МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

КАФЕДРА «ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ»

ЗВІТ

з лабораторної роботи № 1

з навчальної дисципліни

«Поглиблений курс програмування Python»

ВИКОНАВ

Студент групи КН-423с

БРАТАШ Максим

ПЕРЕВІРИВ

викладач

Коваленко М.С

Харків – 2024

Тема: Робота з Jupyter Notebook.

Мета: отримати навички у встановленні,налаштуванні та основах роботи з Jupyter Notebook

Лістинги комірок

# **Greece**

### **General Information**

* **Area:** 131,957 km²
* **Population:** 10,724,599 (2020 est.)
* **Government:** Parliamentary Republic

### **Famous Personalities**

* **Philosophers:**
  + Socrates
  + Plato
  + Aristotle
* **Historical Figures:**
  + Alexander the Great
  + Pericles

### **Cultural Heritage**

* **Ancient Ruins:**
  + Acropolis of Athens
  + Ancient Agora of Athens
  + Temple of Olympian Zeus
* **Museums:**
  + National Archaeological Museum
  + Museum of Cycladic Art

### **Natural Wonders**

* **Islands:**
  + Santorini
  + Mykonos
  + Crete
* **Mountains:**
  + Mount Olympus
  + Mount Parnassus

### **Pictures**

****

### **Links**

* [Official Government Website](https://government.gov.gr/)
* [Greek National Tourism Organization](https://www.visitgreece.gr/)

| **Column 1** | **Column 2** | **Column 3** |
| --- | --- | --- |
| Row 1 | Row 1 | Row 1 |
| Row 2 | Row 2 | Row 2 |
| Row 3 | Row 3 | Row 3 |

In [7]:

**%%HTML**

<svg xmlns **=** "http://www.w3.org/2000/svg" version **=** "1.1" height **=** "100" width **=** "150" >

<g class **=** "greek\_flag\_image" transform **=** "translate(5,5)">

<g class **=** "background">

<rect x **=** "0" y **=** "0" stroke **=** "transparent" stroke-width **=** "0" height **=** "90" width **=** "135" fill **=** "#0D5EAF"></rect>

</g>

<g class **=** "stripes">

<path id **=** "first\_from\_bottom\_blue\_stripe" stroke **=** "#0D5EAF" stroke-width **=** "0" fill **=** "#0D5EAF" d **=** "M0, 80 L0, 90 L 135, 90 L135, 80 L0, 80 z" ></path>

<path id **=** "2nd\_from\_bottom\_white\_stripe" stroke **=** "#FFFFFF" stroke-width **=** "0" fill **=** "#FFFFFF" d **=** "M0, 70 L0, 80 L 135, 80 L135, 70 L0, 70 z"></path>

<path id **=** "third\_from\_bottom\_blue\_stripe" stroke **=** "#0D5EAF" stroke-width **=** "0" fill **=** "#0D5EAF" d **=** "M0, 60 L0, 70 L 135, 70 L135, 60 L0, 60 z"></path>

<path id **=** "fourth\_from\_bottom\_white\_stripe" stroke **=** "#FFFFFF" stroke-width **=** "0" fill **=** "#FFFFFF" d **=** "M0, 50 L0, 60 L 135, 60 L135, 50 L0, 50 z"></path>

<path id **=** "fifth\_from\_bottom\_blue\_stripe" stroke **=** "#0D5EAF" stroke-width **=** "0" fill **=** "#0D5EAF" d **=** "M0, 40 L 0,50 L135, 50 L135, 40 L 0,40"></path>

<path id **=** "sixth\_from\_bottom\_white\_stripe" stroke **=** "#FFFFFF" stroke-width **=** "0" fill **=** "#FFFFFF" d **=** "M0, 30 L0, 40 L 135, 40 L135, 30 L0, 30 z"></path>

<path id **=** "seventh\_from\_bottom\_blue\_stripe" stroke **=** "#0D5EAF" stroke-width **=** "0" fill **=** "#0D5EAF" d **=** "M0, 20 L0, 30 L135, 30 L135, 20 L0,20 z"></path>

<path id **=** "eighth\_from\_bottom\_white\_stripe" stroke **=** "#FFFFFF" stroke-width **=** "0" fill **=** "#FFFFFF" d **=** "M0, 10 L0, 20 L 135, 20 L135, 10 L0, 10 z"></path>

<path id **=** "ninth\_from\_bottom\_blue\_stripe" stroke **=** "#0D5EAF" stroke-width **=** "0" fill **=** "#0D5EAF" d **=** "M0, 0 L0, 10 L 135, 10 L135, 0 L0, 0 z"></path>

</g>

<g class **=** "blue\_canton">

<g class **=** "canton\_background">

<rect x **=** "0" y **=** "0" stroke **=** "transparent" stroke-width **=** "0" height **=** "50" width **=** "50" fill **=** "#0D5EAF"></rect>

</g>

<g class **=** "tau\_cross">

<path id **=** "equal\_armed\_cross" stroke **=** "#FFFFFF" stroke-width **=** "0" fill **=** "#FFFFFF" d **=** "M20, 0 L30, 0 L 30, 20 L50,20 L50,30 L30, 30 L30, 50 L20, 50 L 20, 30 L0, 30 L0, 20 L20, 20 L20, 0z"></path>

</g>

</g>

</g>

</svg>

In [9]:

**def** is\_prime(num):

**if** num **<=** 1:

**return** **False**

**if** num **<=** 3:

**return** **True**

**if** num **%** 2 **==** 0 **or** num **%** 3 **==** 0:

**return** **False**

i **=** 5

**while** i **\*** i **<=** num:

**if** num **%** i **==** 0 **or** num **%** (i **+** 2) **==** 0:

**return** **False**

i **+=** 6

**return** **True**

**def** task5(num):

**if** num **<=** 2 **or** num **%** 2 **!=** 0:

**return** "Число повинно бути парним та більше 2."

**for** i **in** range(2, num **//** 2 **+** 1):

**if** is\_prime(i) **and** is\_prime(num **-** i):

**return** f"Число {num} можна представити як {i}+{num **-** i}"

**return** "Не вдалося знайти представлення числа як суми двох простих чисел."

In [10]:

task5(5)

Out[10]:

'Число повинно бути парним та більше 2.'

In [12]:

task5(6)

Out[12]:

'Число 6 можна представити як 3+3'

In [13]:

task5(28)

Out[13]:

'Число 28 можна представити як 5+23'

In [ ]:

Діаграма активності

+---------------------------+

| Початок функції is\_prime |

+---------------------------+

|

|

|

V

+---------------------------+

| Перевірка на num <= 1 |

+------------+--------------+

|

| Так

|

V

+---------------------------+

| Повернути False |

+---------------------------+

|

|

|

V

+---------------------------+

| Перевірка на num <= 3 |

+------------+--------------+

|

| Так

|

V

+---------------------------+

| Повернути True |

+---------------------------+

|

|

|

V

+---------------------------+

| Перевірка на num % 2 == 0 |

+------------+--------------+

|

| Ні

|

V

+---------------------------+

| Перевірка на num % 3 == 0 |

+------------+--------------+

|

| Ні

|

V

+---------------------------+

| Ініціалізація змінної i |

+------------+--------------+

|

|

|

V

+---------------------------+

| Поки i \* i <= num |

+------------+--------------+

|

|

|

V

+----------------+-------------------+

| Перевірка на num % i == 0 або |

| num % (i + 2) == 0 |

+----------------+-------------------+

|

| Так

|

V

+----------------+-------------------+

| Повернути False |

+----------------+-------------------+

|

|

|

V

+----------------+-------------------+

| Збільшення значення i на 6 |

+----------------+-------------------+

|

|

|

V

+----------------+-------------------+

| Повернути True |

+----------------+-------------------+

| |

| |

V V

+----------------+---+ +-----+---------------+

| Кінець функції | | Початок функції |

| is\_prime | | goldbach\_conjecture |

+------------------+ +-----------------------+

|

|

|

V

+---------------------------+

| Перевірка на num <= 2 або|

| num % 2 != 0 |

+------------+--------------+

|

| Так

|

V

+---------------------------+

| Повернути "Число повинно |

| бути парним та більше 2." |

+---------------------------+

|

|

|

V

+---------------------------+

| Перевірка для i в діапазоні|

| [2, num // 2] |

+------------+--------------+

|

|

|

V

+---------------------------+

| Перевірка на is\_prime(i) |

| та is\_prime(num - i) |

+------------+--------------+

|

| Так

|

V

+---------------------------+

| Повернути "Число num можна|

| представити як i+(num - i)"|

+---------------------------+

|

|

|

V

+---------------------------+

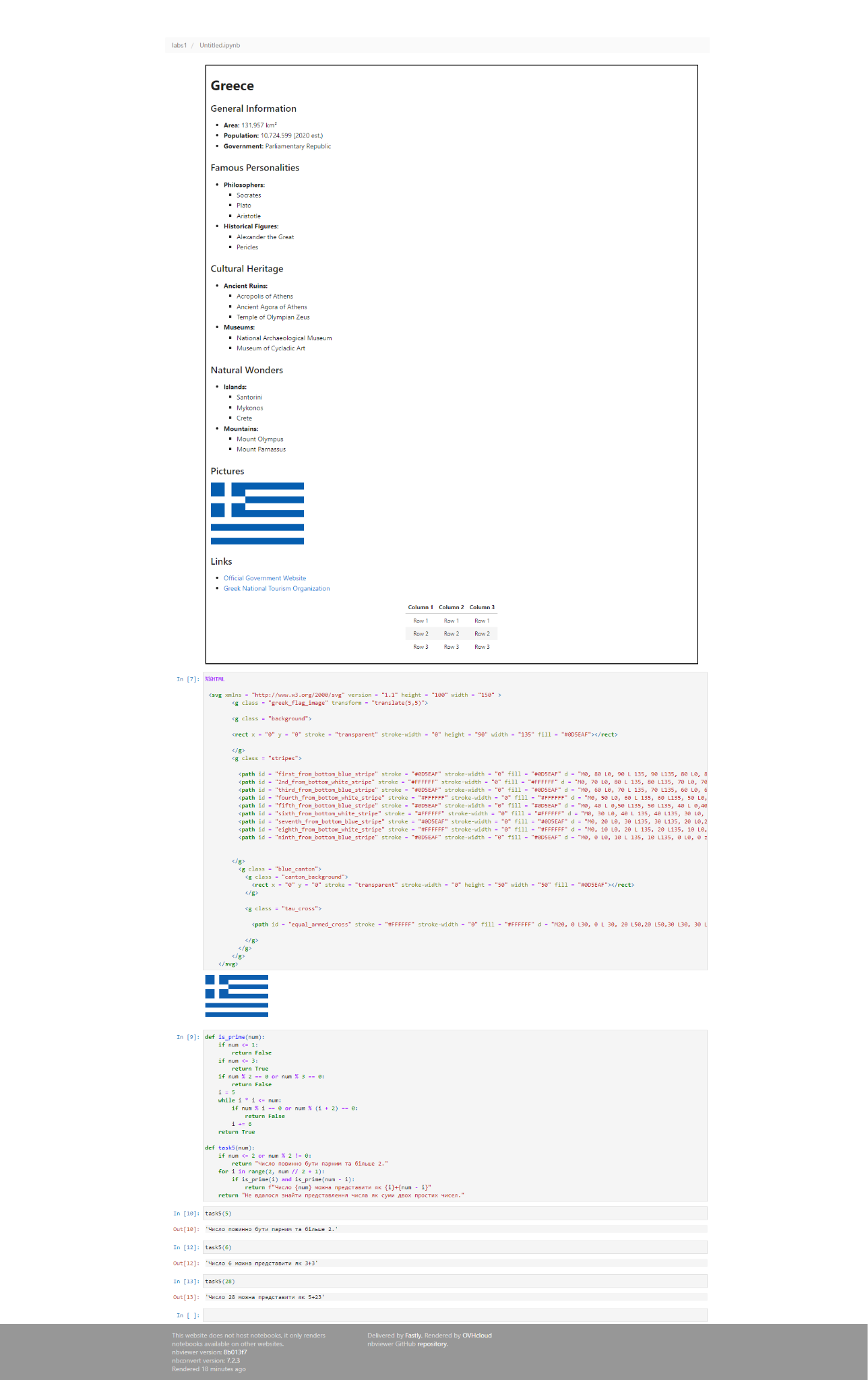
| Повернути "Не вдалося |

| знайти представлення числа|

| як суми двох простих чисел."|

+---------------------------+

Скріншоти блокноту Jupiter з вхідними та вихідними даними.



Посилання на створений блокнот

<https://github.com/kitska/labs1/blob/master/Untitled.ipynb>

Контрольні запитання

1. Jupyter Notebook - це інтерактивне середовище для програмування, яке дозволяє об'єднувати код, текст та графіку в одному документі. Він підтримує багато мов програмування, таких як Python, R, Julia та інші.
2. Існують кілька способів встановлення Jupyter Notebook:
   * Встановлення за допомогою Anaconda, яка включає Jupyter Notebook у свій дистрибутив.
   * Встановлення через pip для Python за допомогою команди pip install notebook.
   * Використання веб-сервісів, таких як Google Colab або Azure Notebooks, які надають доступ до Jupyter Notebook в хмарному середовищі.
3. Для створення окремого оточення для Jupyter Notebook можна використати віртуальне середовище, таке як virtualenv для Python або conda env для Anaconda. Наприклад, для створення віртуального середовища з Python 3 можна виконати команду virtualenv myenv -p python3.
4. Для запуску Jupyter Notebook з відповідної папки за допомогою командного рядка використовується команда jupyter notebook.
5. В середовищі Jupyter Notebook існують два режими:
   * Режим команди (Command mode): Використовується для навігації та виконання дій з клавіатури, таких як видалення, додавання нових комірок та інші.
   * Режим редагування (Edit mode): Використовується для редагування вмісту комірки.
6. Способи запуску комірки на виконання:
   * Натискання кнопки "Run" у верхньому меню.
   * Використання клавіші "Shift + Enter".
   * Вибір команди "Run Cells" у верхньому меню або контекстному меню комірки.
7. Щоб створити комірку з кодом на іншій мові програмування всередині Python Jupyter Notebook, потрібно встановити та налаштувати ядро (kernel) для цієї мови програмування. Після цього можна створити нову комірку та обрати встановлене ядро для її виконання.
8. Мова Markdown - це легка у використанні мова розмітки, яка використовується для форматування тексту у веб-документах. Вона дозволяє створювати заголовки, списки, посилання, таблиці, формули та багато іншого.
9. Для створення формули в новому рядку за допомогою мови LaTeX в Markdown потрібно використати подвійні доларові знаки, наприклад, $$ формула $$.
10. Способи перегляду та запуску блокноту Jupyter, розміщеного на GitHub:
    * Прогляд через GitHub: Можна переглядати блокноти Jupyter, які зберігаються у репозиторії GitHub, просто переходячи за посиланням на файл у веб-браузері.
    * Використання сервісів для візуалізації: Наприклад, GitHub підтримує відображення блокнотів Jupyter прямо на своєму веб-інтерфейсі без необхідності завантаження.
    * Запуск блокноту локально: Завантаження блокноту з GitHub та запуск його локально на власному комп'ютері з використанням Jupyter Notebook.

Висновок

Jupyter Notebook - це потужний інструмент для розробки, тестування та демонстрації програмного коду, а також для створення документів, що поєднують в собі код, текст, графіку та форматовані елементи. Він дозволяє працювати з різними мовами програмування, зокрема з Python, R, Julia та іншими. Встановлення Jupyter Notebook можливе як за допомогою популярних дистрибутивів, таких як Anaconda, так і через pip для Python. Для комфортної роботи з Jupyter Notebook можна створити окреме віртуальне середовище. Використання мови Markdown дозволяє зрозуміло форматувати текс, створювати таблиці, формули та інші структуровані елементи. Крім того, блокноти Jupyter можна легко ділитися через платформу GitHub, переглядати їх прямо на веб-сторінці та запускати локально для подальшої роботи. Всі ці можливості роблять Jupyter Notebook незамінним інструментом для досліджень, навчання та розв'язання різноманітних завдань в області програмування та аналізу даних.