**Strings and numbers are different.**

They work in similar way i.e if I want to duplicate strings or bring them together.

I.e “hello world” \* 2

\ (escape) let’s your quotes be recognized as quotes eith single or double quotes

Example

**>>> "hello world this is a \"quote\" awesome time"**

**'hello world this is a "quote" awesome time'**

**Strings** have built in features like line breaks

Example

**>>> "hello\nworld"**

**'hello\nworld'**

**>>> print("hello\nworld")**

**hello**

**world**

**>>>**

Variables and line breaks

Example

**>>> Fruits = "Apple\nOrange\nBanana"**

**>>> print (Fruits)**

**Apple**

**Orange**

**Banana**

>>>

##########

**Indexing will help put items in a list of column. Count starts from 0...**

***Create index value with fruit names for my\_items variable***

>>> my\_items = ["Juice", "orange", "apple"]

>>> my\_items

['Juice', 'orange', 'apple']

***Display item by using corresponding index number***

>>> my\_items[2]

'apple'

***Create index value with prices for my\_cart variable***

my\_cart = [13, 14, 55, 15]

>>> my\_cart

[13, 14, 55, 15]

***Combine items and cart as product***

>>> my\_products = [my\_items, my\_cart]

>>> my\_products

[['Juice', 'orange', 'apple'], [13, 14, 55, 15]]

***Add additional value to your cart***

>>> my\_cart.append(20)

>>> my\_cart

[13, 14, 55, 15, 20]

***Do the same for list of Items***

>>> my\_items.append("guava")

>>> my\_items

['Juice', 'orange', 'apple', 'guava']

###############

List and Dictionaries:

List is an example of the things you do at home

***Create a table list with name and age***

>>> my\_data = {"Name":"Ade Tayo", "Age":"Thirty Five"}

>>> my\_data["Name"]

'Ade Tayo'

>>> my\_data["Age"]

'Thirty Five'

>>> print(my\_data)

{'Name': 'Ade Tayo', 'Age': 'Thirty Five'}

***Add location to the list***

>>> my\_data = {"Name":"Ade Tayo", "Age":"Thirty Five", "Location":"Texas"}

>>> my\_data

{'Name': 'Ade Tayo', 'Age': 'Thirty Five', 'Location': 'Texas'}

***How to add data with dictionary. Unlike list, you don’t append to add with dictionary. You write it like you create a data.***

>>> my\_data["Occupation"] = "Tech"

>>> my\_data

{'Name': 'Ade Tayo', 'Age': 'Thirty Five', 'Location': 'Texas', 'Occupation': 'Tech'}

>>>

***Look up the dictionary and list***

>>> my\_data.keys()

dict\_keys(['Name', 'Age', 'Location'])

>>> list(my\_data.keys())

['Name', 'Age', 'Location']

***List a data from list using indexing i.e index 1 which is Age***

>>> list(my\_data.keys())[1]

'Age'

>>>

**Note: You use List when you want to keep things in ordering manner or numbers you plan or summing up.**

***If you want a bunch of data that describes something in a consistent way a Dictionary is really good Example.***

**>>> user\_1 = {"Name": "Adetayo Bero"}**

**>>> user\_2 = {"John": "Doe"}**

**>>> my\_users=[user\_1, user\_2]**

**>>> my\_users**

**[{'Name': 'Adetayo Bero'}, {'John': 'Doe'}]**

***List user from the list using index value***

***>>> my\_users[1]***

***{'Name': 'John Doe'}***

***>>>***

***#####################***

**ITERATIONS and LOOPS. Just like above it goes through each individual items in the record.**

**Create a list with numbers in the list. And list each index manually through looping**

**>>> my\_list = [1,2,3,4,5]**

**>>> my\_list[0]**

**1**

**>>> my\_list[1]**

**2**

**>>> my\_list[2]**

**3**

**>>> my\_list[3]**

**4**

**>>> my\_list[4]**

**5**

**Now let’s do it automatically using for loop. for and in are inbuilt into python. my\_var is the variable of my\_list items I created.**

**The 3 dots after I pressed enter means I can continue my statement going forward.**

**>>> for my\_var in my\_list:**

**...**

**When you continue a statement in python you have to use space. i.e 4 spaces using space bar**

>>> for my\_var in my\_list:

... print(my\_var)

...

1

2

3

4

5

>>>

The above prints out the value in the list.

**Another iteration/loop is** for i in “abc”

Indentation error means something with spacing

Print(i)

>>> for i in "abc"

... print(i)

...

a

b

c

>>>

**You cannot loop through a number using the same method i.e for x in 10:**

**To loop through numbers i.e 0 to 10 you’ll use a feature called range**

**>>> for x in range (0, 10):**

**... print(x)**

**...**

**0**

**1**

**2**

**3**

**4**

**5**

**6**

**7**

**8**

**9**

**>>>**