1. Student data entry:
   1. look into ways to have javascript do data validation on the things they submit
   2. probably can only check things like: all inputs given, ID format is correct, etc
   3. redesign finalreport.html so that it’s more obvious what to put where?
2. Instructor setup:
   1. place a text file in the cgi-bin directory containing all the student IDs, one on each line
   2. IDs in the text file are case-insensitive, the script will automatically convert everything to uppercase
   3. Include special instructor ID to validate ID list (“AA00000”?)
   4. Include grade penalty variables at the top of the script file so Caruso can change them?
3. Conceptualization
   1. Flies have 4 genes:
      1. Eye texture
      2. Bristles
      3. Two eye color genes
   2. Eye color genes are always co-dominant with each other, always recessive to WT
   3. Randomize the two remaining genes for dom/rec
   4. Gene linkage should be randomized to: 0.10 <= x < 0.50
      1. This range should be configurable
   5. Randomly assign chromosome locations to all four genes
      1. One gene is always sex-linked, three autosomal
      2. Always one pair of linked genes
   6. Curly/Dicheate parent is always male
      1. Female is homozygous RECESSIVE
4. Back-end:
   1. ~~Figure out how to write to a file on the webspace~~
      1. /afs/umbc.edu/common/bin/enable\_student\_web\_space.sh
      2. Requires running an AFS command on a gl terminal
         1. “fs sa . system:anyuser write” gives anyone write access to the current folder
      3. Run web space script first, then apply write permissions to needed folders
   2. Write each student’s results and data to their own file
      1. Avoids versioning conflicts when multiple people are accessing it at once
   3. Each file will be a json format, with the data being a dictionary containing all the student’s data
      1. If the file doesn’t exist, they haven’t done any crosses yet, and NEED to do their P0 cross first, generate karyotype at this time and store it
      2. Karyotype data is stored with full gene names
         1. ~~Make sure to check that all genes for a particular student have unique first initials~~
      3. Check if crosses have already been ordered, redisplay if so
   4. Students can enter their IDs suffixed with a “t” (“AB12345t” or “AB12345T”) to get practice or test problems, then enter their proper ID (“AB12345”) to do it for real
   5. ~~Have mutant names stored in separate files so Caruso can update as needed.~~
   6. ~~For crosses, simulate ~6000 flies, ~1000 for initial crosses given to the student~~
   7. Use .join() to concatenate strings?

Things to do:

1. ~~Check to make sure that gene names don’t start with the same initials~~
2. Functions
   1. ~~Add chromosome localization crosses~~
   2. ~~Cross stocks function~~
   3. Grading function
      1. Show student a grade breakdown
3. ~~Data output for the student~~
4. Grade report
   1. Email a csv file?
   2. Automatically upload grades to blackboard?
      1. Would require finding a working blackboard API
   3. Log all student results in a csv file
      1. Log their answer? (use semicolon to delimit pairs in the same text box)
5. Input checking
   1. Single-character text boxes on finalreport.html?
   2. ID validation
   3. Error messages (for invalid cross inputs)