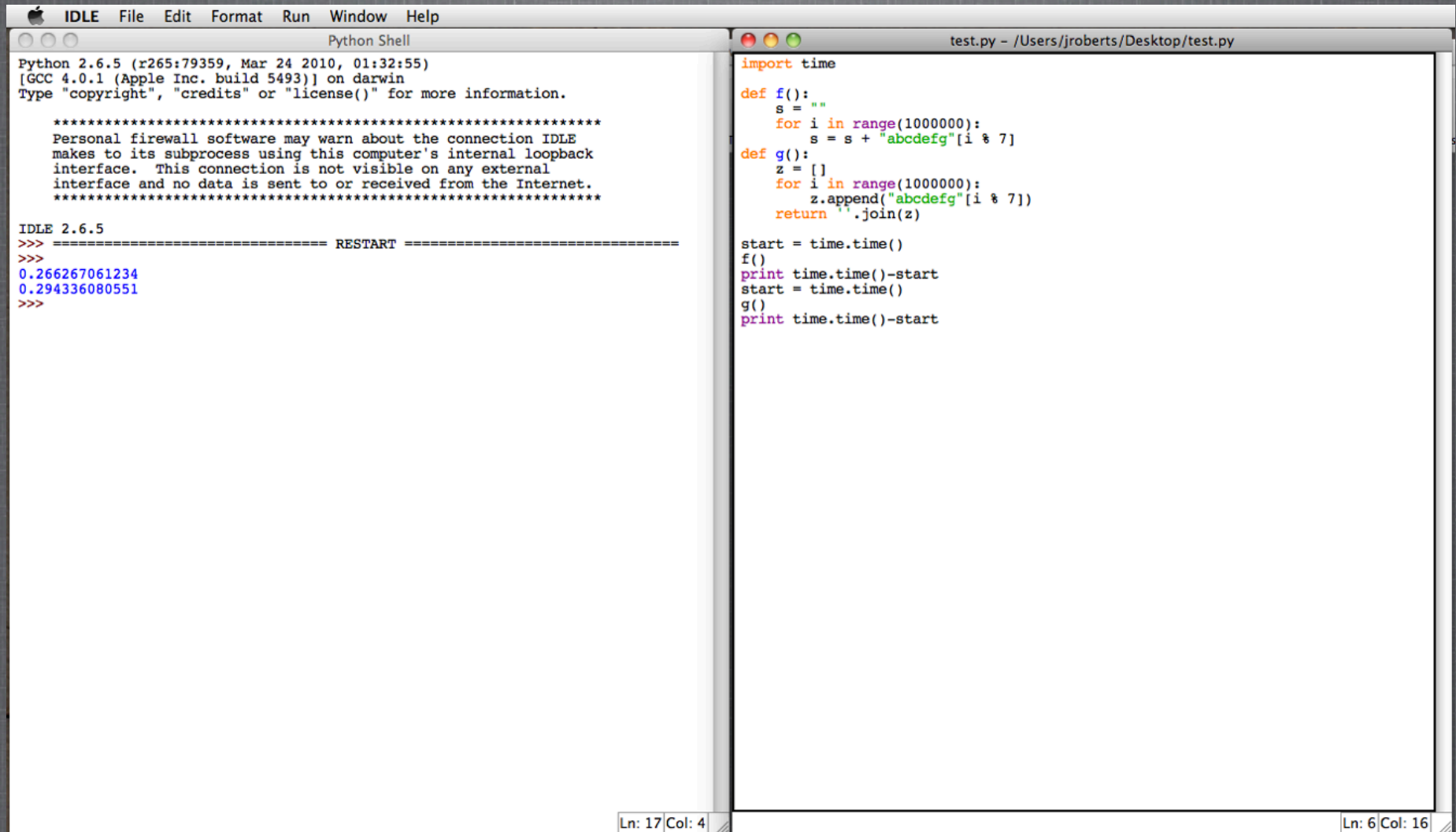


python 101

getting python

<http://www.python.org>

IDLE - the python development environment



The screenshot displays the IDLE Python development environment. The main window is titled 'IDLE' and contains two panes. The left pane, titled 'Python Shell', shows the output of the Python 2.6.5 interpreter. It includes the version information, the GCC compiler details, and a warning about the firewall. Below this, the output of the 'RESTART' command is shown, with two lines of execution time: 0.266267061234 and 0.294336080551. The right pane, titled 'test.py - /Users/jroberts/Desktop/test.py', shows the source code of a Python script. The script defines two functions, 'f()' and 'g()', both of which iterate over a range of 10,000,000 and perform string concatenation. The script also includes timing code to measure the execution time of these functions.

```
Python Shell
Python 2.6.5 (r265:79359, Mar 24 2010, 01:32:55)
[GCC 4.0.1 (Apple Inc. build 5493)] on darwin
Type "copyright", "credits" or "license()" for more information.

*****
Personal firewall software may warn about the connection IDLE
makes to its subprocess using this computer's internal loopback
interface.  This connection is not visible on any external
interface and no data is sent to or received from the Internet.
*****

IDLE 2.6.5
>>> ===== RESTART =====
>>>
0.266267061234
0.294336080551
>>>
```

```
test.py - /Users/jroberts/Desktop/test.py
import time

def f():
    s = ""
    for i in range(1000000):
        s = s + "abcdefg"[i % 7]

def g():
    z = []
    for i in range(1000000):
        z.append("abcdefg"[i % 7])
    return "".join(z)

start = time.time()
f()
print time.time()-start
start = time.time()
g()
print time.time()-start
```

Ln: 17 Col: 4

Ln: 6 Col: 16

Hello World!

python code

```
print()  
print('Hello World!')  
print('Ultimate answer: ', 27 + 15)
```

output

<- blank line

```
Hello World!  
Ultimate answer: 42
```

string

python code

```
x = 'abcde'  
print(x)
```

output

```
abcde
```


integer

python code

```
i = 2  
j = 3  
print('integer divide 2/3: ', i/j)
```

output

```
integer divide 2/3: 0
```


float

python code

```
i = 2.0 #float  
j = 3   # integer  
print('floating point divide 2.0/3: ', i/j)
```

output

```
floating point divide 2.0/3: 0.666666666667
```

sequence - string

python code

```
x = 'abcde'  
print(x[0])  
print(x[-1])  
print(x[1:4])
```

python starts counting at zero

output

```
a  
e  
bcd
```


sequence - tuple

python code

```
x = (1,2,3,4,5)
print(x[0])
print(x[-1])
print(x[1:4])
```

output

```
1
5
(2, 3, 4)
```


sequence - list

python code

```
x = [1,2,3,4,5]  
print(x[0])  
print(x[-1])  
print(x[1:4])
```

output

```
1  
5  
[2, 3, 4]
```


tuple is immutable

python code

```
x = (1,2,3,4,5)  
x[0] = 9
```

output

```
Traceback (most recent call last):  
  File "/Users/jroberts/Desktop/test.py", line 2, in <module>  
    x[0] = 9  
TypeError: 'tuple' object does not support item assignment
```


list is mutable

python code

```
x = [1,2,3,4,5]  
x[0] = 9  
print(x)
```

output

```
[9, 2, 3, 4, 5]
```

lists can have mixed types

python code

```
x = [1,2,3,4,5]  
x[0] = 'a'  
x[1] = (4,5,6)  
print(x)
```

output

```
['a', (4, 5, 6), 3, 4, 5]
```

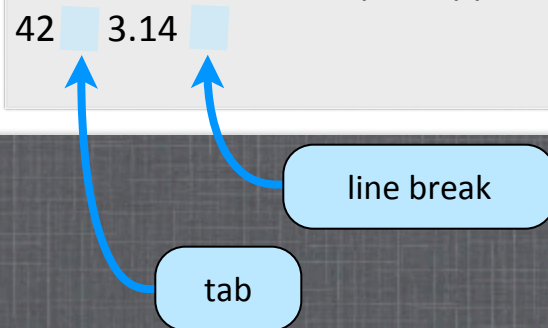

string % formatting

python code

```
answer = 42
pi = 3.141592654
print('the answer is %i and pi is approx. %f' % (answer,pi))
print('the answer is %i and pi is approx. %.2f' % (answer,pi))
print('%i\t%.2f\n' % (answer,pi))
```

output

```
the answer is 42 and pi is approx. 3.141593
the answer is 42 and pi is approx. 3.14
42      3.14
```



line break

tab

functions()

python code

```
print(float(42))  
print(int(5.75))  
print(round(5.75))  
print(int(round(5.75)))  
print(range(5))
```

output

```
42.0  
5  
6.0  
6  
[0, 1, 2, 3, 4]
```


functions from imported libraries

python code

```
import random  
numbers = [1,2,3,4,5]  
random.shuffle(numbers)  
print(numbers)
```

output

```
[2, 5, 4, 1, 3]
```

methods - functions “attached” to objects

python code

```
names = 'Jon Roberts\tErnie Mross'  
print(1, names.split())  
print(2, names.split('\t'))  
print(3, names.split('\t')[0])  
print(4, names.lower().split('\t')[0].split())
```

'\t' is the tab character

output

```
1 ['Jon', 'Roberts', 'Ernie', 'Mross']  
2 ['Jon Roberts', 'Ernie Mross']  
3 Jon Roberts  
4 ['jon', 'roberts']
```


what if?

python code

```
words = ['one','two','three']  
if 'three' in words:  
    print('found three')
```

output

```
found three
```

more if?

python code

```
x = 5
if x > 5:
    print('big x')
else:
    print('x is less than or equal to 5')
```

output

```
x is less than or equal to 5
```


even more if?

python code

```
x = 1
if x == 3:
    print('three')
elif x == 5:
    print('five')
else:
    print('not found')
```

output

```
not found
```

loops

python code

```
for num in range(4):  
    x = num+10  
    print(x)  
print('done')
```

output

```
10  
11  
12  
13  
done
```


more loops

python code

```
nameList = ['jon','ernie','linda']  
for aName in nameList:  
    print('aName -> ', aName)
```

output

```
aName -> jon  
aName -> ernie  
aName -> linda
```

even more loops

python code

```
done = False ; x = 0
while not done:
    x += 3
    if x>9: done = True
    print(x)
```

output

```
3
6
9
12
```


reading files

python code

```
infile = open('tabtext.txt', mode='rU')  
for aLine in infile:  
    print(aLine.split('\t'))  
infile.close()
```

output

```
['line 1', 'abc', 'def\n']  
['line 2', 'ghi', 'jkl']
```

more reading files

python code

```
infile = open('tabtext.txt', mode='rU')  
for aLine in infile:  
    print(aLine.strip().split('\t'))  
infile.close()
```

output

```
['line 1', 'abc', 'def']  
['line 2', 'ghi', 'jkl']
```


writing files

python code

```
data = [[1,2],[3,4]]
outfile = open('tabtextout.txt', mode='w')
for subList in data:
    outfile.write('%i\t%i\n' % (subList[0],subList[1]))
outfile.close()
```

'w' for write, 'a' for append

you must close the file to be certain all data are written

output

```
1 2
3 4
```

line breaks

tabs