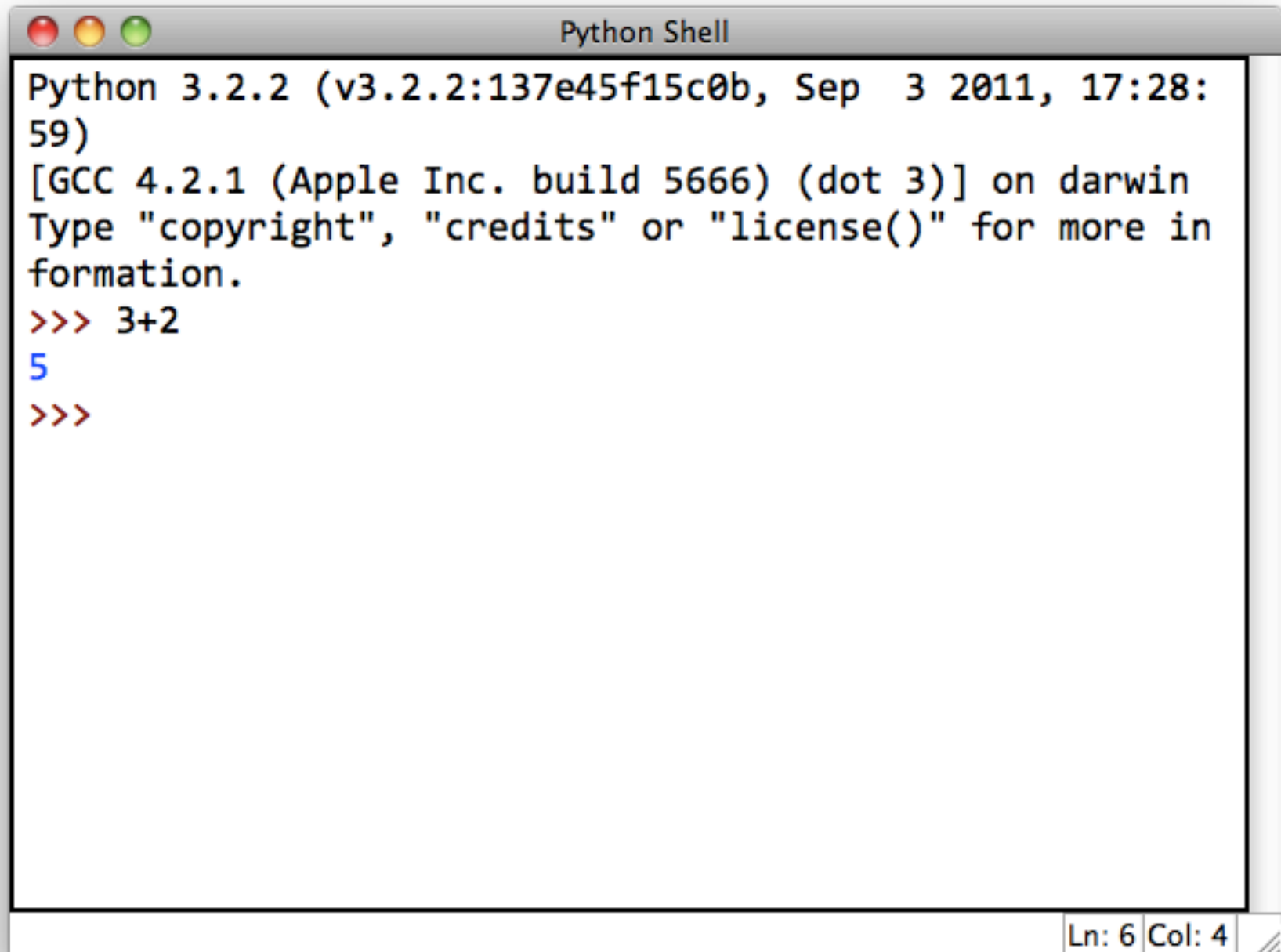


# CS 141, Lecture 3

# Python Shell

A screenshot of a Python Shell window. The window has a title bar with three colored buttons (red, yellow, green) on the left and the text "Python Shell" in the center. The main area contains the following text: "Python 3.2.2 (v3.2.2:137e45f15c0b, Sep 3 2011, 17:28:59)", "[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin", "Type \"copyright\", \"credits\" or \"license()\" for more in formation.", ">>> 3+2", "5", and ">>>". The bottom right corner shows a status bar with "Ln: 6 Col: 4" and a small icon.

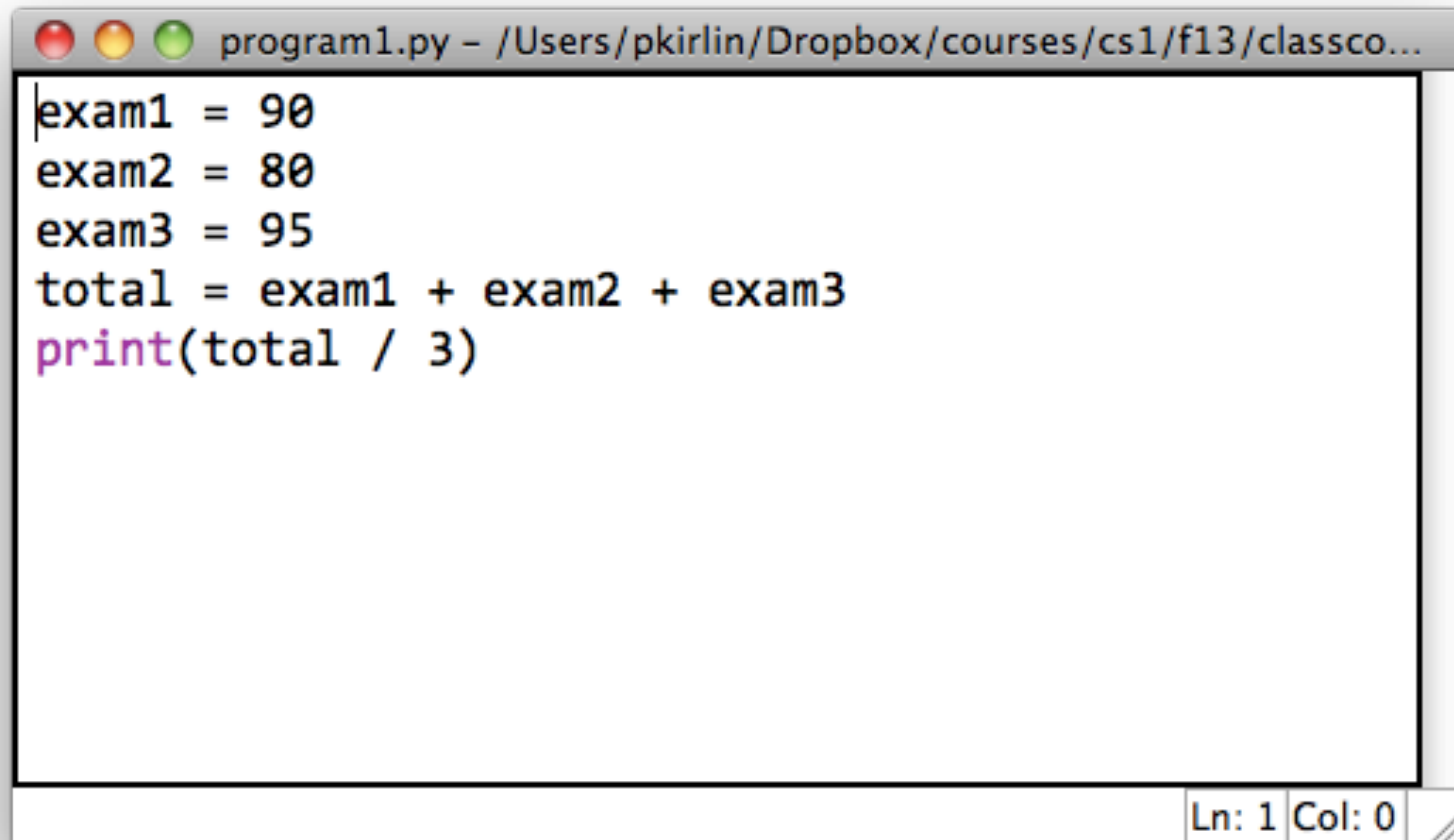
```
Python 3.2.2 (v3.2.2:137e45f15c0b, Sep 3 2011, 17:28:59)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "copyright", "credits" or "license()" for more in
formation.
>>> 3+2
5
>>>
```

Ln: 6 Col: 4

# Python Shell

- Runs single-line "mini-programs"
- Runs each line after you type it and press enter.
- Results of computations are automatically *printed* (displayed) back to you.

# Longer Programs



A screenshot of a Python script editor window. The title bar shows the filename 'program1.py' and the path '/Users/pkirlin/Dropbox/courses/cs1/f13/classco...'. The editor contains five lines of Python code: 'exam1 = 90', 'exam2 = 80', 'exam3 = 95', 'total = exam1 + exam2 + exam3', and 'print(total / 3)'. The 'print' function is highlighted in purple. The status bar at the bottom right shows 'Ln: 1 Col: 0'.

```
program1.py - /Users/pkirlin/Dropbox/courses/cs1/f13/classco...  
exam1 = 90  
exam2 = 80  
exam3 = 95  
total = exam1 + exam2 + exam3  
print(total / 3)  
Ln: 1 Col: 0
```

# Longer Programs

- Code doesn't run until you ask Python to run it.
- Each line executes in order, top to bottom, line by line.
- Lets you run the code over and over without retyping.
- Nothing is automatically printed; to do so you must call the print function.

# Math

- $+$ ,  $-$ ,  $*$ ,  $/$ ,  $**$
- Normal order of operations.
- Use parentheses to change order of operations.

# Variables

```
program1.py - /Users/pkirlin/Dropbox/courses/cs1/f13/classco...  
exam1 = 90  
exam2 = 80  
exam3 = 95  
total = exam1 + exam2 + exam3  
print(total / 3)
```

The variables in this program are exam1, exam2, exam3, and total.

Variables are assigned *values* by using the assignment statement:

*variable = value*

Ln: 1 Col: 0

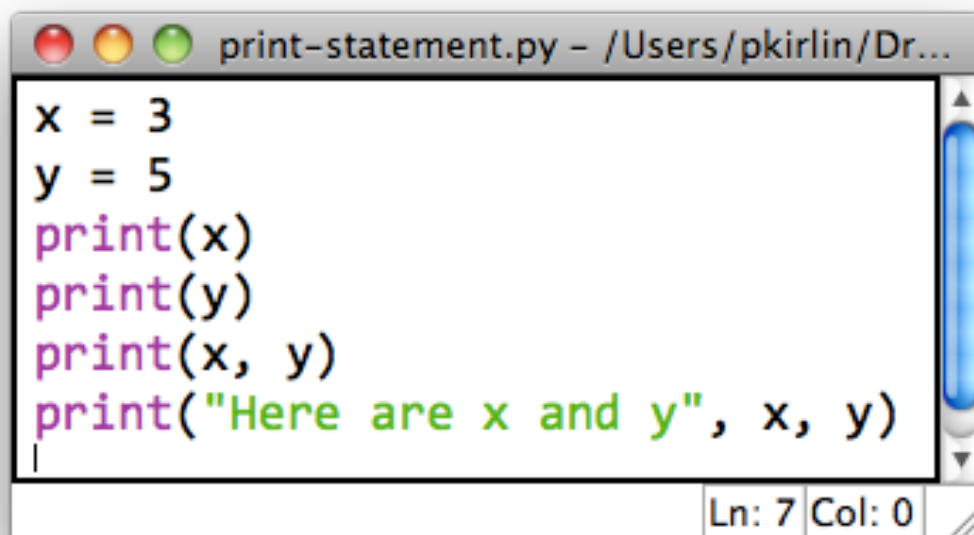
# Print function

- In a "real program" (not the Python Shell), nothing is displayed when you run the program unless you ask.
- Use the print function to do so.



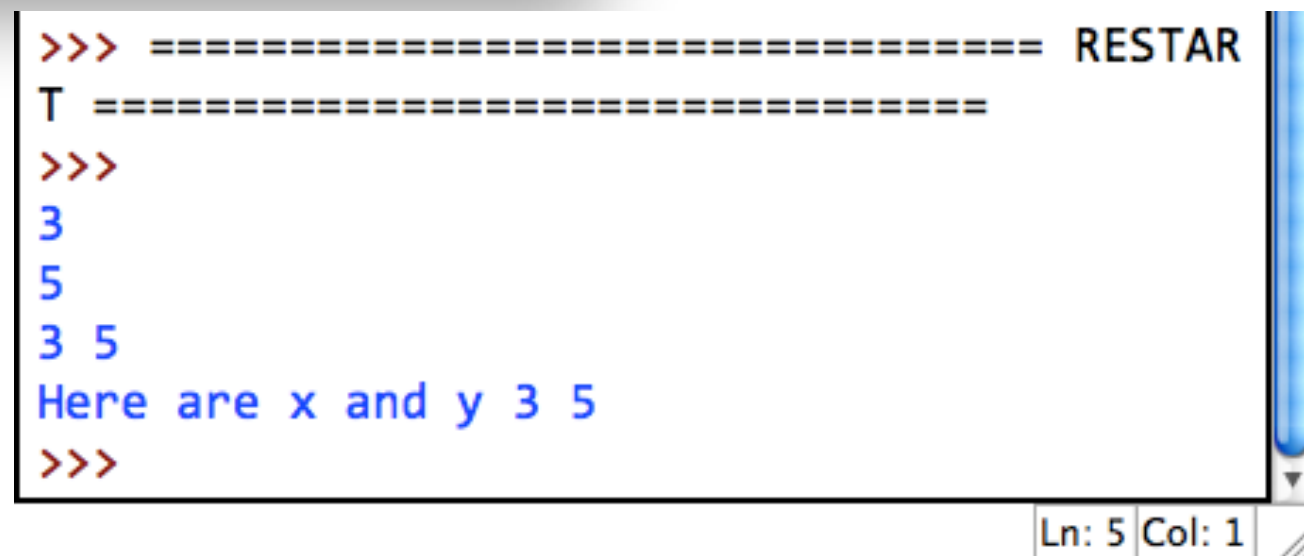
```
print(_____, _____, _____, ...)
```

- Replace the blank spaces above with the name of a variable, or a math expression.
- You can print any number of things at once.
  - Separate each thing you want to print with a comma.
  - Each thing will be displayed with a space in between.
  - If you want to print words, surround the words with double quotes.



```
x = 3
y = 5
print(x)
print(y)
print(x, y)
print("Here are x and y", x, y)
```

Ln: 7 Col: 0



```
>>> ===== RESTART
T =====
>>>
3
5
3 5
Here are x and y 3 5
>>>
```

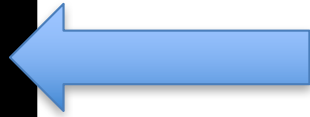
Ln: 5 Col: 1

```
x = 3  
print(x)  
x = 6  
print(x)
```

Computer Memory

Program Output

```
x = 3  
print(x)  
x = 6  
print(x)
```

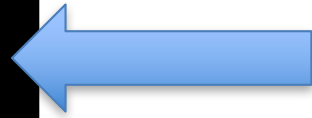


## Computer Memory

<u>Name</u>	<u>Value</u>
x	3

## Program Output

```
x = 3  
print(x)  
x = 6  
print(x)
```



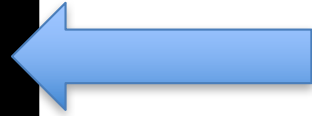
## Computer Memory

<u>Name</u>	<u>Value</u>
x	3

## Program Output

3

```
x = 3  
print(x)  
x = 6  
print(x)
```



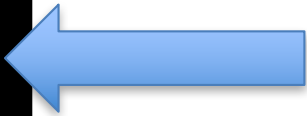
## Computer Memory

<u>Name</u>	<u>Value</u>
x	6

## Program Output

3

```
x = 3  
print(x)  
x = 6  
print(x)
```



## Computer Memory

<u>Name</u>	<u>Value</u>
x	6

## Program Output

3  
6

```
a = 4
b = 5
print(a, b)
a = 3
b = a
print(a, b)
a = b + 1
a = a + 1
print(a, b)
```

```
a = 1
b = 2
a = b
b = a
print(a, b)
```



- Variable names must be all one word (no spaces).
- Must consist of letters, numbers, or `_`.
  - Start with a letter.
- Choose a name that indicates the meaning of the variable.
  - For your grade on an exam: good ideas: `exam`, `exam_score`, `grade`,
  - Bad ideas: `e`, `g`,  
`the_score_i_got_on_the_exam`

- You're working at a fast food restaurant where a burger costs \$3.99 and French fries cost \$1.99.
- Write a program (in a separate file, saved as `burger.py`) that uses two variables to store these two prices.
- Your program should then print out the cost of buying two burgers and three orders of fries.
- If you finish early, make your program add in 9.25% sales tax.

# Data types

- Integers (`int`)
  - Whole numbers; may be negative.
- Floating point numbers (`float`)
  - Any number with a decimal point; may be negative.
- Strings
  - Any sequence of letters, numbers, or punctuation.
  - String literals are always surrounded by quotation marks, single or double.

# Input from the keyboard

- Use a variation of a variable assignment:

- For integers:

```
variable = int(input("Prompt"))
```

- For floats:

```
variable = float(input("Prompt"))
```

- For strings:

```
variable = input("Prompt")
```

- You're working at a fast food restaurant where a burger costs \$3.99 and French fries cost \$1.99.
- Write a program (in a separate file, saved as `burger.py`) that uses two variables to store these two prices.
- **CHANGE:** Make your program ask the user for how many burgers and orders of fries they want, and print the total cost.
- If you finish early, make your program ask the user for the costs of a burger and fries, and the sales tax rate.