **REVIEW**

Loops:

- A programming structure that executes the same block of code over and over
- Python has two types of loops: “while” and “for”
- “While” loop executes a set of statements as long as a condition is true
- “For” loop is used for iterating over a sequence.
- “Range” function is used to loop through a set of code a specified number of times

List Methods:

- strip(): removes any whitespace from the beginning or the end
- len(): returns the length of a string
- upper(): returns the string in upper case
- replace(): replaces a string with another string
- split(): splits the string into substrings if it finds instances of the separator
- append(): Adds an item to the list
- insert(): Adds an item to the specified index
- remove(): removes specified item
- pop(): removes specified index (or the last item if the index is not specified)
- del(): removes specified index
- clear(): empties the list
- copy(): copies a list (can not use list2 = list1 because it will only be a reference to the list. Any changes to the first list will also affect the second list)

Tuple Methods:

count(): Returns the number of times a specified value occurs in a tuple.

Index(): Searches the tuple for a specified value and returns the position where it was found.

Set Methods:

- add(): Adds an item to the set
- update(): Adds multiple items to the set

Arithmetic Operations:

- There are two types of division operations, “/” which produces a float and “//” which produces an integer.
- “%” is a modulus, which returns the remainder of the integer division.
- exponent operators are expressed using two asterisks “**”.
- Augmented assignment operators

Functions:

- A block of code that only runs when it is called.
- You can pass data (aka parameters) into a function.

- A function can return data as a result.
- In python, a function is defined using the “def” keyword.
- To call a function, use the function name followed by parenthesis.

Module:

- A separate file with reusable code. Used to organize code into different files. Kind of like an “expansion” pack that has specific features to be used.
- To import an “math” module, type “import math”.

If Statements:

- “if” refers to when a condition is met.
- “else” refers to when no conditions are met.
- “elif” refers to “else (otherwise) if”. In other words, “if the previous conditions were not true, try this condition”.
- include colon after statement before executing a command

Logical Operators:

- Used in conditions where there are multiple conditions.
- “And”: if all conditions are met
- “Or”: if one condition is met
- “Not”: inverses any boolean value we give it (if we give it “true”, it converts it to false).

Comparison Operators:

- Used when comparing a variable with a value.
- “>”: greater than
- “<”: less than
- “>=”: greater than or equal to
- “<=”: less than or equal to
- “==”: equality (different than one “=” sign, because the single operator changes the value, instead of making a comparison).
- “!=”: not equal to.

Definitions:

Expression: A piece of code that produces a value

Integer: A number without a decimal point

Floating point number (Float): Numbers with decimals

String: A sequence of characters (used to represent text, rather than numbers)

Common Mistakes:

- Forgetting colons
- Invalid
- The first character of a python index is 0 NOT 1.
- The variable's “identifier” can not be casted (assigned int or str or float), however its value can.

1.16.24

Examples: