```
# This script is used to convert the time series data into standardised .csv-
files.

# Import of libraries
import os
import main as settings
import shutil


#
------------------------------------------------------------------------

# Functions

def terminate():
    """
    This function terminates the program.
    """
    print("[INFO] The program has been terminated.")
    exit()


def split_profile(filename, delimiter, split_column):
    print(f'[INFO] Reading file {filename}')
    try:
        length = sum(1 for line in open(os.path.join("data_raw", filename),
"r"))
        with open(os.path.join("data_raw", filename), "r") as file_in:
            print(f'[INFO] Extracting and parsing values')
            for index, line in enumerate(file_in):
                line = line.strip().split(delimiter)
                if(index%20000 == 0):
                    print(f'[INFO][{(index+1)*100/length:5.1f}%] Writing
individual profiles', end="\r")
                with open(os.path.join("data_split", f"{filename.split('.')
[0]}_{int(line[split_column]):05d}.csv"), "a") as file_out:
                    for column, entry in enumerate(line):
                        if(column == 0):
                            file_out.write(f"{entry.strip()}")
                        else:
                            file_out.write(f";{entry.strip()}")
                    file_out.write("\n")
            print(f'[INFO][100.0%] Writing individual profiles')
    except FileNotFoundError:
        print(f'[ERROR] File "{filename}" not found in the folder "data_raw".
Please add the measurement data to the folder "data_raw".')
        terminate()


#
------------------------------------------------------------------------

# Classes

#
------------------------------------------------------------------------

# Beginning of main program

if(__name__=='__main__'):
```

```
    # Delete old files and create new folder
    print(f'[INFO] Deleting old files in folder "data_split"')
    shutil.rmtree("data_split", ignore_errors=True)
    print(f'[INFO] Creating new folder "data_split"')
    os.mkdir("data_split")
    for dataset in settings.datasets:  # Iterate over the datasets
        print(f'[INFO] Splitting dataset "{dataset["filename"]}".')
        split_profile(dataset["filename"], dataset["delimiter"],
dataset["split_column"])
```