

Git 分支管理入门

Git 的分支管理不会导致存储或者内存上的开销，反而维护逻辑缜密的多分支比维护臃肿的单分支简单许多，因此：早用分支，多用分支很重要。

1、创建新分支

模板：

`git branch <分支名>`

例如：

`git branch Starts`

或者还可以在 pycharm 中找到

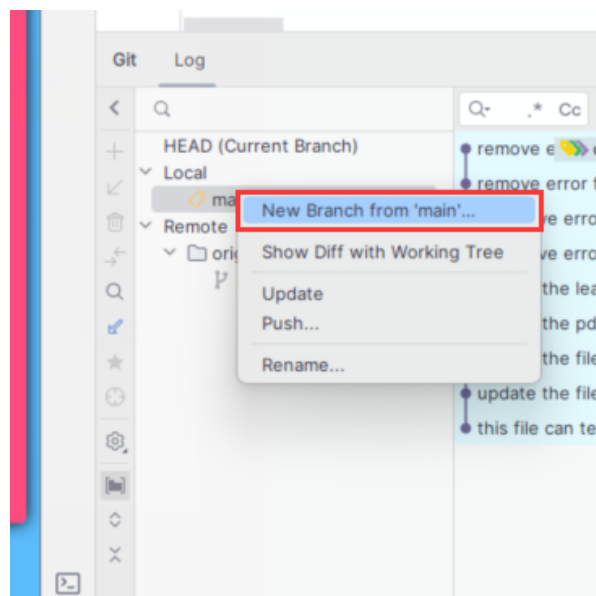


图 1 在 pycharm 中创建新分支

2、切换分支

从分支 A 创建新分支 B 后，当前依旧停留在分支 A 上，要在分支 B 上更新

代码必须先将分支切换到分支 B 上。

模板：

git checkout <分支名>

例如：

git checkout Starts

```
PS F:\git_repository\learn_git> git branch Starts
PS F:\git_repository\learn_git> git checkout Starts
Switched to branch 'Starts'
M      learn_git/solve_error_methods/solve_error_methods.md
PS F:\git_repository\learn_git> 
```

图 2 在命令行中创建新分支，并转移到新分支

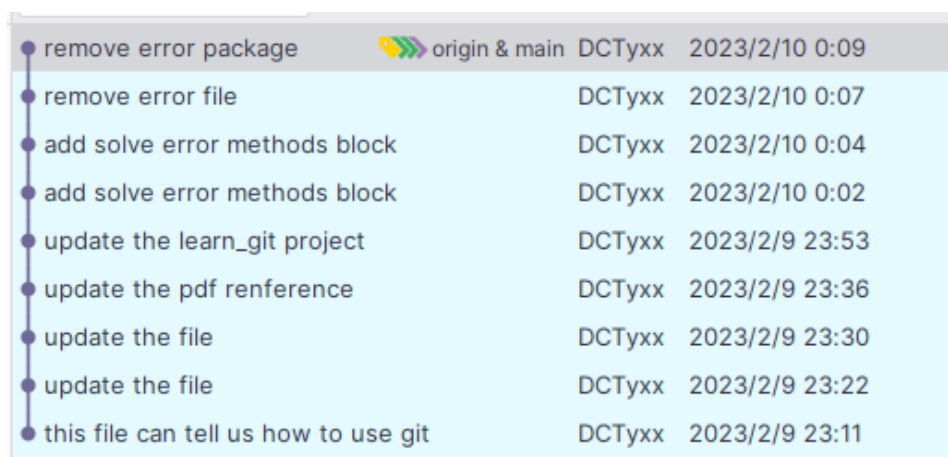


图 3 创建行分支后在 pycharm 中的可视化结果

3、在完成版本更新、迭代后，我们需要将项目合并到同一个分支上。合并有两种范式

合并分支一

我们要想将 A 分支与 B 分支合并。A 操作指令则 B 合并到 A，反之。

模板：

git merge <branch name>

例如：

`git merge Starts` （在 main 分支中操作，将 Starts 分支合并到 main 中）

合并分支二

取出一系列提交记录，“复制”它们，然后在另一个地方逐个放下去。这么看会创造更线性的提交记录。与方法一不同，我们要想将 A 分支与 B 分支合并。A 操作指令则 A 合并到 B，反之。

模板：

`git rebase <branch name>`

例如：

`git rebase main` （在 Starts 分支中操作，将 Starts 分支合并到 main 中）

练习：

- 1、 在 main 分支中创建并切换到一个新的分支 bugfix
- 2、 提交一次（忽略添加文件，下同）
- 3、 切回 main 分支再提交一次
- 4、 切换到 bugfix 分支，rebase 到 main 上
- 5、 切回到 main 分支，rebase 到 bugfix 上
- 6、 在 main 分支中新建并切换到一个新的分支 starts
- 7、 提交一次
- 8、 切回 main 分支

9、 提交一次

10、 将 starts 合并到 main (使用 merge)

解：

```
git branch bugfix
```

```
git checkout bugfix
```

```
git commit
```

```
git checkout main
```

```
git commit
```

```
git checkout bugfix
```

```
git rebase main
```

```
git checkout main
```

```
git rebase bugfix
```

```
git branch starts
```

```
git checkout starts
```

```
git commit
```

```
git checkout main
```

```
git commit
```

```
git merge starts
```

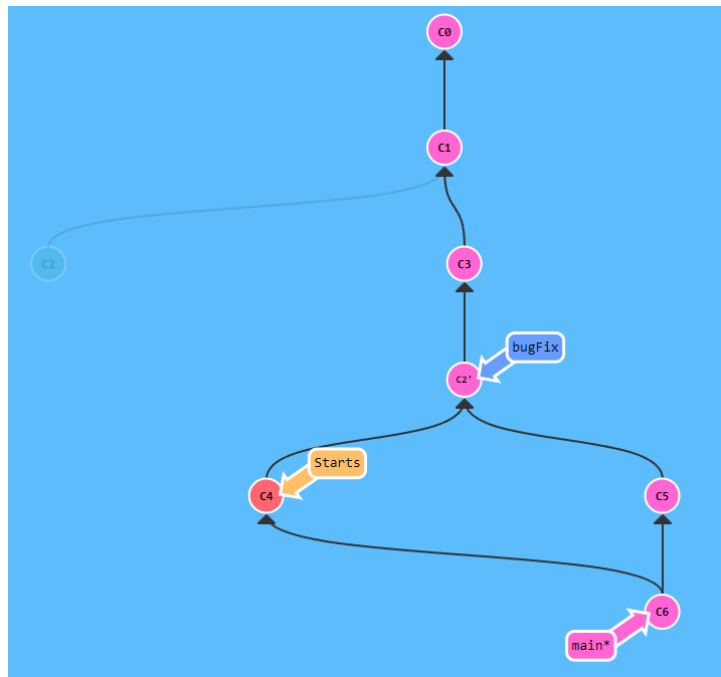


图 4 练习可视化结果