1. Log in to your Sage Cloud account.
   (a) Start the Chrome browser.
   (b) Go to http://cloud.sagemath.com and sign in.
   (c) You should see an existing Project for our class. Click on that.
   (d) Click “New”, call it c07, then click “Sage Worksheet”.

Appetizers

2. Try “1==2”, “1!=2”, “not (1==2)”, “1==2 or 1<2”, “1==2 and 1<2”

3. Find 5%2, 6%2, and 7%4. Explain.

   We have seen the if...else program construction. This is a piece of the full if...elif...else...
   program construction. elseif is short for “else if”.

4. Type in the following program. Try to guess what you’ll get before you evaluate.

   m=31
   if m%3==0:
      print m/3
   elif m%3==1:
      print (m-1)/3
   else:
      print (m-2)/3

5. The following function definition defines a function which returns the largest multiple
   of 2 which goes into an integer x:

   def quotient_mod2(x):
      y=x%2
      if y==0:
         return x/2
      elif y==1:
         return (x-1)/2

   Try quotient_mod2(4), quotient_mod2(5), and quotient_mod2(8).
6. Define and test a function `quotient_mod3(x)` which returns the largest multiple of 3 which goes into an integer `x`.

Even More Lists

7. Let `L = [2,3,5,7,11,13,17,2]`. To find out what kind (or type) of object `L` is, try “type(L)”.


9. Let `S = ['apple', 'pear']`. Find `S[0]`. Find `type(S)`, `type(S[0])`, and `len(S)`. Remember that Python/Sage list indexing begins with 0.

10. Let `M = [[1,2],[1,3],[1,4]]`. Find `len(M)`, `M[2]`, and `M[2][2]`.


    `L[a:]` returns the list with the `L[a]` entry and every other entry of `L` past that one (that is, every other entry with larger index). `L[:b]` returns the list with every entry of list `L` from the beginning up to the one *before* the `b^{th}` entry (so up to and including `L[b-1]`).

11. Try `L = [1, 2, 0, 3, 4, 0, 4, 5]`. Find `L[2:]`. Explain.


13. To find the *index* of the first occurrence of element `a` in list `L` use `L.index(a)`. What should `L.index(2)` be, `L.index(0)`? Now try it.

14. You can concatenate lists just like you concatenate strings in Python. Try `[1,3,5]+[2,4,6]+[100]`. 