- In Python every object in a list can be of different data type
- List == []
- Using the brackets notation [] we can access and play around with lists same as strings
- A list can be modified
- Enumerate function gives index information stored in a tuple ➔ (0, ‘a’)
- Tuples are read only
- A simple, anonymous, one-line function which we can pass to other functions is called as Lambdas
- We can have a list of lists as well
- A Matrix is a 2-dimensional list
- Zeros = [0] * 100 we get a list of 100 zeroes as list items
- Lists can be concatenated
- List() function is used for the creation of list items by iteration
- Negative index like -1 will give last item of the list
- :: is step in Brackets notation ➔ [::2]
- [::-1] causes the reversal of a list
- You can assign variables to different list items
- List unpacking is assigning list items to multiple variables
- *parameter ➔ Python interprets it as a collection of arguments and store it in a list
- We can only unpack a list if we have the sufficient variables to accommodate list items
- Append method is used to insert something at end of a list
- Insert method is used to insert something at a specific position in a list
- Pop method is used to remove something at end of a list
- Remove is used to remove by name
- Del keyword is also good practice
- Clear Method is used to vanish all
- Count method gives the number of occurrence of a list item
- Index method gives the index of a certain list item
- In method is used to verify that a certain item is present in the list
- Sorted gives a new list while sort method modifies because it gets implement over the object list
- Map function gives off an iterable
- Map function uses a lambda function to iterate over lists to produce new lists
- Lambda function is used to reference other functions
- Filter function uses a lambda function to iterate over lists to produce new lists as per certain criteria of filtering list items

- List Comprehensions are python-only and are the best practices to be used instead of Map & Filter functions
- List Comprehension works on one list
- Zip function is used to comprehend two lists in an expanded fashion of tuples
- Stack resembles a stack of real world books
- Stack works on the principle of LIFO ➔ Last in- First out
- Which means if you have a stack of books around, the last book you placed on the top of all books, you can remove that very first out
- A perfect real world example of Stack is the browsing history of your browser
- We can implement Stack in python by using a List
- An empty list is a falsy value
- Queue works on the principle of FIFO ➔ First in-First out
- It resembles a real world queue or line
- Lists are used to implement queues
- Queue has a limitation regarding memory & performance due to the left shifting of each list item when it comes to large data, so we use deque to cope up with all the odds
- Deque auto shifts the list items
- A tuple is basically a read only list
- Tuple function is used to convert a list into a tuple
- We cant modify a tuple however we can concatenate two tuples and multiply a tuple for generation
- A general rule of thumb is we only use tuples in real world scenarios when we try to avoid accidental mishandling of data in software
- When defining a tuple you can exclude the parenthesis
- Swapping variables in python is actually unpacking tuples under the hood
- Arrays perform a bit faster when large collection of numbers is involved as compared to lists
- In Arrays all the items should have same data type unlike lists
- Typecode is a string of single character which determines data type of your array
- Sets is data structure which is actually a collection of items with no duplicates
- Sets are used in powerful mathematical use-cases
- Sets are denoted by curly braces {}
- Sets don't support indexing properties
- Dictionary is actually a collection of Key-Value pairs
- A Real world example of dictionary is the phone book in which we have name of a person as a key and his/her contact details as value
- We can use comprehensions with lists, sets and dictionaries

- Comprehensions can be used instead of 1- Declaring 2- looping 3- Creating
- Tuple Comprehension expressions are basically Generators or Generator objects
- We use generators when we have an infinite stream of data
- Generator has a size of only 120 bytes
- Lists stores all the items whereas generator just spill out a new value at each iteration that’s why we can’t access to all the items working on
- We use unpacking variable to take out individual items from any iterable (*unpacking)
- Pretty printing is a useful module for print purposes