|  |  |
| --- | --- |
|  | ***Study*** |

Read chapter 13 of the following book:

Learning Python:

<http://www.dsf.unica.it/~fiore/LearningPython.pdf>

Then answer the quiz in page 414

1. What are the main functional differences between a while and a for?

For is for a range, while is for a repeated loop, takes a while longer to do a for loop

2. What’s the difference between break and continue?

Break gets out of while loop

Continue gets back to beginning of loop

3. When is a loop’s else clause executed?

After a loop ends

4. How can you code a counter-based loop in Python?

Reduce, increase the counter to end a while /for loop

5. What can a range be used for in a for loop?

(Step, , skip)

Answer from the book

1. The while loop is a general looping statement, but the for is designed to iterate

across items in a sequence or other iterable. Although the while can imitate the

for with counter loops, it takes more code and might run slower.

2. The break statement exits a loop immediately (you wind up below the entire

while or for loop statement), and continue jumps back to the top of the loop (you

wind up positioned just before the test in while or the next item fetch in for).

3. The else clause in a while or for loop will be run once as the loop is exiting, if the

loop exits normally (without running into a break statement). A break exits the

loop immediately, skipping the else part on the way out (if there is one).

4. Counter loops can be coded with a while statement that keeps track of the index

manually, or with a for loop that uses the range built-in function to generate successive

integer offsets. Neither is the preferred way to work in Python, if you need

to simply step across all the items in a sequence. Instead, use a simple for loop

instead, without range or counters, whenever possible; it will be easier to code and

usually quicker to run.

5. The range built-in can be used in a for to implement a fixed number of repetitions,

to scan by offsets instead of items at offsets, to skip successive items as you go, and

to change a list while stepping across it. None of these roles requires range, and

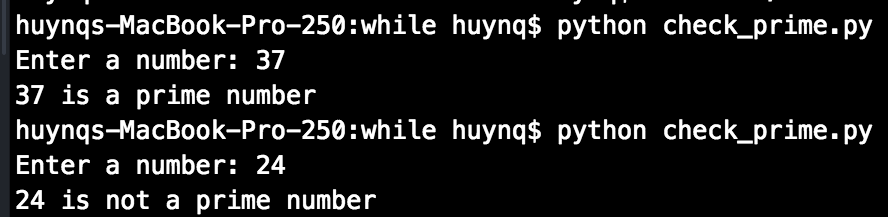
most have alternatives—scanning actual items, three-limit slices, and list comprehensions

are often better solutions today (despite the natural inclinations of ex–C

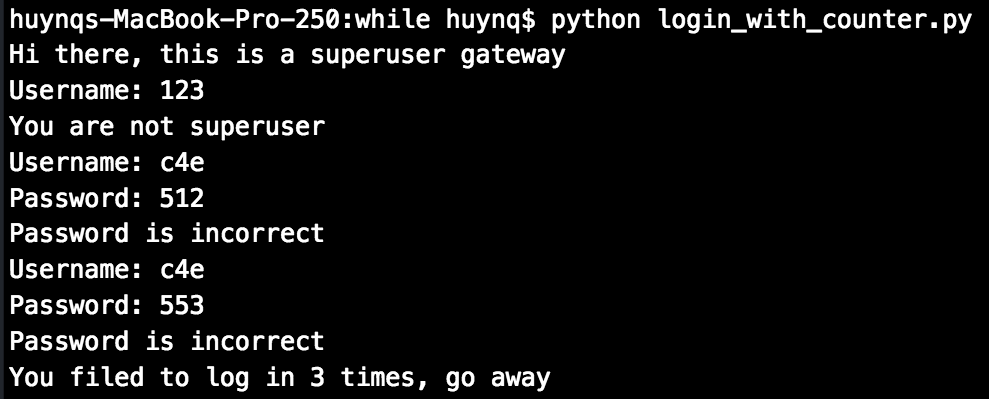
programmers to want to count things!).

|  |  |
| --- | --- |
|  | ***Serious exercises*** |

**Exercise 1**: Write a program to check whether a number is a prime number



**Exercise 2**: Modify the username password in previous session to allow users login at maximum 3 times



**Exercise 3 (Optional)**: Modify guess\_my\_number.py to the opposite scenario, you think of a number and then the program takes a guess, then you tell it where its guess is correct, smaller or larger than the number you’re thinking

