

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ
ФЕДЕРАЦИИ**

**Федеральное государственное автономное
образовательное учреждение высшего
образования
«СЕВЕРОКАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ
УНИВЕРСИТЕТ»**

Кафедра

инфокоммуникаций

Институт цифрового

развития

ОТЧЁТ

по лабораторной работе №3.1

Дисциплина: «Технологии распознавания образов»

Тема: «Работа с IPython и Jupyter Notebook»

Выполнила:

студентка 2 курса

группы Пиж-б-о-21-1

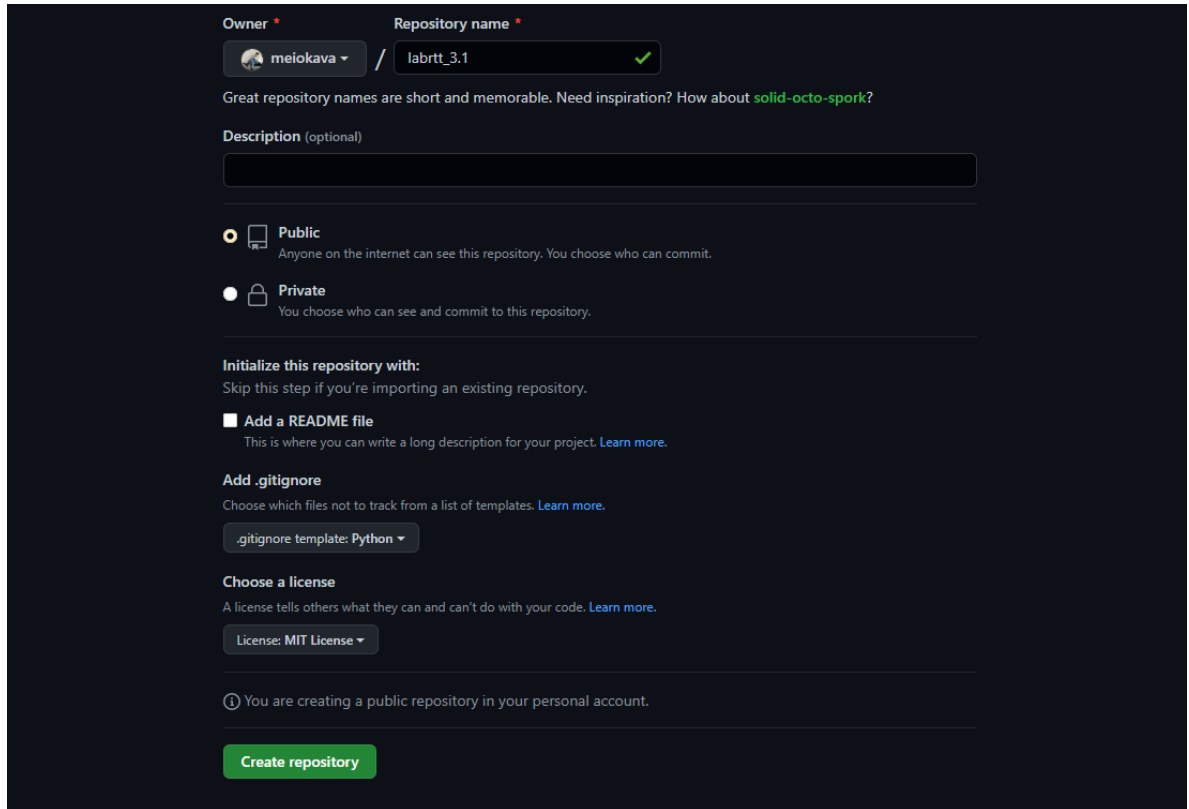
Джолдошова Мээрим

Бекболотовна

Ставрополь 2023

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

1. Был создан общедоступный репозиторий на GitHub, в котором использована лицензия MIT и выбран язык программирования python.



Owner * meiokava / Repository name * labrtrt_3.1 ✓

Great repository names are short and memorable. Need inspiration? How about [solid-octo-spork?](#)

Description (optional)

☒ Public
Anyone on the internet can see this repository. You choose who can commit.

☐ Private
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

☒ Add a README file
This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore
Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: Python ▾

Choose a license
A license tells others what they can and can't do with your code. [Learn more.](#)

License: MIT License ▾

ⓘ You are creating a public repository in your personal account.

Create repository

Рисунок 1 – Создание репозитория

2. Было выполнено клонирование созданного репозитория на рабочий компьютер.

```
C:\Users\мвидео>cd/d C:\gith\lb_3.1

C:\gith\lb_3.1>git clone https://github.com/meiokava/labrtrt_3.1.git
Cloning into 'labrtrt_3.1'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (4/4), done.

C:\gith\lb_3.1>|
```

Рисунок 2 – Клонирование репозитория

3. Организация репозитория в соответствии с моделью ветвления git-flow.

```
C:\gith\lb_3.1>cd/d C:\gith\lb_3.1\labrtt_3.1
C:\gith\lb_3.1\labrtt_3.1>git flow init

Which branch should be used for bringing forth production releases?
- main
Branch name for production releases: [main]
Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?
Feature branches? [feature/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? []
Hooks and filters directory? [C:/gith/lb_3.1/labrtt_3.1/.git/hooks]

C:\gith\lb_3.1\labrtt_3.1>
```

Рисунок 3 – Организация модели ветвления git-flow

4. Был дополнен файл .gitignore правилами для выбранного языка программирования, интерактивной оболочки Jupyter notebook и интегрированной среды разработки.

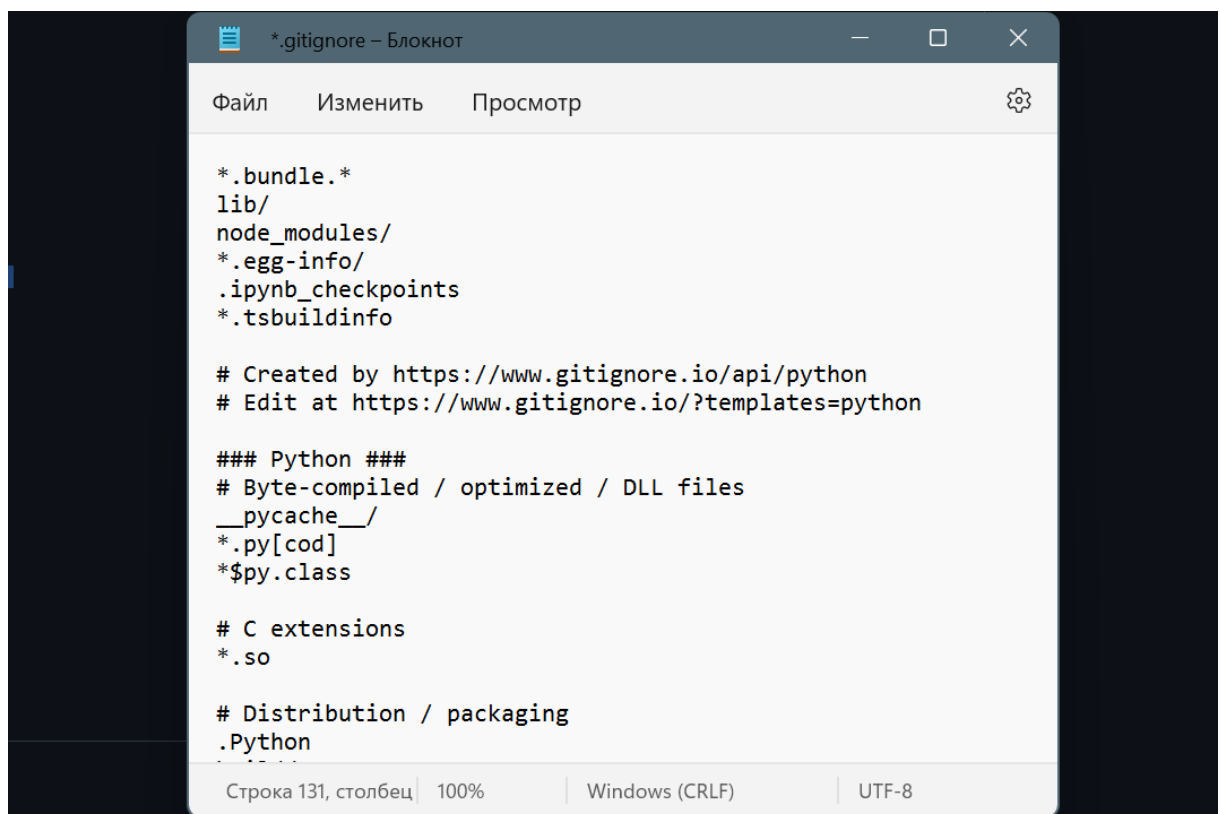


Рисунок 4 – Дополнение файла .gitignore

5. Проработать примеры лабораторной работы.

```
In [2]: 2+3
Out[2]: 5

In [4]: a = 5
        b = 7
        print(a+b)
12

In [6]: n = 7
        for i in range(n):
            print(i*10)
0
10
20
30
40
50
60

In [7]: i = 0
        while True:
            i += 1
            if i > 5:
                break
            print("Testing while")
Testing while
Testing while
Testing while
Testing while
Testing while
```

Рисунок 5 – Проработка примеров

```
In [9]: from matplotlib import pylab as plt
        %matplotlib inline

In [10]: x = [i for i in range(50)]
         y = [i**2 for i in range(50)]
         plt.plot(x, y)

Out[10]: [<matplotlib.lines.Line2D at 0x1f3a0fe1340>]
```

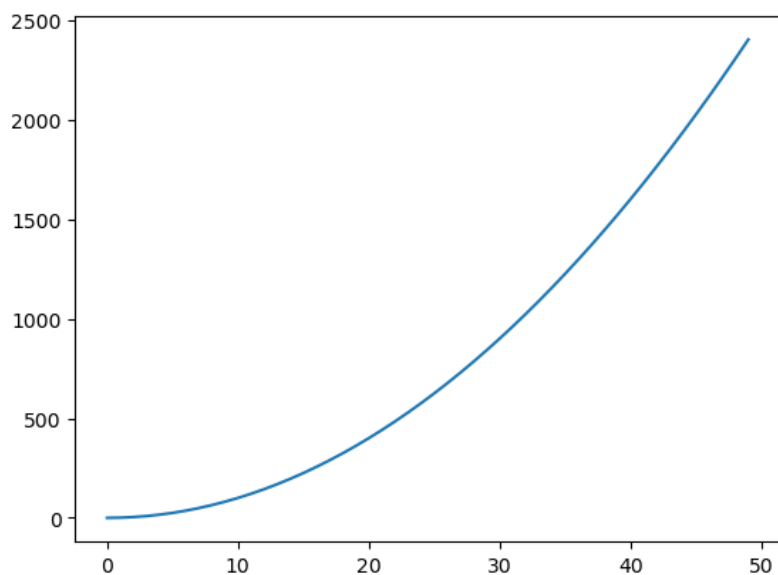


Рисунок 6 – Проработка примеров (продолжение)

```

In [12]: %lsmagic

Out[12]: Available line magics:
%alias %alias_magic %autoawait %autocall %automagic %autosave %bookmark
%cd %clear %cls %colors %conda %config %connect_info %copy %ddir %debug
%dhist %dirs %doctest_mode %echo %ed %edit %env %gui %hist %history
%killbgscripts %ldir %less %load %load_ext %loadpy %logoff %logon %logs
tart %logstate %logstop %ls %lsmagic %macro %magic %matplotlib %mkdir
%more %notebook %page %pastebin %pdb %pdef %pdoc %pfile %pinfo %pinfo2
%pip %popd %pprint %precision %prun %psearch %psource %pushd %pwd %pyc
at %pylab %qtconsole %quickref %recall %rehashx %reload_ext %ren %rep
%rerun %reset %reset_selective %rmdir %run %save %sc %set_env %store %
sx %system %tb %time %timeit %unalias %unload_ext %who %who_ls %whos
%xdel %xmode

Available cell magics:
%%! %%HTML %%SVG %%bash %%capture %%cmd %%debug %%file %%html %%javasc
ript %%js %%latex %%markdown %%perl %%prun %%pypy %%python %%python2 %
%python3 %%ruby %script %sh %svg %sx %system %%time %%timeit %%wr
itefile

Automagic is ON, % prefix IS NOT needed for line magics.

In [13]: %env TEST = 5

env: TEST=5

```

Рисунок 7 – Проработка примеров (продолжение)

```

In [15]: %%time
import time
for i in range(50):
    time.sleep(0.1)

Wall time: 5.44 s

In [16]: %timeit x = [(i*10) for i in range(10)]

2.29 µs ± 23.7 ns per loop (mean ± std. dev. of 7 runs, 100000 loops each)

In [ ]:

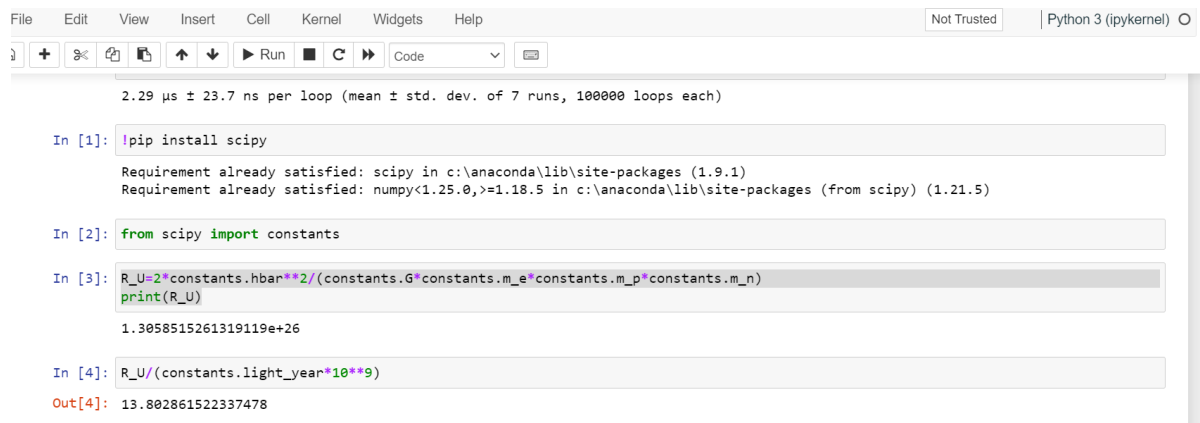
```

Рисунок 8 – Проработка примеров (продолжение)

6. Создать ноутбук, в котором выполнить решение вычислительной задачи (например, задачи из области физики, экономики, математики, статистики и т. д.), условие которой предварительно необходимо согласовать с преподавателем.

Universe Horizon Radius calculus

$$R_U = \frac{2\hbar^2}{G \cdot m_e \cdot m_p \cdot m_n}$$



```
File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

2.29 µs ± 23.7 ns per loop (mean ± std. dev. of 7 runs, 100000 loops each)

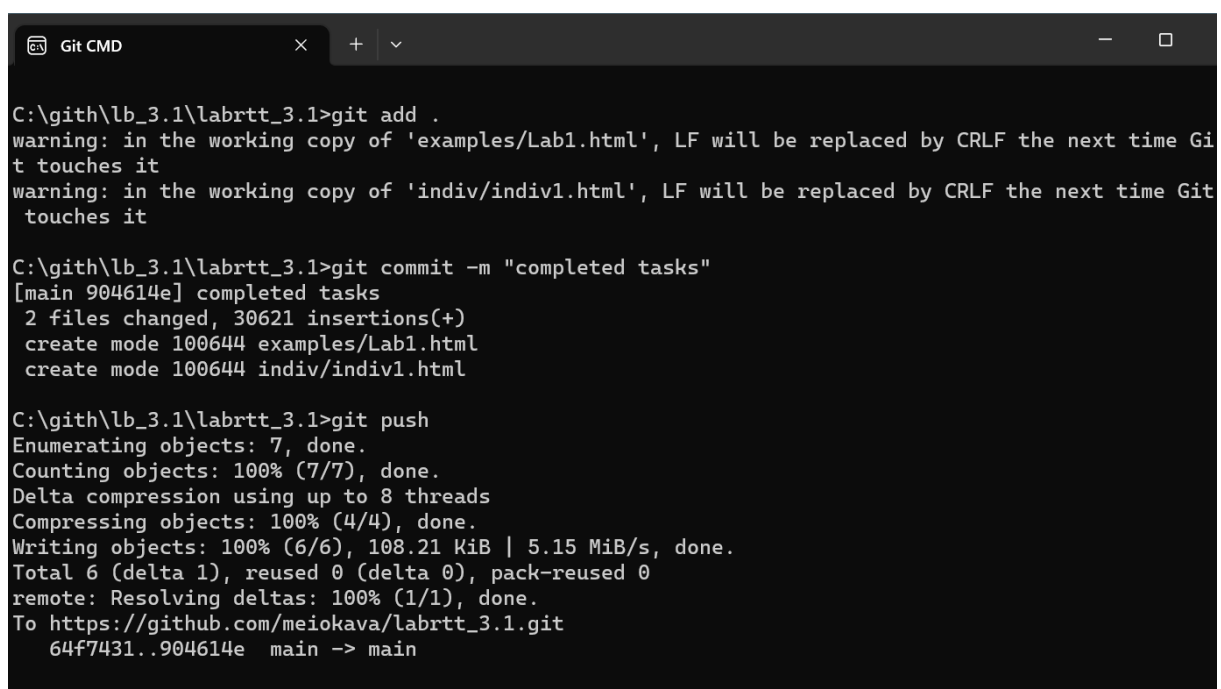
In [1]: pip install scipy
Requirement already satisfied: scipy in c:\anaconda\lib\site-packages (1.9.1)
Requirement already satisfied: numpy<1.25.0,>=1.18.5 in c:\anaconda\lib\site-packages (from scipy) (1.21.5)

In [2]: from scipy import constants

In [3]: R_U=2*constants.hbar**2/(constants.G*constants.m_e*constants.m_p*constants.m_n)
print(R_U)
1.3058515261319119e+26

In [4]: R_U/(constants.light_year*10**9)
Out[4]: 13.802861522337478
```

Рисунок 9 – Работа программы



```
Git CMD
C:\gith\lb_3.1\labrtr_3.1>git add .
warning: in the working copy of 'examples/Lab1.html', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'indiv/indiv1.html', LF will be replaced by CRLF the next time Git touches it

C:\gith\lb_3.1\labrtr_3.1>git commit -m "completed tasks"
[main 904614e] completed tasks
2 files changed, 30621 insertions(+)
create mode 100644 examples/Lab1.html
create mode 100644 indiv/indiv1.html

C:\gith\lb_3.1\labrtr_3.1>git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 108.21 KiB | 5.15 MiB/s, done.
Total 6 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/meiokava/labrtr_3.1.git
64f7431..904614e main -> main
```

Рисунок 10 – Гит коммит и пуш изменений

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?

Ячейки в блокноте Jupyter бывают четырех типов – Code, Markdown и

Raw и Headings.

Содержимое в ячейке Code обрабатывается как инструкции на языке программирования, по умолчанию используется Python.

Ячейки Markdown содержат текст, отформатированный с использованием языка markdown. Доступны все виды функций форматирования, такие как выделение текста жирным шрифтом и курсивом, отображение упорядоченного или неупорядоченного списка, отображение табличного содержимого и т.д.

Содержимое Raw ячейки не оценивается ядром notebook.

Headings-ячейка может использоваться для разбивки блокнота на разделы.

3. Как осуществляется работа с ячейками в Jupyter notebook?

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

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.