Programs (or scripts)

- While we can type expressions directly to a Python interpreter (for example using an interface such as an IDLE shell), in general we will want to include statements in a program file
- Executing an expression from a script will not produce any output; for that we need statements (not expressions), such as
  - `print('ab')`
  - `print('3'*3)`

Providing input

- If we are going to write programs or scripts, we will need a way to incorporate input from a user.
- We use the Python function `raw_input`, as in:

```
>>> name = raw_input('Enter your name: ')
Enter your name: Eric Grimson
>>> print('Are you ' + name + '?')
Are you Eric Grimson?
```

Some simple code

One can use variable names anywhere you might use the expression whose value it holds

```
>>> myString = 'Too much'
>>> weather = 'snow'
>>> print(myString + ' ' +
  weather)
Too much snow
```

A straight line program

Suppose we type the following into a file, and load it into a Python IDLE window

```
x = 3
x = x*x # square value of x
print(x)
y = float(raw_input('Enter a number: '))
print(y*y)
```

Then we observe the following behavior (where I type a 4 below)

```
9
Enter a number: 4
16.0
```

Some observations

- Comments appear after a #
  - These are very valuable, as they help a user understand decisions the programmer has made in creating the program
  - Well commented code should be very readable by a user
- A straight line program simply executes each statement in order, with no variation in order
- Most programs require more sophisticated flow control