

"""

Code Challenge

Name:

copy command

Filename:

copy.py

Problem Statement:

Make a program that create a copy of a file

"""

"""

Code Challenge

Name:

Create a list of absentee

Filename:

absentee.py

Problem Statement:

Make a program that create a file absentee.txt

The program should take max 25 students name one by one

When the user enter a blank line, it should terminate the input

Store all the name of students in the file

Once all the students names have been entered, it should display the list

"""

"""

Code Challenge

Name:

Zoo Management

Filename:

zoo.py

Problem Statement:

Create different functions to :

read the zoo.csv file using readlines and print them

Print in list of animals in groups (elephant / tiger / lion / zebra / kangaroo)

print the total number of water need by elephant / tiger / lion / zebra / kangaroo

print the total number of water needed by all the animals

"""

"""

Code Challenge

Name:

Romeo and Juliet

Filename:

romeo.py

Problem Statement:

Let's start with a very simple file of words taken from the text of

Romeo and Juliet. (romeo.txt)

We will write a Python program to read through the lines of the

```
file
    break each line into a list of words
    and then loop through each of the words in the line,
    and count each word using a dictionary.
"""
```

```
"""
```

Code Challenge

Name:

Last Line

Filename:

lastline.py

Problem Statement:

Ask the user for the name of a text file.

Display the final line of that file.

Think of ways in which you can solve this problem,

and how it might relate to your daily work with Python.

```
"""
```

```
"""
```

Code Challenge

Name:

etc passwd

Filename:

passwd.py

Problem Statement:

This exercise assumes that you have access to a copy of
/etc/passwd,

The file in which basic user information is stored on Unix
computers.

The format is:

```
nobody:*:-2:-2::0:0:Unprivileged User:/var/empty:/usr/bin/false
```

```
root:*:0:0::0:0:System Administrator:/var/root:/bin/sh
```

```
daemon:*:1:1::0:0:System Services:/var/root:/usr/bin/false
```

In other words, each line is a set of fields, separated by colon
(:) characters.

The first field is the username, and the third field is the ID of
the user.

Thus, on my system, the nobody user has ID -2, the root user has
ID 0,

and the daemon user has ID 1.

You can ignore all but the first and third fields in the file.

There is one exception to this format:

A line that begins with a # character is a comment,
and should be ignored by the parser.

For this exercise,

you must create a dictionary based on /etc/passwd,

in which the dict's keys are usernames and the values are the
numeric IDs of those users.

You should then iterate through this dict, displaying one username

and
user ID on each line in alphabetical order.

"""

"""

Code Challenge

Name:

Word count

Filename:

wordcount.py

Problem Statement:

Unix systems contain many utility functions.

One of the most useful to me is wc, the "word count" program.

If you run wc against a text file, it'll count the characters,
words,

and lines that the file contains.

The challenge for this exercise is to write a version of wc in
Python.

However, your version of wc will return four different types of
information

about the files:

Number of characters (including whitespace)

Number of words (separated by whitespace)

Number of lines

Number of unique words

The program should ask the user for the name of an input file,
and then produce output for that file.

"""

"""

Code Challenge

Name:

Image Processing using PIL

Filename:

imgprocess.py

Problem Statement:

Given an image, perform image processing operations.

Keep only one output image i.e perform all tasks on the same image
(override)

and print only the name of your output image with extension name
in the end of your program.

Take the Image name from User (Handle the extension for image file
name in your code)

The image processing features to be provided by your code are:

a. Greyscale

- b. Rotate_90 (Rotate the given image file by 90 clockwise)
- c. Crop (Center) (size = 160(W), 204(H))
- d. Thumbnail - Generate the thumbnail of the given image (size = 75, 75)

"""

"""

Code Challenge

Name:

Reading and Writing CSV

Filename:

csv.py

Problem Statement:

Create a program that reads from one CSV file (/etc/passwd), and writes to another one.

You are to read from passwd file, and produce a file whose contents are the username (index 0) and the user ID (index 2). Note that a record may contain a comment, in which it will not have anything at index 2; you should take that into consideration when writing the file. The output file should use TAB characters to separate the elements.

Thus, the input will look like:

```
root:*:0:0::0:0:System Administrator:/var/root:/bin/sh
daemon:*:1:1::0:0:System Services:/var/root:/usr/bin/false
_ftp:*:98:-2::0:0:FTP Daemon:/var/empty:/usr/bin/false
```

and the output will look like:

```
root      0
daemon    1
_ftp      98
```

"""

Optional

"""

Code Challenge

Name:

SHA-1 Algorithm

Filename:

hash.py

Problem Statement:

Find hash of a file using hashlib library and using SHA-1 algorithm

Hint:

<https://www.programiz.com/python-programming/examples/hash-file>

"""

"""

Code Challenge

Name:

Resolution of an Image

Filename:

resolution.py

Problem Statement:

Find the resolution of any jpeg Image file (width x height)

Hint:

<https://www.programiz.com/python-programming/examples/resolution-image>

"""

"""

Code Challenge

Name:

Different sizes

Filename:

png.py

Problem Statement:

Convert all files PNG in a directory into different sizes

Hint:

`os.listdir('.')` function will list all the files in the current directory

"""