Cheat Sheet 2

Xingyu Zhang

09/22/2021

1.Define: Packages, objects, functions, methods and attributes

Packages –A collection of modules. These modules are premade, importable python programs, each made for a specific function. In essence, packages allow a python user to customize their workspace, tailoring it to their expected operations via their collection of available functions and canned routines from the modules within their chosen packages.

Objects – Objects represent the data input into python. These are the variables, be they floats, integers, characters, lists, or something else. Objects are defined and referred to with a name handle, along with a data type.

Functions – Functions are built-in routines, designed to act on a wide range of objects. To execute a function on an object, use the syntax function(object). Like modules, functions are pre-built blocks of Python code, and installing additional modules and packages will make more functions available.

Methods – Methods are a special type of function, designed for a specific object type, which perform a pre-specified operation on the object. Functions are universal, while methods only work on specific object types. Methods are called by putting a ‘.’ after the object name and putting the name of the method being used after it.

Attributes – Attributes are the inner workings of an object, the definitions and properties behind them. Attributes cover a very broad range of properties, and can be either predefined in the making of an object or function, or they can be completely custom, designed by the user after the fact. The flexibility attributes provide is a key part of object-oriented programming. Like methods, attributes can also be called/defined by placing a ‘.’ after the object name.

2.Lists: What are they, how do you make them, what are some important methods to remember.

1) List is a data type that stores various items with different data types separated by commas.

2) List = [item\_1, item\_2, item\_3, …, item\_n]

3) append, remove, reverse, insert, sort, count, pop, del, index, extend

3. indexing: summarize how indexing and slicing works for lists and give examples of the different ways to access things (such as 1d array)

1) indexing starts from 0 to n-1 in a forward direction but starts from -1 in the opposite direction. Slicing is a way to pick a sequential element from a list

2) examples: my\_list=[a,b,c]

4.Conditional statements: Whats the syntax examples of different types of conditionals and conditional nesting.

1)Conditional statements are used to determine whether a statement is true, only executing the code following it if the statement is true. If the statement is false, alternative code may or may not be executed instead. Conditionals always begin with an if, else, or elif statement.

2)example:

a=5

if a<7:

print(‘a<7’)

5.For Loops: What are these for and what's the syntax

A for loop is used to iterate over a sequence, such as a list, tuple, and string.

Example:

i=5

For i<7:

Print(‘i’)

i=i+1

6.List comprehensions What are these and what's the syntax

List comprehension is used to create a new list based on existing iterables, such as list, tuple, strings, etc.