

PYTHON ASSIGNMENT

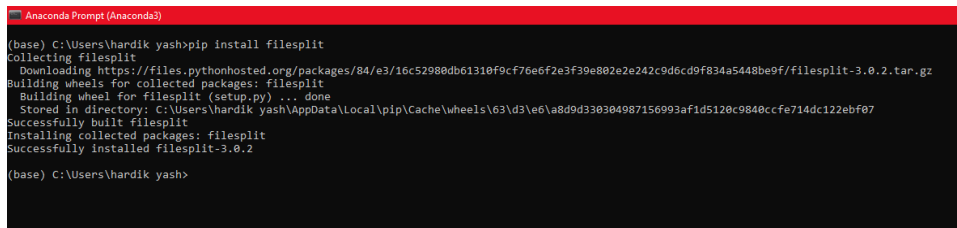
YASH SHAH

Intern | IPS

yashah@informatica.com

Library used Filesplit

In Anaconda prompt - > *pip install filesplit*



```
Anaconda Prompt (Anaconda3)
(base) C:\Users\hardik yash>pip install filesplit
Collecting filesplit
  Downloading https://files.pythonhosted.org/packages/84/e3/16c52988db61310f9cf76e6f2e3f39e802e2e242c9d6cd9f834a5448be9f/filesplit-3.0.2.tar.gz
Building wheels for collected packages: filesplit
  Building wheel for filesplit (setup.py) ... done
  Stored in directory: C:\Users\hardik yash\AppData\Local\pip\Cache\wheels\63\d3\e6\8d9d330304987156993af1d5120c9840ccfe714dc122ebf07
Successfully built filesplit
Installing collected packages: filesplit
Successfully installed filesplit-3.0.2
(base) C:\Users\hardik yash>
```

Python code –

from fsplit.filesplit import Filesplit

fs = Filesplit()

filedir=input("Enter the complete file path")

outdir=input("Enter the complete path of the directory where you want to store the output files")

choice=int(input("Do you want to split the file in \n 1.KB's \n 2.MB's \n 3.GB's \n"))

if(choice == 1):

x=int(input("Enter the split size for KB split"))

*x=x*1024*

elif(choice == 2):

x=int(input("Enter the split size for MB split"))

*x=x*1024*1024*

elif(choice == 3):

x=int(input("Enter the split size for GB split"))

*x=x*1024*1024*1024*

else:

print("Wrong Input")

f_size=x

print(f_size)

def split_cb(f, s):

print("file: {0}, size: {1}".format(f, s))

fs.split(file=filedir, split_size=f_size, output_dir=outdir, callback=split_cb)

Committing to Git –

```
hardik yash@Hardik-Yash MINGW64 ~/Python_assignment (main)
$ git add file_split.ipynb
warning: LF will be replaced by CRLF in file_split.ipynb.
The file will have its original line endings in your working directory

hardik yash@Hardik-Yash MINGW64 ~/Python_assignment (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   file_split.ipynb

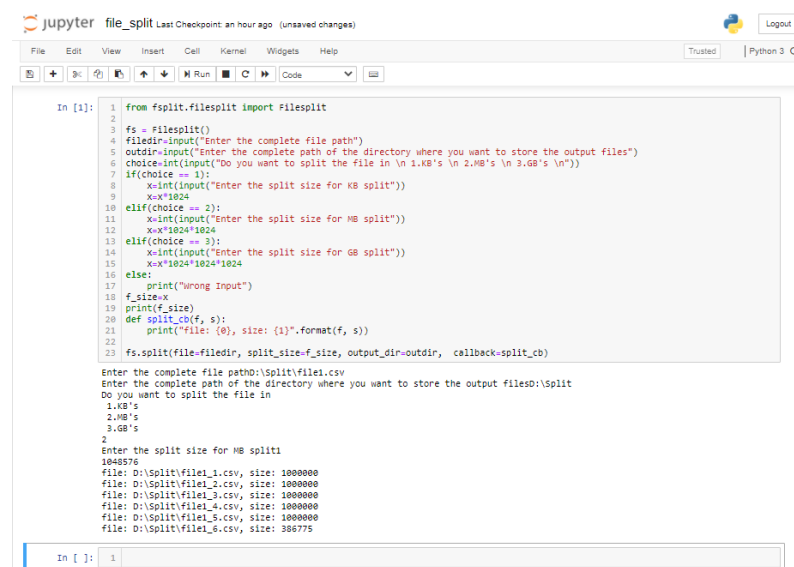
hardik yash@Hardik-Yash MINGW64 ~/Python_assignment (main)
$ git commit -m "Moving online"
[main 590d35e] Moving online
 1 file changed, 88 insertions(+)
 create mode 100644 file_split.ipynb

hardik yash@Hardik-Yash MINGW64 ~/Python_assignment (main)
$ git push
Logon failed, use ctrl+c to cancel basic credential prompt.
remote: Invalid username or password.
fatal: Authentication failed for 'https://github.com/Yash-Doraemon/Python_assignment.git/'

hardik yash@Hardik-Yash MINGW64 ~/Python_assignment (main)
$ git push
Logon failed, use ctrl+c to cancel basic credential prompt.
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 1.05 KiB | 215.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/Yash-Doraemon/Python_assignment.git
 41cad25..590d35e  main -> main

hardik yash@Hardik-Yash MINGW64 ~/Python_assignment (main)
$ |
```

Sample output –



The screenshot shows a Jupyter Notebook window titled "file_split" with a "Last Checkpoint: an hour ago (unsaved changes)" status. The interface includes a top bar with "File", "Edit", "View", "Insert", "Cell", "Kernel", "Widgets", and "Help" menus, along with "Trusted" and "Python 3" indicators. Below the menu bar is a toolbar with icons for running, saving, and other actions. The main area displays a Python script in a code cell, which is being executed. The script defines a function to split a file into smaller chunks based on user input. The output of the script is shown in the console, indicating the file path, the directory where the output files will be stored, and the size of the files. The script also prompts the user to enter the complete file path and the directory where the output files will be stored.

```
In [1]: 1 from fsplit.filesplit import Filesplit
        2
        3 fs = Filesplit()
        4 filedir=input("Enter the complete file path")
        5 outdir=input("Enter the complete path of the directory where you want to store the output files")
        6 choice=int(input("Do you want to split the file in \n 1.KB's \n 2.MB's \n 3.GB's \n"))
        7 if(choice == 1):
        8     x=int(input("Enter the split size for KB split"))
        9     x=x*1024
        10 elif(choice == 2):
        11     x=int(input("Enter the split size for MB split"))
        12     x=x*1024*1024
        13 elif(choice == 3):
        14     x=int(input("Enter the split size for GB split"))
        15     x=x*1024*1024*1024
        16 else:
        17     print("wrong Input")
        18 f_size=x
        19 print(f_size)
        20 def split_cb(f, s):
        21     print("file: {0}, size: {1}".format(f, s))
        22
        23 fs.split(file=filedir, split_size=f_size, output_dir=outdir, callback=split_cb)

Enter the complete file pathD:\Split\file1.csv
Enter the complete path of the directory where you want to store the output filesD:\Split
Do you want to split the file in
1.KB's
2.MB's
3.GB's
2
Enter the split size for MB split
1048576
file: D:\Split\file1_1.csv, size: 1000000
file: D:\Split\file1_2.csv, size: 1000000
file: D:\Split\file1_3.csv, size: 1000000
file: D:\Split\file1_4.csv, size: 1000000
file: D:\Split\file1_5.csv, size: 1000000
file: D:\Split\file1_6.csv, size: 386775
```