

Tutorial 3 Git + JavaScript

Tutorial 3

Git Practice

JS Practice

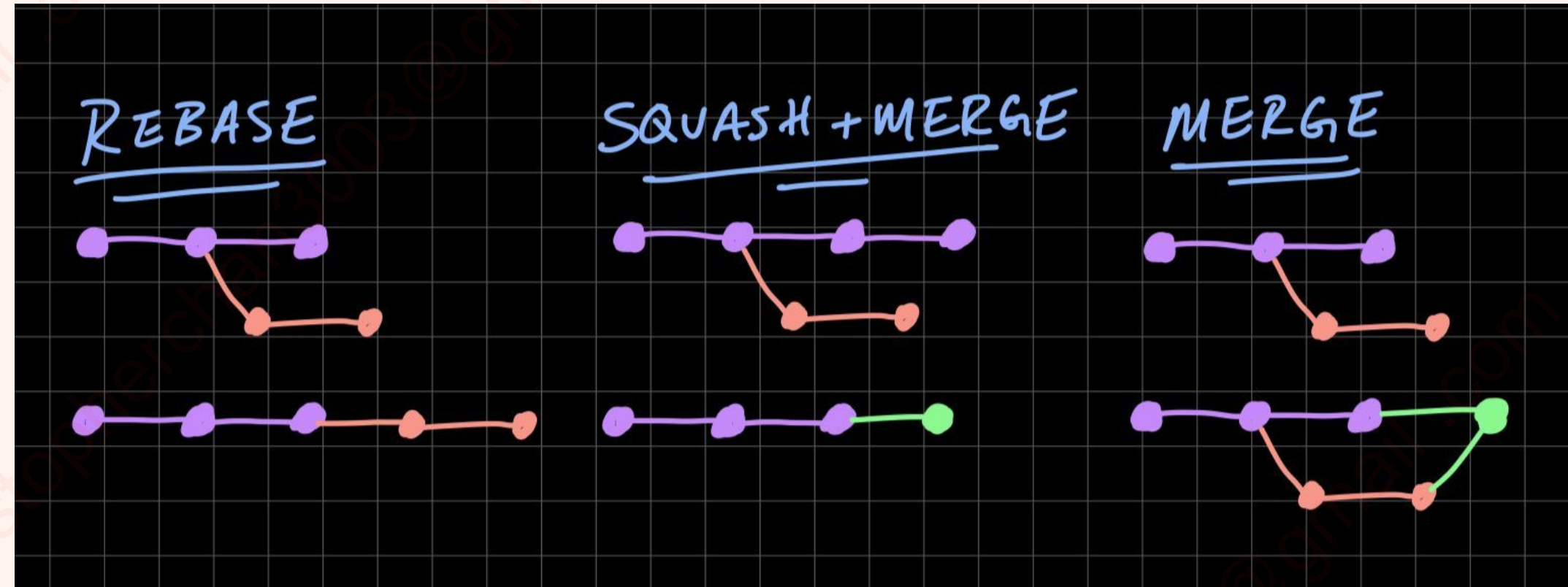
Assignment 3 review

Assignment 4 review

Q & A

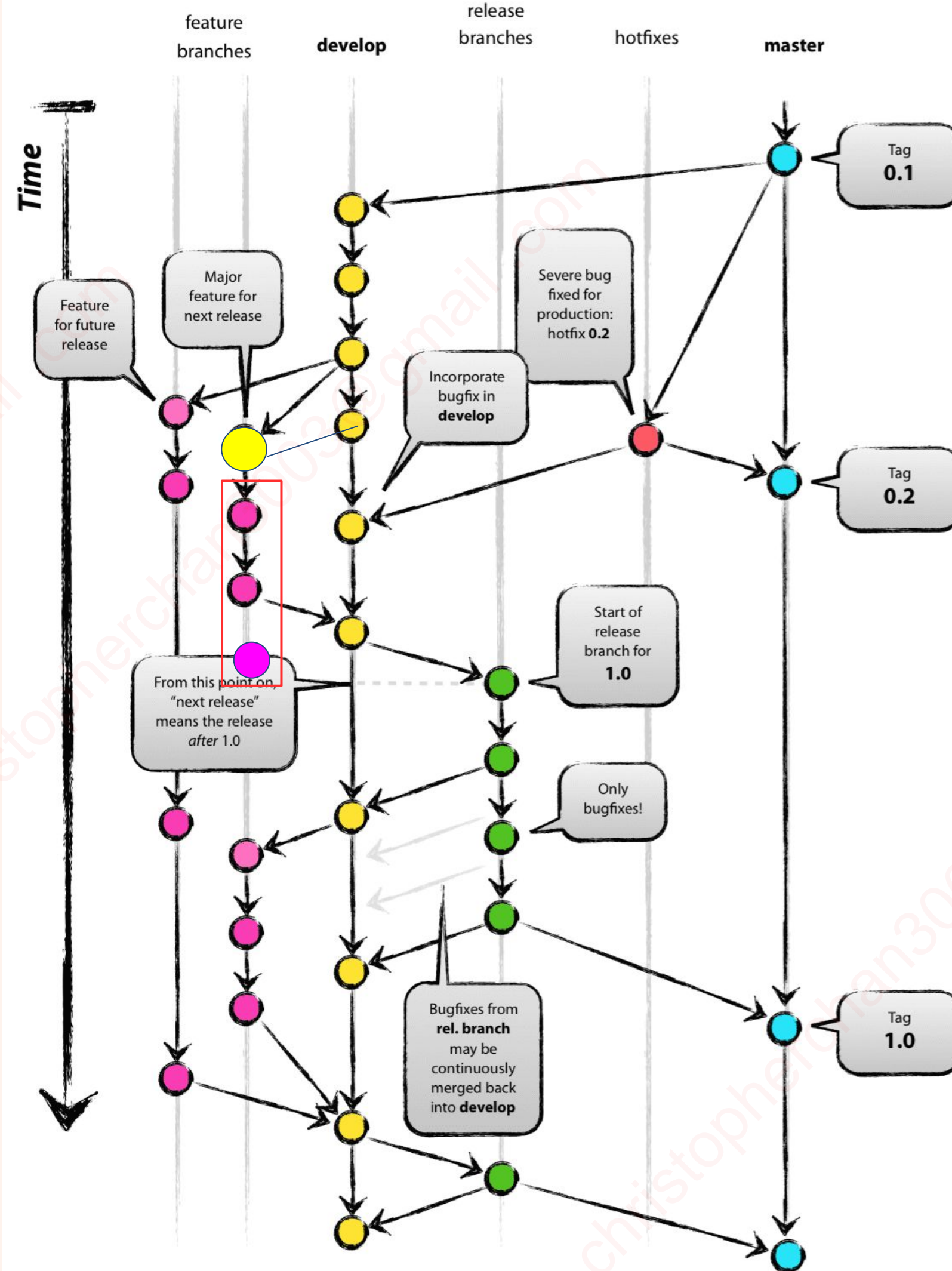
Git Practice

1. git clone xxx.git
2. git config (config user)
3. git status (check files status)
4. git diff
5. git add (. / myfile.txt)
6. git commit
7. git commit -am "feat: my feature"
8. git reset (撤销commit记录)
9. git revert (用新的commit撤销)
10. git stash (缓存当前工作进度)
11. git branch
12. git switch
13. git checkout (-b = branch+switch)
14. git merge (合并代码)
15. git rebase (合并代码并且变基)
16. git pull
17. git push



Gitflow Workflow

1. 什么是 git flow ?
2. 为什么我们需要和原因 git flow ?
3. 大概概述它是如何工作的
4. Git init Repo
5. 协作创建新的 feature 分支, 模拟日常开发流程。
6. 实践如何在 feature 分支上进行 add、commit 和 push 操作。
7. 正确 commit message
8. **创建合并冲突**
 - a. 学员故意制造分支间的冲突, 以模拟实际工作中可能遇到的情况。
 - b. 学习如何使用 Git 命令来处理和解决合并冲突。
9. **合并到 dev 和 master**
 - a. 练习将 feature 分支合并到 dev 分支, 再从 dev 分支合并到 master。
 - b. 讨论何时应该合并到 master, 以及合并请求 (MR/PR) 的最佳实践。
 - c. 正确 commit message



Git Workflow

git init

vim readme.md (title)

git add readme.md

git commit -am "init: repo init"

git branch develop

git branch feature/feature-1

git branch feature/feature-2

git switch feature/feature-1

vim reamde.md (title 1)

git add.

git commit -m "feat: title1"

git switch develop & git merge feature/feature-1

git switch feature/feature-2

vim reamd.md (title 2)

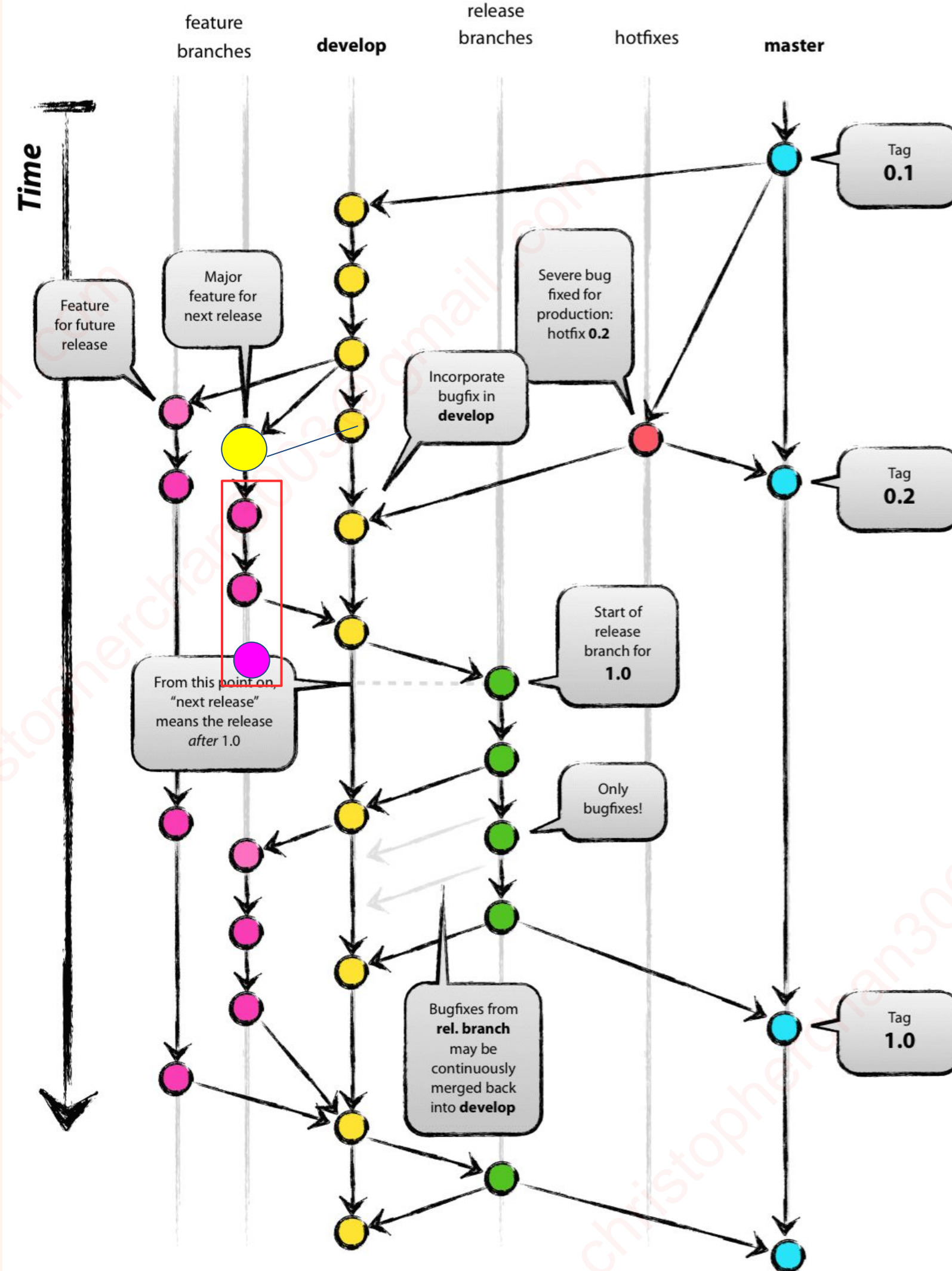
git add. & git commit -m "feat: title2"

git rebase develop (resolve conflicts)

git add .

git rebase --continue

git switch develop



Git Workflow

Branch:

feature/JR-230-add-page-title

bugfix/xxx

release/xxx

hotfix/xxx

main/xxx

Git commit message:

feat: add page title

fix: fix bugs for button

docs: update readme file

chore: xxx

refactor: xxx

test: add unit test

<https://www.conventionalcommits.org/en/v1.0.0/>

1.23.753

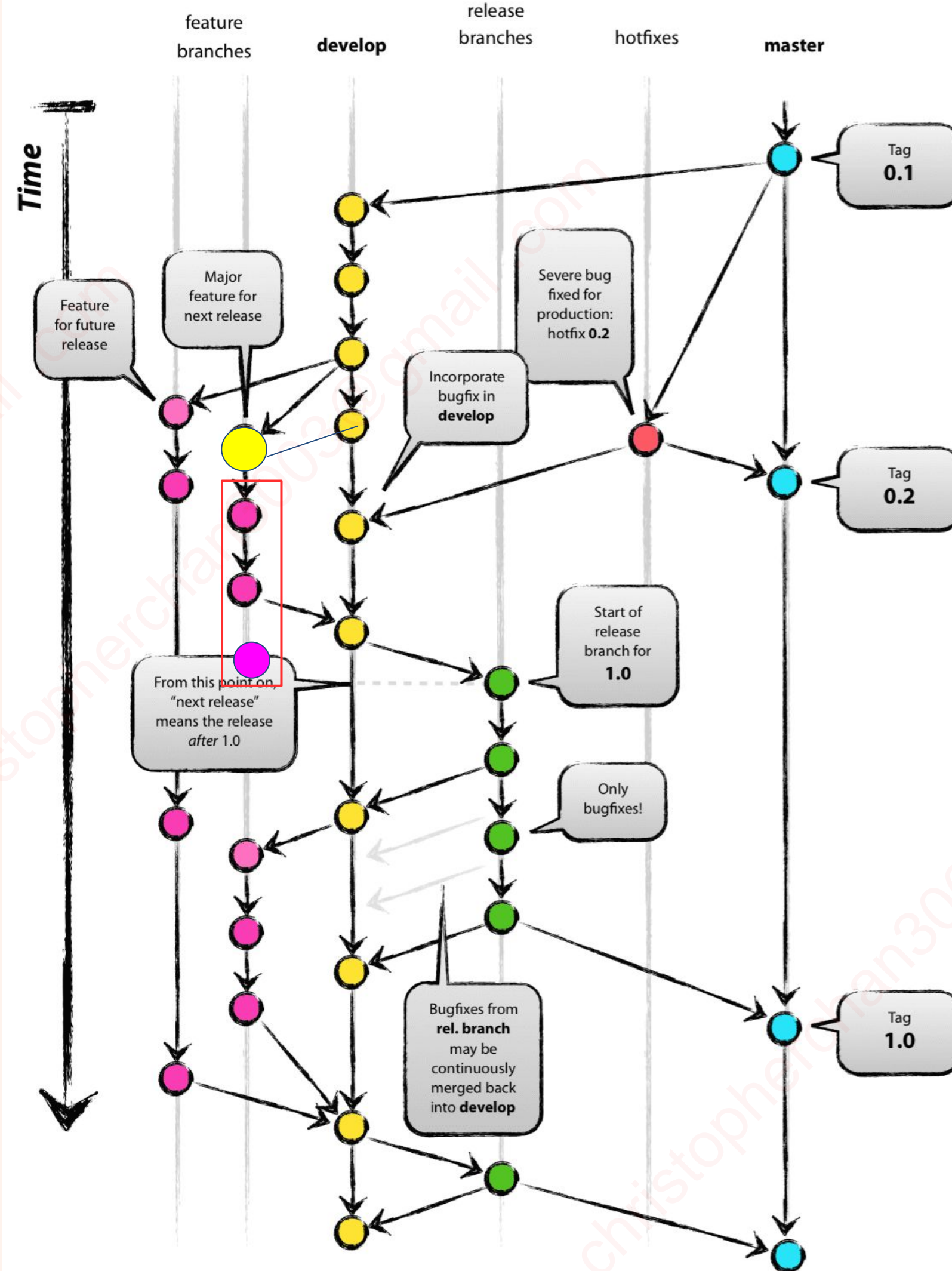
[Major].[Minor].[Patch]

Semantic Versioning

fix 1

fix 2

改动1



JS Practice

1. 用函数实现计算的基本功能

- a. `add(a, b)`
- b. `subtract(a, b)`
- c. `multiply(a, b)`
- d. `divide(a, b)`
- e. `square(a)`
- f. `isEven(a)` ←**
- g. `isPrime(a)` ←**

质数Prime定义为在大于1的自然数中, 除了1和它本身以外不再有其他因数。

Assignment 3 Review

作业要求

你需要为一个简单的购物清单编写一个小程序。

1. 数据与变量:

创建一个购物清单数组, 例如

2. `let shoppingList = ["牛奶", "鸡蛋", "面包"];`

3. 数组应用:

- 向购物清单中添加两样物品, 并输出更新后的清单。
- 从购物清单中删除最后一项物品, 并输出结果。

4. 条件与循环:

- 创建一个函数, 当购物清单中的物品超过5项时, 在控制台输出"你的购物车满了"。

5. 使用循环语句遍历购物清单, 将每一项物品在控制台上以编号的形式输出。例如:

1. 牛奶
2. 鸡蛋
3. 面包

1. 函数与对象:

- 创建一个函数, 该函数接受物品名称作为参数, 并返回该物品是否在购物清单中。
- 创建一个购物物品对象, 其中包括物品名称、价格和数量。例如:

```
let item = { name: "牛奶", price: 12, quantity: 1 };
```


Assignment 4 Review

作业要求:

1. Create a GitHub repo called 'jr-git-intro' and clone it to your local.
2. Inside your local repo, create a new file called 'first.md' and add some text to it. Commit the change to your local using the commit message "feat: JR-101 Set up the first file".
3. Push the branch (master) to your remote.
4. Create two other files 'second.md' and 'third.md'. Add some text to it. Commit the change using the commit message "JR-102: Set up the second and third file".
5. Remove 'third.md' and commit the change using the commit message "JR-103: Refactor and clean up redundant code".
6. Create a branch called bugfix/JR-104-fix-words-in-second-file
7. In this new branch, change the first line of 'second.md' to something else. Commit the change with the message "JR-104: Modify first line of second file"
8. Switch back to master branch, make a commit to change the first line of 'second.md' to something different to what you did in Step 7. Commit the change with the message "JR-105: Modify first line of second file in a different way".
9. Merge branch bugfix/JR-104-fix-words-in-second-file into master. Resolve the conflicts to accept changes in the bugfix branch.
10. Delete the bugfix branch and push the new commits in master to GitHub.
11. Check out branch feature/JR-106-create-home-page, add a new file 'index.html' with **Home Page** in the code. Commit the change with message "JR-106: Create the home page"
12. Check out master branch, add a new file 'index.html' with **About** in the code. Commit the change with message "JR-107: Create the about section". Note this is to simulate changes made by your colleague. In your work, it's rare that you will ever need to work in the master branch. Push the new commit to GitHub.
13. Push the new branch feature/JR-106-create-home-page to remote and make a pull request into master on GitHub. Make sure you write meaningful pull request title.
14. Use the git rebase to resolve merge conflicts and result in a fast-forward merge in the remote repository.
15. Merge the PR and delete the feature branch.
16. Pull the latest change in master to your local
17. Provide a graphic history of your current local Git repo. Take a screenshot of it and put it in your Git repo. Commit the change in master with message "JR-108: Add screenshot" and push it to GitHub.
18. Add the url to your GitHub Repo in the excel spreadsheet provided.

Assignment 4 Review

作业要求:

1. xxx
2. T
3. F
4. T
5. T
6. F
7. F
8. T, but it depends, local feature branch -> reset, public branch -> revert

Q & A

git reset HEAD~1 默认--mixed 清空暂存区但是保留工作区的改动, but --hard 舍弃未提交的更改(慎用)
CLI vs GUI:先熟悉每个命令以及作用, 再熟悉了之后再去通过图形化工具减少繁琐和易出错的命令

Thanks!