esade

Advanced Python

Class 3

Pepe Bonet Giner

12th January 2024

Index Class 3

Topic 1: Recap Class 2

Topic 2: GitHub I

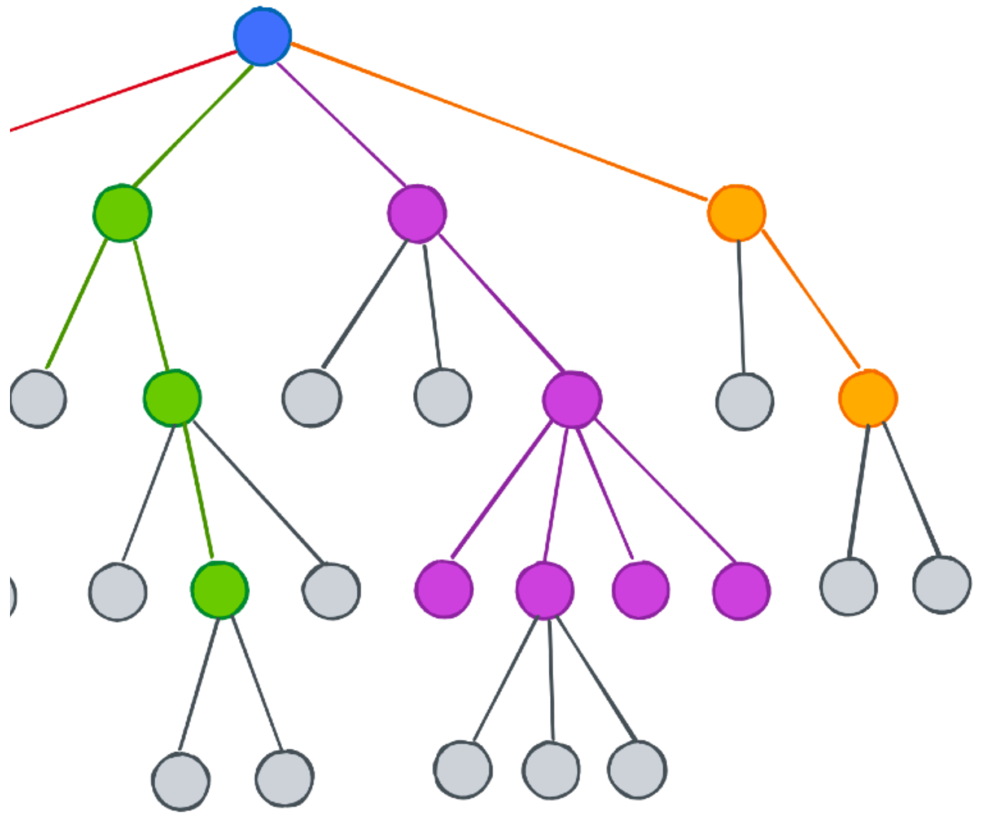
Topic 3: Bringing Class 1 & 2 to Github

esade

Topic 1: Recap Class 2

The 4 things we saw

CLI Parser (Click)



Debugger (PDB)

Code Working

Code Working

Code Working

Code Working

Add debugger (stop the code)

Inspect the problem

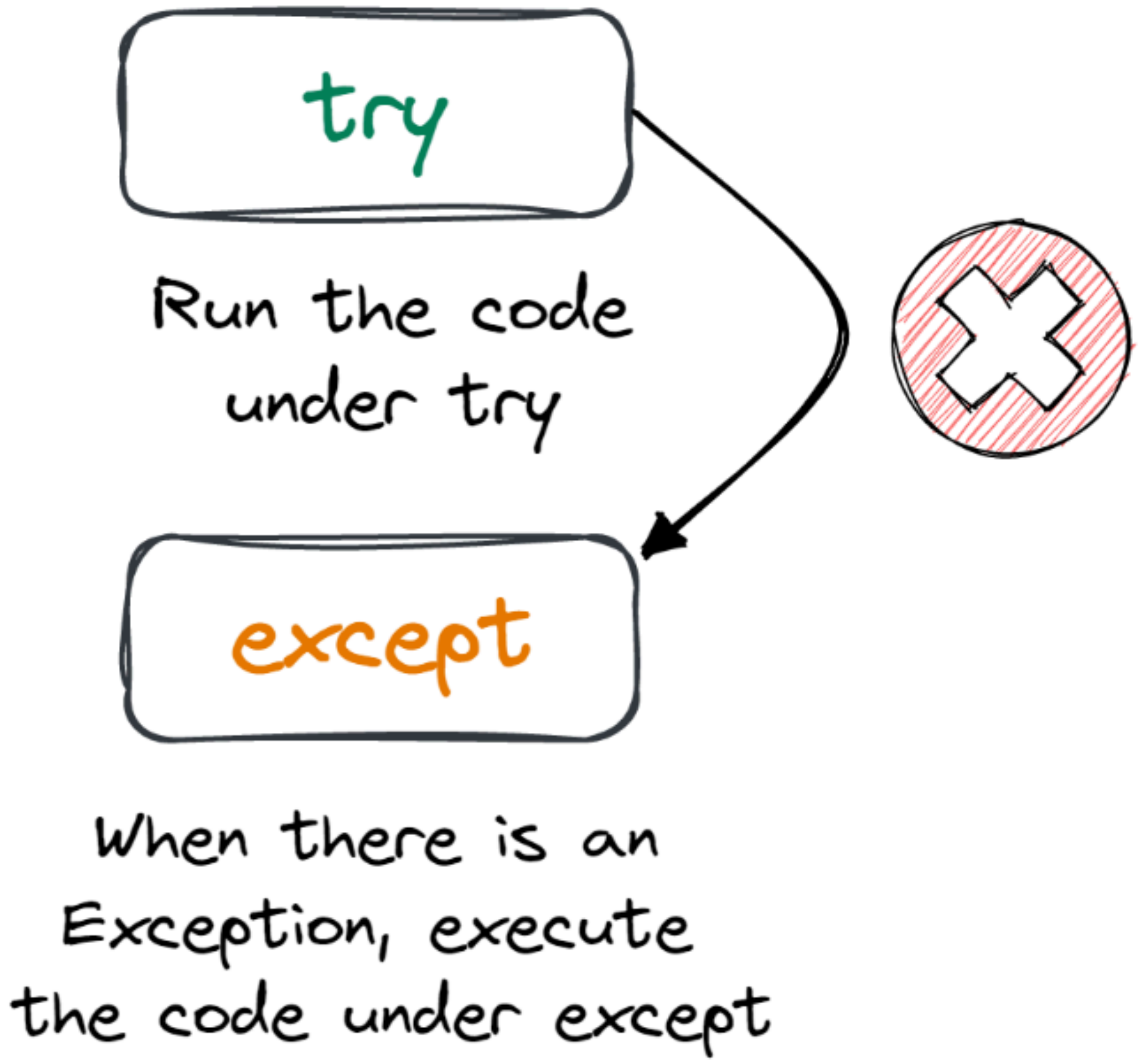
Solve it

Code Not Working —> Code Working

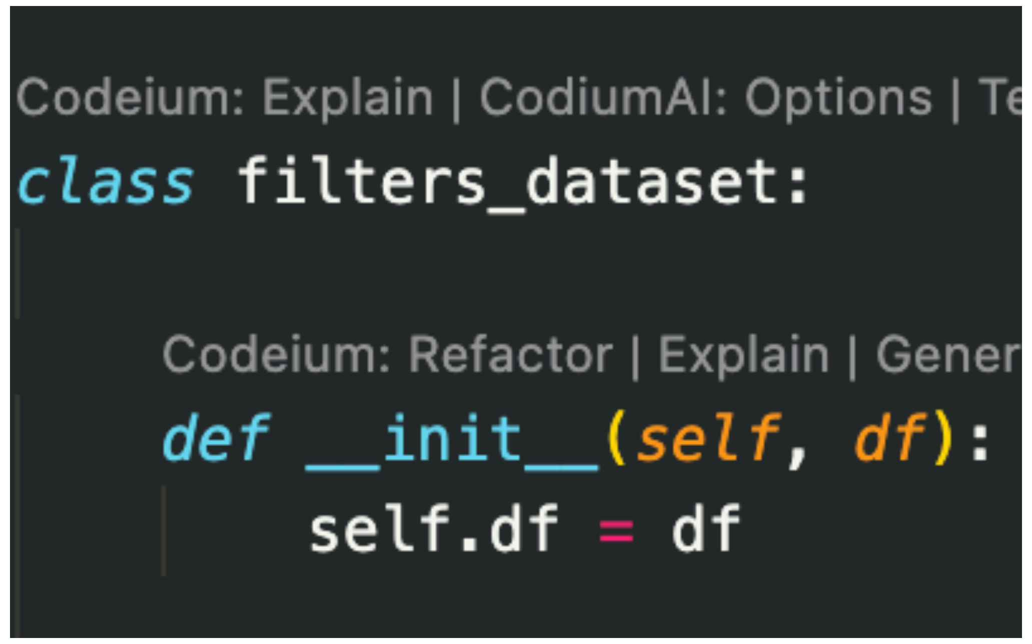
More Code

More code

Try & Except



Classes

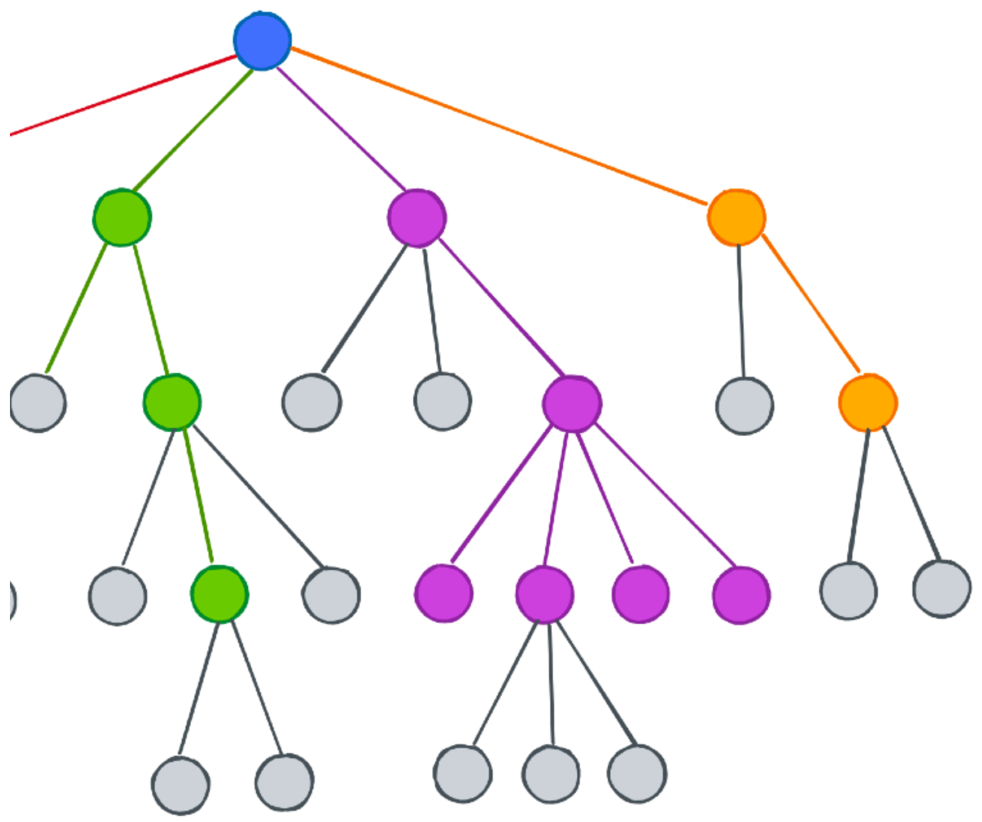


```
Codeium: Explain | CodiumAI: Options | Te
class filters_dataset:

    Codeium: Refactor | Explain | Gener
    def __init__(self, df):
        self.df = df
```

The 4 things we saw

CLI Parser (Click)



Debugger (PDB)

Code Working

Code Working

Code Working

Code Working

Add debugger (stop the code)

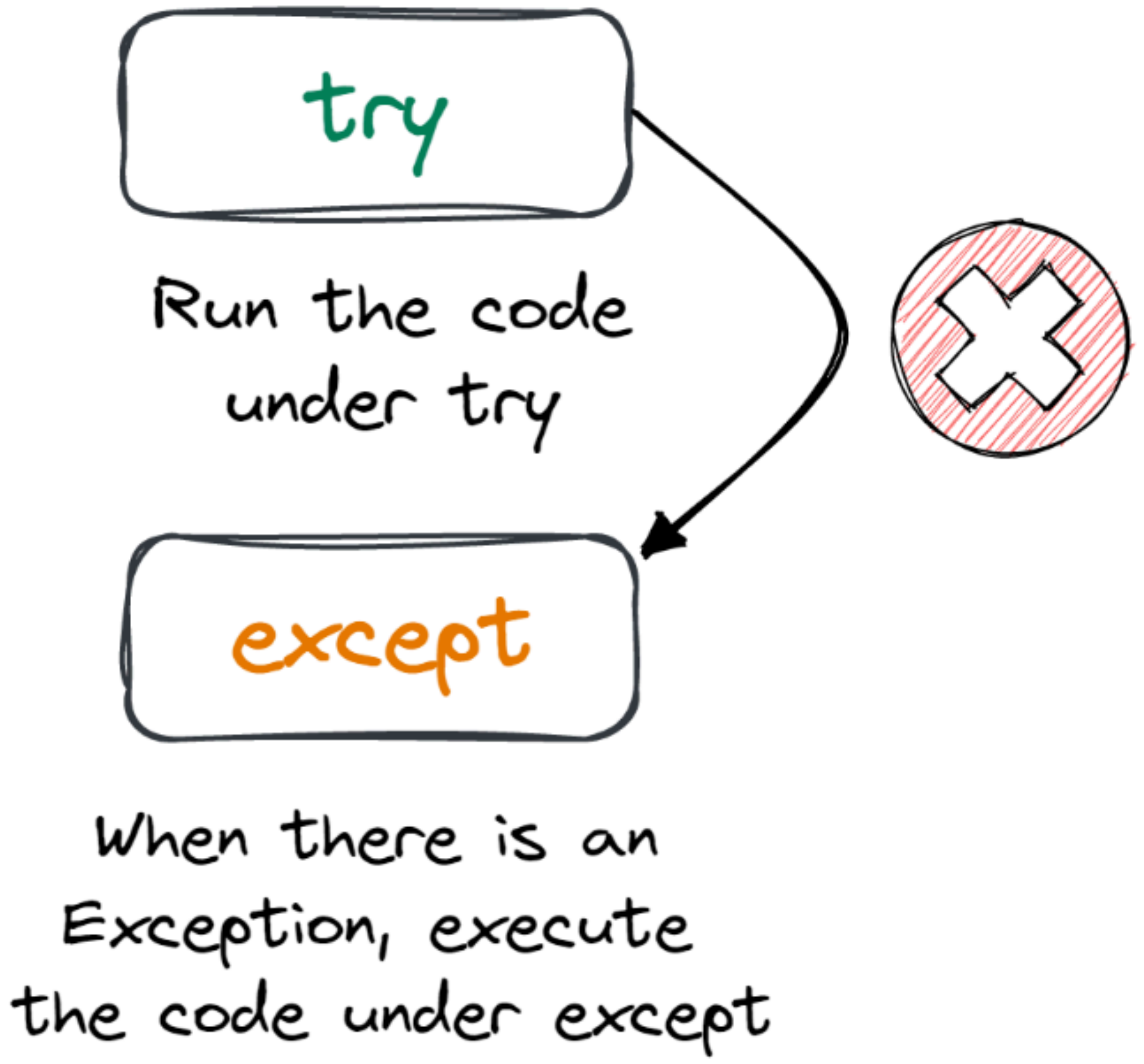
- Inspect the problem
- Solve it

Code Not Working —> Code Working

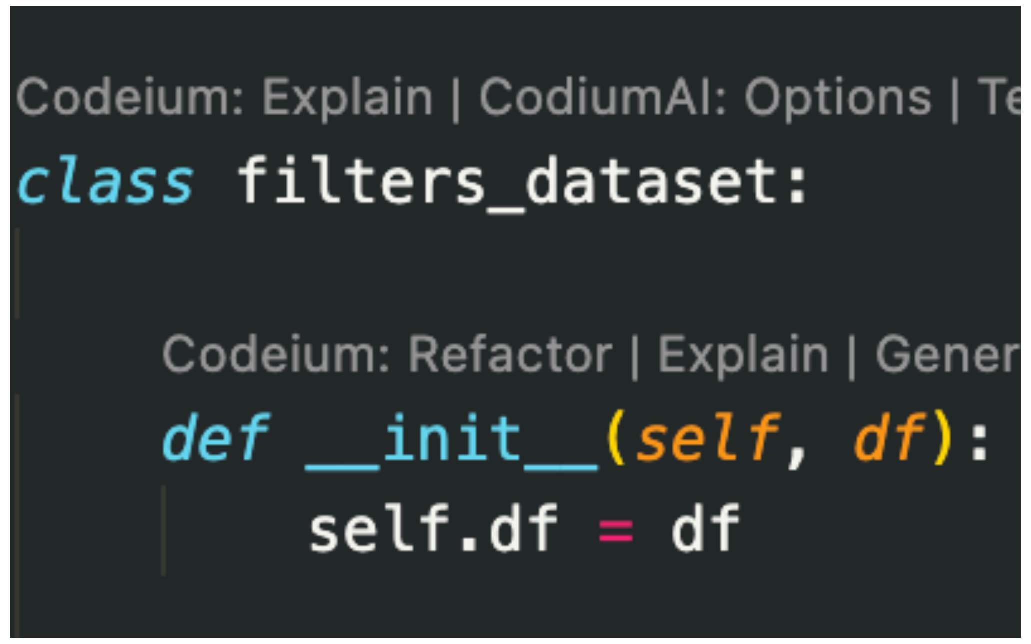
More Code

More code

Try & Except



Classes



```
Codeium: Explain | CodiumAI: Options | Te
class filters_dataset:

    Codeium: Refactor | Explain | Gener
    def __init__(self, df):
        self.df = df
```

Why are they important? What are their building blocks? Whiteboard Recap

Topic 2: GitHub I

The Development Environment

"If you have time to set up only one piece of infrastructure well, make it the development environment"

Ville Tuulos

IDE

- Editor where you write the code

Visual Studio

Sublime Text

Code Versioning

- Store & track different versions and stages of your code

git

GitHub

GitLab

CI/CD

- Continuous Improvement & Continuous Development

GitHub Actions

circleci

Code Versioning - GitHub

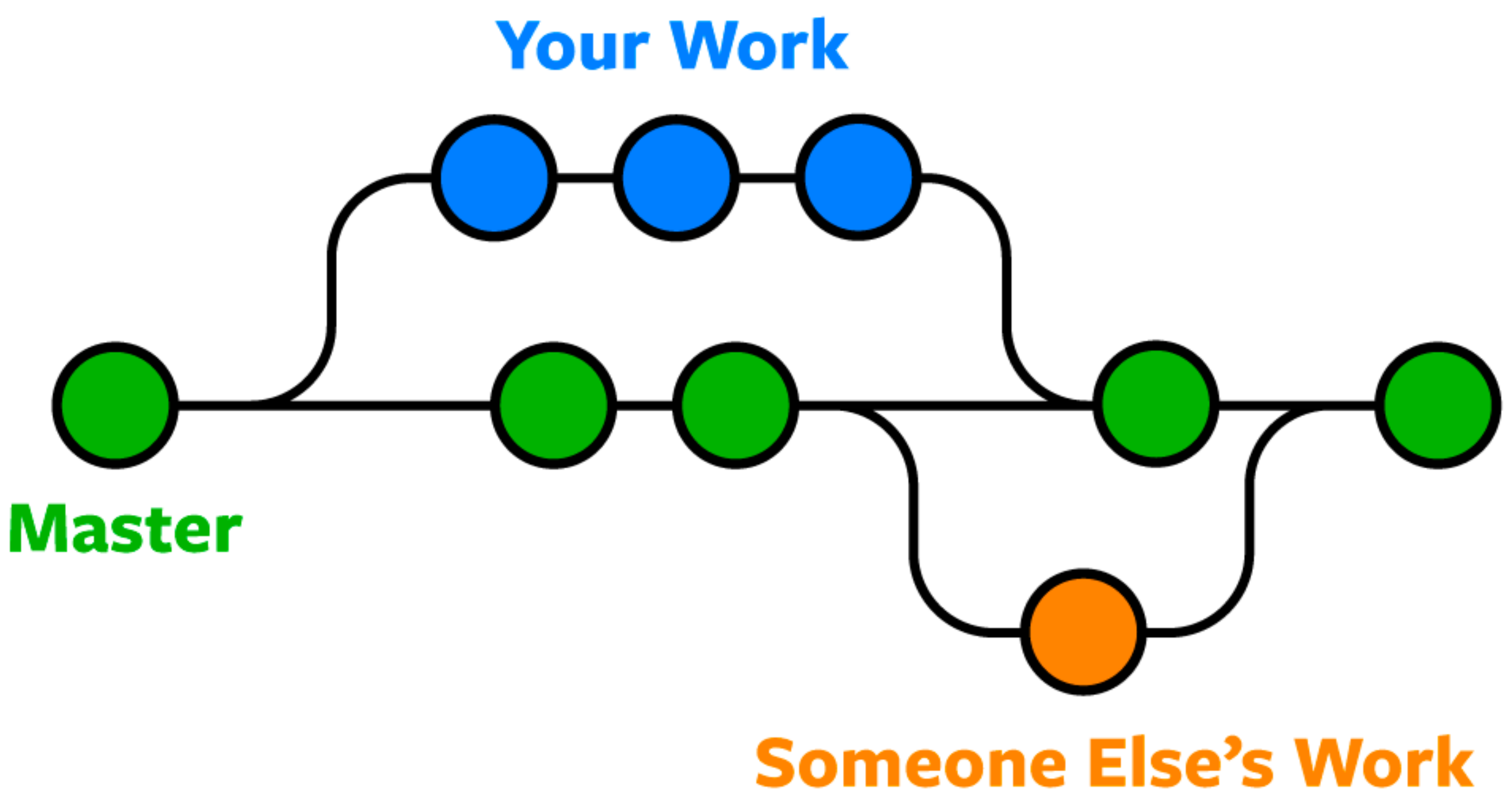
- **Versioning:** Allows developers to track changes to their codebase over time.

- **Collaboration:** It enables multiple developers to work on the same project concurrently, managing code modifications, merging changes, and resolving conflicts efficiently.

- **Backup:** Remote code repository, hosting your project's code and providing a secure backup

- **Continuous Integration and Deployment (CI/CD):** Automated build and testing processes. Ensure code quality, and allows for smooth deployment to production environments.

GitHub



Exercise

- I assume all of you have a GitHub account already created and you downloaded GitHub Desktop
- Let’s login in Github and in GitHub Desktop

GitHub

Desktop

Exercise

- Create a new repository to store all your files and the changes you make to them

GitHub

Desktop

What is .gitignore for?

- .gitignore is a file used in all git repositories to ignore files that you never want to put up in your remote repository (web / GitHub)
- Examples:
    - Secret information (Passwords etc)
    - Large files

Exercise

- Share the repository with me. Add me as a collaborator

GitHub

Desktop

Exercise

- Add a folder named scripts in the newly created folder and start a new python script

GitHub

Desktop

Exercise

- Commit new changes and push to your remote repository (The web)
- Explain everything step by step

GitHub

Desktop

Exercise

- Create a new branch and publish it to your remote repository

GitHub

Desktop

Exercise

- Make a change to one files in this new branch.
- Commit the changes in the file to the new branch

GitHub

Desktop

Exercise

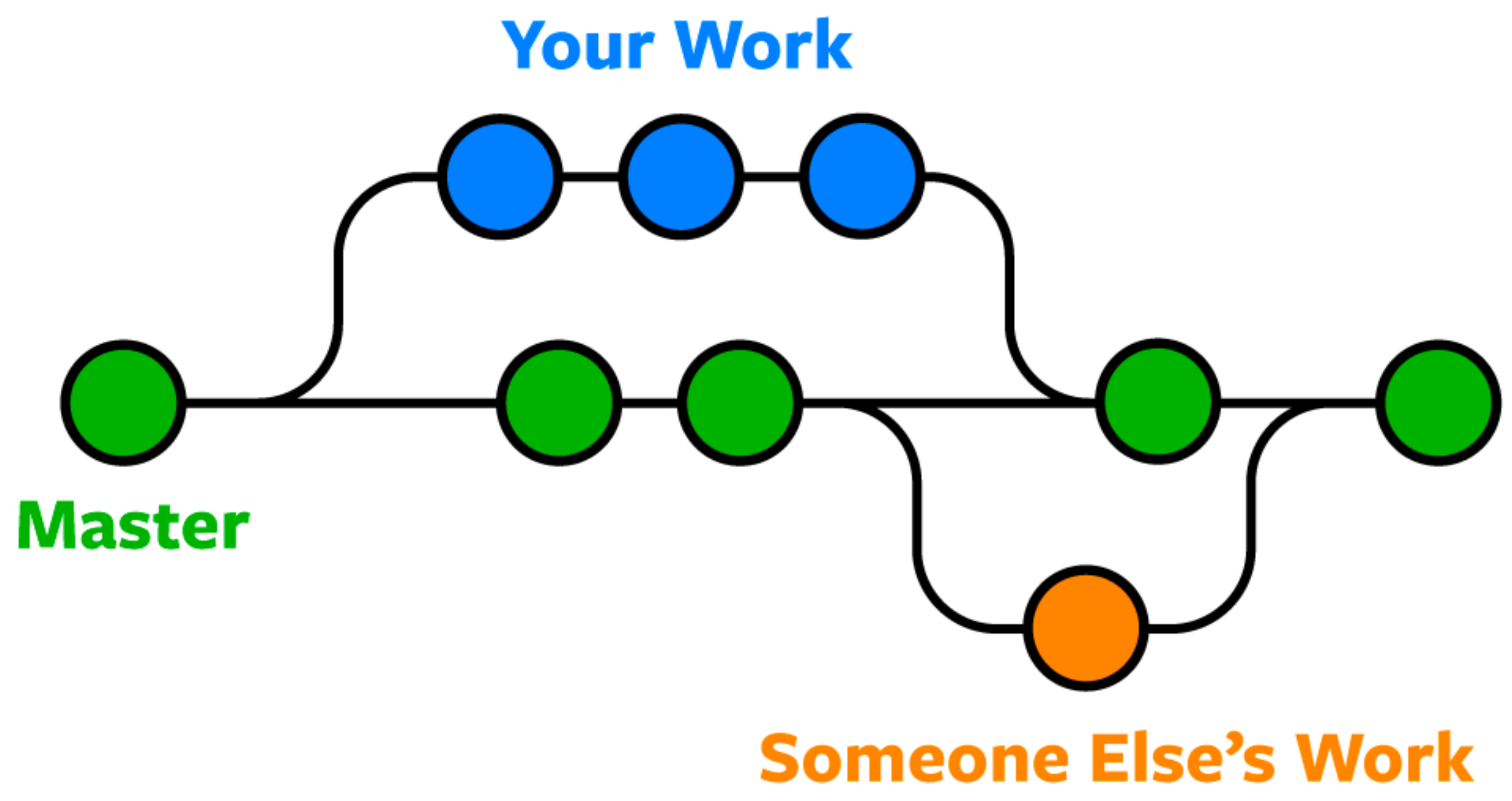
- Push to your new branch
- Preview a pull request
- Create a pull request and merge your new branch with master
- Please Explain properly

GitHub

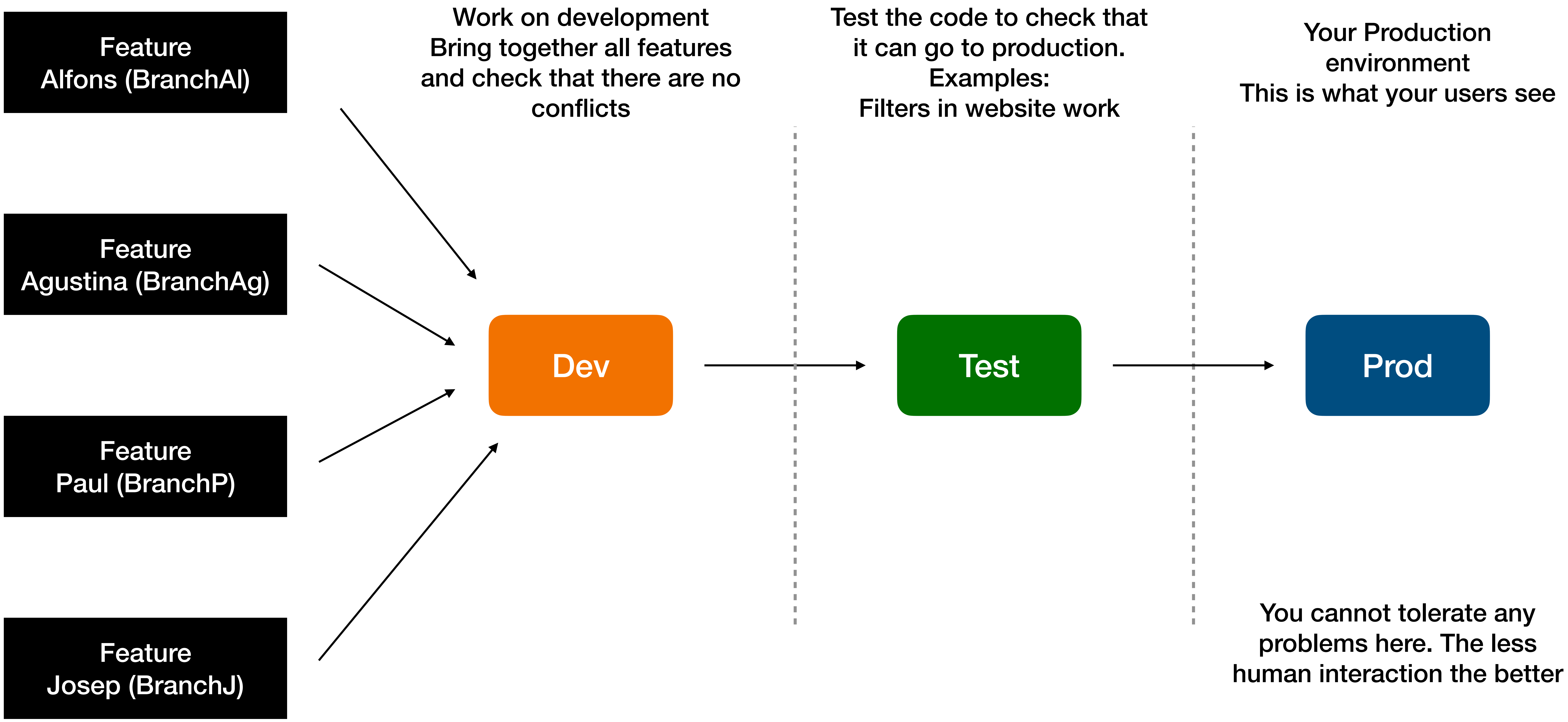
Desktop

Take home message —> You gotta use GitHub (or similar)

- **If you manage any type of data team or want to in the future**
- **If you want to do any data project**
- **If you want to properly learn how to work on an ML project**
- **If you plan on collaborating with people**
- **If you want to create your own product to share on the internet**



The infrastructure of a product we are building @ Hynts



Feature
Alfons (BranchAl)

Feature
Agustina (BranchAg)

Feature
Paul (BranchP)

Feature
Josep (BranchJ)

Work on development
Bring together all features and check that there are no conflicts

Dev

Test the code to check that it can go to production.
Examples:
Filters in website work

Test

Your Production environment
This is what your users see

Prod

You cannot tolerate any problems here. The less human interaction the better

Some Comments about commits

- Commit Regularly
- Use Meaningful Commit Messages
- Try using branches for different features, bug fixes, or experiments.
- Merge Changes Carefully
- Push regularly to GitHub for backup

esade

Topic 3: Class 1 & 2
To Github

Exercise

- Download dataset FilmGenreStats from the Moodle
- Let’s create a script within our folder that is connected to Github. This script should have:
  - Classes
  - Click
  - Try & Except
  - Maybe some import pdb;pdb.set_trace()
  - Every meaningful step we take should correspond with a commit