SSE
Driven By Creativity & Innovation

TU WIEN
TECHNISCHE UNIVERSITÄT WIEN
Vienna | Austria

CSE & INSO Collaboration

Meeting 1 - An introduction to the world of git

Goals

- Collaborate together
- Strengthen the partnership between CSE & INSO
- Learn from each other
- Create easy to use Java Lab Code

Program overview

- Tool check
- Intro to git
- git vocabulary
- Live Demo together
- Advanced git
- Live Demo together

Tools

- Zoom
- GitHub Desktop and GitHub Account
- IntelliJ Community
- TeamViewer

git - /gɪt/

What does Wikipedia say?

- Git is a software for **tracking changes** in any set of files, usually used for **coordinating work among programmers** collaboratively developing source code during software development.
- Its goals include speed, data integrity, and support for distributed, non-linear workflows (thousands of parallel branches running on different systems).

Who uses git?

and many more…

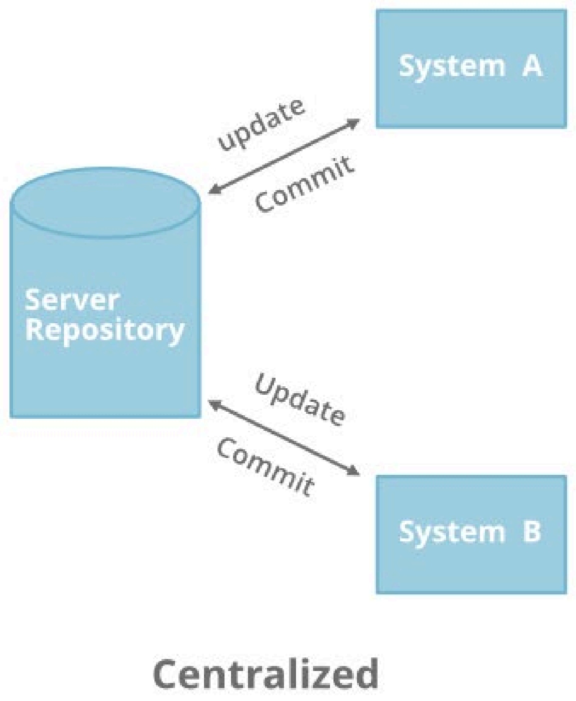
History

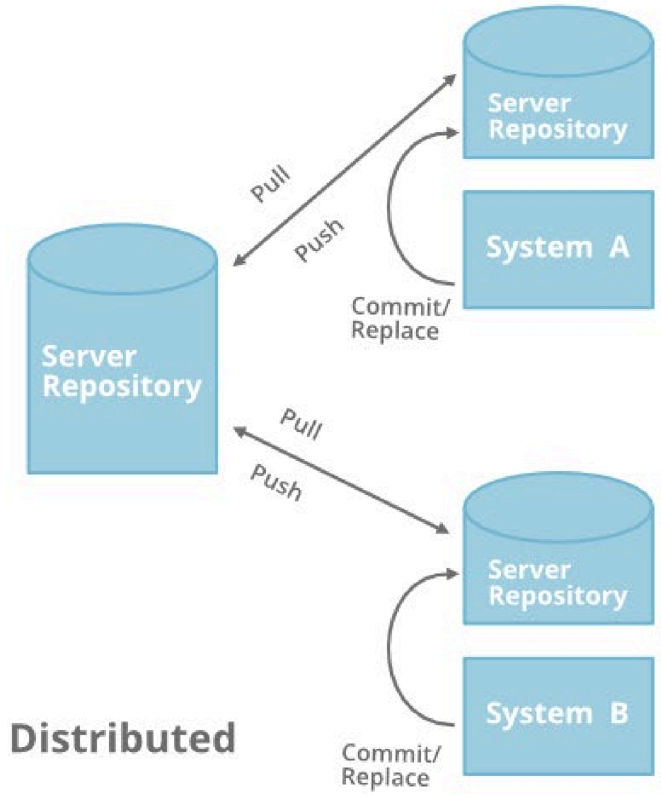
- Git was originally authored by Linus Torvalds in 2005 for development of the Linux kernel, with other kernel developers contributing to its initial development.
- Distributed version control system
- https://git-scm.com/

Distributed Version Control System

- is a form of version control in which the complete codebase, including its full history, is mirrored on every developer's computer.
- Compared to centralized version control
  - automatic management branching and merging
  - speeds up most operations
  - ability to work offline
  - does not rely on a single location for backups

Centralized vs Distributed





What does git do?

- Tracks code changes
- Tracks who made changes
- Enables coding collaboration

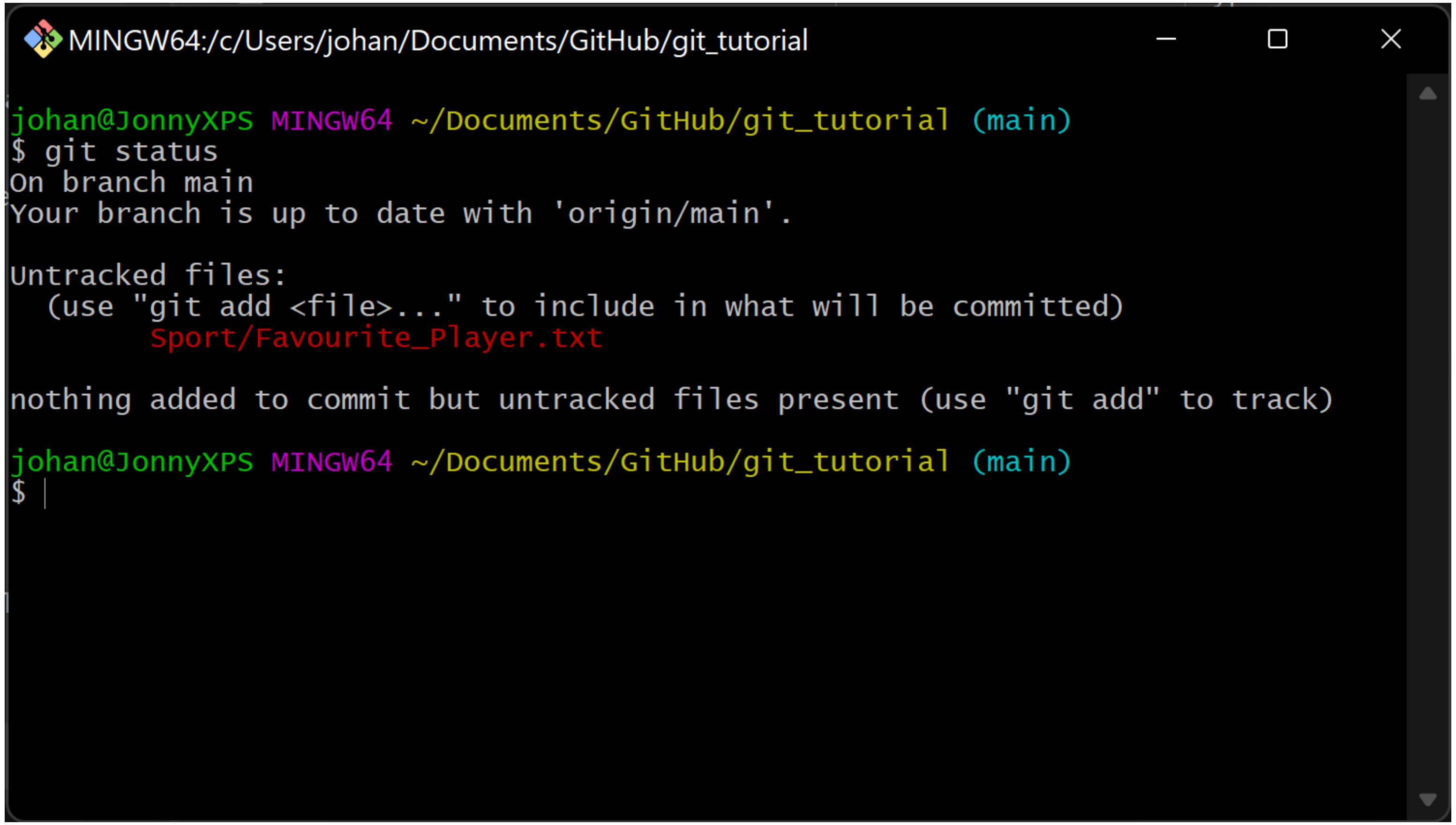
Why git?

- Developers can work together from anywhere in the world.
- Developers can see the full history of the project.
- Developers can revert to earlier versions of a project.

GitHub

How to use git?

- Terminal

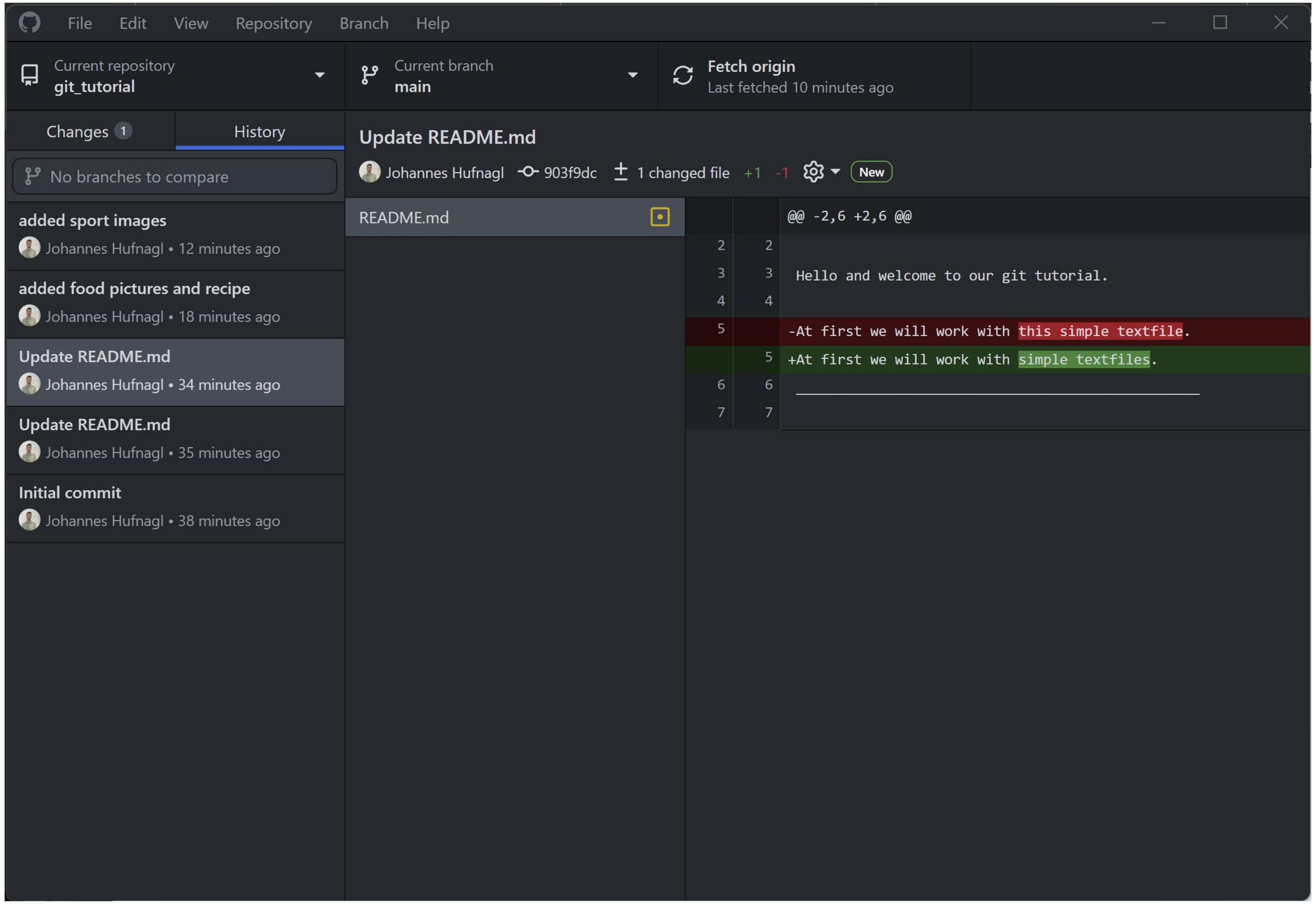


MINGW64:/c/Users/johan/Documents/GitHub/git_tutorial

```
johan@JonnyXPS MINGW64 ~/Documents/GitHub/git_tutorial (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Sport/Favourite_Player.txt

nothing added to commit but untracked files present (use "git add" to track)

johan@JonnyXPS MINGW64 ~/Documents/GitHub/git_tutorial (main)
$ 
```

How to use git?

- GitHub Desktop



added sport images
Johannes Hufnagl • 12 minutes ago

added food pictures and recipe
Johannes Hufnagl • 18 minutes ago

Update README.md
Johannes Hufnagl • 34 minutes ago

Update README.md
Johannes Hufnagl • 35 minutes ago

Initial commit
Johannes Hufnagl • 38 minutes ago

Update README.md

Johannes Hufnagl 903f9dc 1 changed file +1 -1 New

README.md

```
@@ -2,6 +2,6 @@
2 2
3 3   Hello and welcome to our git tutorial.
4 4
5    -At first we will work with this simple textfile.
  5  +At first we will work with simple textfiles.
6 6
7 7
```

git vocabulary

- Repository
- Clone
- Add
- Commit
- Push

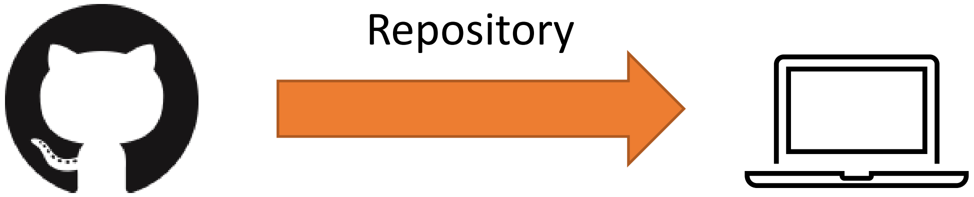
Repository

- Rough explanation: contains all your data (source code files) and their versions (history)
- Remote repository -> on the server
- Local repository -> on the client

git clone

- Clones (creates a copy) a repository from the server to the client (laptop, desktop computer, smartphone, …)

```
git clone https://github.com/RIT-at-SSE/git_tutorial
```



git add

- Makes (changed/added/removed) files ready to commit

Add all files of the current directory

```
git add .
```

Add a specific file

```
git add Main.java
```

git commit

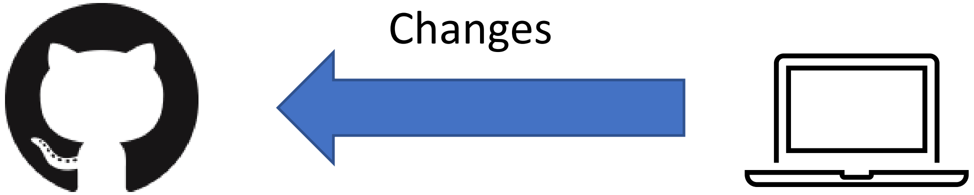
- Makes (changed/added/removed) files ready on the local repository to push to the remote repository.
- Adds a message to those files for explaining collaborators (your colleagues) what was changed/implemented/fixed

```
git commit –m “short and precise commit message”
```

git push

- Pushes (synchronizes) all changes of the current branch from the local repository to the remote repository

```
git push
```



Live Demo together

Let’s try it out

Tutorial 1

- Add your own file to the repository

1. Clone the repository
2. Create a new file locally (“YourName.txt”), in your local repository
3. Add some information about yourself
   (Name, Birthdate, Hobbies, Favorite food, Link of a trailer of your favorite movie)
4. Save the file
5. Add the file to the git stage
6. Commit your file
7. Push your changes to the server

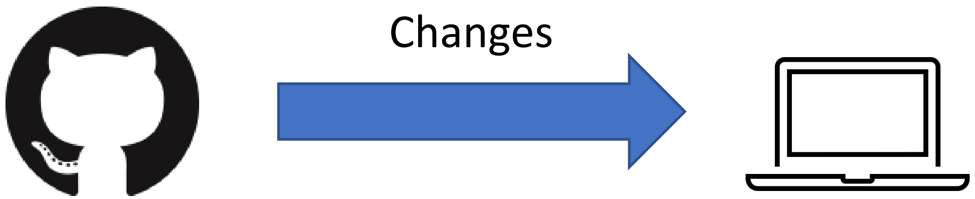
More git vocabulary

- Pull
- Status

git pull

- Pulls (synchronizes) all changes of the current branch from the remote repository to the local repository

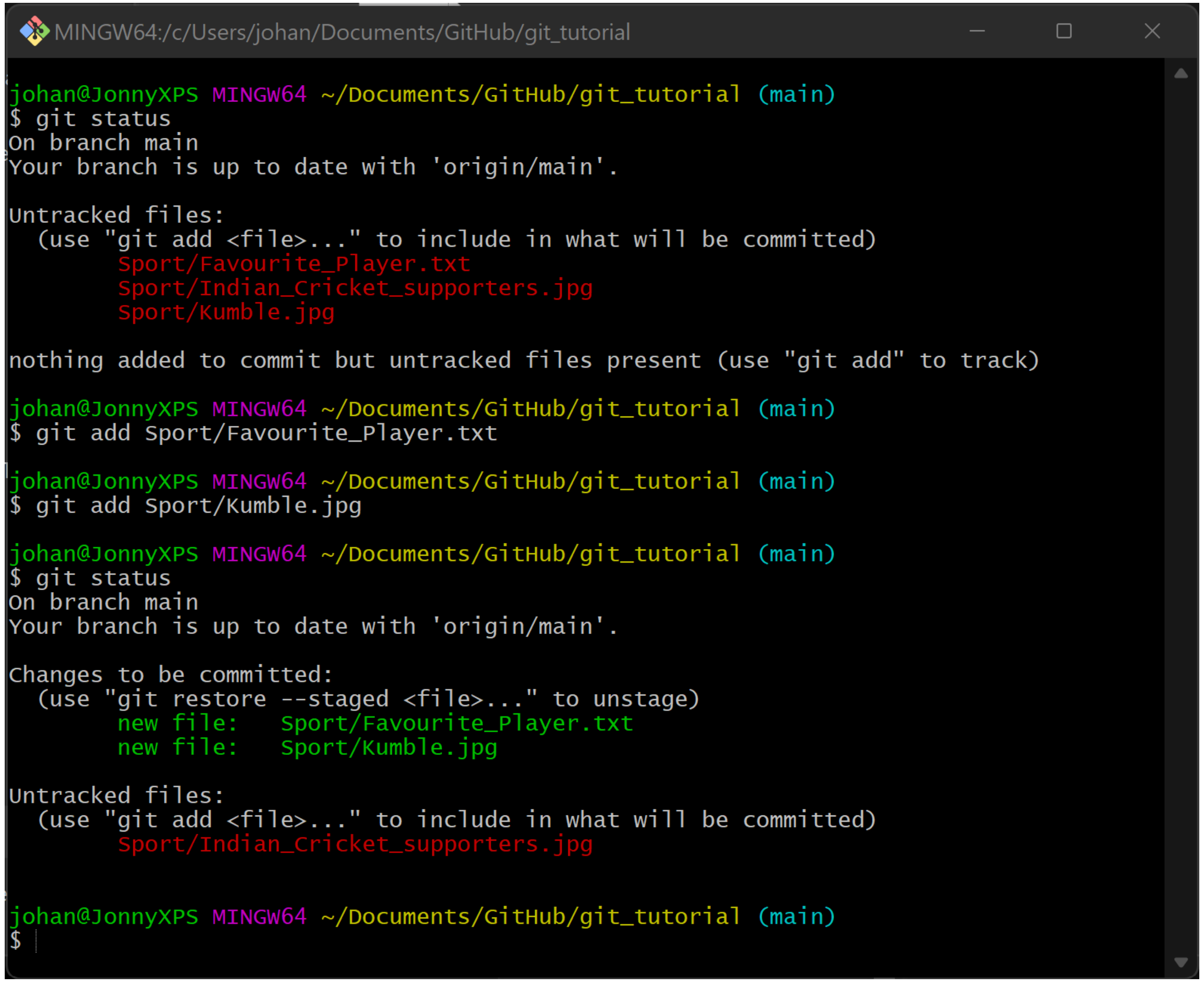
```
git pull
```



git status

- If you are not sure what's the current status -> check the status

```
git status
```



```
johan@JonnyXPS MINGW64 ~/Documents/GitHub/git_tutorial (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Sport/Favourite_Player.txt
        Sport/Indian_Cricket_supporters.jpg
        Sport/Kumble.jpg

nothing added to commit but untracked files present (use "git add" to track)

johan@JonnyXPS MINGW64 ~/Documents/GitHub/git_tutorial (main)
$ git add Sport/Favourite_Player.txt

johan@JonnyXPS MINGW64 ~/Documents/GitHub/git_tutorial (main)
$ git add Sport/Kumble.jpg

johan@JonnyXPS MINGW64 ~/Documents/GitHub/git_tutorial (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   Sport/Favourite_Player.txt
        new file:   Sport/Kumble.jpg

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Sport/Indian_Cricket_supporters.jpg


johan@JonnyXPS MINGW64 ~/Documents/GitHub/git_tutorial (main)
$
```

Live Demo together

Let’s get the files from each other

Tutorial 2

- Get the files from the other team members

1. Pull the changes from the server
2. Look through the repository and add a image or file wherever you want
3. (Look at the status)
4. Commit your new file
5. Push your changes to the server

How to use git for coding?

- Let us start using git for computer science

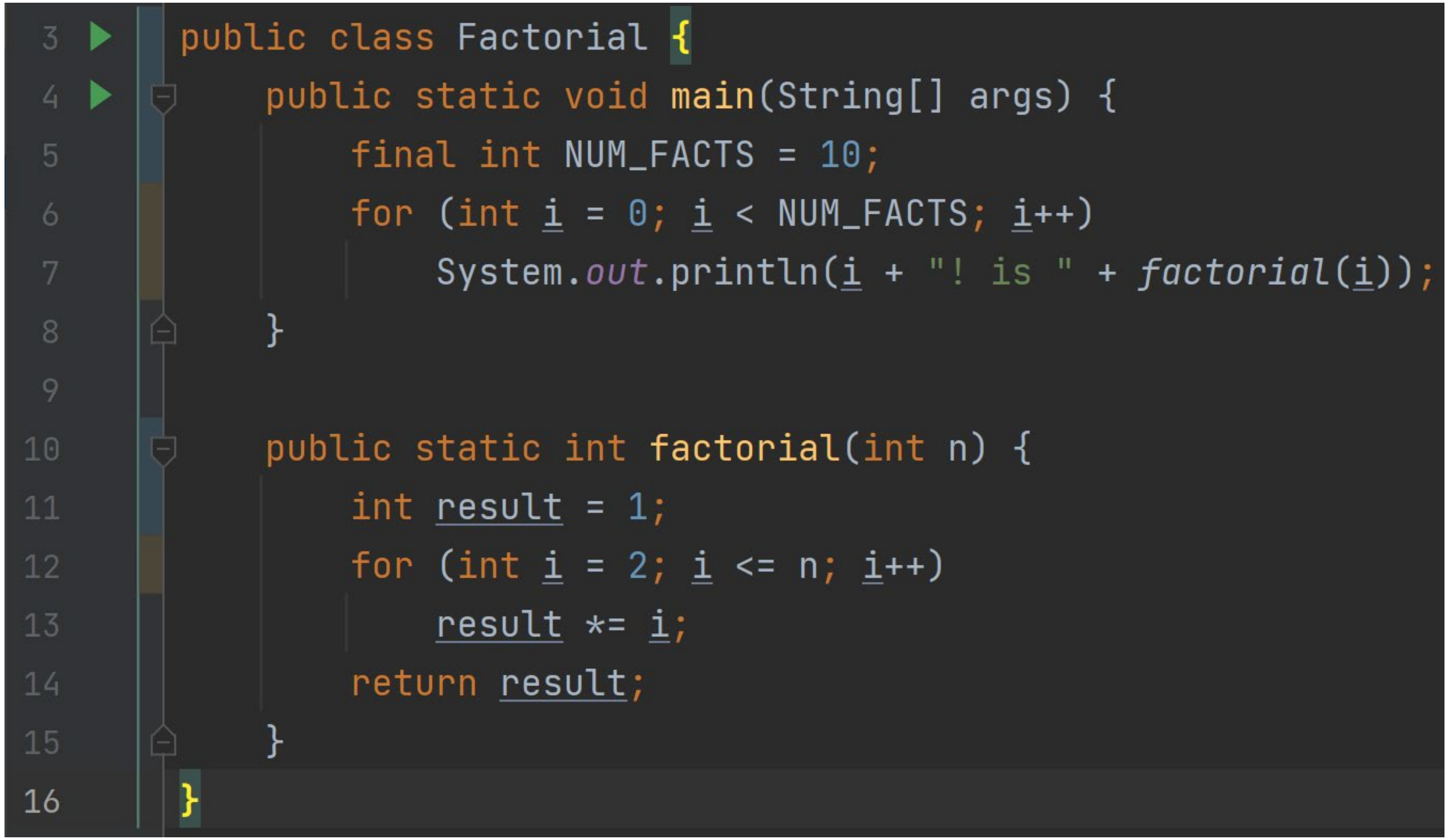
- A Repository can be opened with a editor of our choice, we will use:

  IntelliJ Community Edition

Why use IntelliJ as an editor?

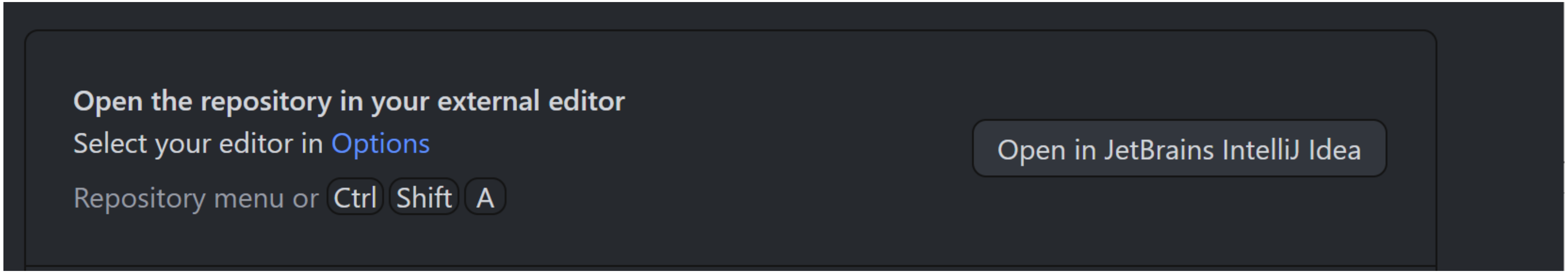
- Learning programming in theory is hard,
  learning programming in practice is even harder
- People who use an editor can learn faster and more efficient
- Using an editor enables run and see the outputs of code

Example: What does this code do?



```java
public class Factorial {
    public static void main(String[] args) {
        final int NUM_FACTS = 10;
        for (int i = 0; i < NUM_FACTS; i++)
            System.out.println(i + "! is " + factorial(i));
    }

    public static int factorial(int n) {
        int result = 1;
        for (int i = 2; i <= n; i++)
            result *= i;
        return result;
    }
}
```

Open Repository in IntelliJ



Open the repository in your external editor

Select your editor in Options

Repository menu or Ctrl Shift A

Takeaways

- git a version control system
- git vocabulary learned (Repository, clone, commit, push, pull)
- How to use GitHub Desktop
- IntelliJ Community

Next time

- Working together with git branches and merges
- Creation of a Java Lab Repository
- Discussion about structure

More information available here:

- https://www.w3schools.com/git/default.asp?remote=github
- https://docs.github.com/en/desktop
- YouTube is a great resource

Credits

- Raimund Rittnauer
- https://www.w3schools.com/git/default.asp?remote=github
- https://www.oracle.com/java/technologies/javase/jdk17-archive-downloads.html