

- -> notebooks from this lecture: <https://github.com/ine-rmotr-curriculum/ds-content-python-under-10-minutes>
- -> functions
- -> def <- this is the keyword to define functions
- -> function parameters use the return keyword
- -> the function should return something
- -> this is the result which is returned / written down
- -> the function should still return something
- -> even if you don't use the return keyword, it will still return something -> none
- -> **passing parameters**
 - -> variable lengths arguments <- you can pass as many arguments as you want
 - -> modulo / operations
 - -> standard operations
 - -> boolean operators -> >= for example
 - -> there are dynamically types operators
 - -> not and or operators
 - -> defining them according to the operators
 - -> else / if
 - -> Python does not have a switch statement
- -> **to loop through a list**
 - -> you can loop through arrays in Python
 - -> we don't have lists in Python
 - -> we iterate over collections
 - -> correct a range element and iterate over it
 - -> for each
 - -> for name in names
 - -> the names are associated with elements in the list
- -> **while loops**
 - -> for loops are preferred
 - -> this avoids infinite loops
- -> **collections in Python**
 - -> primitive / fundamental ones
 - -> three elements -> one integer, one string and one boolean
 - -> Python supports mixed types in the collections
 - -> avoid mixing types and connections
 - -> revisit the code if you have too many different types
 - -> Python is zero indexed
 - -> you can index starting from the end
 - -> this is done using negative indices
 - -> you can also use .append
 - -> you can add new elements to a tuple, for example
 - -> you can check if the element is part of the list using a boolean test
 - -> you can't modify a tuple
- -> **dictionaries**
 - -> a two value map
 - -> hash tables in JavaScript
 - -> associating values to names
 - -> he creates a list, copies the elements and then stored the information about the customers in

that list

- -> the problem with this is that the customers are accessed using indices, rather than keys in a dictionary
- -> dictionaries are collections of values
- -> you don't just index it by the position
- -> the name, email, age, and is the customer is a subscriber or not
- -> in a dictionary, we can access the element by either of those (rather than just the index of the customer / person)

- -> **sets**

- -> these are a type of data structure
- -> these are used commonly in Python, but not as commonly in other languages
- -> sets and dictionaries are unordered data structures <- you don't know the order of those data structures
- -> sets are bags which contain elements -> there is no order to them
- -> the membership operation
- -> the process of checking something is quick
- -> when you correct the set, you correct the elements
- -> elements aren't repeated in a set -> it contains unique values
- -> unique values are simple to check -> for example by checking a membership organisation
- -> sets are useful when checking for members

- -> **question**

What is the main difference between lists and tuples in Python?

- options

- Tuples are immutable. <- This one
- Lists are ordered.
- Tuples are unordered.