

The screenshot shows the PyCharm IDE interface. The top toolbar includes icons for Project, Run, Debug, Settings, and a minus sign. The top tabs show 'cc.py', 'test.py', and 'TEST01.py'. The left sidebar displays the project structure for 'pyqt5' at 'D:\pycharm\pyqt5', including folders 'build' and 'dist', and a 'venv library root' containing files like 'apk_path', 'bg.jpg', 'cc.ico', 'cc.py', 'cc.spec', 'icon.jpg', 'name.txt', and 'script_path'. The main editor window shows the code for 'TEST01.py' with line numbers 1 through 11. The code includes a '@profile' decorator, a 'test()' function that iterates over a list [1, 2, 3, 4] and prints each element, and a main block that calls 'test()' if the script is run directly. The bottom terminal window shows the output of the memory profile, including line numbers, memory usage, and line contents. A red arrow points to line 6 in the terminal output.

```
1 @profile
2 def test():
3     list = [1, 2, 3, 4]
4     it = iter(list)
5     for i in it:
6         print(i)
7
8
9 if __name__ == "__main__":
10     test()
11
if __name__ == "__main__":
```

Terminal: Local x +

2
3
4
Filename: TEST01.py

Line #	Mem usage	Increment	Line Contents
1	14.285 MiB	14.285 MiB	@profile
2			def test():
3	14.285 MiB	0.000 MiB	list = [1, 2, 3, 4]
4	14.285 MiB	0.000 MiB	it = iter(list)
5	14.293 MiB	0.000 MiB	for i in it:
6	14.293 MiB	0.008 MiB	print(i)

14.293-14.285=0.008MiB

The screenshot shows the PyCharm IDE interface. The top toolbar includes icons for Project, Run, Debug, Settings, and a minus sign. The top tabs are labeled 'pyqt5', 'test.py', and 'TEST01.py'. The left sidebar shows the Project view with a tree structure: 'pyqt5' (D:\pycharm\pyqt5) containing 'build', 'dist', 'cc.exe', 'venv library root', 'apk_path', 'bg.jpg', 'cc.ico', 'cc.py', 'cc.spec', 'icon.jpg', 'name.txt', and 'script_path'. The main editor window displays the code in 'test.py':

```
1 @profile
2 def test():
3     list = [1, 2, 3, 4]
4     for i in list:
5         print(i)
6
7 if __name__ == "__main__":
8     test()
9
```

The bottom panel shows the Terminal with the output of the script. The first part of the terminal output is:

```
1
2
3
4
Filename: test.py
```

The second part of the terminal output is a table showing memory usage:

Line #	Mem usage	Increment	Line Contents
1	14.281 MiB	14.281 MiB	@profile
2			def test():
3	14.281 MiB	0.000 MiB	list = [1, 2, 3, 4]
4	14.289 MiB	0.000 MiB	for i in list:
5	14.289 MiB	0.008 MiB	print(i)

The bottom status bar shows the current environment: '(venv) D:\pycharm\pyqt5>'. The left sidebar also shows 'Structure' and 'Favorites' views.

14.289-14.281=0.008MiB

解释：迭代器其实就是一个for循环

The screenshot shows the PyCharm IDE with a project named 'pyqt5'. The file explorer on the left lists various files and folders. The main editor displays the code in 'test.py':

```
1 @profile
2 def test():
3     for line in open("./test.txt").readlines():
4         print(line)
5
6 if __name__ == "__main__":
7     test()
8
```

The terminal window shows the output of the memory profiler:

```
4
5
Filename: test.py

Line #    Mem usage    Increment    Line Contents
=====
1      14.340 MiB    14.340 MiB    @profile
2
3      14.355 MiB     0.008 MiB        for line in open("./test.txt").readlines():
4      14.355 MiB     0.008 MiB            print(line)

(venv) D:\pycharm\pyqt5>python -m memory_profiler test01.py
1
```

A red arrow points to the memory usage values for lines 1 and 4, highlighting the difference of 0.015 MiB.

14.355-14.340=0.015M

The screenshot shows the PyCharm IDE interface. The top toolbar includes icons for Project, Run, Debug, and other standard IDE functions. The left sidebar displays the Project Explorer with the following structure:

- pyqt5 D:\pycharm\pyqt5
 - build
 - dist
 - venv library root
 - apk_path
 - bg.jpg
 - cc.ico
 - cc.py
 - cc.spec
 - icon.jpg
 - name.txt
 - script_path
 - simulator_path

The main editor window shows the file TEST01.py with the following code:

```
1 @profile
2 def test():
3     for line in open("./test.txt"):
4         print(line)
5
6
7 if __name__ == "__main__":
8     test()
9
```

The bottom panel shows the Terminal window with the output of the program:

```
2
3
4
5
Filename: TEST01.py
```

Below the terminal, the memory usage profile is displayed as a table:

Line #	Mem usage	Increment	Line Contents
1	14.297 MiB	14.297 MiB	@profile
2			def test():
3	14.312 MiB	0.008 MiB	for line in open("./test.txt"):
4	14.305 MiB	0.008 MiB	print(line)

A red arrow points to the row for line 4, which shows a memory usage of 14.305 MiB and an increment of 0.008 MiB.

14.305-14.297=0.008M

The screenshot shows the PyCharm IDE with a project named 'pyqt5'. The left sidebar displays the file explorer with a 'venv library root' folder containing various files like 'apk_path', 'bg.jpg', 'cc.ico', 'cc.py', 'cc.spec', 'icon.jpg', 'name.txt', 'script_path', and 'simulator_path'. The main editor window shows a Python script with the following code:

```
1 @profile
2 def test():
3     with open("./test.txt") as f:
4         for line in f:
5             print(line)
6
7
8 if __name__ == "__main__":
9     test()
10
```

The terminal window at the bottom shows the command `(venv) D:\pycharm\pyqt5>python -m memory_profiler TEST01.py` and its output:

```
a
2
Filename: TEST01.py

Line #    Mem usage    Increment    Line Contents
=====
1      14.301 MiB     14.301 MiB   @profile
2
3      14.301 MiB      0.000 MiB       with open("./test.txt") as f:
4      14.309 MiB      0.000 MiB           for line in f:
5      14.309 MiB      0.008 MiB               print(line)
```

A red arrow points to line 5 in the terminal output, highlighting the memory increment of 0.008 MiB for the `print(line)` statement.

14.309-14.301=0.008M

解释：使用对象本身作为迭代对象，能明显节省内存