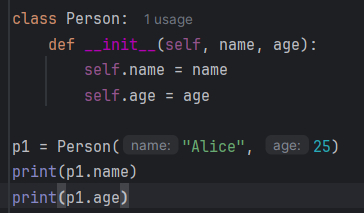
## DAY 6: MORNING ASSESSMENT

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

It sets up the initial state of the object.

It allows you to pass arguments when creating an object, and assigns those values to object attributes.



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

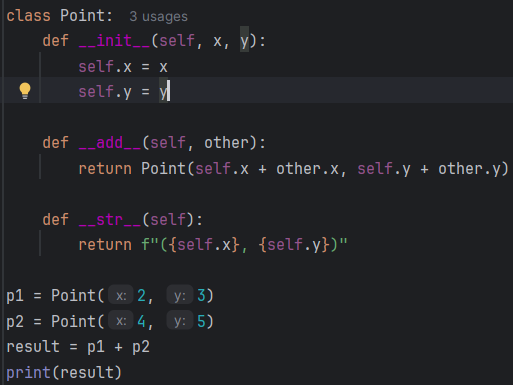
1. \_\_str\_\_() – For Users (Readable)

* Called by str(object) or when using print(object)
* Goal: Return a user-friendly or nicely formatted string

2. \_\_repr\_\_() – For Developers (Unambiguous)

* Called by repr(object) or in interactive interpreter
* Goal: Return a developer-friendly, often unambiguous string (ideally valid Python code to recreate the object)

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

output: (6, 8)

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

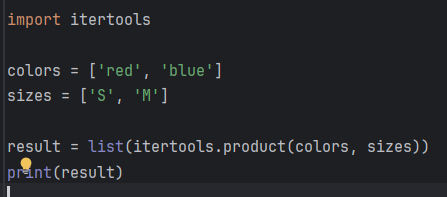
\_\_enter\_\_(self): Called at the start of the with block

\_\_exit\_\_(self, exc\_type, exc\_value, traceback): Called at the end of the with blockand is used to clean up

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

itertools.product() is used to compute the Cartesian product of input iterables. It returns tuples containing every possible combination of elements — similar to nested loops.

itertools.product(\*iterables, repeat=1)



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

Use permutations() when order matters (like arranging letters).

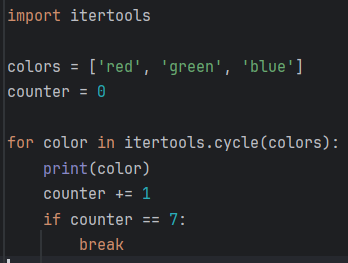
Use combinations() when order doesn't matter (like choosing lottery numbers).

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

The itertools.chain() function is used to combine multiple iterables into a single sequence, allowing you to iterate over them as if they were one.

itertools.chain(iterable1, iterable2, ...)

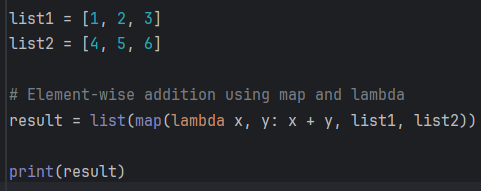
8. Write a code snippet using itertools.cycle().



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

The map() function applies a given function to each item in an iterable and returns a map object.

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

  
output: [5, 7, 9]

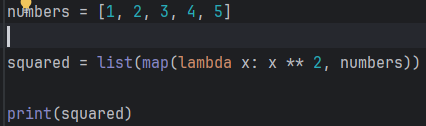
11. What is the difference between map() and filter() functions?

Use map() when you want to transform data.

Use filter() when you want to select specific data.

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

Yes, map() works with lambda functions in Python.

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

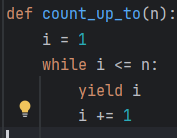
13. What is a generator function in Python? How is it defined?

A generator function is a special type of function that yields values one at a time, instead of returning them all at once

Uses the yield keyword instead of return.

Automatically creates an iterator.

Keeps its state between calls.

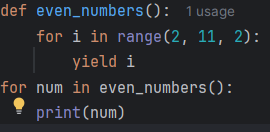


14. How does yield differ from return in a function?

Use return when you want to send back one final result.

Use yield when you want to generate a sequence of results, especially in loops or large datasets.

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



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

StopIteration exception: This signals that the generator is exhaused and no more items to yield

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

An iterator is an object that represents a stream of data — it returns one item at a time using the next() function until all items are exhausted.

An iterable is any object you can loop over, such as lists, tuples, strings, sets, and generators.

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

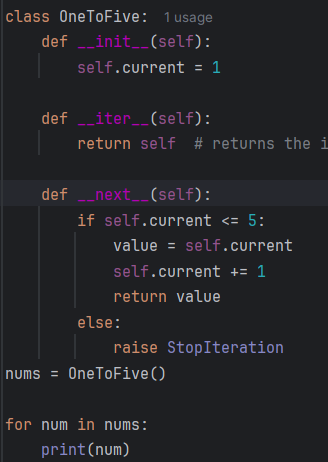
1. \_\_iter\_\_(self)

* Should return the iterator object itself.
* Called when an iteration is initialized

2. \_\_next\_\_(self)

* Should return the next item in the sequence.
* Must raise a StopIteration exception when items are exhausted.

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



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

The built-in iter() function is used to get an iterator from an iterable