

## Упр. - Сборка и запуск образа

- Сохраните этот код как Dockerfile в директории с вашим скриптом analyze.py и файлом data.csv. у меня файл data\_month.csv

Этот компьютер > System (C:) > Пользователи > 79181 > n.brykovskaya > module\_2 > 2Docker > 1Docker

Имя

analyze.py  
data\_month.csv  
Dockerfile

```
import pandas as pd
import matplotlib.pyplot as plt

data = pd.read_csv("data_month.csv")
print('Информация о датафрейме:\n\
=====')
print(data.info())
print('\nданные:
\n=====')
print(data)
plt.rcParams['figure.figsize'] = [12, 4]
plt.plot(data['month_id'],
         data['amount'],
         color='red',
         alpha=0.7,
         marker='o',
         linestyle='-',
         linewidth = 3)
plt.grid(alpha = 0.3, color = 'grey')
plt.xlabel('Месяц')
plt.ylabel('Продажи')
plt.title('Продажи по месяцам')

plt.savefig("graph.png")

print('The end')
```

Dockerfile – Блокнот

Файл Правка Формат Вид Справка

```
FROM python:3.9

WORKDIR /app
# Устанавливаем библиотеки
RUN pip install pandas matplotlib
# Копируем код и данные в образ
COPY analyze.py /app/
COPY data_month.csv /app/
# Запускаем скрипт при старте контейнера
CMD ["python", "analyze.py"]
```

n.brykovskaya > module\_2 > 2Docker > 1Docker >

Имя	Дата изменения	Тип
.ipynb_checkpoints	11.03.2025 9:28	Папка с
analyze.py	11.03.2025 9:28	Исходны
data_month.csv	11.03.2025 9:19	Файл Mi
Dockerfile	11.03.2025 9:37	Файл

- Откройте терминал, перейдите в эту директорию и выполните команду:

“ **docker build -t my-analysis-app .** ” -Эта команда соберет образ с именем my-analysis-app. Точка . в конце команды указывает текущий каталог в качестве контекста сборки.

```
C:\Users\79181\n.brykovskaya\module_2\2Docker\1Docker>docker build -t my-analysis-app .
[+] Building 8.1s (5/10)                                docker:desktop-linux
=> [internal] load build definition from Dockerfile      1.1s
=> => transferring dockerfile: 191B                     0.0s
=> [internal] load metadata for docker.io/library/python:3.9 3.5s
=> [auth] library/python:pull token for registry-1.docker.io 0.0s
=> [internal] load .dockerignore                        1.0s
=> => transferring context: 2B                          0.0s
=> [1/5] FROM docker.io/library/python:3.9@sha256:5ea663a1c6ba266fdcac5949d1d2ea364ce30a2da92a3d 2.3s
=> => resolve docker.io/library/python:3.9@sha256:5ea663a1c6ba266fdcac5949d1d2ea364ce30a2da92a3d 0.8s
=> [internal] load build context                        1.8s
=> => transferring context: 733B                        0.0s
```

```
<=> [2/5] WORKDIR /app
=> [3/5] RUN pip install pandas matplotlib
=> [4/5] COPY analyze.py /app/
=> [5/5] COPY data_month.csv /app/
=> exporting to image
=> => exporting layers
=> => exporting manifest sha256:9dbd07bb4bb4f367545904d7fb56675631b7fcdc3369649990441bd30b65c090
=> => exporting config sha256:2c28c9ce76ea072360246359d54fbe6c9a9198a57a1b3e520955c511082696ae
=> => exporting attestation manifest sha256:4ecbbcea46fcede3a7d9c854a075c8039e38ddee62655e121e22
=> => exporting manifest list sha256:cc2248d113f89a311b9920dd27dd86f9bf795c8fdcfce8e8321828870a6f
=> => naming to docker.io/library/my-analysis-app:latest
=> => unpacking to docker.io/library/my-analysis-app:latest

C:\Users\79181\n.brykovskaya\module_2\2Docker\1Docker>
```

Теперь, для того, чтобы посмотреть список всех загруженных образов, нужно выполнить:

docker images

```
C:\Users\79181\n.brykovskaya\module_2\2Docker\1Docker>docker images
REPOSITORY          TAG         IMAGE ID      CREATED       SIZE
my-analysis-app     latest     cc2248d113f8  About a minute ago  1.87GB

C:\Users\79181\n.brykovskaya\module_2\2Docker\1Docker>
```

Теперь откройте новое окно терминала (не закрывая и не отключаясь от текущего), и выполните команду docker ps -просмотр всех запущенных образов

```
Microsoft Windows [Version 10.0.19045.5487]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Users\79181>docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS        NAMES
C:\Users\79181>
```

ничего не запущено

- Запустите контейнер из созданного образа:

docker run -it --rm my-analysis-app

но чтобы сохранить выгрузку файла нужно код такой:

**docker run -it -v ./app --rm my-analysis-app**

-v ./app – это чтобы сохранить из app в текущую директорию

```
C:\Users\79181\n.brykovskaya\module_2\2Docker\1Docker>docker run -it --rm my-analysis-app  
C:\Users\79181\n.brykovskaya\module_2\2Docker\1Docker>_
```

-Эта команда запустит контейнер, выполнит скрипт analyze.py внутри контейнера, и по завершении удалит контейнер.  
-Результаты анализа (например, файл graph.png) будут сохранены в вашей локальной директории.

```
C:\Users\79181\n.brykovskaya\module_2\2Docker\1Docker>docker run -it -v ../app --rm my-analysis-app  
Информация о датафрейме:  
=====
```

1

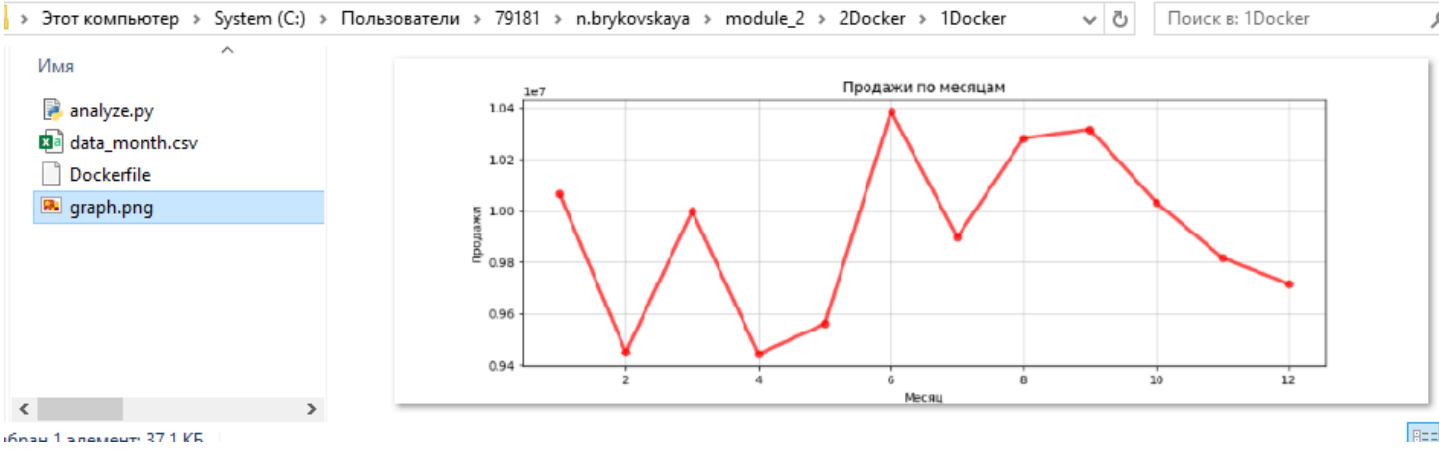
2

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 12 entries, 0 to 11  
Data columns (total 2 columns):  
#   Column      Non-Null Count  Dtype  
---  ---      -  
0   month_id    12 non-null    int64  
1   amount      12 non-null    int64  
dtypes: int64(2)  
memory usage: 320.0 bytes  
None  
  
данные:  
=====
```

	month_id	amount
0	1	10065653
1	2	9451536
2	3	9997897
3	4	9444852
4	5	9561238
5	6	10385066
6	7	9897312
7	8	10281610
8	9	10315691
9	10	10033283
10	11	9817268
11	12	9716343

```
The end  
C:\Users\79181\n.brykovskaya\module_2\2Docker\1Docker>_
```

проверим папку



```
C:\Users\79181\n.brykovskaya\module_2\2Docker\1Docker>pip freeze > requirements.txt  
C:\Users\79181\n.brykovskaya\module_2\2Docker\1Docker>exit_
```