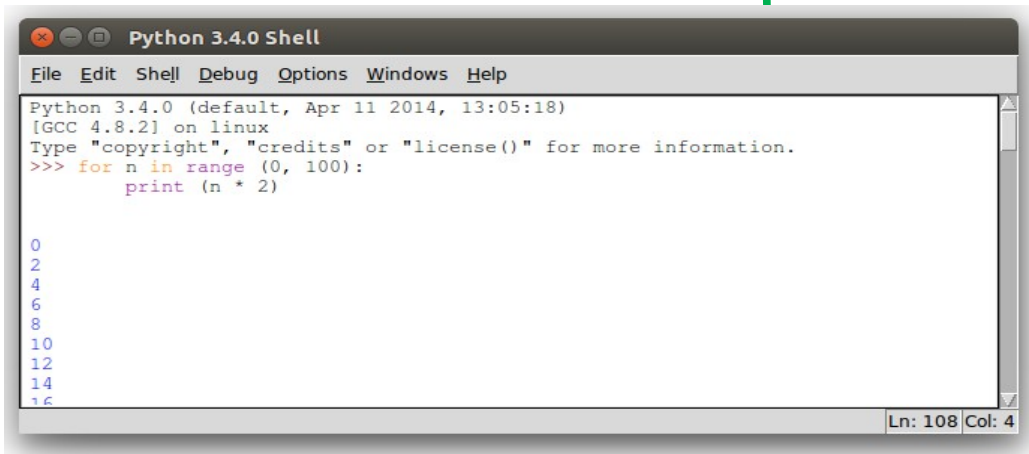


Loops



```
Python 3.4.0 Shell
File Edit Shell Debug Options Windows Help
Python 3.4.0 (default, Apr 11 2014, 13:05:18)
[GCC 4.8.2] on linux
Type "copyright", "credits" or "license()" for more information.
>>> for n in range(0, 100):
      print(n * 2)
0
2
4
6
8
10
12
14
16
Ln: 108 Col: 4
```

Do you know your two times table up to 100?

Loops can be used to repeat instructions. We are going to use a kind of loop called a "for loop" to repeat a command 100 times and print out the 2 times table up to 100 times.

Enter the command:

`for n in range(0,100):` and press Enter.

You will notice that the cursor has become indented, just like when we made decisions using "if" statements. Every indented command that is underneath the line "for n in range..." will be repeated.

We will now type the command: `print(n * 2)`.

The commands on your screen should now look like this:

```
for n in range(0, 100):
    print(n * 2)
```

Now press Enter. Nothing happened! This is because Python is waiting to see if you are going to enter any more commands to be repeated in this for loop. Press Enter again and you should see the answers of the two times table print right up to the answer of 100 times 2.

Can you see how you can use commands like this to display the 13 times table?

Lets try the commands

```
for n in range(0, 100):
    print(n * 13)
```

What would you need to do to output any other times table, for example the 5 times table or 7 times or even the 27 times table?

We can combine this looping method with random lists of words and random choices to make a program that can seem to create original stories.

Project 4:

Python story maker

Program Recipes: Silly story generator.

1. We are going to use `random.choice`, some lists of words and loops to make a program that can create stories. You will use the program file:
`"silly_storyteller.py"`

2. This is quite a long program. Your teacher will explain to you how the program works. Read each line of the program and listen carefully while the teacher explains what the program is doing.

3. Run the program by clicking Run → Run Module (or press F5).

The program will ask you to input answers to some questions like:

What is your name? Bob

Hello Bob are you sitting comfortably?

Yes

Then we shall begin.

Would you like a story about a boy, a girl, an alien, a monster or something else? alien

What day is it? Wednesday

5. The program should now display a silly story.

Run the program a few more times, you will notice that the stories become quite repetitive.

We need to improve this, one way to do this is to add more words to the lists of: boy's names, girl's names, other names, places, emotions, senses, thoughts and so on.

For example, we can add to the list of places like this:

`places = ["castle", "town"]` can be added to like this: `places = ["castle", "town", "beach", "island"]`

Add as many suitable words as you can to all of the different lists, run the program more times.

Can you see how adding the right words to the right lists helps your program to seem to become a better story teller?

6. Your story teller program combines just three computer programming principles, loops, lists and random choices. Using these few basic principles it can seem to generate an original and creative (if rather silly and childish) story.

Look carefully at the lines of Python code that are in the for loop that combine random choices of the different word lists together to make two story paragraphs. Can you think of any ways that you can improve the range of stories your program can generate by adding more lines to create more paragraphs? Perhaps you could also add further lists of different kinds of word as well? Discuss some of these ideas with your teacher and class mates.