1. Log in to your Sage/Cocalc account.
   
   (a) Start the Chrome browser.
   (b) Go to \texttt{http://cocalc.com} and sign in.
   (c) You should see an existing Project for our class. Click on that.
   (d) Click “New”, call it \texttt{c08}, then click “Sage Worksheet”.

2. **Warm-ups.** What will the following commands return in Sage? Answer and then use Sage to check.
   
   (a) \texttt{is\_prime(2**(2**4)+1)}
   (b) \texttt{is\_prime(2**(2**5)+1)}
   (c) \texttt{L=[1, 2, 1, 2, 1]}
   (d) \texttt{len(L)}
   (e) \texttt{S=set(L)}
   (f) \texttt{len(S)}
   (g) \texttt{[k**k for k in L]}

**More Lists**

A list in Sage is a \textit{mutable} object. Its entries can be changed.

3. Evaluate \texttt{L[0]} (you will get the 0th entry of \texttt{L}; it should be 1).

4. \texttt{L[0]} can be changed. Evaluate \texttt{L[0]=5}. Now evaluate \texttt{L}.

5. Let \texttt{L=[2,3,3,3,2,1,8,6,3]}. Try \texttt{L.sort()}. What does Sage do?

6. Try \texttt{L.append(4)}. Evaluate \texttt{L}.

7. Try \texttt{L.remove(3)}. What do you get?

8. Try \texttt{sum(L)}. What do you get? What did Sage do?

9. Try \texttt{prod(L)}. What do you get? What did Sage do?

10. Try \texttt{map(cos, [0, pi/4, pi/2, 3*pi/4, pi])}. What do you get? Explain.

11. Try \texttt{map(factorial,[1,2,3,4,5])}. What do you get? Explain?
12. What do you think the value of sum(map(exp,[1,2,3,4,5])) will be? Try it.

**Tuples**

A *tuple* is a Sage object, similar to a list, but with curved brackets instead of square brackets. These include pairs like (2,3), triples like (4,5,6), etc.

13. Let \( t=(2,3) \). Then evaluate \( t \), and evaluate \( \text{type}(t) \).

14. You can find the entries in a tuple just like you can with a list. Try \( t[0] \), \( t[1] \) and \( t[2] \).

A tuple in Sage is an *immutable* object. You can’t change it.

15. Try to change the 0\(^{th}\) entry of \( t \). Evaluate \( t[0] = 5 \).

**Anonymous Functions**

An *anonymous function* is just a function without an associates name and, thus, not meant for reuse. They can be useful in order to save defining an auxiliary function (especially if you have no intention of using that auxiliary function for anything else).

16. We want to triple all the entries of some list. We could do the following:

\[
\begin{align*}
  f(x) &= 3*x \\
  \text{map}(f,[4,5,6])
\end{align*}
\]

Equivalently we could use a “lambda construction” and define an *anonymous function*:

\[
\text{map}(\lambda x: \, 3*x,[4,5,6])
\]

Now define an anonymous function to square every element of \([4,5,6]\).

17. Define a function \( \text{filter evens}(L) \) that takes a list \( L \) of integers as input and returns a new list which is exactly the even integers that were in \( L \).

18. Suppose you only intended to do this once ever. Then you don’t need to define a function. If you wanted to filter the even integers from \( L = [1, 2, 3, 4, 5, 6] \) you could do the following:

\[
\text{filter}(\lambda x: \, x\%2==0, \, L)
\]