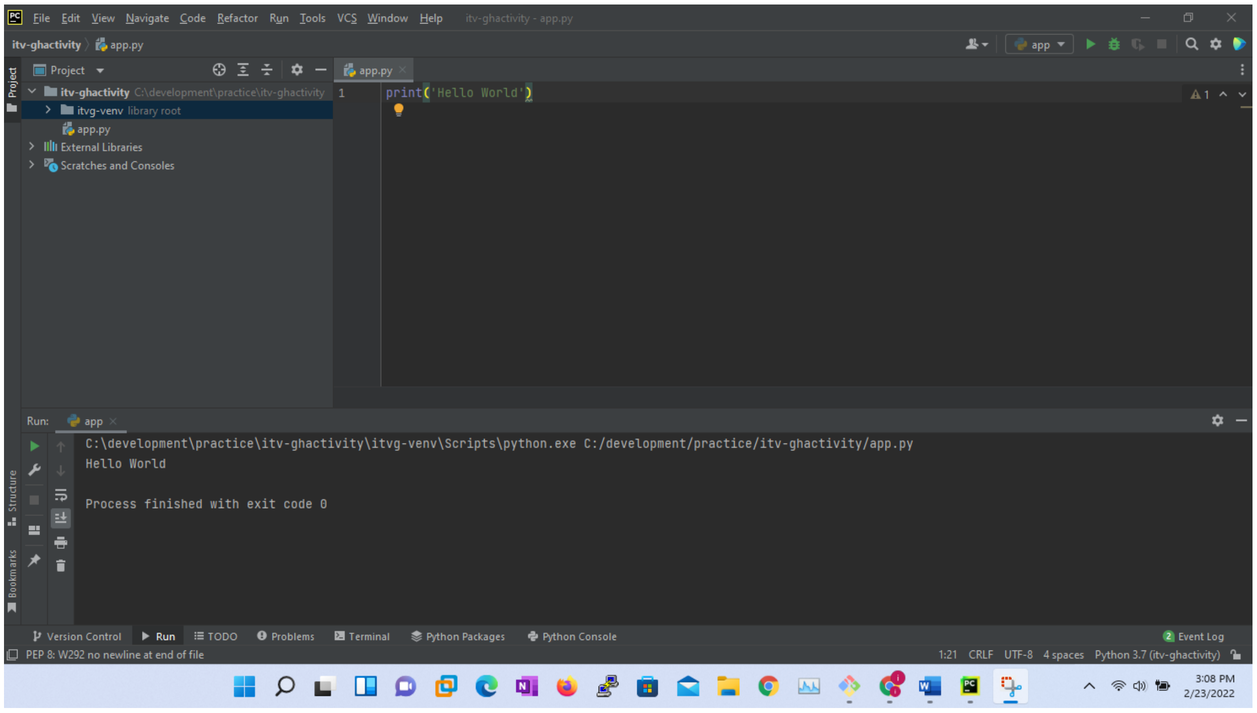
Setup Virtual Environment and Install Pyspark:

Here are the commands used to set up a virtual environment and install Pyspark.

```
python3 -m venv itvg-venv
source itvg-venv/scripts/activate
pip install pyspark
```

Here is how we can get started with local development of data engineering pipelines using Spark.

- Create Virtual Environment - `python3.7 -m venv itvg-venv`
- Activate virtual environment - `source itvg-venv/bin/activate`
- Install PySpark for local development - `pip install pyspark==2.4.*`
- Open using PyCharm and make sure appropriate virtual environment is used from the virtual environment which we have setup.

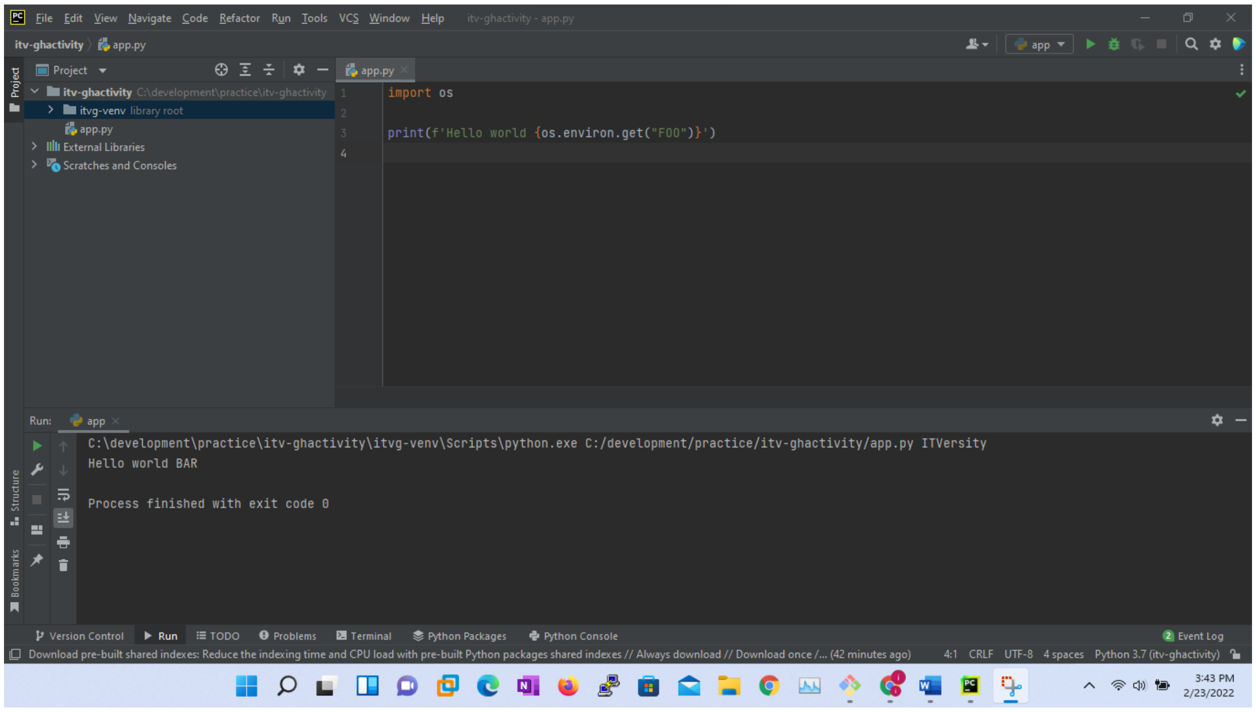


```
print('Hello World')
```

```
C:\development\practice\itv-ghactivity\itvg-venv\Scripts\python.exe C:/development/practice/itv-ghactivity/app.py
Hello World

Process finished with exit code 0
```

PEP 8: W292 no newline at end of file

Figure 1Created new environment with pyspark and python and opened in pycharm



```
import os

print(f'Hello world {os.environ.get("FOO")}')
```

```
C:\development\practice\itv-ghactivity\itvg-venv\Scripts\python.exe C:/development/practice/itv-ghactivity/app.py ITVersity
Hello world BAR

Process finished with exit code 0
```

Figure 2 Passed the environment variable from operating system into the program

Read data from files:

Let us develop the code to read the data from files into Spark Dataframes.

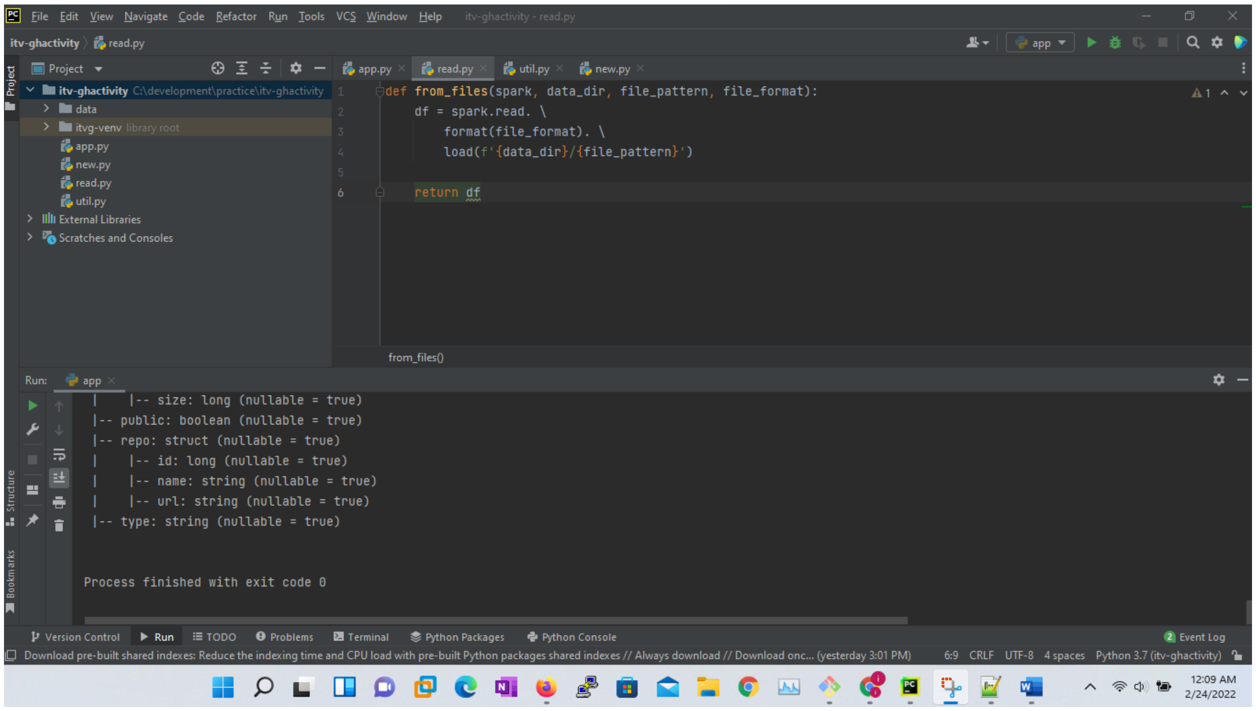
Create directory for data and copy some files into it.

```
mkdir -p data/itv-github/landing/ghactivity

cd data/itv-github/landing/ghactivity

wget https://data.gharchive.org/2021-01-13-0.json.gz

wget https://data.gharchive.org/2021-01-14-0.json.gz

wget https://data.gharchive.org/2021-01-15-0.json.gz
```

Create a Python program by name read.py. We will create a function by name from_files. It reads the data from files into Dataframe and returns it.

```
def from_files(spark, data_dir, file_pattern, file_format):

  df = spark. \

      read. \

      format(file_format). \

      load(f'{data_dir}/{file_pattern}')
```

```
    return df
```



```
def from_files(spark, data_dir, file_pattern, file_format):
    df = spark.read. \
        format(file_format). \
        load(f'{data_dir}/{file_pattern}')

    return df
```

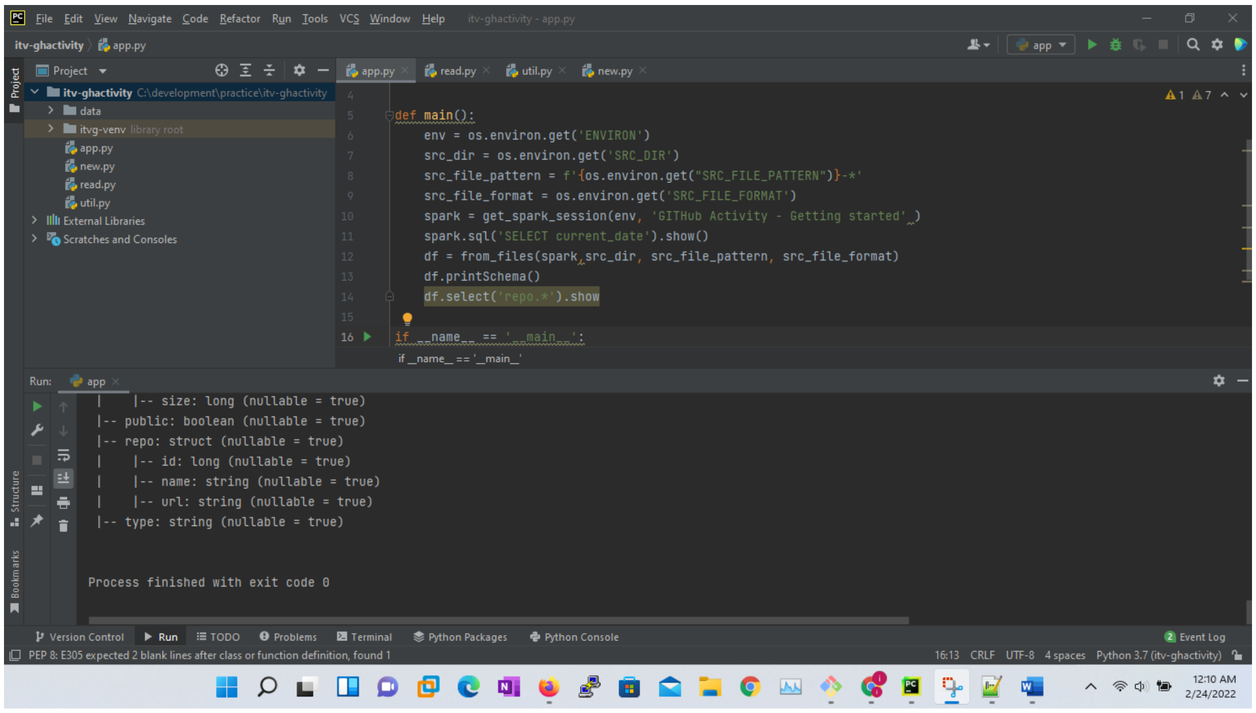
```
 |    |-- size: long (nullable = true)
 |-- public: boolean (nullable = true)
 |-- repo: struct (nullable = true)
 |    |-- id: long (nullable = true)
 |    |-- name: string (nullable = true)
 |    |-- url: string (nullable = true)
 |-- type: string (nullable = true)


Process finished with exit code 0
```

Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built Python packages shared indexes // Always download // Download onc... (yesterday 3:01 PM)

Figure 3read.py

Call the program from app.py. For now review schema and data.

```
import os
from util import get_spark_session
from read import from_files
 def main():
    env = os.environ.get('ENVIRON')
    src_dir = os.environ.get('SRC_DIR')
    file_pattern = f"{os.environ.get('SRC_FILE_PATTERN')}-*"
    src_file_format = os.environ.get('SRC_FILE_FORMAT')
    spark = get_spark_session(env, 'GitHub Activity - Reading Data')
    df = from_files(spark, src_dir, file_pattern, src_file_format)
    df.printSchema()
    df.select('repo.*').show()
 if __name__ == '__main__':
    main()
```



```
def main():
    env = os.environ.get('ENVIRON')
    src_dir = os.environ.get('SRC_DIR')
    src_file_pattern = f'{os.environ.get("SRC_FILE_PATTERN")}-*'
    src_file_format = os.environ.get('SRC_FILE_FORMAT')
    spark = get_spark_session(env, 'GITHub Activity - Getting started')
    spark.sql('SELECT current_date').show()
    df = from_files(spark,src_dir, src_file_pattern, src_file_format)
    df.printSchema()
    df.select('repo.*').show

if __name__ == '__main__':
```

```
 |    |-- size: long (nullable = true)
 |-- public: boolean (nullable = true)
 |-- repo: struct (nullable = true)
 |    |-- id: long (nullable = true)
 |    |-- name: string (nullable = true)
 |    |-- url: string (nullable = true)
 |-- type: string (nullable = true)


Process finished with exit code 0
```

PEP 8: E305 expected 2 blank lines after class or function definition, found 1

Process data using Spark APIs:

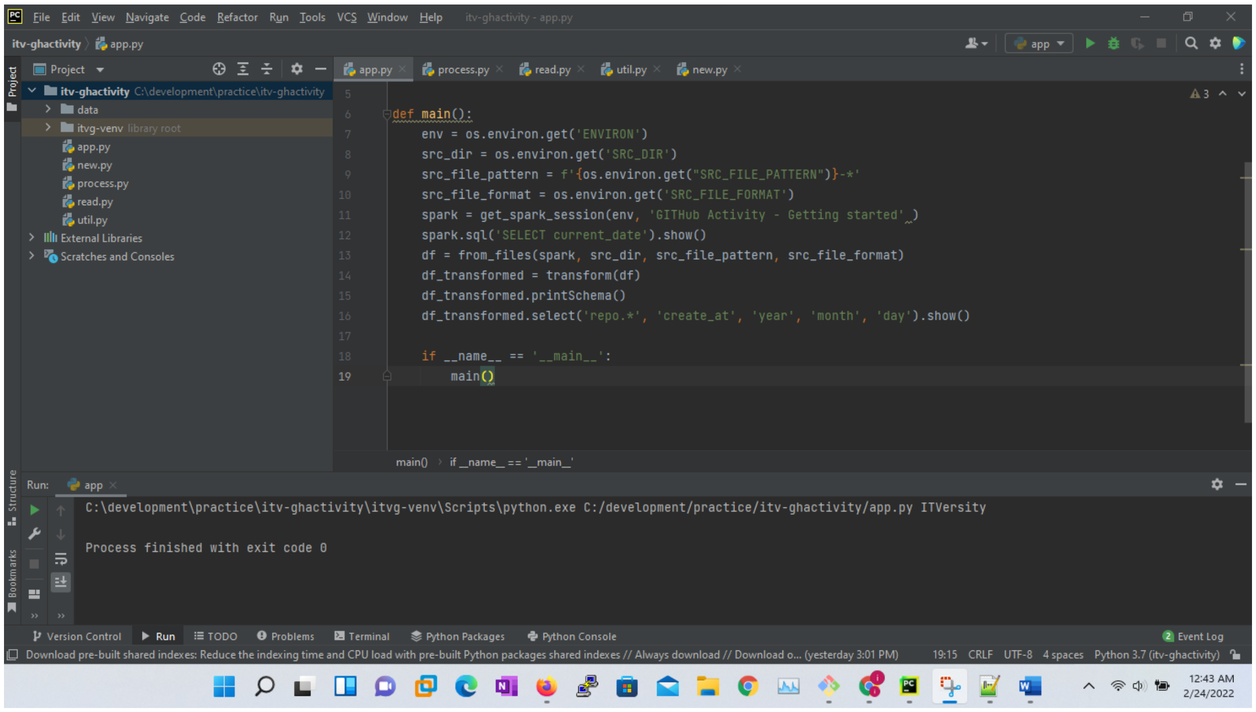
Create a Python program by name process.py. We will create a function by name df_transform. It partitions the Dataframe using specified field.

```
from pyspark.sql.functions import year, \
    month, dayofmonth

def transform(df):
    return df.withColumn('year', year('created_at')). \
        withColumn('month', month('created_at')). \
        withColumn('day', dayofmonth('created_at'))
```

Call the program from app.py. For now review schema and data.

```
import os
from util import get_spark_session
from read import from_files
from process import transform
```

```
def main():
  env = os.environ.get('ENVIRON')
  src_dir = os.environ.get('SRC_DIR')
  file_pattern = f"{os.environ.get('SRC_FILE_PATTERN')}-*"
  src_file_format = os.environ.get('SRC_FILE_FORMAT')
  spark = get_spark_session(env, 'GitHub Activity - Partitioning Data')
  df = from_files(spark, src_dir, file_pattern, src_file_format)
  df_transformed = transform(df)
  df_transformed.printSchema()
  df_transformed.select('year', 'month', 'day').show()
```


File Edit View Navigate Code Refactor Run Tools VCS Window Help itv-ghactivity - app.py
itv-ghactivity app.py
Project
itv-ghactivity C:\development\practice\itv-ghactivity
data
itvg-venv library root
app.py
new.py
process.py
read.py
util.py
External Libraries
Scratches and Consoles
app.py process.py read.py util.py new.py

```
def main():
    env = os.environ.get('ENVIRON')
    src_dir = os.environ.get('SRC_DIR')
    src_file_pattern = f'{os.environ.get("SRC_FILE_PATTERN")}-*'
    src_file_format = os.environ.get('SRC_FILE_FORMAT')
    spark = get_spark_session(env, 'GITHub Activity - Getting started' )
    spark.sql('SELECT current_date').show()
    df = from_files(spark, src_dir, src_file_pattern, src_file_format)
    df_transformed = transform(df)
    df_transformed.printSchema()
    df_transformed.select('repo.*', 'create_at', 'year', 'month', 'day').show()

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

```
C:\development\practice\itv-ghactivity\itvg-venv\Scripts\python.exe C:/development/practice/itv-ghactivity/app.py ITVersity

Process finished with exit code 0
```

Version Control Run TODO Problems Terminal Python Packages Python Console Event Log

Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built Python packages shared indexes // Always download // Download o... (yesterday 3:01 PM) 19:15 CRLF UTF-8 4 spaces Python 3.7 (itv-ghactivity)