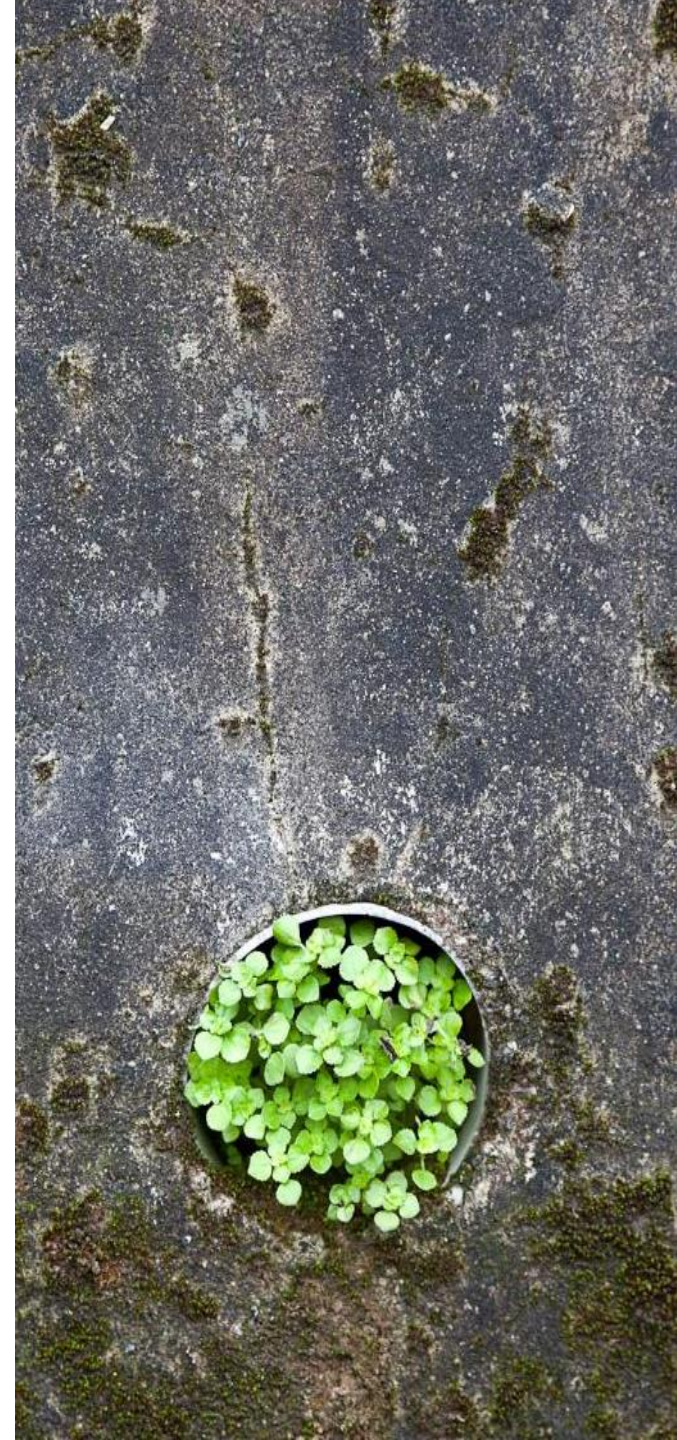






eleven Hackathon

Introduction to shared code

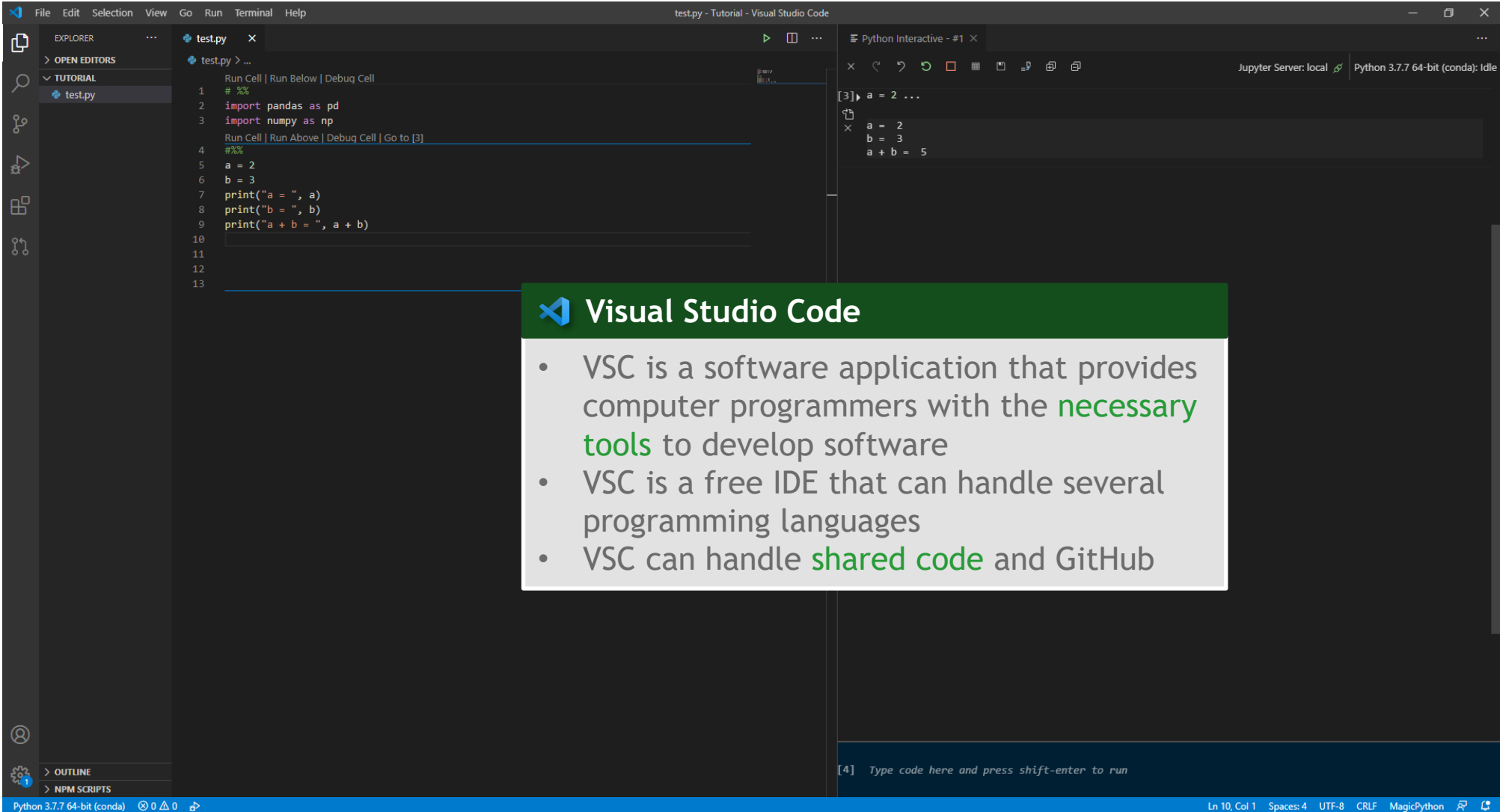
To the attention of the Data Sciences
& Business Analytics master students
February 8th, 2021



AGENDA

- I  Why using VSC ?
- II  Why using shared code ?
- III  What is GitHub and how to use it ?
- IV  How to use GitHub with VSC?

Eleven's IDE is Microsoft's visual studio code



The screenshot displays the Visual Studio Code IDE interface. The main editor window shows a file named `test.py` with the following code:

```
1 # %%  
2 import pandas as pd  
3 import numpy as np  
4  
5 a = 2  
6 b = 3  
7 print("a = ", a)  
8 print("b = ", b)  
9 print("a + b = ", a + b)  
10  
11  
12  
13
```

The right-hand side of the interface features the Python Interactive window, which shows the execution of the code:

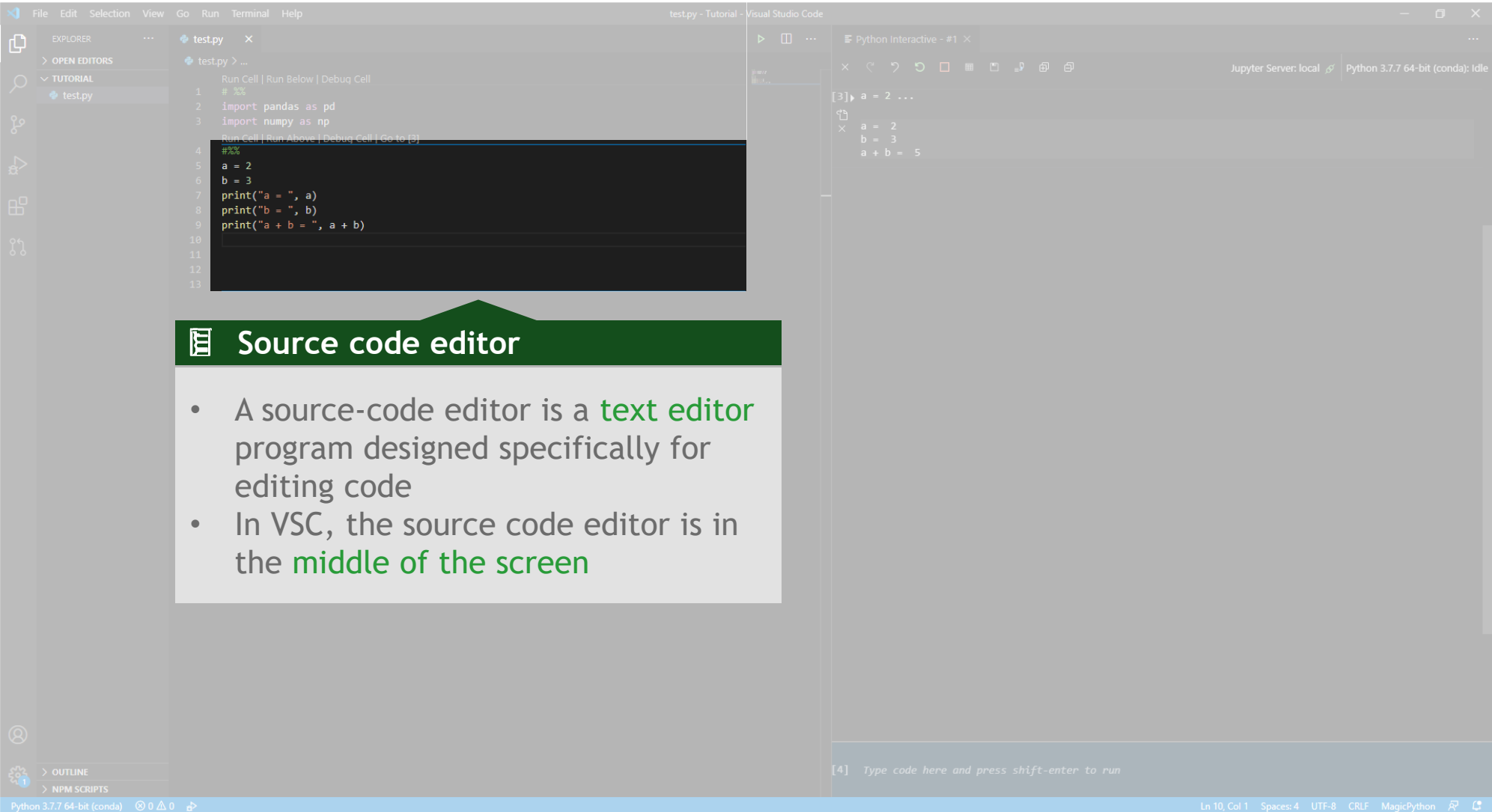
```
[3] a = 2 ...  
a = 2  
b = 3  
a + b = 5
```

Below the code editor, there is a green box with the Visual Studio Code logo and the text "Visual Studio Code". To the right of this box is a list of bullet points:

- VSC is a software application that provides computer programmers with the **necessary tools** to develop software
- VSC is a free IDE that can handle several programming languages
- VSC can handle **shared code** and GitHub

The status bar at the bottom of the IDE shows "Python 3.7.7 64-bit (conda)" and "Ln 10, Col 1 Spaces: 4 UTF-8 CRLF MagicPython".

The source code editor is where **code will be written**



The screenshot displays the Visual Studio Code (VSC) interface. On the left, the Explorer sidebar shows a file named 'test.py' under a 'TUTORIAL' folder. The main editor window shows the source code of 'test.py', which includes comments and Python code for importing pandas and numpy, and printing variables a and b. On the right, the 'Python Interactive' window is open, showing the execution of the code: 'a = 2', 'b = 3', and 'a + b = 5'. The status bar at the bottom indicates 'Python 3.7.7 64-bit (conda)' and 'Ln 10, Col 1'.

Source code editor

- A source-code editor is a **text editor** program designed specifically for editing code
- In VSC, the source code editor is in the **middle of the screen**

Build automation tools enable code lines execution



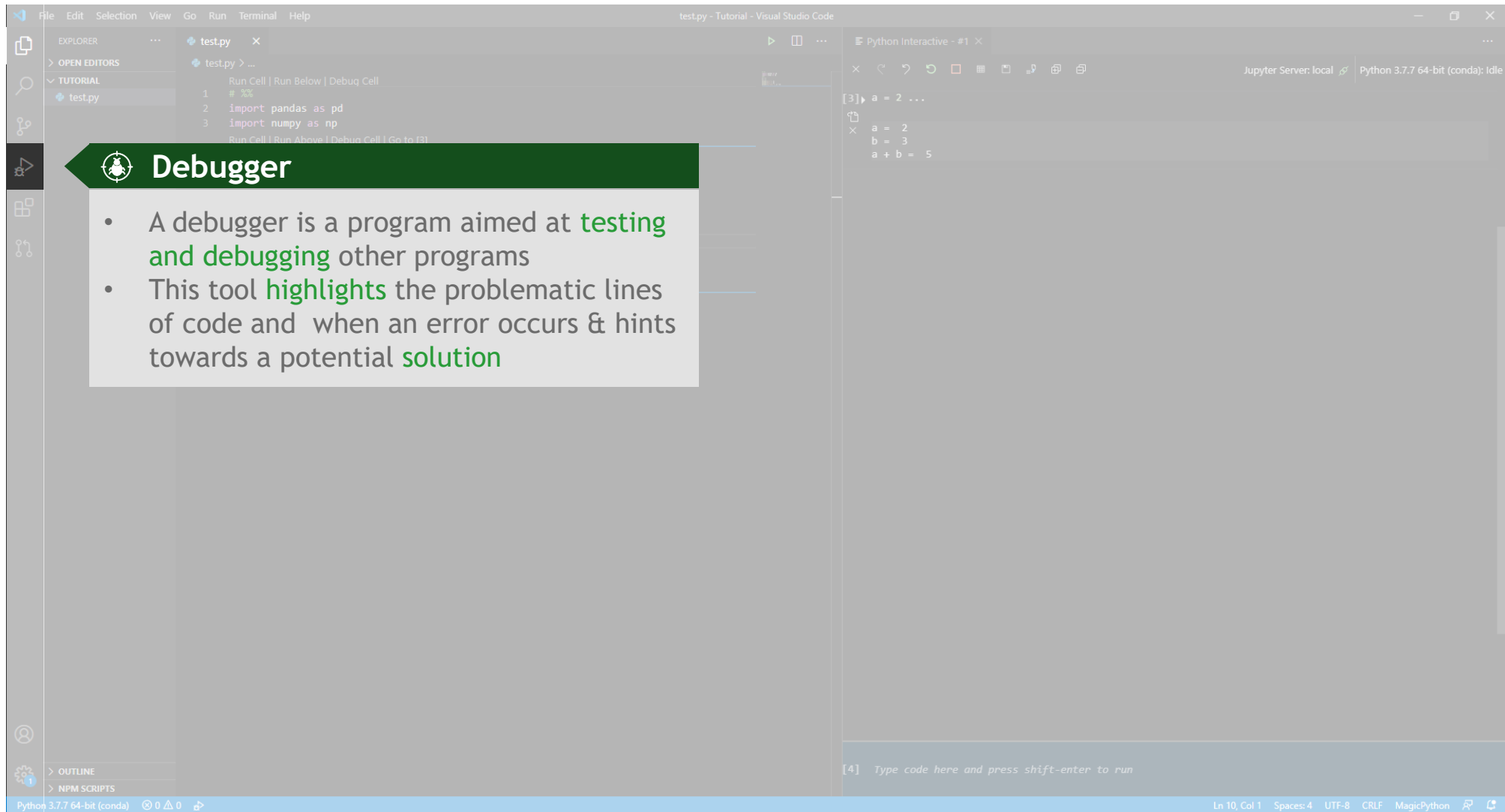
Build automation tools

- Build automation tools seamlessly create a piece of **software** from a given code
- The highlighted part is the **result of the code** written in the previous slide being run

```
[3] a = 2 ...  
a = 2  
b = 3  
a + b = 5
```

```
[4] Type code here and press shift-enter to run
```

The Debugger aims at helping the programmer **fix mistakes**



Debugger

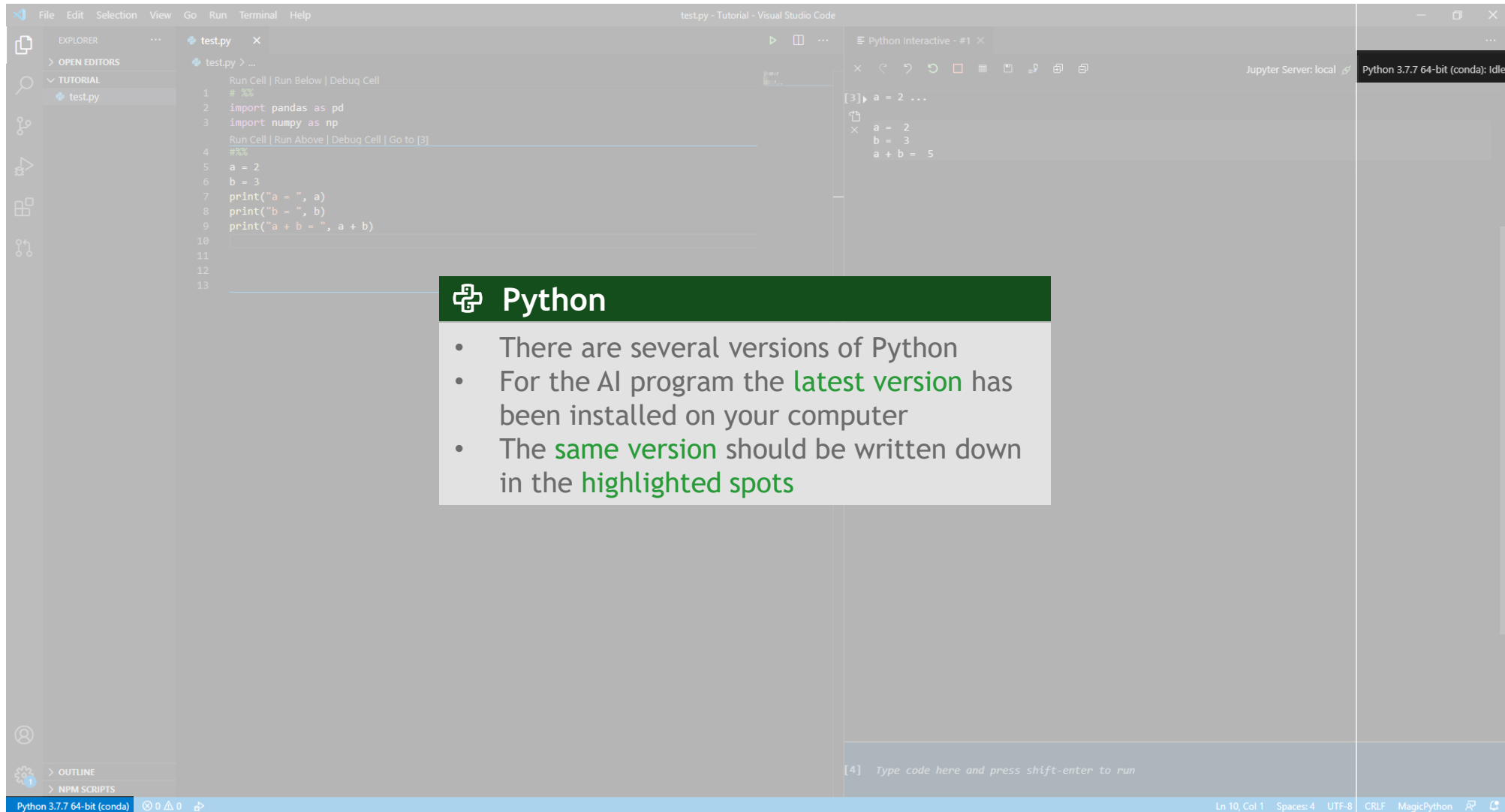
- A debugger is a program aimed at **testing and debugging** other programs
- This tool **highlights** the problematic lines of code and when an error occurs & hints towards a potential **solution**

```
test.py
1 # %%
2 import pandas as pd
3 import numpy as np
```

```
Python Interactive - #1
[3] a = 2 ...
a = 2
b = 3
a + b = 5
```

Ln 10, Col 1 Spaces: 4 UTF-8 CRLF MagicPython

This hackathon will focus on python programming language



```
1 # %%
2 import pandas as pd
3 import numpy as np
4
5 a = 2
6 b = 3
7 print("a = ", a)
8 print("b = ", b)
9 print("a + b = ", a + b)
```

Python

- There are several versions of Python
- For the AI program the **latest version** has been installed on your computer
- The **same version** should be written down in the **highlighted spots**

AGENDA

I



Why using VSC ?

II



Why using shared code ?

III



What is GitHub and how to use it ?

IV

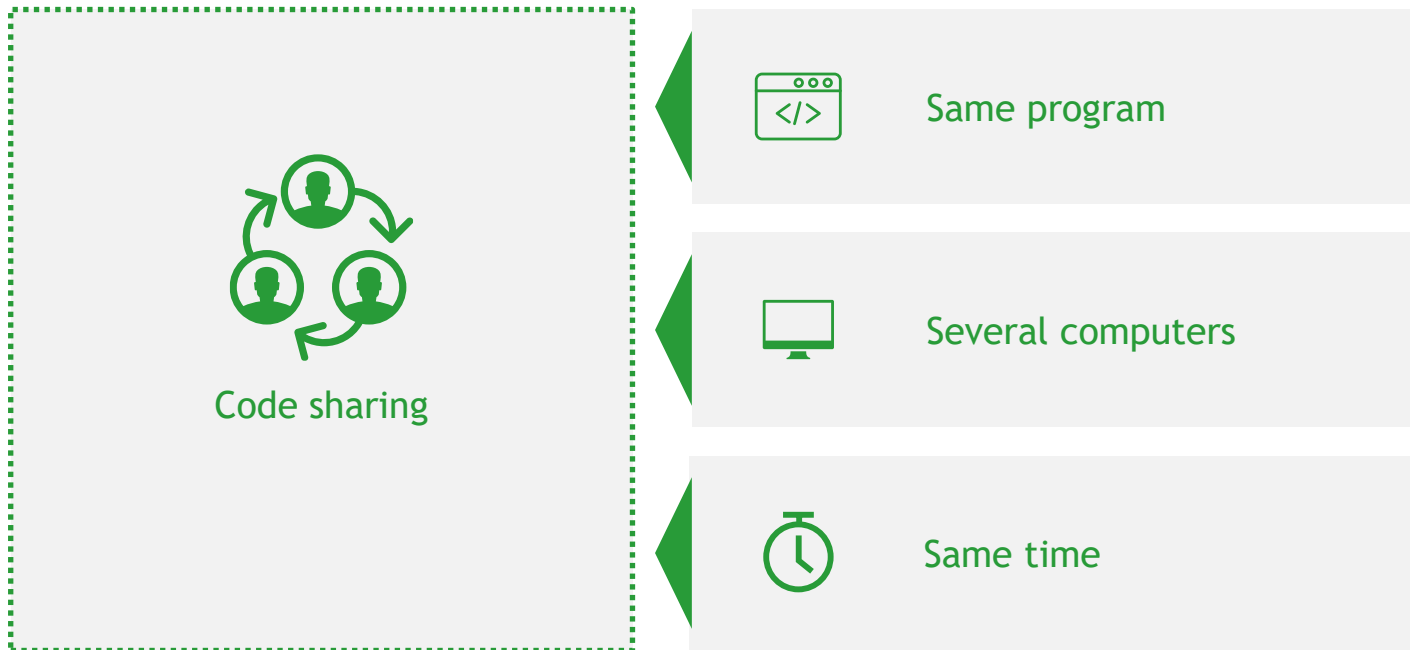


How to use GitHub with VSC?

Code sharing enables to **work with others** on the same code



Code sharing has three main characteristics



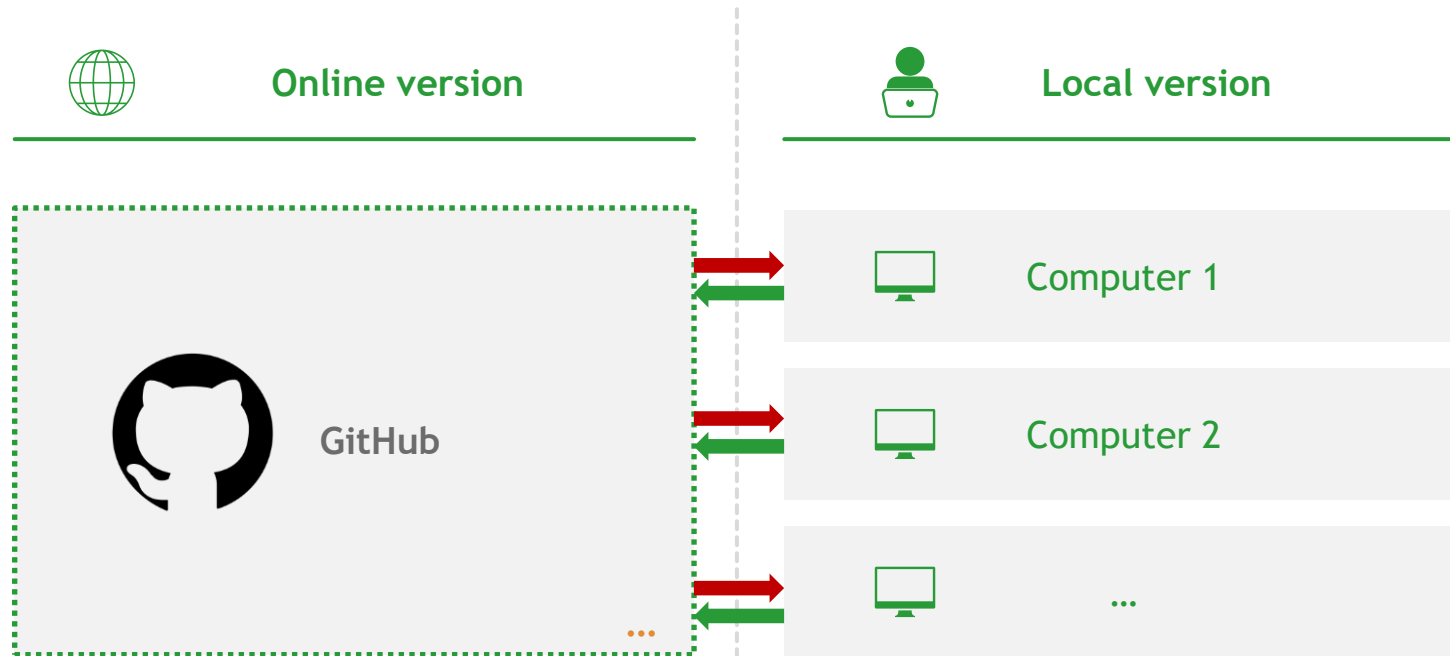
Key learnings

- Code sharing is like a **google docs** but for programmers
- Code sharing gives a **team of developers** the opportunity to work on different aspects of the same code at the same time

Code sharing manages interactions between Online & local versions



Architecture of a shared code network



Key learnings

- The action of **downloading** the online version on laptop is called a “pull”
- The action of **uploading** the local version is called a “push”

Diagram Legend







Pull



Push

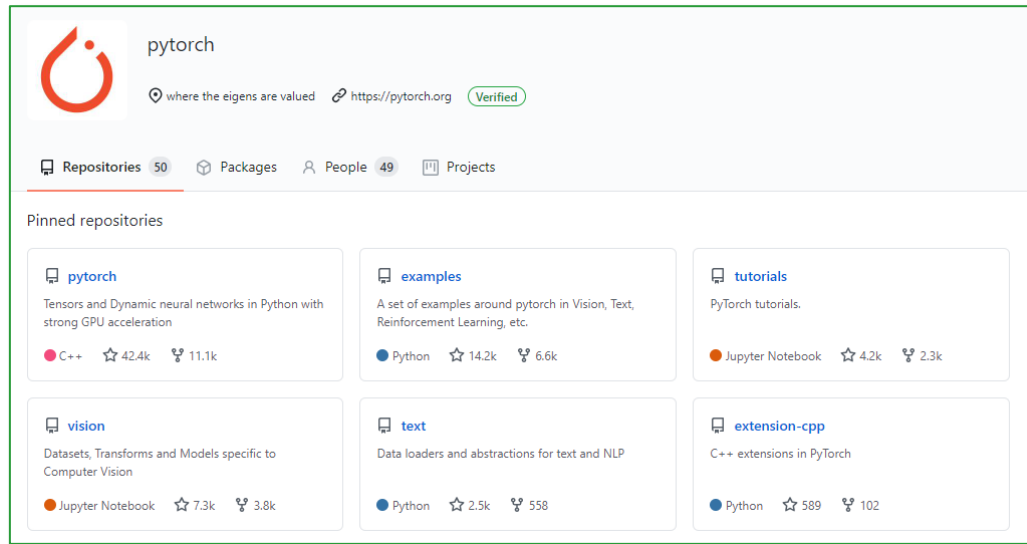
AGENDA

- I  Why using VSC ?
- II  Why using shared code ?
- III  What is GitHub and how to use it ?
- IV  How to use GitHub with VSC?

Github is a website that **hosts** a team's or an individual's **shared projects**



A Github account is structured with repositories



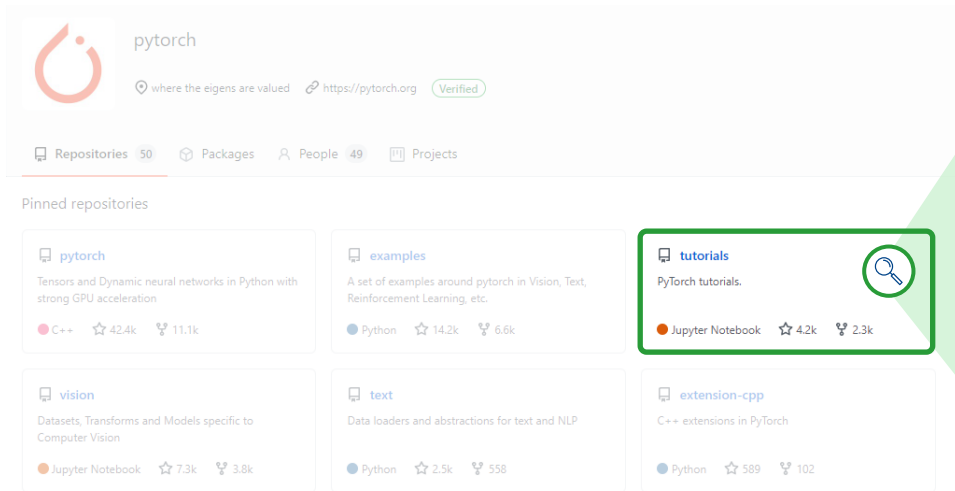
Structure of the Pytorch's GitHub

- This picture is taken from PyTorch's account, a famous deep learning library
- A repository is a Git word to design a project
- 6 repositories can be seen in this image:
 - Pytorch
 - Examples
 - Tutorials
 - Vision
 - Text
 - Extension-cpp

Github is a website that **hosts** a team's or an individual's **shared projects**



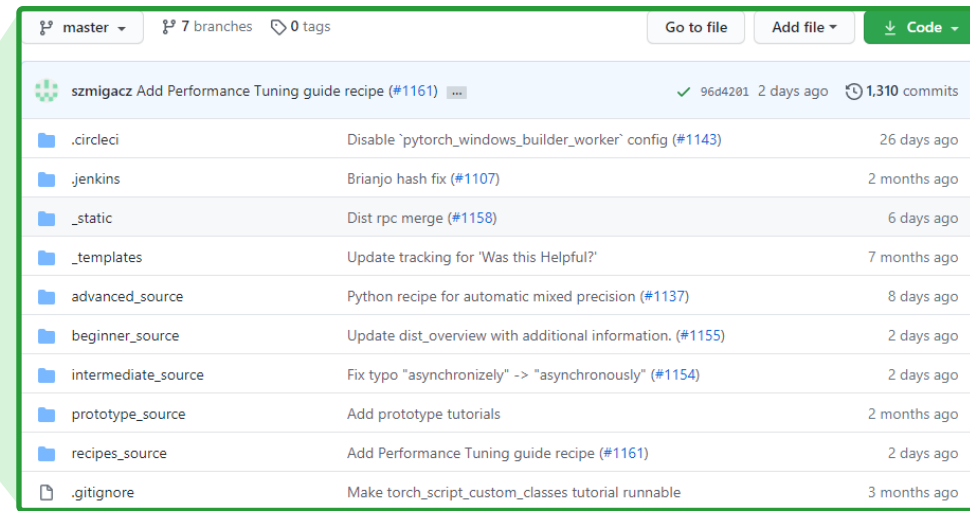
A Github account is structured with repositories



Structure of the Pytorch's Github



A repository is structured with folders



Structure of the tutorials' repository with several folders



Key learnings

- A Github account is structured with **repositories**
- A repository contains different **folders** and files
- A repository can be **remote** (online) or **local**

Tutorial: Create a GitHub account



Steps to create an account

1. Go to Github website:
“https://github.com/”
2. Click the “Sign up” button
3. Choose a username
4. Fill your email address
5. Choose a password



Illustration of steps 3 to 5

Join GitHub

Create your account

Username *

3

Email address *

4

Password *

5

Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter.

[Learn more.](#)

Tutorial: Create a GitHub repository



Steps to create a repository

1. Click “Repositories”
2. Click the “New” button
3. Enter the name of the repository (see next image)
4. Click “Add a README file”
5. Click “Create repository”



Illustration of steps 3 to 5

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Repository template
Start your repository with a template repository's contents.

Owner * greg-lep **Repository name *** first-repository ✓

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

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.](#)

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

This will set `master` as the default branch. Change the default name in your [settings](#).

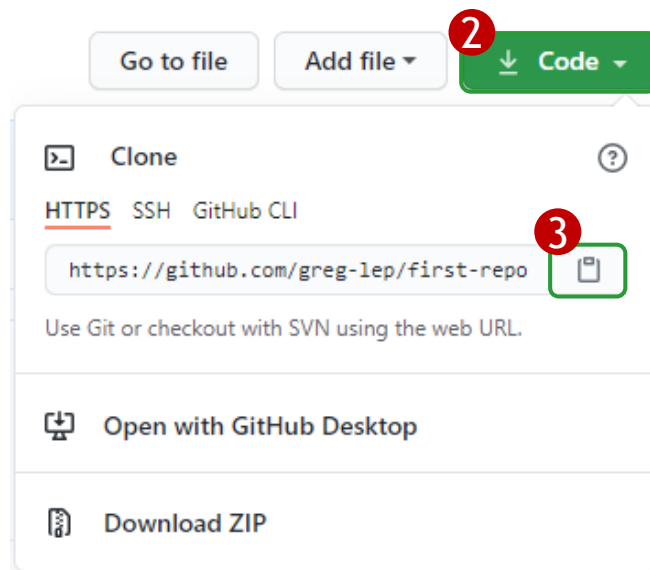
AGENDA

- I  Why using VSC ?
- II  Why using shared code ?
- III  What is GitHub and how to use it ?
- IV  How to use GitHub with VSC?

Tutorial: Create a local version of this repository

Copy the link of the repository

1. Open the repository “first-repository”
2. Click the “Code” button
3. Click the diskette button to copy the HTTPS link of the repository



Clone the repository

4. Open VSC and open a new Terminal
 - › Click the “Terminal” button (top of VSC’s screen)
 - › Click the “New Terminal”
5. Create a “Github” folder in your “Documents” folder and then navigate through it



```
cd Documents  
mkdir Github  
cd Github
```

6. Write on the prompt command “git clone” followed by the HTTPS link of the repository
 - › The new folder created in the Github folder is the clone of “first-repository” (Github repository)

Tutorial: Synchronize the local repository with VSC & and Push changes



Copy the link of the repository

1. Open VSC
2. Open in VSC the folder “first-repository”
3. Create a python file named “test.py”
4. Write the following code line and save it

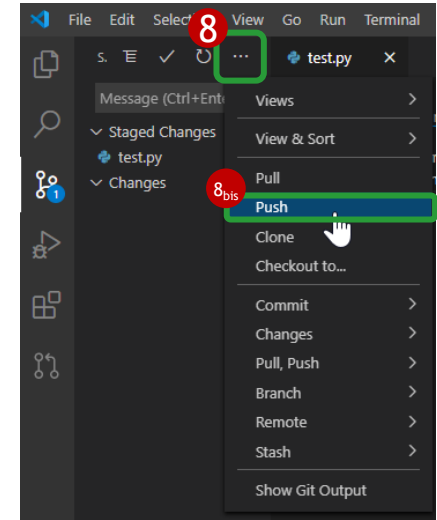
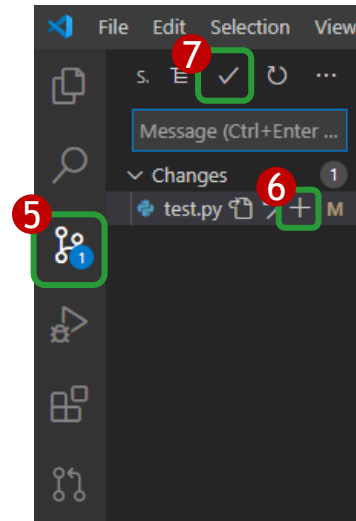


```
import numpy as np  
a = np.linspace(1, 10, 50)  
print(a)
```

5. Click the third icon where a “1” appeared
6. Stage “test.py”
7. Commit the stage and write: “Add first Python file”
8. Push the commit
9. Go on Github website to watch the adding of the Python file



Illustration of steps 5 to 8



Key learnings

- A **stage** moves modified files from the working directory to a **staging area** ready to be committed
- A **commit** moves staging area's files to the **local git directory**

Tutorial: Pull changes from the remote repository

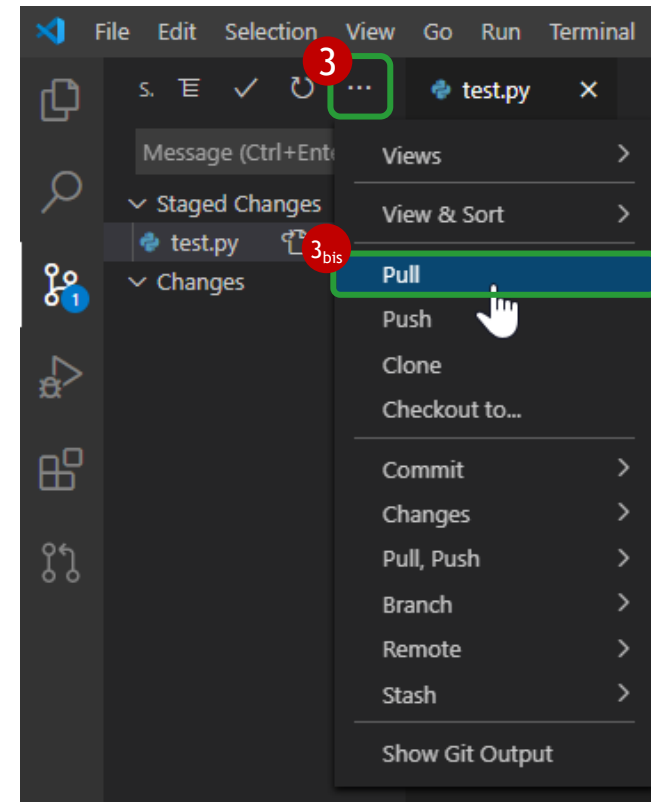


Copy the link of the repository

1. Go to GitHub website on the repository “first-repository”
2. Add a new python file named “test_pull.py”
 - Click “Add file”
 - Click “Create new file”
 - Fill the file name: “test_pull.py”
 - Click “Commit new file” at the bottom of the web page
3. Pull locally the new python file
4. Check on VSC the presence of the file “test_pull.py”



Illustration of step 3



Branches are useful for **solution versioning** as well as new ideas testing **without affecting the whole program**

- A **branch** is a way of **isolating** some development work without affecting other branches of the repository
- The **default branch** is called “**master**”
- Once you are satisfied with the changes in your branch you can **merge** a **branch** with another

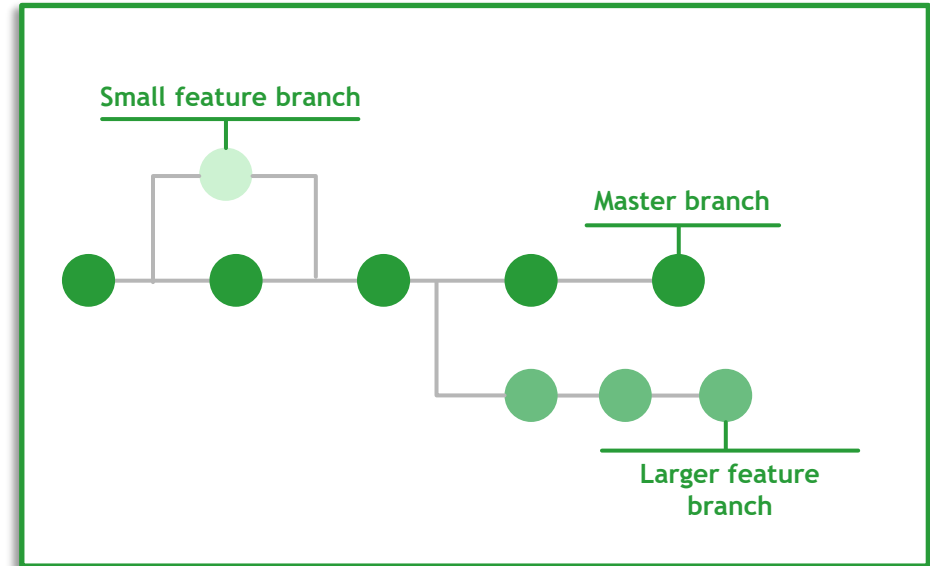


Illustration of how branches work



Key learnings

- GitHub can only get you so far, you need to **learn how to use it efficiently**
- **Commits** must be **frequent and well documented**
- **Communication** remains key to collaborating on a coding project

Tutorial: Create a new branch and merge it with the master branch



Steps to pull a repository

1. Click the branch sign
2. Choose “Create a new branch”
3. Call this new branch “test”
4. Check that the current branch is the “test” branch
5. Modify the python file “test.py”
6. Merge the branch “test” with the “master” branch
 - Click “...” > “branch” > “Merge Branch”
 - Choose the branch “test”
7. Check that the file “test.py” on the master branch has been modified



Illustration of steps 1 to 6

