BRO Academy

GIT

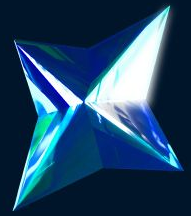
<h1>In this LESSON</h1>

- Remember previous lesson
- Why we need GIT?
- How to work together
- Interactive game (practice)

Let’s remember HTML

Link to Menti

BRO Academy



Working with Version Control Systems

HTML course: Lesson 4

Lesson Plan

1

How the development process works

2

What is a version control system and why is it used

3

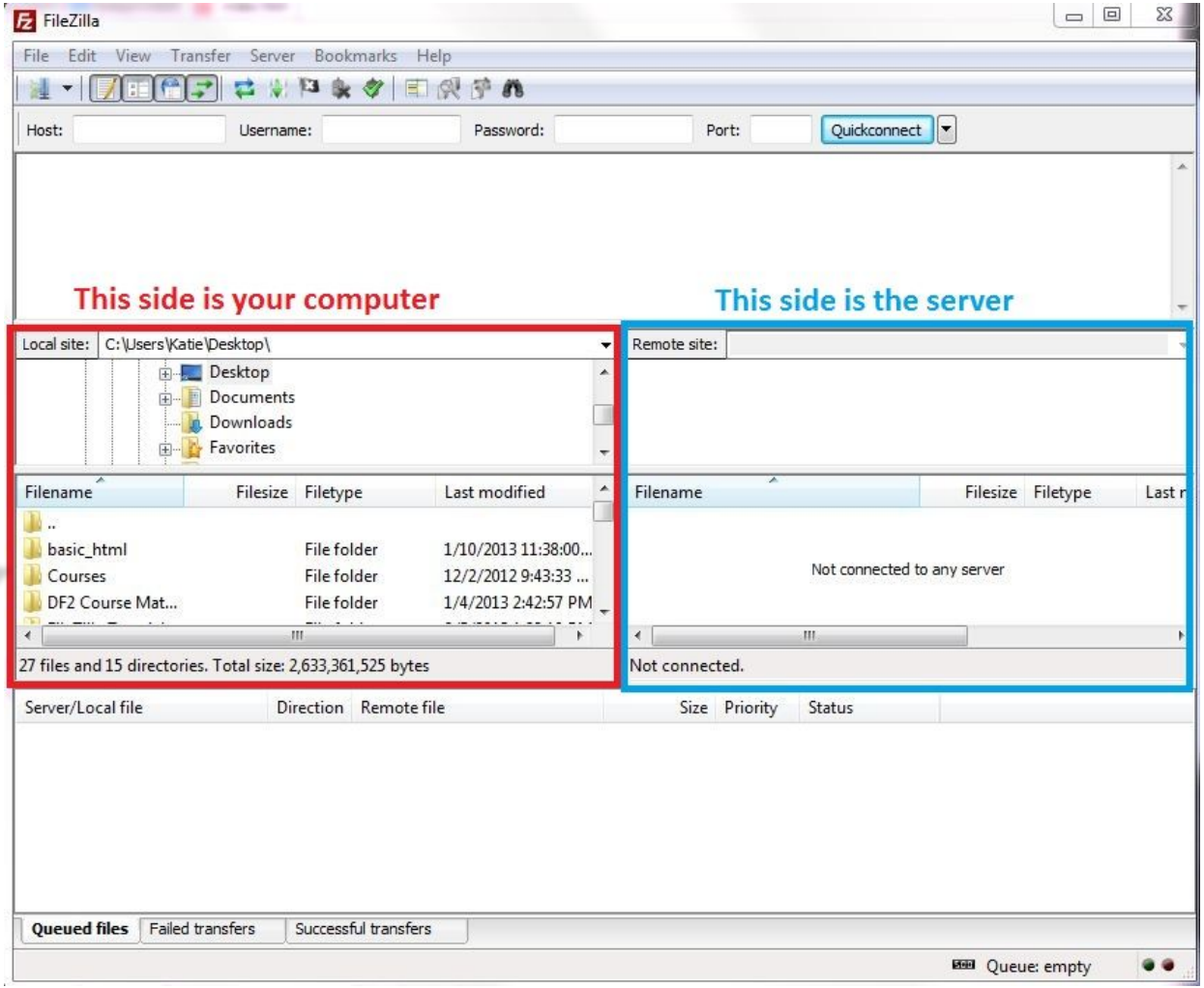
Tools for working with Git

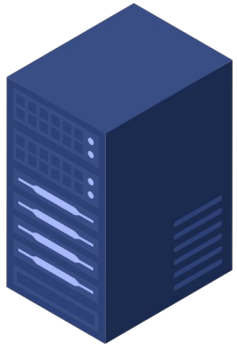
4

Basic commands for working


WHAT IF
WE DIDN’T
USE GIT

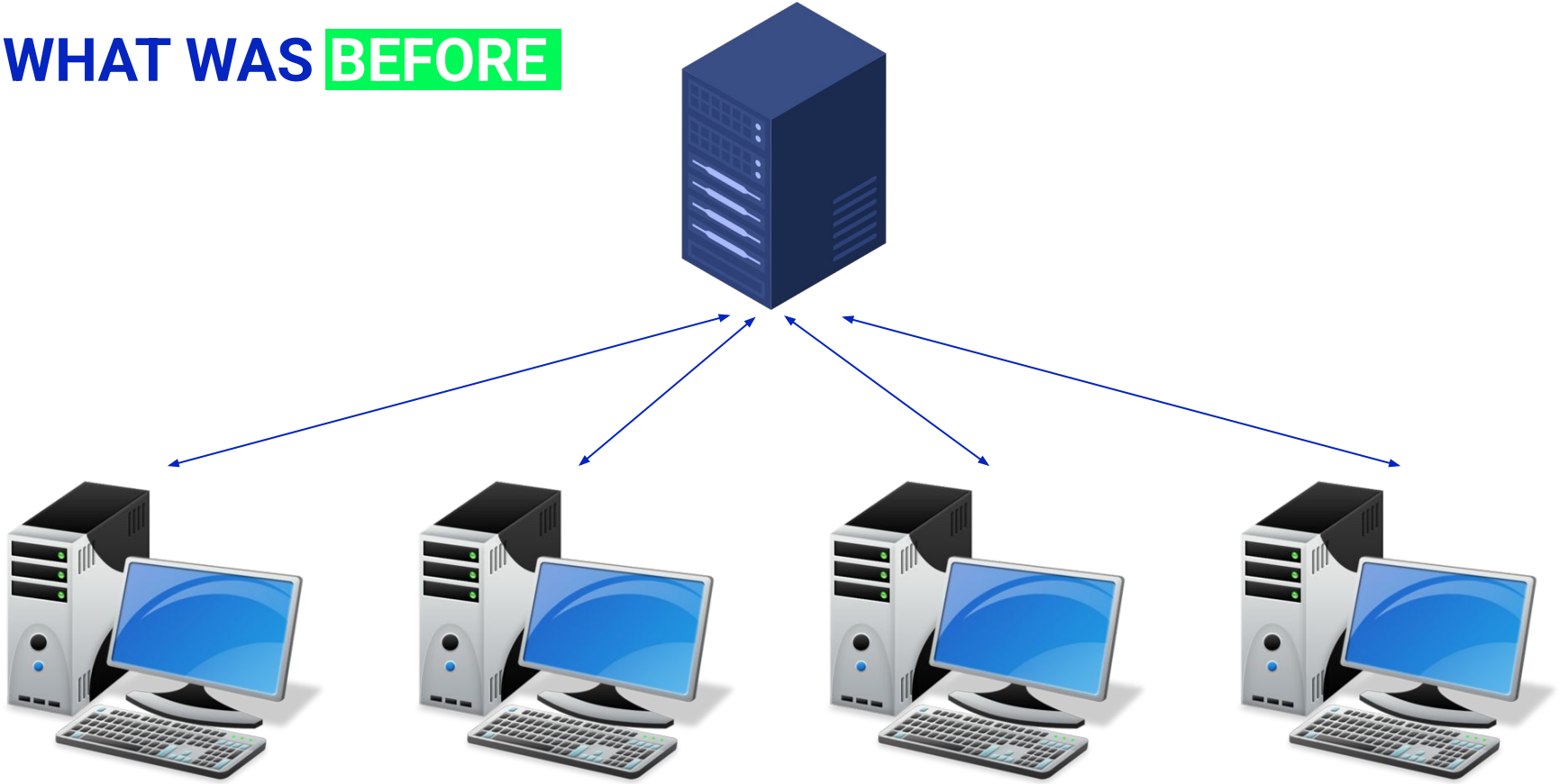
PC


FileZilla
File Edit View Transfer Server Bookmarks Help
Host: Username: Password: Port: Quickconnect
This side is your computer
This side is the server
Local site: C:\Users\Katie\Desktop\
Desktop
Documents
Downloads
Favorites
Filename Filesize Filetype Last modified
basic_html File folder 1/10/2013 11:38:00...
Courses File folder 12/2/2012 9:43:33 ...
DF2 Course Mat... File folder 1/4/2013 2:42:57 PM
27 files and 15 directories. Total size: 2,633,361,525 bytes
Remote site:
Filename Filesize Filetype
Not connected to any server
Not connected.
Server/Local file Direction Remote file Size Priority Status
Queued files Failed transfers Successful transfers
Queue: empty

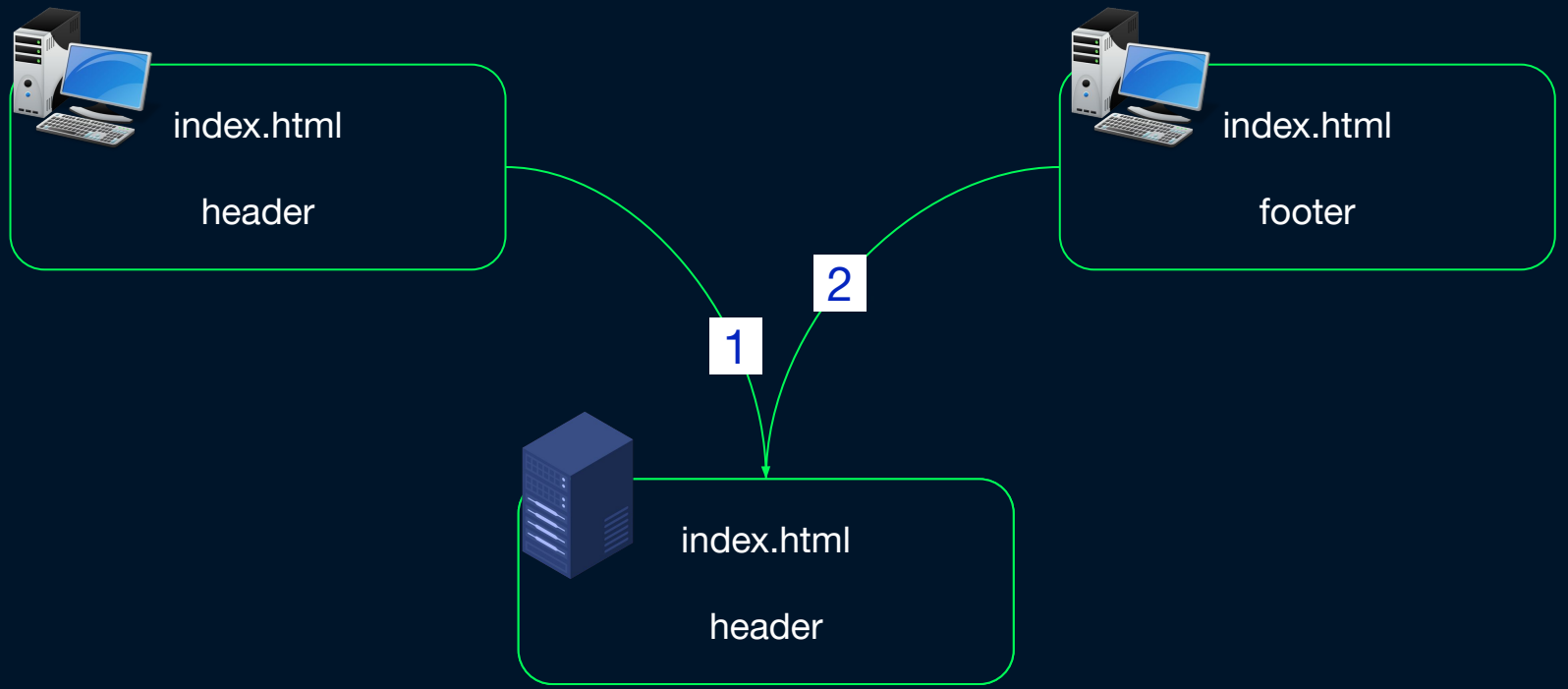


Server

WHAT WAS BEFORE

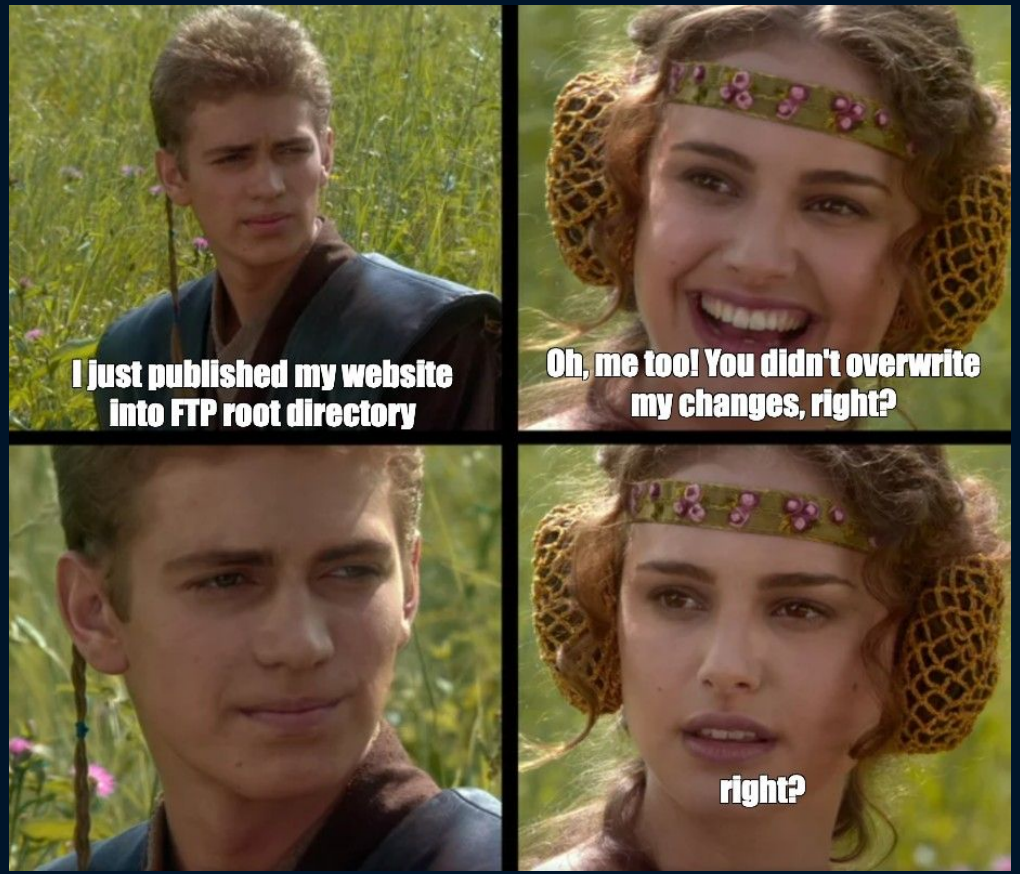


Overrides problem



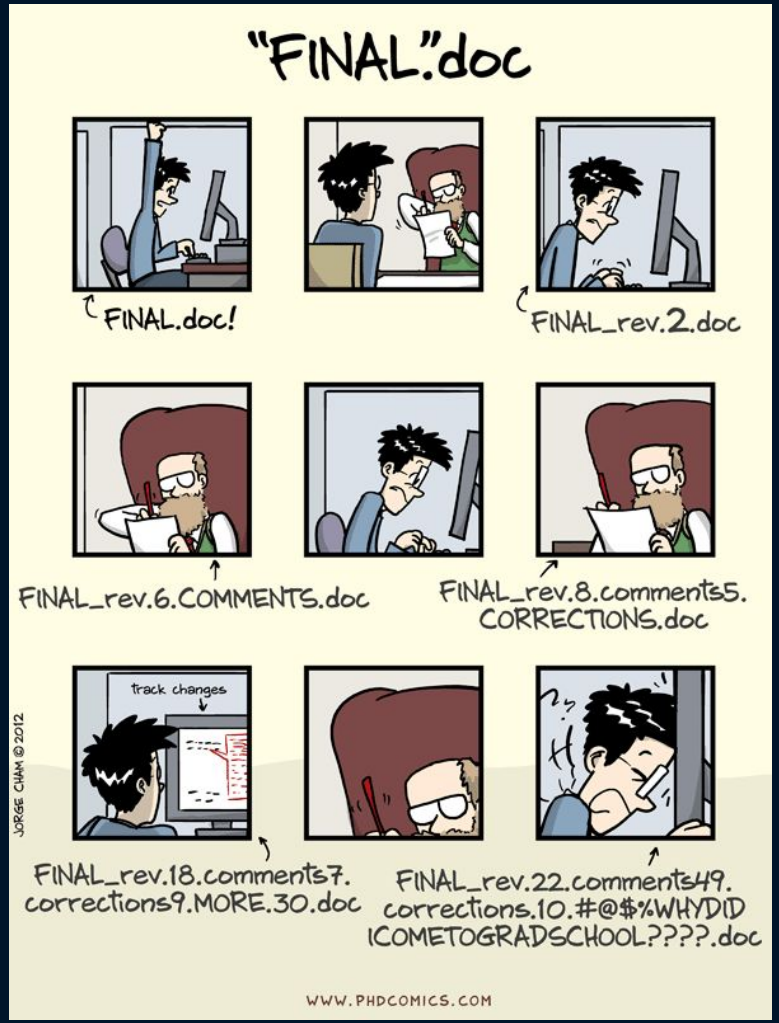
Overrides problem

How to work together?



History problem

How to rollback pre-pre-pre-previous version?

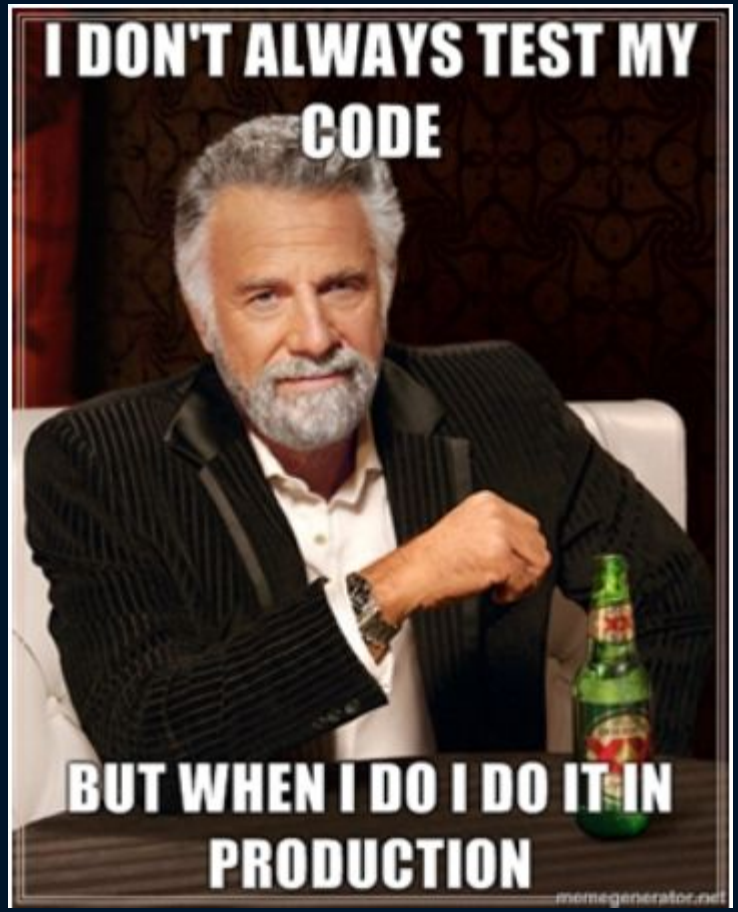


History problem

something about backups

Quality problem

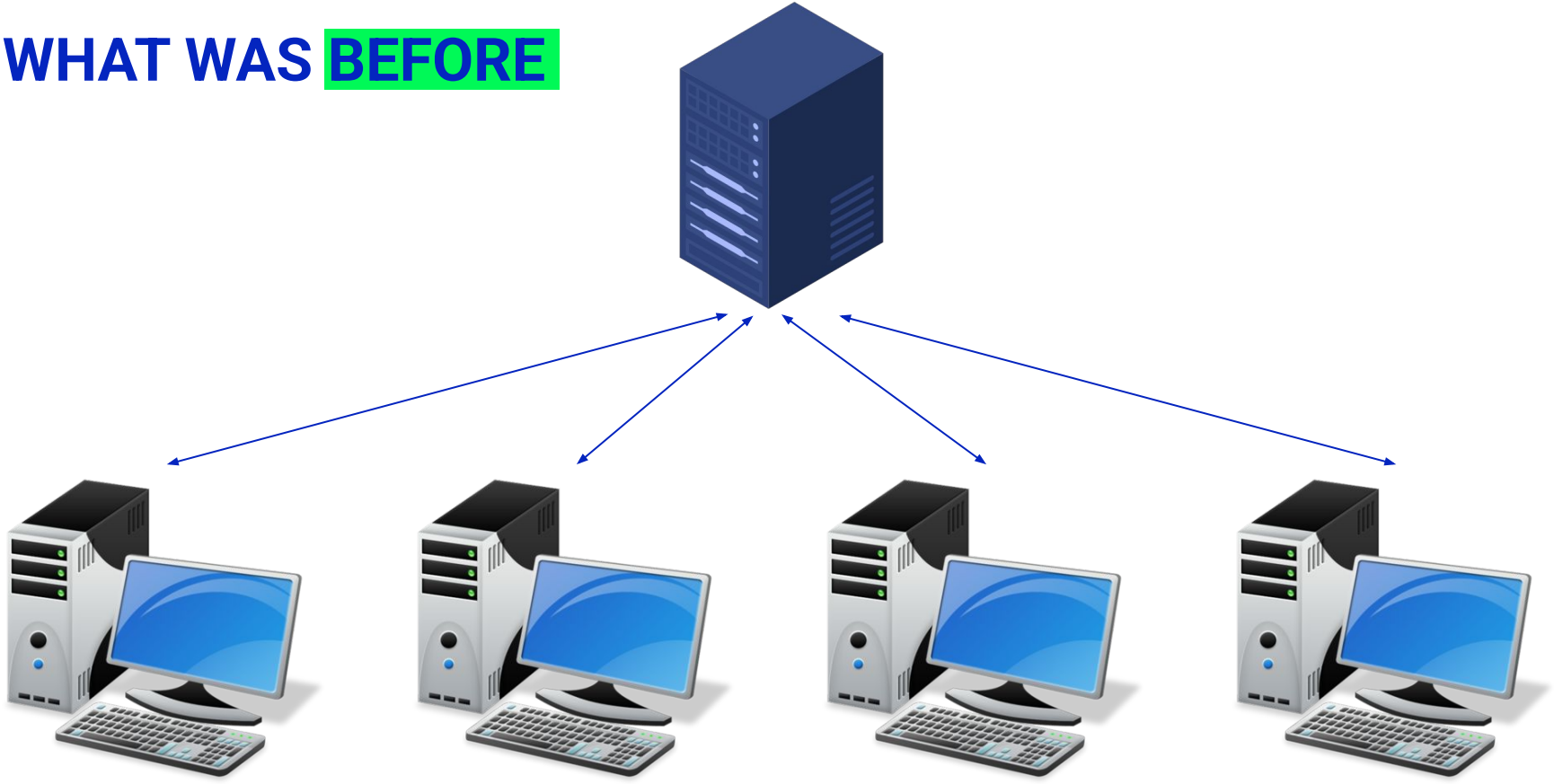
How to be sure in code and product quality?



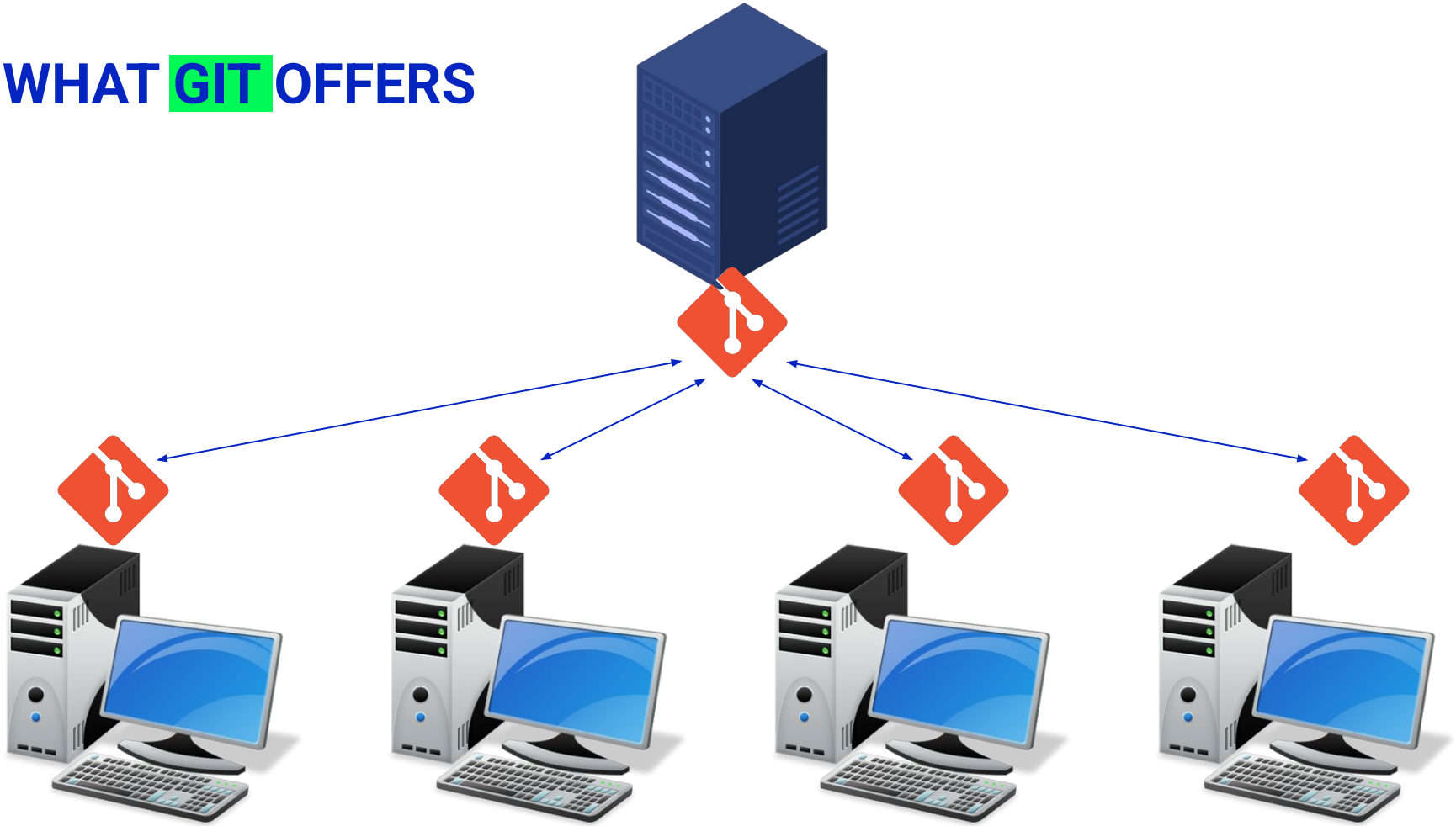
GIT

HOW IT SOLVES PROBLEMS

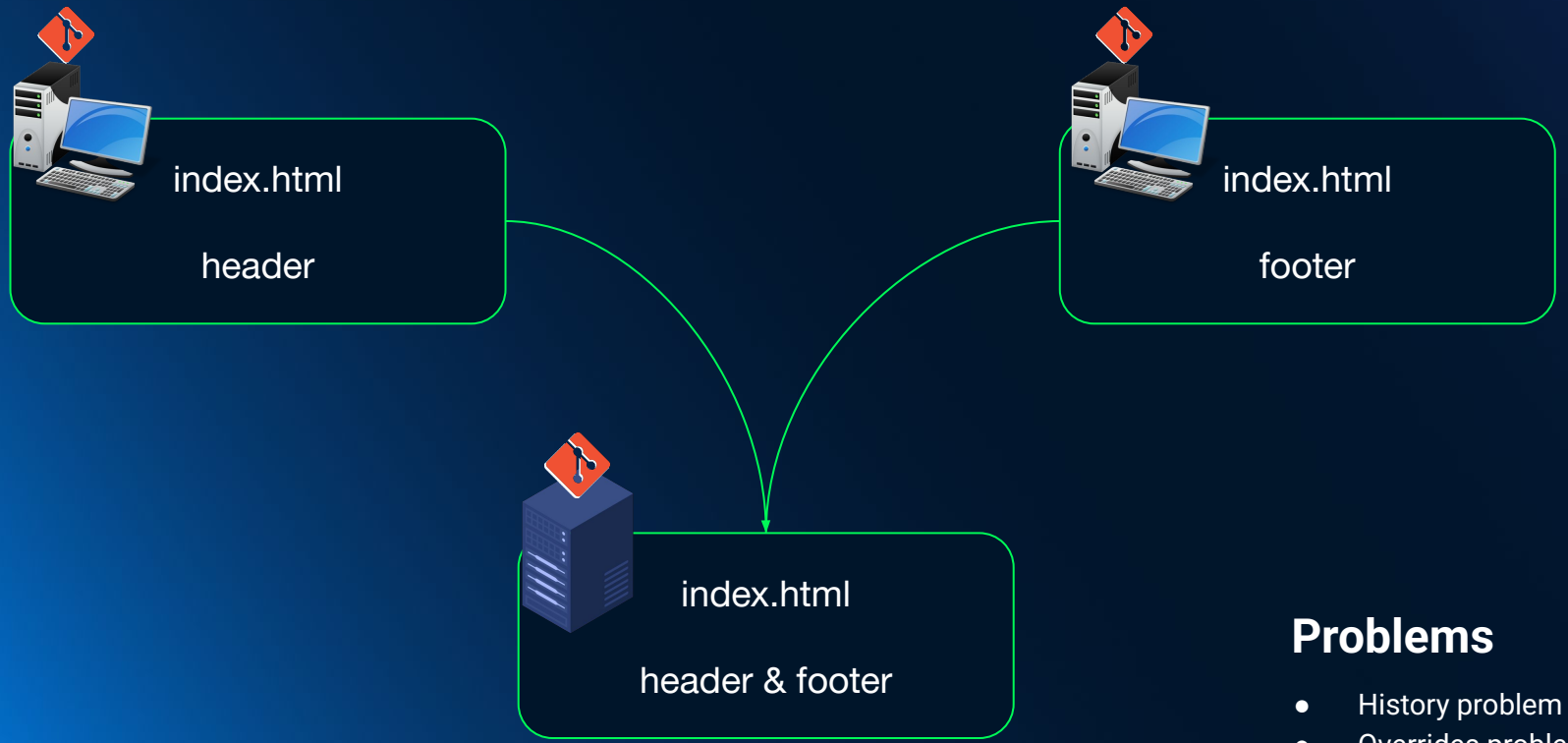
WHAT WAS BEFORE



WHAT GIT OFFERS

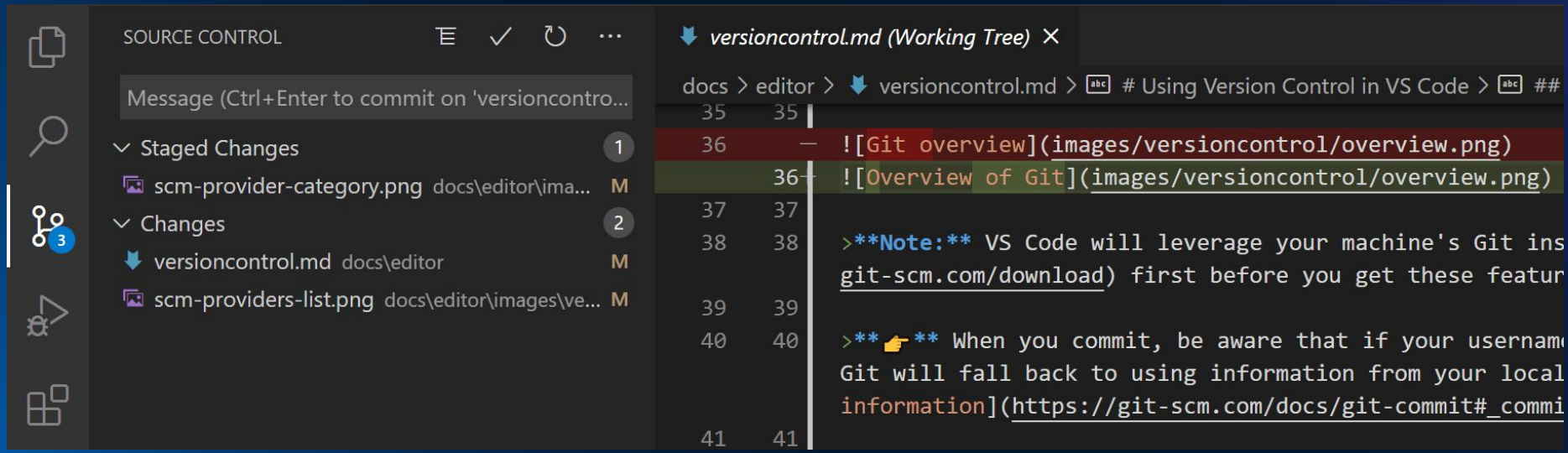


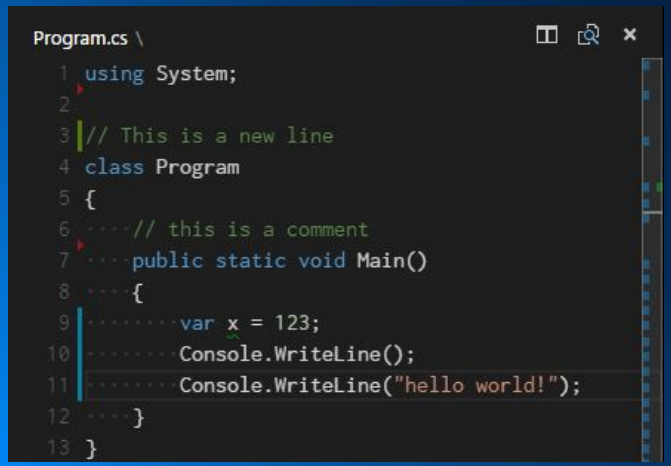
GIT IS ABOUT CHANGES



Problems

- History problem
- Overrides problem ✅
- Quality problem

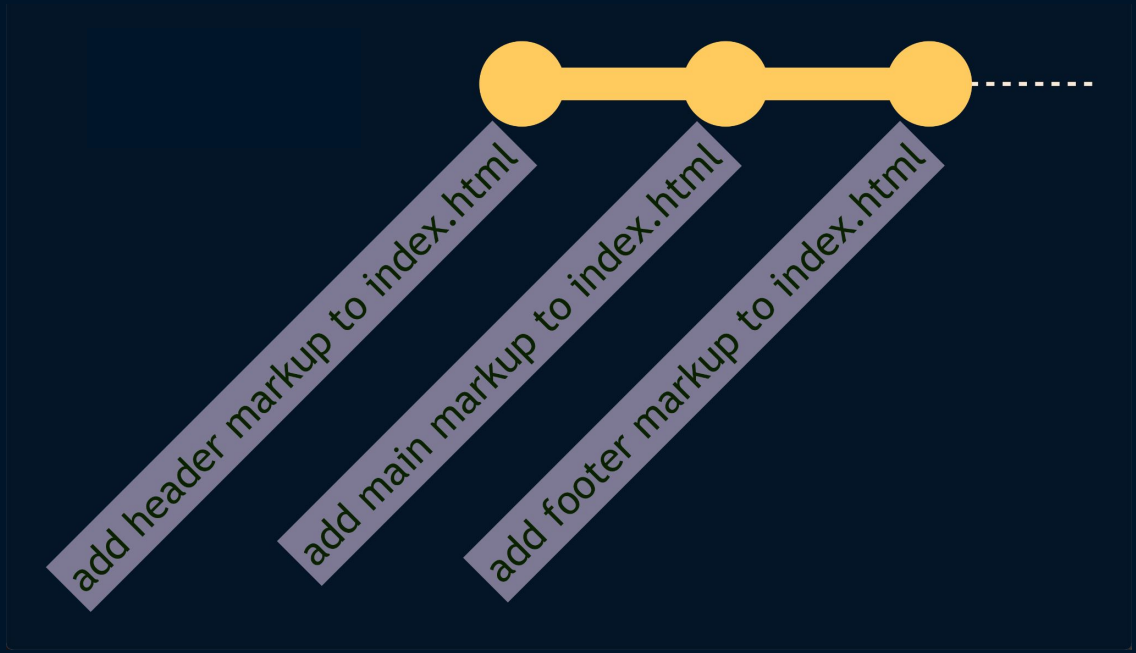

SOURCE CONTROL
Message (Ctrl+Enter to commit on 'versioncontro...
Staged Changes 1
scm-provider-category.png docs\editor\ima... M
Changes 2
versioncontrol.md docs\editor M
scm-providers-list.png docs\editor\images\ve... M
versioncontrol.md (Working Tree)
docs > editor > versioncontrol.md > # Using Version Control in VS Code > ##
35 35
36 ![Git overview](images/versioncontrol/overview.png)
36 ![Overview of Git](images/versioncontrol/overview.png)
37 37
38 38 >**Note:** VS Code will leverage your machine's Git ins
git-scm.com/download) first before you get these featur
39 39
40 40 >**👉** When you commit, be aware that if your usernam
Git will fall back to using information from your local
information](https://git-scm.com/docs/git-commit#_commi
41 41


Program.cs
using System;
// This is a new line
class Program
{
// this is a comment
public static void Main()
{
var x = 123;
Console.WriteLine();
Console.WriteLine("hello world!");
}
}

CHANGES SEPARATED PER COMMITS INTO HISTORY

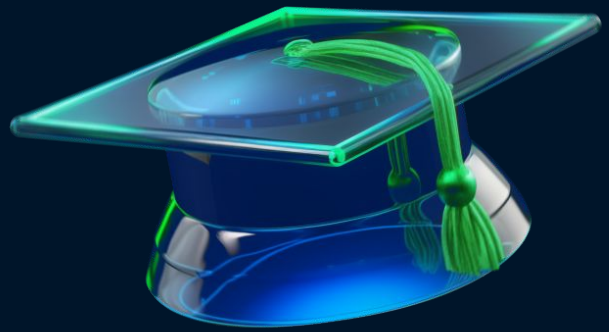
GIT HISTORY

Each change is stored as a commit
and together in a row they're
the git history

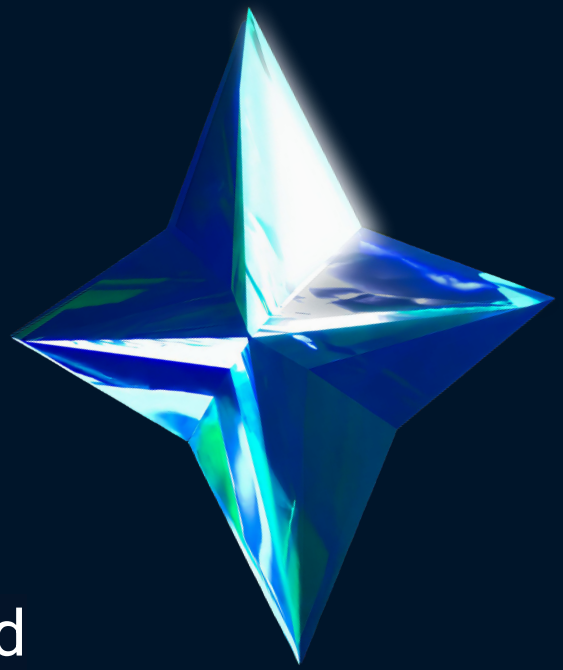


COMMIT RULES

- **Use a descriptive commit message**
  ✅ good: "fix validation in contact form", ❌ bad: "fix", "sdfsdf"
- **Use the imperative mood**
  ✅ good: "fix validation", ❌ bad: "fixed validation"
- **Make each commit a logical unit**
  don't commit more than described
- **Make atomic commits**
  logical units, the smallest possible size
- **Strive to use no more than 60 symbols**
  to be visible in GitHub UI



TAKE CARE OF TEAMMATES

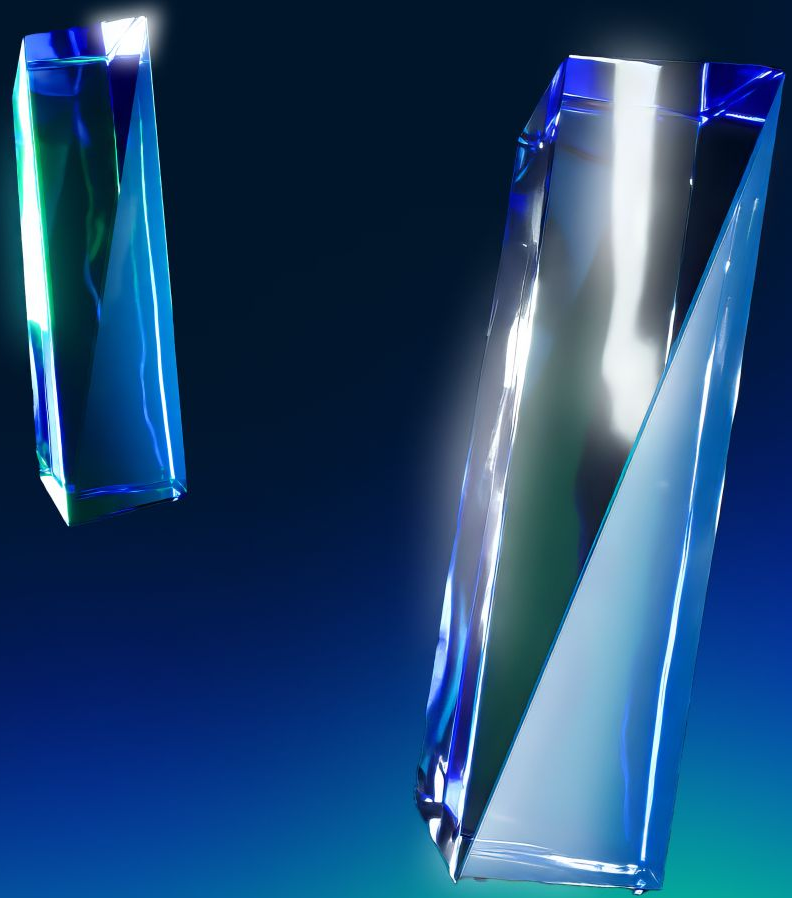


What or how to commit

- Pull before all new commits
- Coordinate with your co-workers
- Remember that the tools are line-based

SO LET'S USE GIT

BRO Academy



BREAK TIME

LET’S PLAY A GAME

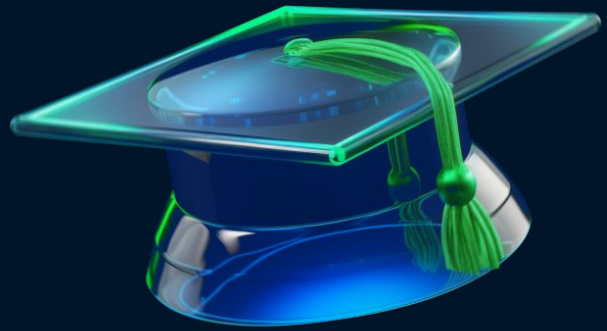
This is a funny kids **game**, maybe you know it.
Each player must write a response to a question using their imagination.
Encourage **creativity and humor** to make the final stories more entertaining.
Theme for this play session: **fairytale.**

1. **Fork** game repository and **clone** to your computer.
2. You have your own number. **Answer the question** with the same line number in `README.md` and **commit and push** the changes to your repository.
3. Create a **pull request** from your repository to the original repository.
4. Next we will **merge** the pull requests and get the text.

Try not to look at others' PRs to maintain the suspense.

Summary

1. What a version control system is and why it is needed.
2. What Git and GitHub are.
3. How to work with repositories.
4. Basic Git terms.
5. The process of working with Git on the course project.



Homework

Made two iterations of pull requests

1. **Git lesson (add readme)**
    - 1.1. Create a branch with your task about git lesson
    - 1.2. Add yourself and your mentor into README.md
    - 1.3. Commit and push these changes
    - 1.4. Create a pull request and set your mentor as a reviewer
    - 1.5. Wait until your mentor will merge your branch into target branch
2. **Save results of HTML lesson**
    - 2.1. Move back to target branch
    - 2.2. Create another branch with your task about html lesson
    - 2.3. Add your HTML files (index.html, blog.html)
    - 2.4. Commit and push
    - 2.5. Create a pull request and set your mentor as a reviewer
    - 2.6. Wait until your mentor will merge your branch into target branch

Quality Criteria for HTML Course

❤️ **The site should have a GitHub repository**

❤️ **There's no system file**

No .DS_Store or Thumbs.db files in the repository

❤️ **There are no commits in the target branch**

Target branch (usually main) contains only merge commits

💛 **Commit messages are descriptive**

✅ good: "fix validation in contact form" ❌ bad: "fix", "sdfsdf"

💛 **Use the imperative mood**

✅ good: "fix validation" ❌ bad: "fixed validation"

💛 **Each commit is a logical unit**

don't commit more than described

💚 **Commits are atomic**

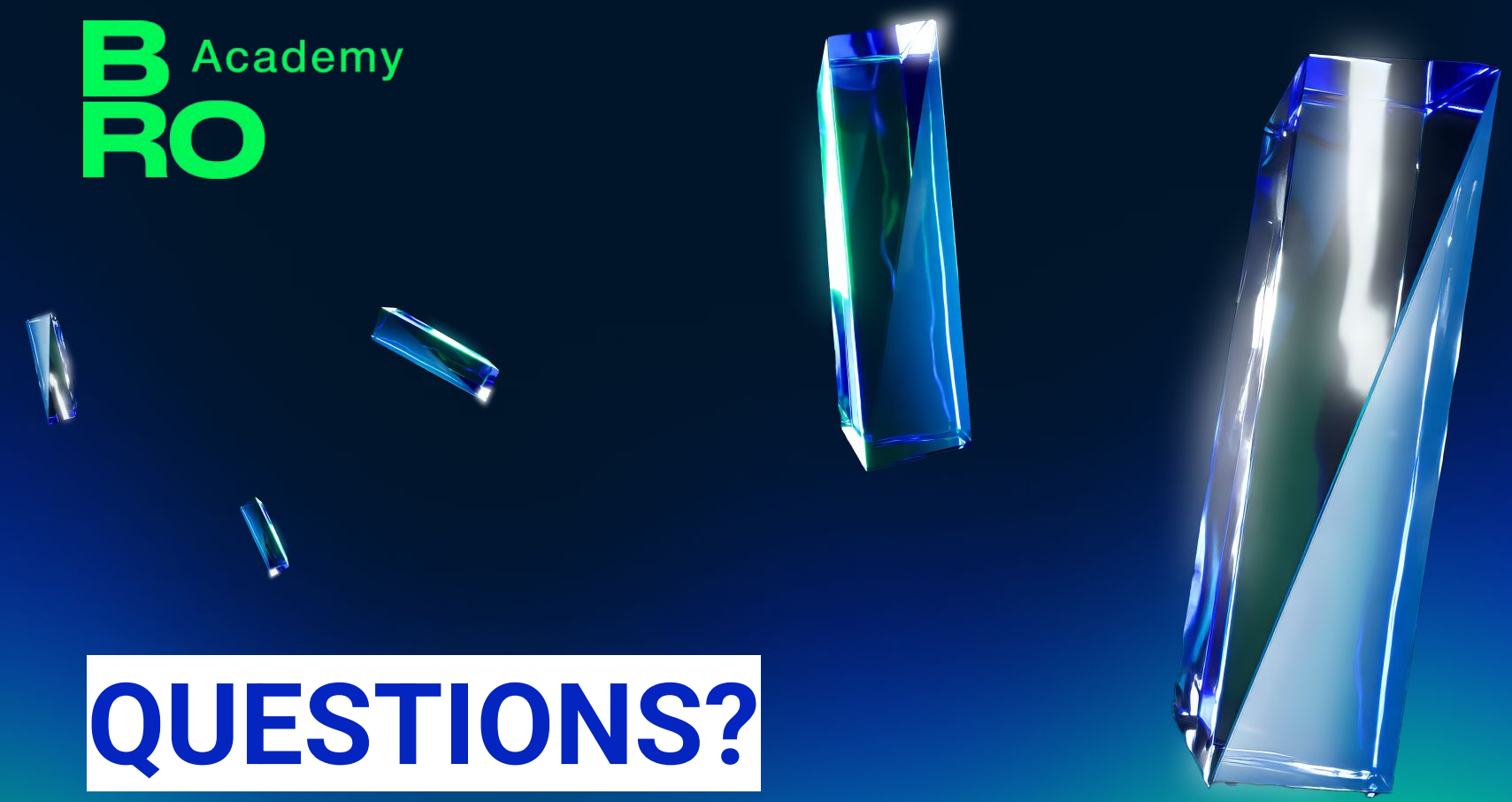
logical units, the smallest possible size

💚 **Commit message no longer than 60 symbols**

to be visible in GitHub UI

❤️Mandatory for passing the course
💛Required for the highest grade
💚Optional

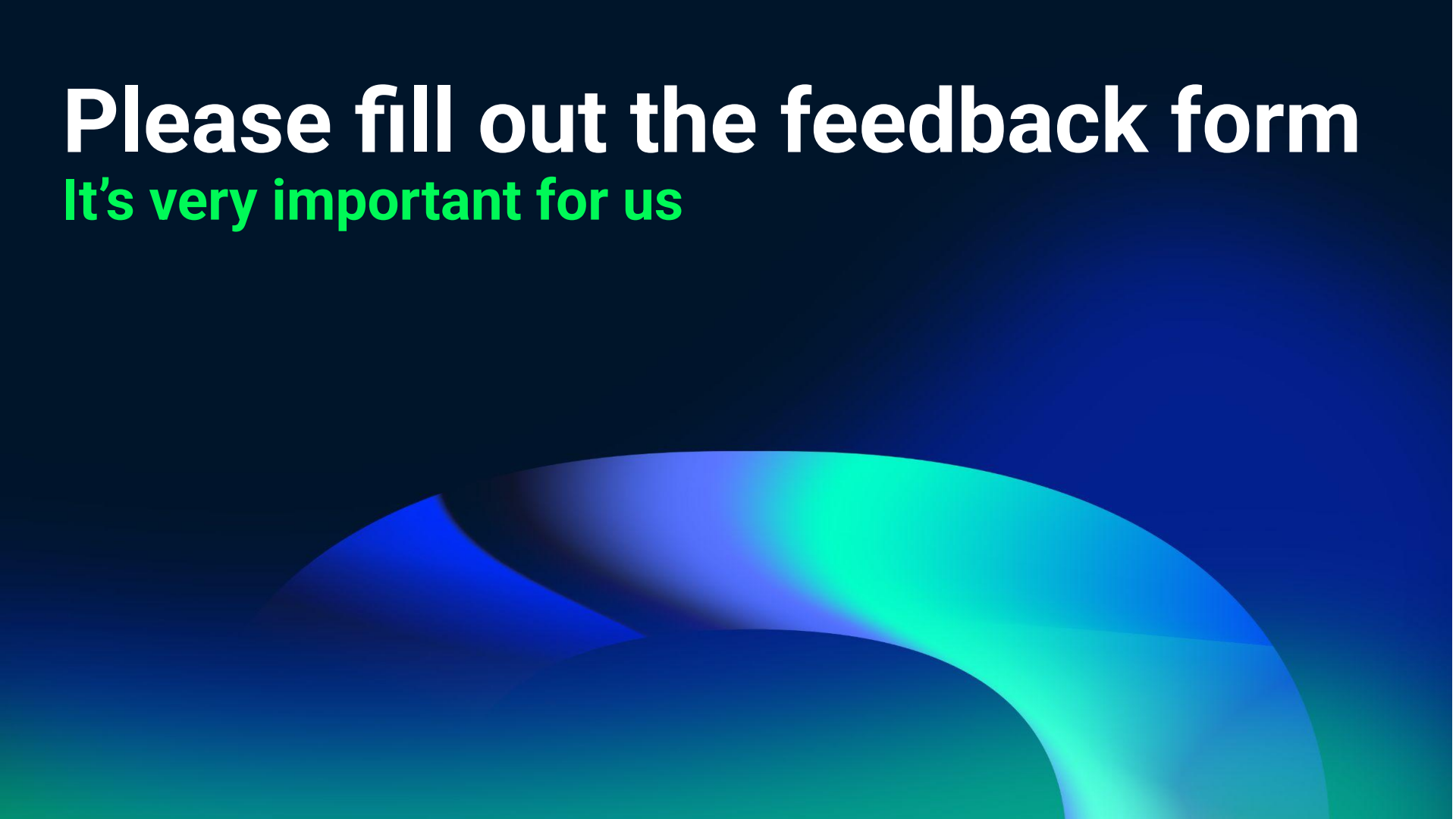
BRO Academy



QUESTIONS?

Please fill out the feedback form

It’s very important for us




THANK YOU!
Have a good evening!