Jason Schlottman

HWRS 401

9/19/2021

Cheat Sheet 2 -Python\_basics

**Define:**

* **Packages:**

These are the imported extensions which provide prebuilt coded suites contained within modules which allow users to work with functions that are user-friendly and help create new projects without starting from scratch. An example of a python package is the pandas install package.

* **Objects:**

An object is the type of data that belongs to a class. Since python is object-oriented, we can work with many types of objects such as variables, tuples, lits, etc. to write code and define arguments within functions based on these objects of varying types.

* **Functions:**

A function is a command in python which calls upon variables or arguments and returns or controls a certain process to be carried out based on whether the function defines the outcome.

* **Methods:**

A method is similar to a function where it is a grouping of code but it must be set within a class and it must be called on an object. An example is the .append method which can be used for things like lists to return an outcome or result.

* **Attributes:**

An attribute is a variable that can belong to all types of objects within the data type or method. These attributes apply to all data types within the class.

**Describe:**

* **Lists:**

Lists are python built-in data types used to store several items within a single variable. Lists are made by defining any variable and entering the arguments within []. For example, my\_list=[a,b,c]

* **Indexing:**

Indexing refers to the "indexed" value within a list or set where the arguments are assigned specific values ranging from 0:'end value'. For example, in the created list above, a=0, b=1, c=2. We may call upon a indexed value in a list by saying for example if my\_list(0)=7".

* **Conditional Statements:**

Conditional statements describe a number of various commands in python which set boundaries on returns and apply these conditions to the return. A few examples are "if","else" "if not" , etc.

* **For Loops:**

For loops are used to define a variable and loop that statement over decided conditionals to return a set of returns based on the defined inputs. This block of code may sequence a list, tuple.dictionary,sequence,etc. Syntax example is "for i in range:"

* **List comprehensions:**

List comprehensions allow shorter smaller comprehensions using an existing list and it's elements rather than creating an entire new list. For example, we may write only one line of code rather than writing a new block with a conditional statement if we want to return a set from the existing list with only a portion of the returns based on a certain condition.

Ex) list=[the,them,then]

For I in list:

If "m" in list

New\_list=[I for I in list if "m" in list]

Return: "them"