

gutenzahler Design Journal

Gary Khodayari 13th Mar 2022

[Github Link](#)

Intial Plans

- I will be creating this program using python3.
- I want this program to be able to download books from [gutenberg.org](#) as well as accepting local files as input
- Bar and pie graph as options
- System command line arguments should be used not python input function
- I will develop and test the program on WSL

Design Process

sys python libraries allows the python program to read system arguments with the `argv` list. since `argv` is a list, i can use a for loop and go through all the arguments given.

[gutenberg.org](#) doesnt have a API so i will have to use the normal hyperlink and user interface navigation.

[gutenberg.org](#) uses the bookid scheme to navigate through their files on their server.

text files are store in the following format:

[gutenberg.org/files/<bookid>/<bookid>-0.txt](#)

but sometime in this format:

[gutenberg.org/files/<bookid>/<bookid>.txt](#)

Example:

<https://gutenberg.org/files/1/1-0.txt>

other files such as epub and pdf files can be found in the following instead:

[gutenberg.org/cache/epub/<bookid>/](#)

[gutenberg python library](#) (<https://github.com/c-w/gutenberg>) will handle the navigation and file downloads this will make the programming much much easier. this library uses BSD-DB that is not included in linux. it will have to be downloaded before anyone can use the program.

[matplotlib](#) (<https://matplotlib.org/>) will handle the graphing needs. since matplotlib also has a stem plotting tool i will be adding that as an option as well.

```
matplotlib.pyplot.stem(x,y)
matplotlib.pyplot.bar(x,y)
matplotlib.pyplot.pie(values, labels)
```

program should sanatize the arguments and give error if in sufficient arguments are used.

if no arguments are used, the help command will appear the main body of the program will be in a `while True` loop and errors or program completion will break the loop

no threading is needed.

multiple graphing options are accepted and are displayed in the order of bar --> pie --> stem

for the alphabet and keeping count i will make a dictionary with 26 letters. every letter read will be switched to uppercase so i wont need to have to seperate dictionaries. every charecter is switched to uppercase and is compared with the dictionary. if it exist then the +1 is added to the count.