РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций

«Работа с IPython и Jupyter Notebook»

Отчет по лабораторной работе № 3.1

по дисциплине «Программирование на Python»

Выполнил студент группы ИВТ-б-о-21-1

Эсеналиев Арсен.

«10» марта 2023г.

Подпись студента _____

Работа защищена « »_____20__г.

Проверил Воронкин Р.А. _____
(подпись)

Ставрополь 2023

Цель работы: исследовать базовые возможности интерактивных оболочек IPython и Jupyter Notebook для языка программирования Python.

Порядок выполнения работы:

1. Создал общедоступный репозиторий на GitHub, в котором использована лицензия MIT и язык программирования Python.

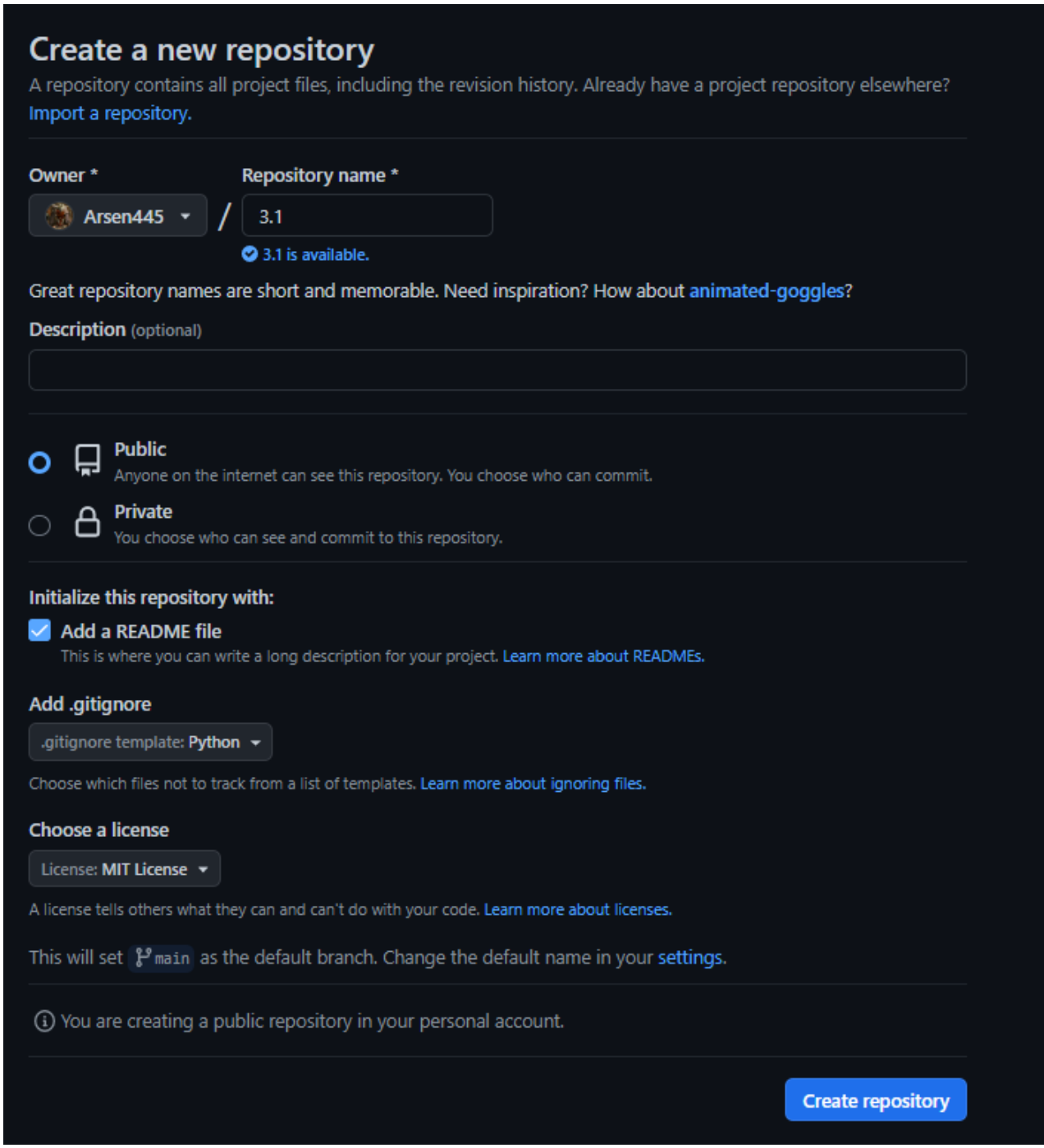

Рисунок 1 - Создание репозитория

2. Выполните клонирование созданного репозитория.

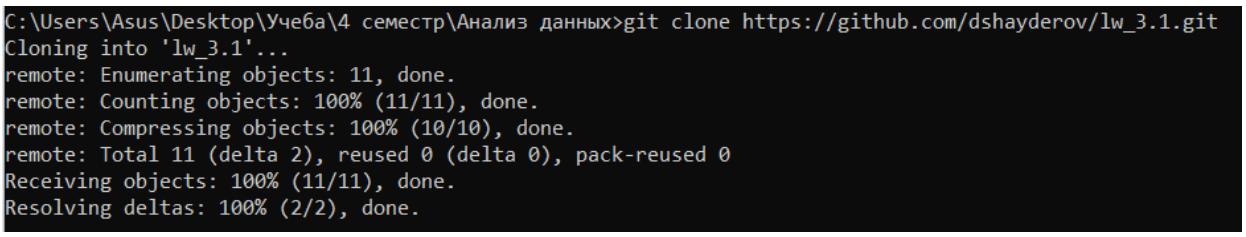

Рисунок 2 - Клонирование репозитория

3. Организуйте свой репозиторий в соответствие с моделью ветвления git-flow.

```
C:\Users\Asus\Desktop\Учеба\4 семестр\Анализ данных\lw_3.1>git checkout -b develop
Switched to a new branch 'develop'

C:\Users\Asus\Desktop\Учеба\4 семестр\Анализ данных\lw_3.1>
```

Рисунок 3 - Ветвление по модели git-flow

4. Проработать примеры лабораторной работы.

Пример 1. Выставите свойство “Code”, введите в ячейке “2 + 3” без кавычек и нажмите Ctrl+Enter или Shift+Enter, в первом случае введенный вами код будет выполнен интерпретатором Python, во втором – будет выполнен код и создана новая ячейка. Если у вас получилось это сделать, выполните еще несколько примеров.

Рисунок 4 - Результат выполнения примера 1

Пример 2. По умолчанию, графики не выводятся в рабочее поле ноутбука. Для того, чтобы графики отображались, необходимо ввести и выполнить следующую команду: *%matplotlib inline*. Пример вывода графика.

```
Ввод [1]: from matplotlib import pylab as plt
          %matplotlib inline

Ввод [2]: x = [i for i in range(50)]
          y = [i**2 for i in range(50)]
          plt.plot(x, y)

Out[2]: [<matplotlib.lines.Line2D at 0x1f18ca62cd0>]
```

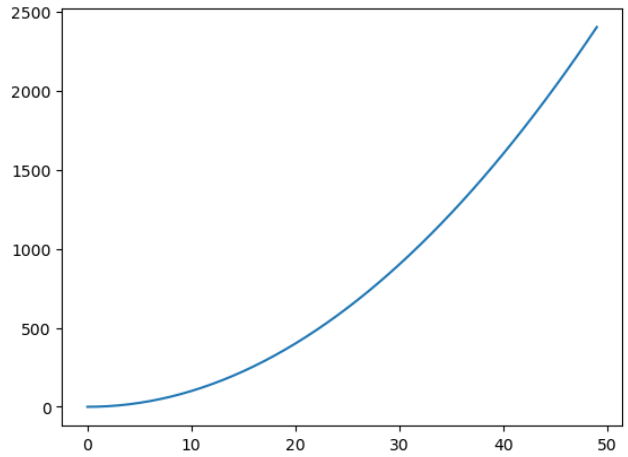

Рисунок 5 - Результат выполнения примера 2

Важной частью функционала Jupyter Notebook является поддержка магии. Под магией в IPython понимаются дополнительные команды, выполняемые в рамках оболочки, которые облегчают процесс разработки и расширяют ваши возможности.

Пример 3. %%time позволяет получить информацию о времени работы кода в рамках одной ячейки.

```
Ввод [1]: %%time
          import time
          for i in range(50):
              time.sleep(0.1)

          Wall time: 5.05 s
```

Рисунок 6 - Результат выполнения примера 3

Пример 4. %timeit запускает переданный ей код 100000 раз (по умолчанию) и выводит информацию о среднем значении трех наиболее быстрых прогонах.

```
Ввод [1]: %timeit x = [(i**10) for i in range(10)]
          2.35 µs ± 230 ns per loop (mean ± std. dev. of 7 runs, 100000 loops each)
```

Рисунок 7 - Результат выполнения примера 4

5. Создать ноутбук, в котором выполнить решение вычислительной задачи.

В тот момент, когда мимо станции со скоростью v м/с проходил товарный состав, от платформы в том же направлении отошёл пассажирский поезд. Через какое время пассажирский поезд догнал товарный, если пассажирский двигался с ускорением a м/с2 , а товарный – равномерно?

индивидуальное задание

Есть 6 городов, которые нужно посетить, но маршрут задаётся случайным образом

```
Задача - обойти все города, но с произвольным начальным городом и случайным порядком остальных городов.
```

```
Ввод [2]: cities = ['Москва', 'Санкт-Петербург', 'Казань', 'Сочи', 'Екатеринбург', 'Владивосток']
```

```
Ввод [3]: def visit(city):
              print(f"Посетили город: {city}")
```

```
Ввод [4]: import random

          def visit_cities(cities):
              # Выбираем случайный начальный город
              start_city = random.choice(cities)
              print(f"Начинаем путешествие с города: {start_city}")

              # Убираем начальный город из списка городов
              cities.remove(start_city)

              # Перемешиваем оставшиеся города
              random.shuffle(cities)

              # Добавляем начальный город в начало списка
              cities.insert(0, start_city)

              # Посещаем все города в новом порядке
              for city in cities:
                  visit(city)
```

```
Ввод [5]: visit_cities(cities)

          Начинаем путешествие с города: Москва
          Посетили город: Москва
          Посетили город: Владивосток
          Посетили город: Санкт-Петербург
          Посетили город: Казань
          Посетили город: Екатеринбург
          Посетили город: Сочи
```

Рисунок 8 - Результат выполнения индивидуального задания

Контрольные вопросы:

1. **Как осуществляется запуск Jupyter notebook?**

Jupyter Notebook входит в состав Anaconda. Для запуска Jupyter Notebook перейдите в папку Scripts (она находится внутри каталога, в котором установлена Anaconda) и в командной строке наберите:

> ipython notebook

В результате будет запущена оболочка в браузере.

2. Какие существуют типы ячеек в Jupyter notebook?

Если это код Python, то на панели инструментов нужно выставить свойство “Code”.

Если это Markdown текст – выставить “Markdown”.

3. Как осуществляется работа с ячейками в Jupyter notebook?

Если ваша программа зависла, то можно прервать ее выполнение выбрав на панели меню пункт

Kernel -> Interrupt.

Для добавления новой ячейки используйте Insert->Insert Cell Above и Insert->Insert Cell Below.

Для запуска ячейки используете команды из меню Cell, либо следующие сочетания клавиш:

Ctrl+Enter – выполнить содержимое ячейки.

Shift+Enter – выполнить содержимое ячейки и перейти на ячейку ниже.

Alt+Enter – выполнить содержимое ячейки и вставить новую ячейку ниже.

4. Что такое "магические" команды Jupyter notebook? Какие "магические" команды Вы знаете?

Важной частью функционала Jupyter Notebook является поддержка магии. Под магией в IPython понимаются дополнительные команды, выполняемые в рамках оболочки, которые облегчают процесс разработки и расширяют ваши возможности.

Для работы с переменными окружения используется команда %env.

Запуск Python кода из “.py” файлов, а также из других ноутбуков – файлов с расширением “.ipynb”, осуществляется с помощью команды %run.

%%time позволяет получить информацию о времени работы кода в рамках одной ячейки.

%timeit запускает переданный ей код 100000 раз (по умолчанию) и выводит информацию о среднем значении трех наиболее быстрых прогонах.

5. Самостоятельно изучите работу с Jupyter notebook и IDE PyCharm и Visual Studio Code.

Приведите основные этапы работы с Jupyter notebook в IDE PyCharm и Visual Studio Code.

PyCharm

1. Сначала вы должны создать новый проект.
2. В этом проекте создайте новый файл ipynb, выбрав File> New…> Jupyter Notebook. Это должно открыть новый файл записной книжки.
3. Если у вас не установлен пакет Jupyter Notebook, над вновь открытым файлом ipynb появится сообщение об ошибке. Сообщение об ошибке гласит: «Пакет Jupyter не установлен», и у вас будет опция «Установить пакет jupyter» рядом с ним.
4. Нажмите «Установить пакет jupyter». Это запустит процесс установки, который вы можете просмотреть, щелкнув запущенные процессы в правом нижнем углу окна PyCharm.
5. Чтобы начать изучение Jupyter Notebook в PyCharm, создайте ячейки кода и выполните их.
6. Выполните ячейку кода, чтобы запустить сервер Jupyter. По умолчанию сервер Jupyter использует порт 8888 по умолчанию на локальном хосте. Эти конфигурации доступны в окне инструментов сервера. После запуска вы можете просмотреть сервер над окном исходного кода, а рядом с ним вы можете просмотреть ядро, созданное как «Python 2» или «Python 3».
7. Теперь вы можете получить доступ к вкладке переменных в PyCharm, чтобы увидеть, как значения ваших переменных меняются при выполнении ячеек кода. Это помогает при отладке. Вы также можете установить точки останова в строках кода, а затем щелкнуть значок

«Выполнить» и выбрать «Debug Cell» (или использовать сочетание клавиш Alt+Shift+Enter), чтобы начать отладку.

Visual Studio Code

- Если у вас еще нет существующего файла Jupyter Notebook, откройте VS Code Command Palette с помощью сочетания клавиш CTRL+SHIFT+P (Windows) или Command+SHIFT+P (macOS) и запустите команду «Python: Create Blank New Jupyter Notebook».
- Если у вас уже есть файл Jupyter Notebook, это так же просто, как просто открыть этот файл в VS Code. Он автоматически откроется с новым нативным редактором Jupyter.

Вывод: были исследованы базовые возможности интерактивных оболочек IPython и Jupyter Notebook для языка программирования Python.