# Python Practice Questions on Magic Methods, Itertools, Map, Generators, Iterators

## Magic Methods (4 Questions)

1. What is the purpose of \_\_init\_\_() magic method in a Python class?

The \_\_init\_\_() method is the constructor in Python. It is called automatically when a new object of a class is created. It is used to initialize the instance variables.

1. How does \_\_str\_\_() differ from \_\_repr\_\_() in Python classes?

class Person:

def \_\_init\_\_(self, name):

self.name = name

1. Write a simple example of overloading the \_\_add\_\_() magic method.

class Point:

def \_\_init\_\_(self, x, y):

self.x = x

self.y = y

def \_\_add\_\_(self, other):

return Point(self.x + other.x, self.y + other.y)

1. Which magic methods are required to make an object context manager?

\_\_enter\_\_(self)

\_\_exit\_\_(self, exc\_type, exc\_val, exc\_tb

## Itertools (4 Questions)

1. What is the use of itertools.product()? Give an example.

It returns the Cartesian product of input iterables

from itertools import product

print(list(product([1, 2], ['a', 'b'])))

1. How does itertools.permutations() differ from itertools.combinations()?

from itertools import \*

print(list(permutations('AB', 2)))

print(list(combinations('AB', 2)))

1. Explain the purpose of itertools.chain().

It is used to convet multiple iteration into a single iteration

from itertools import chain

print(list(chain([1, 2], [3, 4])))

1. Write a code snippet using itertools.cycle().

from itertools import cycle

count = 0

for item in cycle(['A', 'B']):

print(item)

count += 1

if count >6:

break

## Map Function (4 Questions)

1. How does the map() function work in Python? What does it return?

The map() function applies a given function to each item of an iterable and returns a map object (which is an iterator).

1. Write a code snippet to add two lists element-wise using map().

A = [1,2,3,4]

B=[5,6,7,8]

Res = map(lambda x+y : a,b)

Print(list(res))

1. What is the difference between map() and filter() functions?

map() transforms items and filter() selects items based on a condition

1. Can map() work with lambda functions? Give an example.

Yes .

Example : A = [1,2,3,4]

B=[5,6,7,8]

Res = map(lambda x+y : a,b)

Print(list(res))

## Generators (4 Questions)

1. What is a generator function in Python? How is it defined?

A generator function yields values one at a time using the yield keyword and maintains state between calls.

1. How does yield differ from return in a function?

yield pauses the function saving its state, and resumes from there on the next call and return ends the function completely and sends back a value.

1. Write a simple generator to yield even numbers up to 10.

Def evenno():

For I in range (11):

If i%2==0

Yield i

for num in even\_no():

print(num)

1. What happens if you call next() on a generator after it is exhausted?

StopIteration exception is raised.

## Iterators (4 Questions)

1. What is an iterator in Python? How is it different from an iterable?

An iterator is an object with a \_\_next\_\_() method and remembers its state.  
An iterable is an object that can return an iterator using \_\_iter\_\_().

1. Which two magic methods must be implemented for a class to be an iterator?

\_\_iter\_\_(self) and \_\_next\_\_(self)

1. Write a simple iterator class that returns numbers from 1 to 5.

class CountFive:

def \_\_init\_\_(self):

self.num = 1

def \_\_iter\_\_(self):

return self

def \_\_next\_\_(self):

if self.num <= 5:

val = self.num

self.num += 1

return val

else:

raise StopIteration

1. How does the iter() function work on a list?

lst = [1, 2, 3]

it = iter(lst)

print(next(it))

print(next(it))

print(next(it))