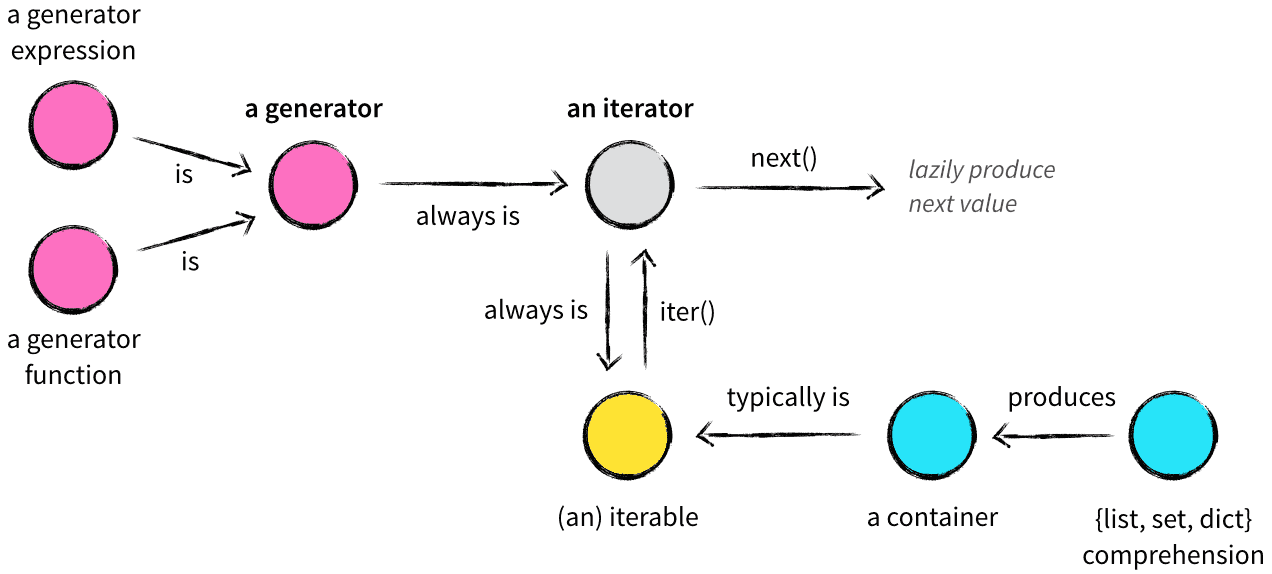
1. What is the main difference between list and Iterable? **Mark: 2**

**Ans:** Iterable is a simple representation of a series of elements that can be iterated over. It does not have any iteration state such as a "current element". Instead, it has one method that produces an Iterator. An Iterator is the object with iteration state

1. What methods can be used to find if the variable name exists?  **Mark: 2**

**Ans:** locals and globals

1. What is the difference between Iterables, iterator and generator? Provide an example for each **Mark: 6**



Site: <http://nvie.com/posts/iterators-vs-generators/>

1. Write a generator for Fibonacci series. **Mark: 5**

def fib(n):

a, b = 0, 1

for \_ in range(n):

yield a

a, b = b, a + b

num = 10

print(list(fib(num)))

1. Print the following using decorators, where only ‘**A-Shanti**’ is printed by the function. **Mark: 4**

{Hello}

A-Priya

Manish Gupta

Neha

A-Shanti

(/Hello}

1. What is Insertion Sort? Implement it in python **Mark: 2+5**
2. Convert the following codes in functional programming. **Mark: 6**

def flatten\_list(a, result=None):

"""Flattens a nested list.

>>> flatten\_list([ [1, 2, [3, 4] ], [5, 6], 7])

[1, 2, 3, 4, 5, 6, 7]

"""

if result is None:

result = []

for x in a:

if isinstance(x, list):

flatten\_list(x, result)

else:

result.append(x)

return result

1. Write a function flatten\_dict to flatten a nested dictionary by joining the keys with “.” character. **Mark: 3**

>>> unflatten\_dict({'a': 1, 'b.x': 2, 'b.y': 3, 'c': 4})

{'a': 1, 'b': {'x': 2, 'y': 3}, 'c': 4}

1. Write a function treemap to map a function over nested list. **Mark: 6**

>>>treemap(lambda x: x\*x, [1, 2, [3, 4, [5]]])

[1, 4, [9, 16, [25]]]

1. Please write the class for the following code: **Mark: 3**

>>>Class XYZ:

>>> <MAGIC CODE>

>>> x = XYZ()

>>> x.name = “Mayank”

>>> x.a = “Hello”

>>> print(x.a)

Hello Mayank

>>> x.a(“World”)

>>> print(x.a)

Hello World

Ans:

class XYZ: