*Steps Arch 02 – Week 7*

**Step 01:**

**What to learn?**

* 1: Function will always have priority for your code
* 2: A anonymous function is lambda. You can define it by following it up with a parameter. Then use “:”, returns the value after x \* x to the caller. The function of Lambda x : x \*x is assigned to a value square. To call it like a named function**.**
* 3: *what are namespace and scope?*

Namespace: a namespace is a collection of currently defined symbolic names with information about the object that each names reference. For example, you can think of namespaces like dictionaries. Where the keys are the objects, and the values are the names.

Four types of namespaces:

* Built – in 🡪 contains all the built-in objects in python.
* Global 🡪 contain the names defined of the level of the main program.
* Enclosing
* Local

The scope of a name is the region of a program in which the name has meaning

For now we will use the scope of a function with the types of namespaces.

If you search for example (X) in your code, the scope will run according to the following order:

1: Local 🡪 if you look for x inside a function, this will run first

2: Enclosing 🡪 if x is not inside the local function, but it appears in a function that resides inside another function.

3: Global 🡪 if the two above won’t result to anything, the interpreter will go through the Global scope next

4: Built -in 🡪 If x can’t be fined everywhere, the interpreter will go through the built-in scope.

**Step 02:**

**What to learn?**

* 1: for using the .append code. This will add something to the list.
* 2: by using the sort()/sorted() code. The sort will sort the list for you. The sorted will sort the list and makes a copy of it. So it will not change the original list. You can also sort tuples.
* using sort(), inside the parameters must be the tuple
* 3: Examples

List() : oldlist = [1,2,3,4,5,6]

newlist = list(oldlist)

print(newlist)

print(oldlist)

Slicing: list = [1,2,3,4,5,6,7,8,9]

list2 = list[:]

print(list)

print(list2)

Copy():list = [1,2,3,4,5,6,7,8,9]

list2 = list.copy()

print(list)

print(list2)

Deepcopy():import copy

list = [1,2,3,4,[5,6],7,8,9]

list2 = copy.deepcopy(list)

print(list2)

* 4: by using a new list with the code: New\_list = [expression for item in iterable]. Tuple comprehension is not possible.

**Step 03**

**What to learn?**

* 1: with copy() you can copy all the keys and values of the dictionary. Copy() is only useful if the dictionary is immutable (shallow copy). If you want

*Steps arch 02 – week 5*

*Step 01*

**What to learn?**

* 1: a function is a named piece of code separated from all others. Function can take any number and type of input parameters and returns any number and type of output.
* 2: The main elements of a function are: The keyword def, a function name to uniquely identify the function, the parameters where pass through values to a function(arguments), and finally the “:” to make the function work.
* 3: A function can be called through the use of the function name. For example if you have a function named def Weather\_forecast():, you can call the function by using Weather\_forecast().
* 4: If a function works it will always return a value, the value can be True or False if you don’t ask for a specific value. You can return a value by using return (Name of value).
* 5: Arguments are the values you pass through the function. When you call a function the values of the arguments are copied to their corresponding parameters.

Step 02

**What to learn?**

* 1: a tuple is a sequence type, that contains 0 or 2 elements. Unlike strings tuple can have different element types, in fact each element can be any python type object. Tuples are immutable. A tuple is defined by the ().
* 2: you can combine tuples by using the “+” syntax. If you want to compare tuples, you can use the comparison syntax “==”. With this you can compare the tuples to look if they are equal.
* 3: you can iterate over tuples by using for ….. in …….
* 4: You can’t modify a tuple, tuples are immutable, however you can modify a tuple by creating a new one and use the combine function.

Step 03 o

**What to learn?**

* 1: A list is a data structure in python. That is mutable and can be changed. A list is ordered in a sequence of elements. A list is defined as characters between quotes, list is defined by having elements between the square brackets.
* 2: the split function splits a string to a list.
* 3: With offset you can specify a single value from the string. With slice you can get multiple values. The pros for offset are that you can get a specific value, and for slice is that you can get multiple.
* 4: you can add new elements to a list by using: .append() and with .insert()
* 5: you can modify elements of a list by using offset or slice
* 6: by using the for-in- technique

*Arch 02 Week 6*

Step 01

**What to learn?**

* 1: a dictionary is like a list, but the order doesn’t matter. Inside the list you have unique Keys that associate with a value. Dictionaries are mutable, so you can change the values, keys, positions and so forth. You can create a dictionary by using the curly brackets ({}), inside those brackets are the Keys: Value.
* 2: Changing dictionaries is easy. Refer to the item by the key and assign a value, if the key already exists, then the key will be replaced by the new one.
* 3: You can get the value of a key by using the name of the dictionary followed by the key itself. For example, Dict(“Key”). If the key is not present, then you will get a key error. Another way is to loop through the dictionary, by using a “Key in Dictionary” format. Another option is to use the get() function. This goes as follow, Dict.get(“Key”). The last option is to get all the keys by using the .values() function. It’s better for this method to also create a list for the asked values.
* 4: The Keys are assigned to a value; with the item you will get both the key and value. You can see it as if the items are both the keys and values combined.
* 5: del : You can use this when you want to remove a certain item in the dictionary. .pop: you can remove a specific item in the dictionary. Clear(): you clear the whole dictionary of all the keys, items and values
* 6: You can iterate over a dictionary by using the for and in method.

Step 02

**What to learn?**

* 1: Positional arguments: values that are copied to their corresponding parameters in order. If you don’t want the confusion of the positional arguments, you can use the keyword arguments. With this you can specify the arguments with their corresponding parameters.
* 2: The default is used when the user does not provide a corresponding argument
* 3: Docstrings are documentations you can add to a function definition by including a string at the beginning of the function body. This can be useful for adding rich formatting.

Step 03

**What to learn?**

* 1: Set is like a dictionary, but without the values and only the keys. Sets also remove the duplicates from it. U want to use sets if you want to know that something exists. Sets are defined by the following code set().
* 2: You can add elements to a set by using the .add() function. If you want to remove a element in a set you can use the .remove() function.
* 3: You can iterate over a set by using the for in method.
* 4: Assume two sets S1 and S2. How can one specify the following operations on S1 and S2 in Python:
  + Intersection of S1 and S2 : By compering if the two sets have the same common elements. This can be done by using the intersection() method.
  + Union of S1 and S2: With union you can look for member in both sets. By using “.union()”.
  + Difference S1 and S2: you can use if you want to find out wich members are in the first set but not the second. This can be done by using “.difference()”.
  + Is S1 a subset of S2 (or vice-versa)?: You can check this if you want to know if all the members of S1 are in S2. You can check this by using the “<=” or “.issubset()”. If you want to check if all the member of S2 are in S1 you can use the “>=” or “issuperset”.

For example: S1 <= S2, S1.issubset(S2). Which will return True if its right.

S2 >= S1 , S2.issuperset(S1). Which will also return True if its right.