APPENDIX
DATA COLLECTION TOOLS

Developed by Professor Garrison LeMasters, the following Python scripts were used to scrape the pertinent data from the Club RealDoll forum. For more information, please refer to the **Methodology** chapter of this thesis.

file.prep.py

```
# LeMasters 2018 / Georgetown CCT
# Code for MA thesis project
# This runs before the parsing routines
# CC BY 4.0

import os
import shutil


# ### This routine identifies the downloaded files we need in order
# to sort the wheat from the chaff.  It renames the .html files based on their
# current directory (which originally indicated the thread's subject),
# and copies everything relevant to research into a new flat folder.

def print_realDoll_files(realDoll_directory, realDoll_extensions=['html']):

    # Get the absolute path of the realDoll_directory parameter
    realDoll_directory = os.path.abspath(realDoll_directory)

    # Get a list of files in realDoll_directory
    realDoll_directory_files = os.listdir(realDoll_directory)

    # Traverse through all files
    for filename in realDoll_directory_files:
        # set filepath to the directory + the filename
        filepath = os.path.join(realDoll_directory, filename)

        # Check if it's a normal file or another directory
        if os.path.isfile(filepath):

        # Check if the file has an extension of realdoll pages
          #  for realDoll_extension in realDoll_extensions:
            if filename !='index.html':
                if filename[:5] !='page-':
                    continue

            print_realDoll_files.counter += 1
```

```
            # Print its name
            tempName = (('{0}'.format(filepath)[48:-11]))
            tempCounter = str(print_realDoll_files.counter)
            targetpath=output_directory + "/output/" + tempName + tempCounter + '.html'
            shutil.copy(filepath,targetpath)

        elif os.path.isdir(filepath):
            # We got a directory, enter into it for further processing
            print_realDoll_files(filepath)


realDoll_directory = os.getcwd()
temp = os.path.join(realDoll_directory, 'try_2')
realDoll_directory = temp
output_directory = os.getcwd()
realDoll_name='index'

print('\n -- Looking for realDoll data in "{0}" --\n'.format(realDoll_directory))

# Set the number of processed files equal to zero
print_realDoll_files.counter = 0

# Start Processing
print_realDoll_files(realDoll_directory)

# We are done. Get. Out.
print('\n -- {0} realDoll File(s) found in directory {1} --'.format
    (print_realDoll_files.counter, realDoll_directory))
```

parse.output.py

```
# 2018 garrison lemasters georgetown
# CC BY 4.0
# Python 3.n, jupyter notebook


from bs4 import BeautifulSoup
import pandas as pd
import re
import glob
import os

def parseDollPage(filename, pageNumber):
    postIDz = []
```

```
    postAuthorz = []
    postQuotz = []
    postMsgz = []
    postDatez = []
    postSourcez = []

    serialA = ['robot-development-updates.27','RD']
    serialB = ['harmony-app-general-discussion.28','GD']
    serialC = ['harmony-app-dev-team-announcements.26','TA']
    serialD = ['ai_beta_tester','AI']
    serialE = ['harmony-app-bug-reports.25','BR']
    serialF = ['doll-gallery.8','08']
    serialG = ['realdoll-discussion.18','18']
    serialH = ['non-doll-discussion.6','06']
    serialI = ['the-lab-suggestions-future-tech.13','13']

    file = open(filename)
    dataObject = file.read()
    soup = BeautifulSoup(dataObject,'html.parser')

# used to be n2 = soup.find(serialTwo)
    nA = dataObject.find(serialA[0])
    nB = dataObject.find(serialB[0])
    nC = dataObject.find(serialC[0])
    nD = dataObject.find(serialD[0])
    nE = dataObject.find(serialE[0])
    nF = dataObject.find(serialF[0])
    nG = dataObject.find(serialG[0])
    nH = dataObject.find(serialH[0])
    nI = dataObject.find(serialI[0])
    origin = '??'

    # This part is all very clunky --
    # a last-minute fix.

    if nA != -1:
        origin = serialA[1]
    else:
        if nB != -1:
            origin = serialB[1]
        else:
            if nC != -1:
                origin = serialC[1]
            else:
                if nD != -1:
                    origin = serialD[1]
```

```
            else:
                if nE != -1:
                    origin = serialE[1]
                else:
                    if nF != -1:
                        origin = serialF[1]
                    else:
                        if nG != -1:
                            origin = serialG[1]
                        else:
                            if nH != -1:
                                origin = serialH[1]
                            else:
                                if nI != -1:
                                    origin = serialI[1]
print(origin)
origURL = soup.find('link', rel = 'canonical').attrs['href']
# postUserpage = soup.find("a", class_="username").attrs['href']
allMsgs = soup.find_all("li", class_="message")

newFileName_temp = soup.find("h1").text

# fix annoying (ANNOYING!) presence of unsafe chrs
newFileName = newFileName_temp.replace("/","-")
newFileName = newFileName + "_" + str(pageNumber)

for item in allMsgs:

    postID = item.attrs['id']
    postAuthor = item.attrs['data-author']
    postMsg = item.find('article')   # temp change --.text
    testDate = item.find('span', class_='DateTime')
    if testDate != None:
        postDate = testDate.text
    else:
        postDate = "n.d."
    postSource = origin

#    quotationSection = item.find('aside')
    quotationSection = item.find('aside')

    if quotationSection != None:
        quotationLevelTwo = quotationSection.find('div')
        if quotationLevelTwo != None:
        # collect the post ID
```

```
            target = quotationSection.find('a', class_='AttributionLink')
            if target != None:
               t = target.get('href',None)
               for aside in postMsg.findAll('aside'):
                  aside.decompose()
                  # We only need the last 10 characters of that ID
               postAttrib = t[-10:]
            else:
               postAttrib = '(edited)'
         else:
            postAttrib = '(sig.)'
      else:
         postAttrib = ''
      postQuotz.append(postAttrib.strip())
      postIDz.append(postID.strip())
      postAuthorz.append(postAuthor.strip())
      postMsgz.append(postMsg.text.strip())
      postDatez.append(postDate.strip())
      postAttrib = ''
      postAttrib_temp = ''

      # .strip() removes unpleasant HTML leftovers as each item is catalogued
      panda_LingLing = pd.DataFrame({'postID': postIDz,
                     'date': postDatez,
                     'author': postAuthorz,
                     'post': postMsgz,
                     'ref': postQuotz,
                      'forum': postSource})
      f = newFileName + '.csv'

      panda_LingLing.to_csv(f)

   print(file)
   file.close()


currentDir = os.getcwd()
outputPageNumber = 0
path = currentDir + "/output/*.html"
print(path)
for fname in glob.glob(path):
   print(fname)
   parseDollPage(fname, outputPageNumber)
   outputPageNumber = outputPageNumber + 1
```