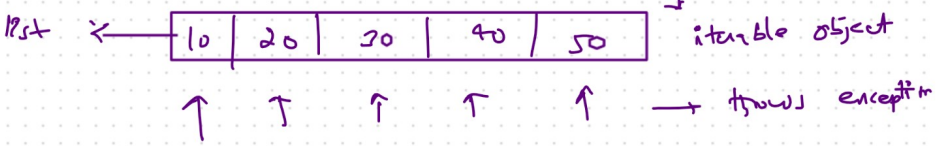
Iterators & Generators

Iterator: An iterator is an object in python that allows you to traverse through a sequence (list, tuple, string) one element at a time

It has 2 methods

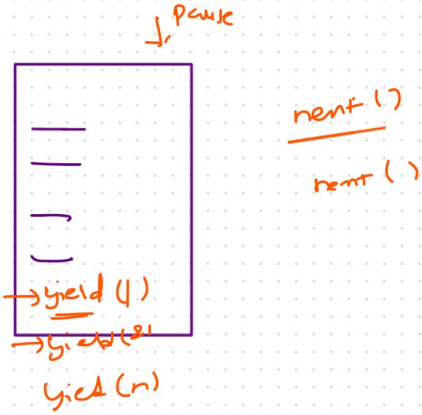
(1) __iter__() : returns the iterator object

(2) __next__() : return the next value in the sequence, if there no more items it raises an exception [StopIteration]



Generator:

def: It is a special type of iterator, we can use to create custom iterators using a function with help of yield keyword, each time yield is called the function "pause" & remembers its state, resuming from the next time it is called



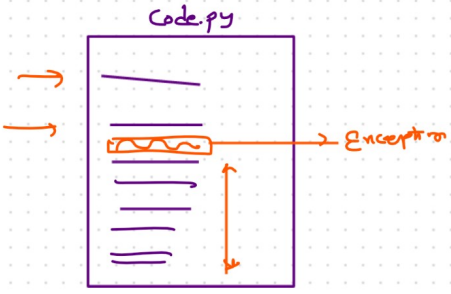
When to use?

(1) When working with large dataset or stream of data

(2) When you need to process data on-the fly without storing it all in memory

(3) D.E when you want to create pipelines for data transformation

Exception Handling



Try - Except Block :

The fundamental mechanism for handling exceptions in python

```
try :
    ...
except Exception as e :
    print(e)
```

- Multiple Except Blocks
- Grouped Exceptions
- Common Exception
- Else Clause
- finally Clause
- Raising Exception
- Custom Exceptions