

Python: Quarto

Прикладна аналітика при розробці IT

Ігор Мірошниченко

КНУ імені Тараса Шевченка, ФІТ

Прикладна аналітика при розробці IT



QUARTO

Прикладна аналітика при розробці ІТ



QUARTO

Quarto — це науково-технічна видавнича система з відкритим кодом, яка ґрунтуються на універсальному конверторі документів Pandoc та використовує мову розмітки Markdown.

Це універсальний інструмент для тих, хто пише на [R](#), [Python](#), [Julia](#) та [Observable JavaScript](#).

Прикладна аналітика при розробці IT



НАЗВА

Розробники **Quarto** хотіли використати ім'я, яке мало деяке значення в історії видавничої справи.

Вибір пав на **Quarto** (від латинського *quārtō*, скорочено *Qto*, *4to* або *4º*) — це формат книги або брошури в одну четвертину топографічного листа.

На кожній стороні листа при цьому поміщається 4 сторінки книги (8 сторінок на один лист).



МОЖЛИВОСТІ QUARTO

Quarto являє собою текстовий документ спеціального формату [.qmd](#), який можна скомпілювати у різноманітні документи:

- Документи у форматах:
 - HTML
 - PDF
 - ePub
 - MS Word
 - OpenOffice
- Презентації у форматах:
 - reveal.js (HTML)
 - PowerPoint (MS Office)
 - Beamer (LaTeX/PDF)
- Веб-сторінки:
 - Practical Deep Learning від fast.ai
 - nbdev від fast.ai
 - Kypc Julia Workshop for Data Science
- Книги, приклади яких можна переглянути за посиланнями:
 - Python for Data Analysis
 - R for Data Science
 - Visualization Curriculum
- Інтерактивні документи:
 - Observable reactive JavaScript
 - Shiny вебфреймворк
 - Інтерактивні віджети Jupyter

ВСТАНОВЛЕННЯ QUARTO

Для роботи з Quarto потрібно:

1. Встановити інтерфейс командного рядка Quarto (CLI) під актуальну операційну систему (Windows, Linux або Mac OS).
2. Встановити plug-in для інтегрованого середовища розробки (IDE). На даний момент підтримуються:
 - VS Code
 - RStudio
 - Jupyter
 - Neovim
 - Text Editor

РОБОТА З ПРОЕКТАМИ

Створити Quarto-проект можна декількома шляхами:

1. За допомогою **командного рядка**:

terminal

```
1 # веб-сайт
2 quarto create-project mysite --type website
3
4 # блог
5 quarto create-project myblog --type website:blog
6
7 # книга
8 quarto create-project mybook --type book
```

Прикладна аналітика при розробці ІТ

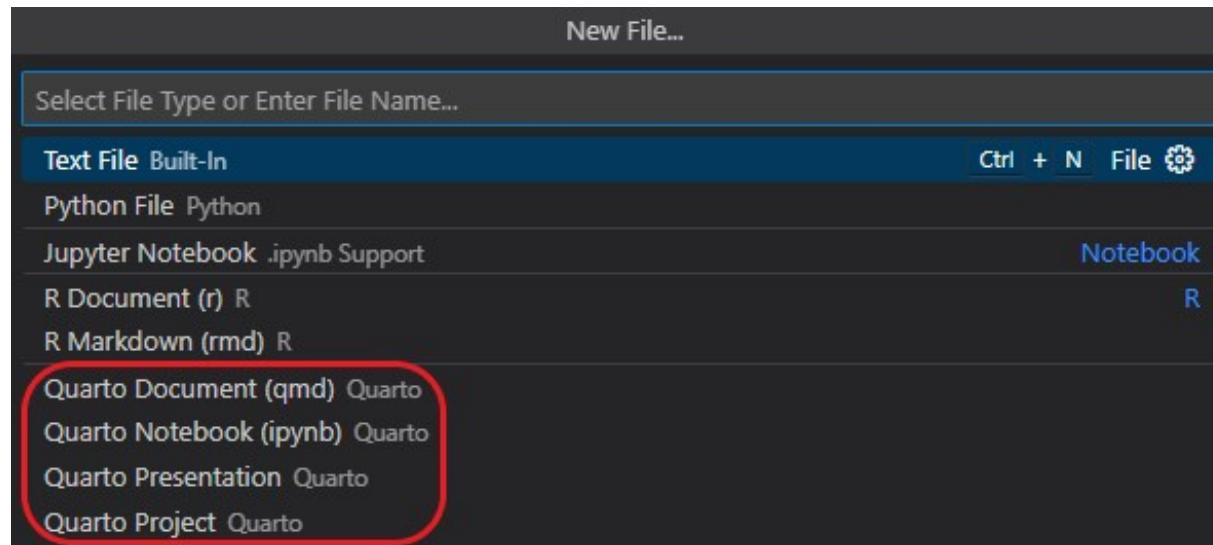


РОБОТА З ПРОЕКТАМИ

Створити Quarto-проект можна декількома шляхами:

2. Засобами **VS Code**:

File -> New File -> ... або Ctrl+Win+Alt+N

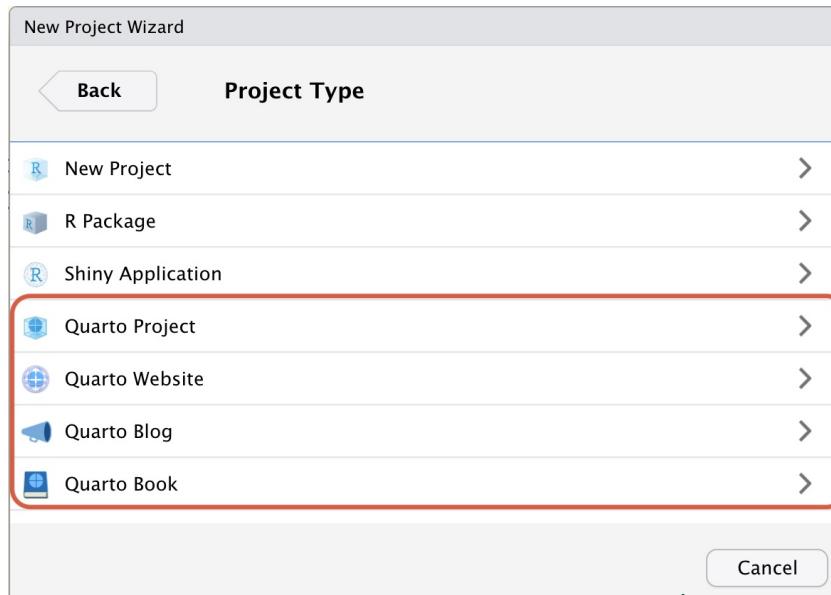


РОБОТА З ПРОЕКТАМИ

Створити Quarto-проект можна декількома шляхами:

3. Засобами RStudio:

File -> New Project... -> New Directory



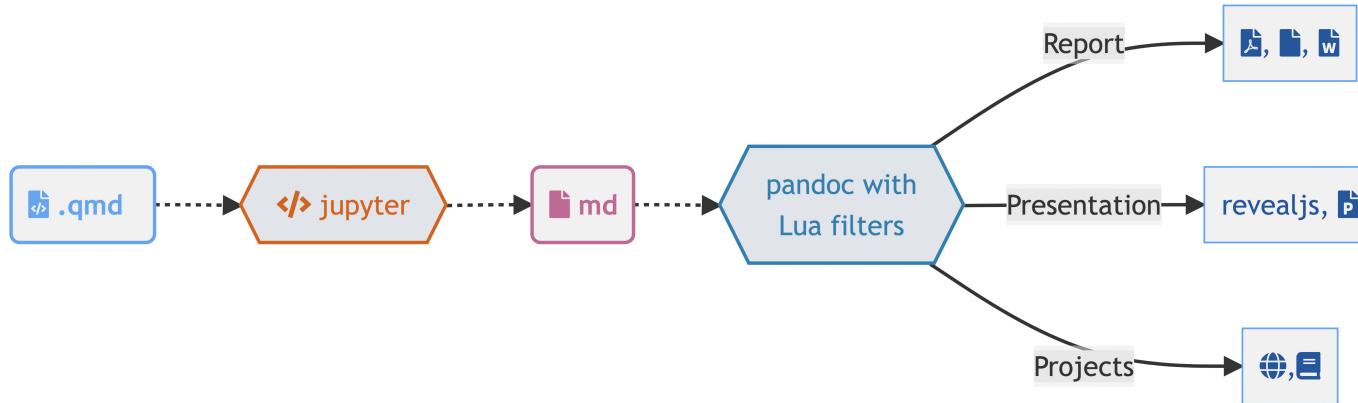
Прикладна аналітика при розробці IT



ЯК ПРАЦЮЄ QUARTO

Спочатку **Knitr** (для R) або **Jupyter** (для Python або Julia) виконує всі фрагменти коду **.qmd**-файлу і створює новий **markdown (.md)** документ, який включає в себе код і всі його результати.

Далі **.md**-файл оброблюється **Pandoc** для перетворення у різноманітні формати файлів (**HTML**, **PDF**, **Word** тощо).



РЕНДЕРІНГ

- За допомогою кнопки **Render** в VS Code.



- За допомогою командного рядка:

```
terminal
1 quarto render <input> --to <format>
2
3 # Наприклад:
4 quarto render document.qmd --to docx
```

A screenshot of a terminal window titled 'terminal'. It contains four numbered lines of text demonstrating the 'quarto render' command. Line 1 shows the basic command structure with placeholders for input and output formats. Line 3 is a note in Russian: '# Наприклад:' (For example:). Line 4 shows a specific example where 'document.qmd' is rendered to 'docx' format.

СТРУКТУРА ДОКУМЕНТІВ QUATRO

Quatro документи складаються з трьох базових складових:

- YAML-шапки¹
- Тексту з використанням розмітки Markdown
- Чанків (анг. *chunks*) з кодом

1. YAML це рекурсивний акронім YAML Ain't Markup Language («YAML — не мова розмітки»). У назві відображені історія розвитку: на ранніх етапах мова називалася Yet Another Markup Language («Ще одна мова розмітки») і навіть розглядалася як конкурент XML. Італієць пізніше  а переїменував мову з метою акцентувати увагу на пакетах а не на позбавлені покоментів.

YAML-ШАПКА

YAML-шапка знаходитьться нагорі документу і відділена трьома дефісами (`---`) зверху та знизу.

В ній зберігається мета-інформація документу: назва, дата створення, автор, інформація щодо роботи коду, контенту і процесу рендерінгу.

```
1 ---
2 title: "Науково-видавнича система Quarto"
3 subtitle: "Мала академія наук України"
4 author: "Igor Miroshnichenko"
5 date: today
6 date-format: iso
7 institute: КНЕУ::ІІТЕ
8 language: _language-ukr.yml
9 format:
10 revealjs:
11   theme: [simple, custom.scss]
12   logo: img/logo.png
13   chalkboard: true
14   slide-number: true
15   toc: true
16   toc-title: ЗМИТ
17   transition: fade
18   mouse-wheel: true
19   highlight-style: github
20   mainfont: metropolis
21   toc-depth: 1
22 editor: source
23 execute:
24   echo: true
25 jupyter: python3
26 ---
```

Прикладна аналітика при розробці IT



MARKDOWN

Ця частина документу йде одразу після YAML-шапки і складає основну частину документу.

[Markdown](#) — це популярна і зручна мова розмітки.

Ви непевно зустрічали її в README.md-файлах репозиторіїв [GitHub](#), а також у [Telegram](#)-повідомленнях.

```
1 *Зродились ми великої години* \
2 **з пожеж війни і полум'я вогнів.**\
3 ***Плекав нас біль за долю України,***\
4 `Зростив в нас гнів` і лють на ~~ворогів
```

**Зродились ми великої години
3 пожеж війни і полум'я вогнів.
Плекав нас біль за долю України,
Зростив в нас гнів і лють на ворогів!**

```
1 $$ x = \{-b \pm \sqrt{b^2 - 4ac} \over 2a\} $$
```

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$



ЧАНКИ З КОДОМ

Chunks – це блоки, які відділяються від тексту потрійними зворотніми лапками `` ` (анг. backtick) на початку та в кінці. У фігурних дужках вказується мова програмування на якій необхідно виконати код.

```
1 ````{python}
2 print('Fly, you fools!')
3 ````
```

Fly, you fools!

Результат чанку за замовчування виводиться сразу після нього, але все це можна налаштувати.

```
1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
4
5 x = np.arange(-4, 4, .012)
6 y = np.arange(-4, 4, .012)
7 X, Y = np.meshgrid(x, y)
8 Z = 1 - np.abs(X) - np.sin(Y**2)
9 W = 1 + Y - np.cos(X**2)
10
11 fig = plt.figure(figsize=(4, 4))
12 ax = fig.add_subplot(projection='polar')
13 plt.scatter(Z, W, alpha=.03, s=0.2)
14 plt.axis('off')
15 plt.show()
```



Прикладна аналітика при розробці IT



Прикладна аналітика при розробці ІТ

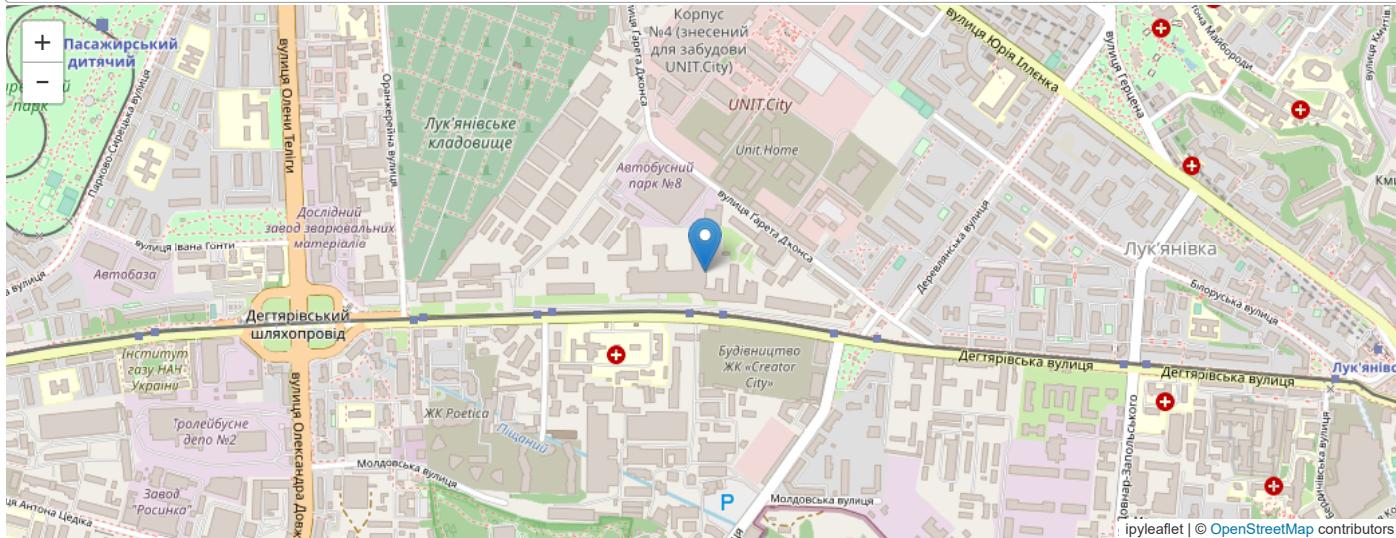


ЧАНКИ З КОДОМ

```

1 from ipyleaflet import Map, Marker, basemaps, basemap_to_tiles
2 m = Map(
3     center=(50.46361774858406, 30.460139871139447),
4     zoom=15
5 )
6 m.add_layer(Marker(location=(50.46361774858406, 30.460139871139447)))
7 m

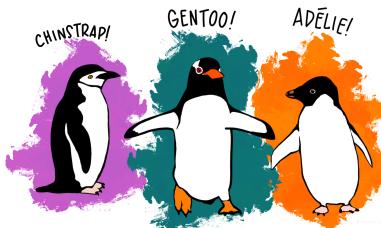
```



Прикладна аналітика при розробці IT



ЧАНКИ З КОДОМ



```
1 from palmerpenguins import load_penguins
2
3 df = load_penguins()
4 df
```

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex	year
0	Adelie	Torgersen	39.1	18.7	181.0	3750.0	male	2007
1	Adelie	Torgersen	39.5	17.4	186.0	3800.0	female	2007
2	Adelie	Torgersen	40.3	18.0	195.0	3250.0	female	2007
3	Adelie	Torgersen	NaN	NaN	NaN	NaN	NaN	2007
4	Adelie	Torgersen	36.7	19.3	193.0	3450.0	female	2007
...
339	Chinstrap	Dream	55.8	19.8	207.0	4000.0	male	2009
340	Chinstrap	Dream	43.5	18.1	202.0	3400.0	female	2009
341	Chinstrap	Dream	49.6	18.2	193.0	3775.0	male	2009
342	Chinstrap	Dream	50.8	19.0	210.0	4100.0	male	2009
343	Chinstrap	Dream	50.2	18.7	198.0	3750.0	female	2009

344 rows × 8 columns

Прикладна аналітика при розробці ІТ

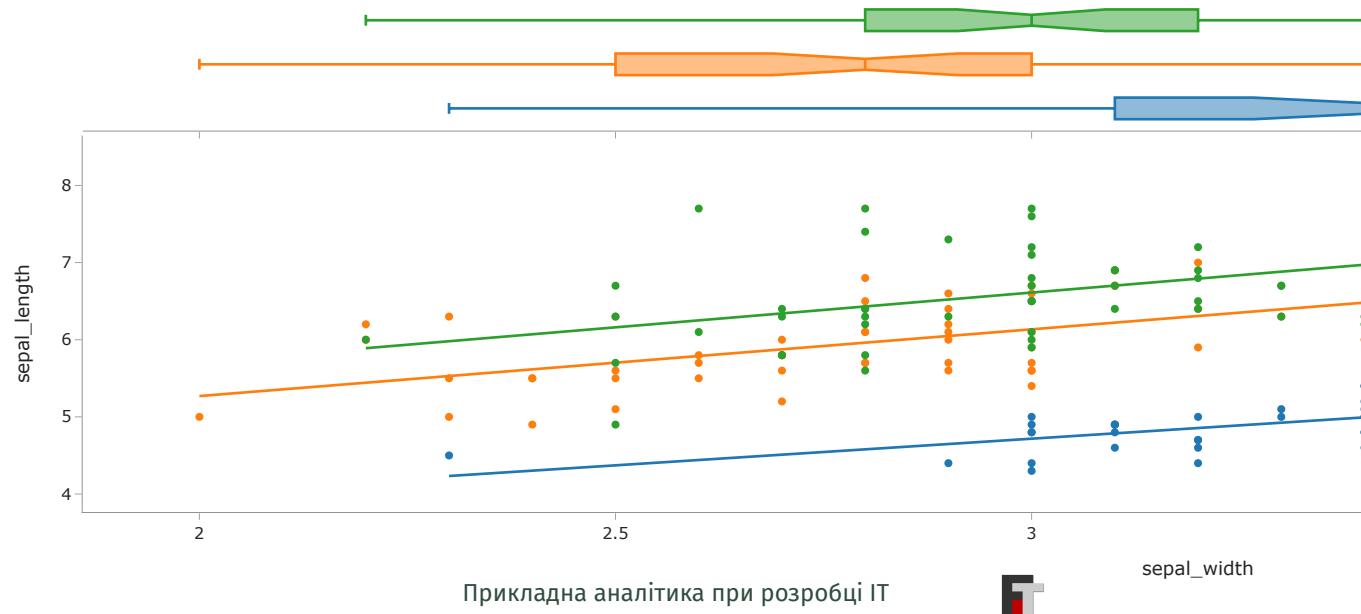


ЧАНКИ З КОДОМ

```

1 import plotly.express as px
2 import plotly.io as pio
3 df = px.data.iris()
4 fig = px.scatter(df, x="sepal_width", y="sepal_length",
5                   color="species",
6                   marginal_y="violin", marginal_x="box",
7                   trendline="ols", template="simple_white")
8 fig.show()

```



Прикладна аналітика при розробці ІТ



ДЯКУЮ ЗА УВАГУ!

- ✉️ Матеріали курсу
- ✉️ ihor.miroshnychenko@kneu.ua
- 👉 Data Mirosh
- 🔗 [@ihormiroshnychenko](https://www.linkedin.com/in/ihormiroshnychenko)
- 👤 [@aranaur](https://www.instagram.com/aranaur)
- 🏡 aranaur.rbind.io

Прикладна аналітика при розробці ІТ

