List = [] ordered and changeable :

Some methods of lists :

Append() , extend(),remove(),sort(),reverse(),clear(),count(),index(), insert()

And pop().

For indexing:

arr[start:end:steps]

Set = {} unordered and immutable :

Slicing can’t be done

Has only immutable data

Items must be unique

Some methods of sets :

Clear(),union(),add(),copy(),remove(),discard(),pop(),update(),…

Tuple = () ordered and unchangeable:

Faster than lists.

Some methods of tuples() :

Count(),index().

dictionary = a collection of {key:value} pairs # ordered and changeable.

Dictionary items consist of key : value .

Key have to be immutable and unique.

Value can be any type of data.

Dictionary is not ordered so you can access it,s elements by key.

Some methods of dictionary :

Print(dict.keys()) ## to get all keys in your dictionary.

Print(dict.values()) ## to get all values in your dictionary.

Items() , clear(), update(), copy(), setdefault(), pop(), popitem(),…

We can use dir() method with any of those data structures to know every Single method of it and help() method to help us knowing the function of Each method.

---

module :

a file containing code you want to include in your program .

use ‘import’ to include the module you want (built_in or made by you).

Or we can use some methods or constants from the module not all of it :

From module_name import module_constant or module_methods

Also we can give the module an alies .

Like : import numpy as np .

Random modcule :

Some methods of it:

Random.randint(num1,num2) # to generate a random integer number between Num1 and num 2

random.shuffle() # to rearrange the order of elements of array or something

random. Choice().

Functions : a block of reusable code.

To write a function we use word **def** after in the **function name**

Like : def function_name():

Some code.

Return : statement used to nd the function and send the result back to the caller .

Now to call a function we write the function name and pass some data to it

If the function takes any arguments

If we don’t know how many arguments the user will input we just could use *

Like : def show_details(name,*skills):

print(f"hello {name} your skills are")

for skill in skills :

print(skill)

but here the data type of skills is tuple.

show_details("mahmoud","python","c")

show_details("marwan","python","c","html","css")

defult arguments : default values for certain parameters and this make the function more readable and reduce the argument numbers .

preceded arguments : an argument preceded by an identifier

like : print(“hello”,end” ”)

here the word **end** precedes the argument value.

file handling.

"a" Append Open File For Appending Values, Create File If Not Exists

"r" Read [Default Value] Open File For Read and Give Error If File is Not Exists

"w" Write Open File For Writing, Create File If Not Exists

"x" Create Create File, Give Error If File Exists

copyfile() = copies contents of a file

copy() = copyfile() + permission mode + destination can be a directory

copy2() = copy() + copies metadata (file's creation and modification times) import shutil