CURVES 2015: SnapPy worksheet 1 Tuesday, June 23, 2015.

SnapPy website. You can download SnapPy from http://snappy.computop.org; the documentation contained there can also be found on the Help menu in SnapPy itself.
The math behind SnapPy. See the excellent paper: J. Weeks, *Computation of Hyperbolic Structures in Knot Theory*, arXiv:0309407

1. (a) Load the manifold $v1234$ and name it V.

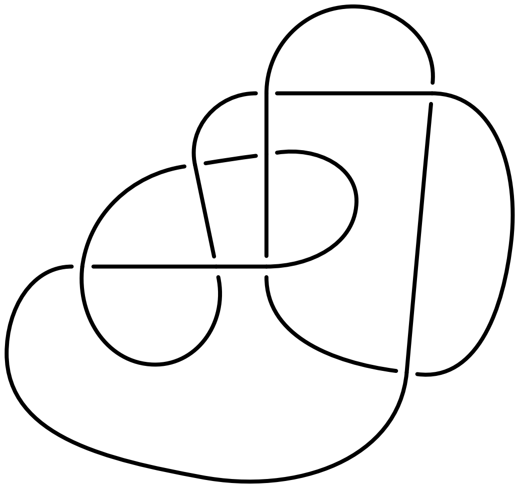
   (b) Use the browser to find the volume, Dirichlet domain, and symmetry group of V.

   (c) Like any manifold in SnapPy, the object V is really a particular *triangulation* of this hyperbolic manifold. Back at the command line, determine the number of tetrahedra in the triangulation V. Hint: Use tab completion by typing `V.<tab-key>`.

   (d) The manifold V has one cusp. Back the browser, do Dehn filling along the meridian curve. What closed manifold do you get?

2. (a) Use SnapPy to find the name in the Rolfsen table for the link shown at right.

   (b) Is the projection at right the same as the one that's stored in SnapPy?



3. In the morning session, I mostly focused on manifolds with cusps, but SnapPy also works with closed manifolds. In particular, it comes with the Hodgson-Weeks census of small-volume closed hyperbolic 3-manifolds, which is called `OrientableClosedCensus`.

   (a) Use the "?" operator to find out more about `OrientableClosedCensus`; in particular, how many manifolds are in it?

   Also, interrogate the orientable *cusped* census to get ideas on how to select various types of manifolds for the later parts of this question.

   (b) Closed manifold in SnapPy are represented as Dehn fillings on cusped manifolds. You can do Dehn filling in the browser, via the `dehn_fill` method, or as part of the specification that you given to `Manifold`. For example, typing `A = Manifold('4_1(1,2)')` gives the closed 3-manifold which is $\frac{1}{2}$-Dehn surgery on the figure 8 knot. Use the method `is_isometric_to` to show that A is the sixth manifold the `OrientableClosedCensus`. Warning: In Python, lists are numbered starting from 0 rather than 1.

   (c) Find the unique manifold M in the original `OrientableClosedCensus` whose volume is between 3.0 and 3.1 and whose first homology is $\mathbb{Z}/3\mathbb{Z} \oplus \mathbb{Z}/3\mathbb{Z} \oplus \mathbb{Z}/3\mathbb{Z}$.

   (d) Find a description of M as Dehn surgery on a 3-component link in S^3. Hint: Unfill the cusp in the default description of M and then drill out the shortest geodesic twice.

4. Here's how you get the exterior of a randomly chosen 14-crossing prime knot:

```
knots = HTLinkExteriors(cusps=1, crossings=14)
M = knots.random()
```

   (a) Python uses the `len` function to access the length of any list-like object, so do `len(knots)` to see how many such knots there are.

   (b) Try creating the Dirichlet domain for M at the command line. Most of the time you will get an error message saying that this failed! (If not, pick a different example which does fail for the rest of this problem ;-).

   (c) By default, the hyperbolic structure on M is computed using standard double-precision floating-point numbers (about 15 decimal digits). It turns out this isn't enough to find the Dirichlet domain for a manifold this complicated. Use the `high_precision` method of M to upgrade it to a `ManifoldHP` called H.

   (d) Compute the volumes of M and H. Are the answers consistent with the hyperbolic structure on H being computed to quad-double precision?

   (e) Try computing the Dirichlet domain D using H, which will most likely succeed though it typically takes a few seconds.

   (f) Interrogate D to get a pretty picture and find out how many faces and vertices D has.