# Collections Module

## LESSON 5

CODE FIRST GIRLS

NANODEGREE → ENGINEERING MODULE

# AGENDA

**01** Collections in Python

**02** Collections Module in Python

**03** Specialised Collections data types

**04** Practice

# COLLECTIONS IN PYTHON

{ A **Container (***or collection***)** is an object that is used to store different objects and provide a way to access the contained objects and iterate over them }

LIST

TUPLE

SET

DICTIONARY

- A container stores collection of data

# COLLECTION MODULE IN PYTHON

COLLECTIONS MODULE

The **Collections module** in Python provides different types of containers.

COUNTER

NAMEDTUPLE

ORDEREDDICT

DEQUE

DEFAULTDICT

USERDICT

CHAINMAP

USERLIST

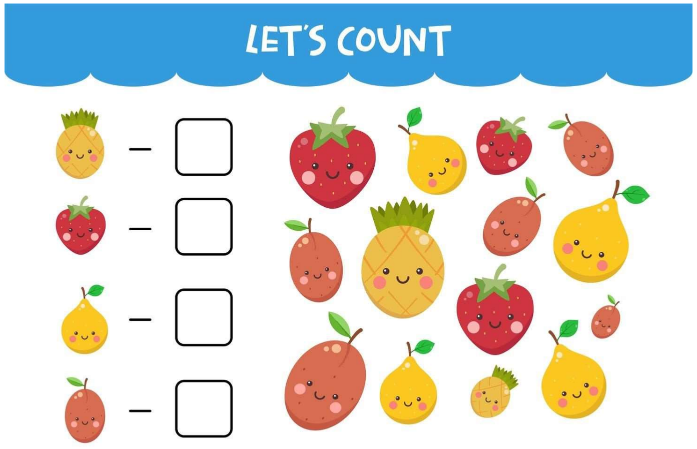
USERSTRING

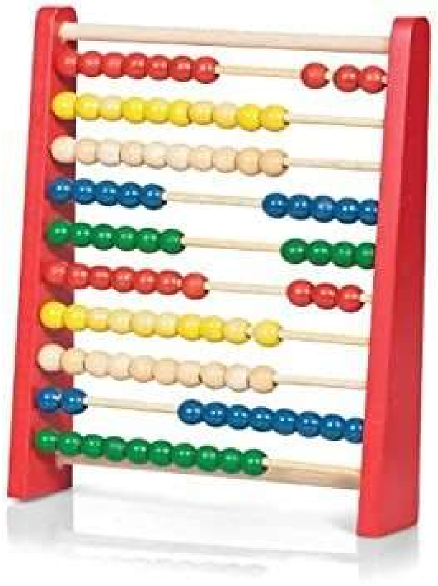
- **New data structures**

# COUNTER

- A counter is a sub-class of the dictionary.
- It keeps the count of the elements in an iterable in the form of an unordered dictionary.
- Key represents the element in the iterable and value represents the count of that element in the iterable.

```
class collections.Counter([iterable-or-mapping])
```

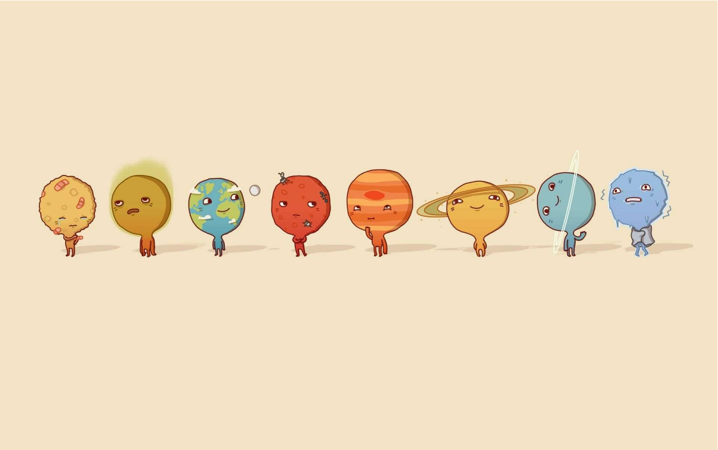




# OrderedDict

- An OrderedDict is also a sub-class of dictionary.
- But unlike dictionary, it remembers the order in which the keys were inserted.

```
class collections.OrderedDict()
```



# DefaultDict

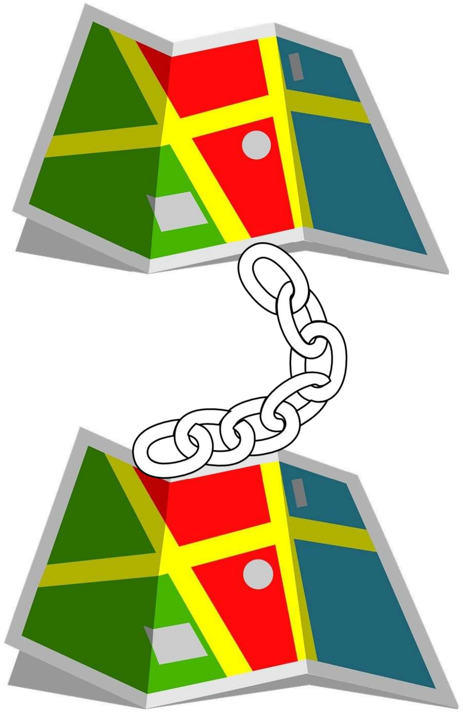
- DefaultDict is also a sub-class to dictionary.
- It is used to provide some default values for the key that **does not exist** and **never raises a KeyError**.
- *default_factory* is a function that provides the default value for the dictionary created.
- If this parameter is absent then the KeyError is raised.

```
class collections.defaultdict(default_factory)
```

# ChainMap

- A ChainMap encapsulates many dictionaries into a single unit and returns a list of dictionaries..
- Values from the whole ChainMap can be accessed using the key name.
- Keys and values can also be accessed by using the keys() and values() method.
- A new dictionary can be added by using the **new_child()** method.
- The newly added dictionary is added at the beginning of the ChainMap.

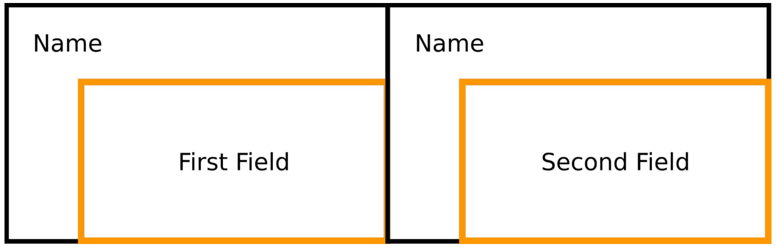
```
class collections.ChainMap(dict1, dict2)
```



# NamedTuple

- A NamedTuple returns a tuple object with names for each position.
- Consider a tuple names student where the first element represents first name, second represents last name and the third element represents the course.
- Suppose for calling first name instead of remembering the index position you can actually call the element by using the first name argument.

```
class collections.namedtuple(typename,
field_names)
```



# DEQUE (Doubly Ended Queue)

- Deque is the optimized list for quicker append and pop operations from both sides of the container.

- Elements in deque can be inserted from both ends.

- To insert the elements from right append() method is used and to insert the elements from the left appendleft() method is used.

- Elements can also be removed from the deque from both the ends.

- To remove elements from right use pop() method and to remove elements from the left use popleft() method.

```
class collections.deque(list)
```

# UserDict | UserList | UserString

- When an object has ‘User’ prefix in its name, it means that it acts as a wrapper around the original object (a list, a dict, a string).

- It is used when someone wants to create their own list/dict/string with **some modified or additional functionality**.

```
class collections.UserDict([initial data])

class collections.UserList([list])

class collections.UserString(seq)
```

**DEMO & EXERCISES**

# COLLECTIONS MODULE EXERCISES

CODE FIRST GIRLS

# THANK YOU!