Python Argparse (Command line)

* Objective:
  + While debugging my custom test case generator, i found a really ease code which i can create my own custom command line argument options which i can pass into the python execution call, and this is the summary of how i did it.
* Staff:
  + Lead Programmer: Samil Chai
  + Junior Programmer: Nick Jang
    - Email: [nickjang114@gmail.com](mailto:nickjang114@gmail.com)
* Start date: 2017/10/02
* End date: 2017/10/02
* Reference:
  + [WY\_PYTHON\_ACCESS\_GOOGLE\_SPREADSHEET\_WITH\_OAUTH](https://docs.google.com/document/d/1tjjrUvoaPH0H2Ec1McGZulHH27z_gWw_h31cDG-orD8/edit)

1. Intro:
   1. During debugging with my custom test case generator following Python OAuth tutorial [WY\_PYTHON\_ACCESS\_GOOGLE\_SPREADSHEET\_WITH\_OAUTH](https://docs.google.com/document/d/1tjjrUvoaPH0H2Ec1McGZulHH27z_gWw_h31cDG-orD8/edit) i found an interesting code where i can create my own command line argument options like (eg. “--version, --help, -v, -l” … ) to pass in the specific command line arguments when executing the Python script.
2. Add command line argument options manually:
   1. After researching i found some good example on how to add them:
      1. Googling “[--logging\_level {DEBUG,INFO,WARNING,ERROR,CRITICAL}]”
      2. <https://stackoverflow.com/questions/26130741/using-argparse-with-google-admin-api>
   2. Create a sample Python test file called “argparsetest.py”
   3. So i can add the command line options like this:
      1. Add following code to “argparsetest.py”

import argparse

parser = argparse.ArgumentParser(parents=[tools.argparser])

# Command line argument options:

**parser.add\_argument('-app','--application\_name', help='OAuth 2.0 client ID name')**

**parser.add\_argument('-ssurl','--google\_spreadsheet\_url', help='Google Spreadsheet URL')**

**parser.add\_argument('-feed','--feed\_name', help='Name of the spreadsheet for processing')**

# Parse the command line arguments.

flags = parser.parse\_args(sys.argv[1:])

print("flags.application\_name: ", flags.application\_name)

print("flags.google\_spreadsheet\_url: ", flags.google\_spreadsheet\_url)

print("flags.feed\_name: ", flags.feed\_name)

* + 1. Where i added following options with function “argparse::add\_argument()”:
       1. “-app” option “--application\_name”
          1. Print description “APPLICATION\_NAME” when not prompted.
          2. Which i can provide the Application Name of the OAuth client ID name.
       2. “-ssurl” option “--google\_spreadsheet\_url”
          1. Which i can provide the Google Spreadsheet URL to process with.
          2. Print description “GOOGLE\_SPREADSHEET\_URL” when not prompted.
       3. “-feed” option “--feed\_name”
          1. Which i can provide the Google Spreadsheet feed name to process with.
          2. Print description “FEED\_NAME” when not prompted.
  1. Then i will get this output when i type “--help” to command line:
     1. $ Python argparsetest.py

[-app APPLICATION\_NAME]

[-ssurl GOOGLE\_SPREADSHEET\_URL]

[-feed FEED\_NAME]

* + 1. This means that i successfully added the command line options to the parser.
  1. Then i can pass the argument by executing with proper flags like this:
     1. **$ python argparsetest.py**

**-app WY\_PROJ\_BLENDER\_EDITOR\_OAUTH\_CLIENT**

**-ssurl https://docs.google.com/spreadsheets/d/1WReZYyjIMcfau-9TfERLZtquK7Tx1kRhHKCTfv1u184/edit#gid=0** **-feed Sheet1**

1. Final working sample code:

import argparse

parser = argparse.ArgumentParser(parents=[tools.argparser])

# Command line argument options:

parser.add\_argument('-app','--application\_name', help='OAuth 2.0 client ID name')

parser.add\_argument('-ssurl','--google\_spreadsheet\_url', help='Google Spreadsheet URL')

parser.add\_argument('-feed','--feed\_name', help='Name of the spreadsheet for processing')

# Parse the command line arguments.

flags = parser.parse\_args(sys.argv[1:])

print("flags.application\_name: ", flags.application\_name)

print("flags.google\_spreadsheet\_url: ", flags.google\_spreadsheet\_url)

print("flags.feed\_name: ", flags.feed\_name)