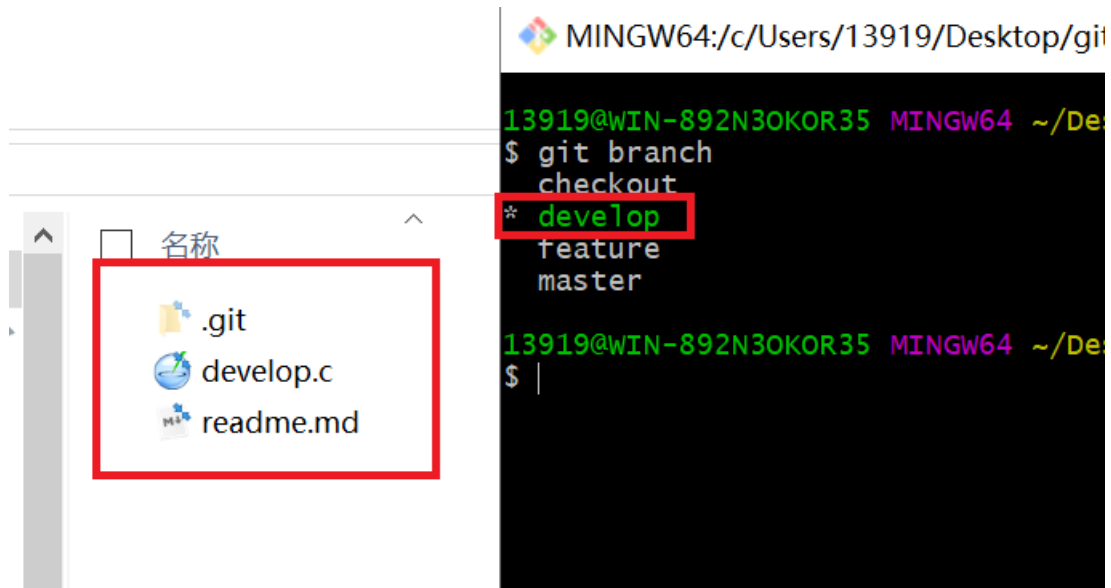


## 疑惑一：

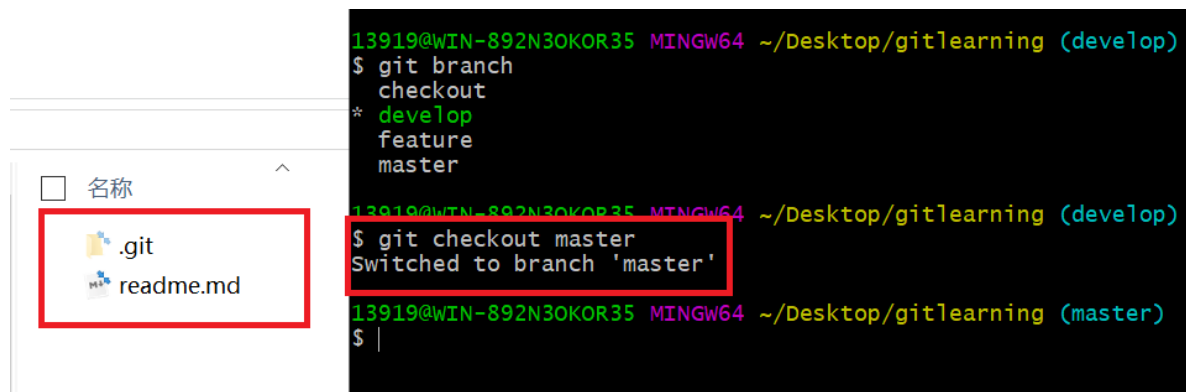
疑惑：

远端repository,保存的是所有分支push的内容,如果我要改feature branch上的一个文件(该文件),我如何知道它是特性分支还是开发分支。

1. 理解git的关键：切换到不同的分支，git文件夹所展示的文件也会变化。git文件夹，只显示当前分支下的文件。
- 当前分支为develop分支，文件夹中有两个文件



- 当从develop分支切换至master分支时，只显示develop分支下的文件。



## 疑惑二：

从develop分支拉取feature branch，从feature branch拉person branch,那么特性分支改变，如何同步到个人分支上

1. `git pull origin develop(父分支)` # 拉取父分支的文件

## 疑惑三：

如何本地创建分支，并推送至远端？

1. `git push origin test_A`

# 本地创建test\_A分支并推送至远端

## 疑惑三与四:

本地add添加的内容如何回退  
本地commit提交的内容，如何回退

1. 可以使用git撤销命令，使用撤销命令需要小心。可能会使你的更新被撤销

## 疑惑五

如何更新本地仓库

1. `git pull origin master`

**第一步: 创建远端仓库，并且在远端仓库仓库，添加 readme文件**

**第二步: 本地创建文件夹 gitlearning，之后 git init (git 初始化)，之后在本地拉取远端仓库**

```
# 拉取远端仓库
git pull git@github.com:xixihaha123-xjh/gitlearning.git
# 查看当前分支
$ git branch
* master
# 查看远端仓库发现为空
13919@WIN-892N3OKOR35 MINGW64 ~/Desktop/gitlearning (master)
$ git remote
# 增加一个新的远程仓库
13919@WIN-892N3OKOR35 MINGW64 ~/Desktop/gitlearning (master)
$ git remote add origin git@github.com:xixihaha123-xjh/gitlearning.git
# 查看远端仓库
13919@WIN-892N3OKOR35 MINGW64 ~/Desktop/gitlearning (master)
$ git remote
origin
```

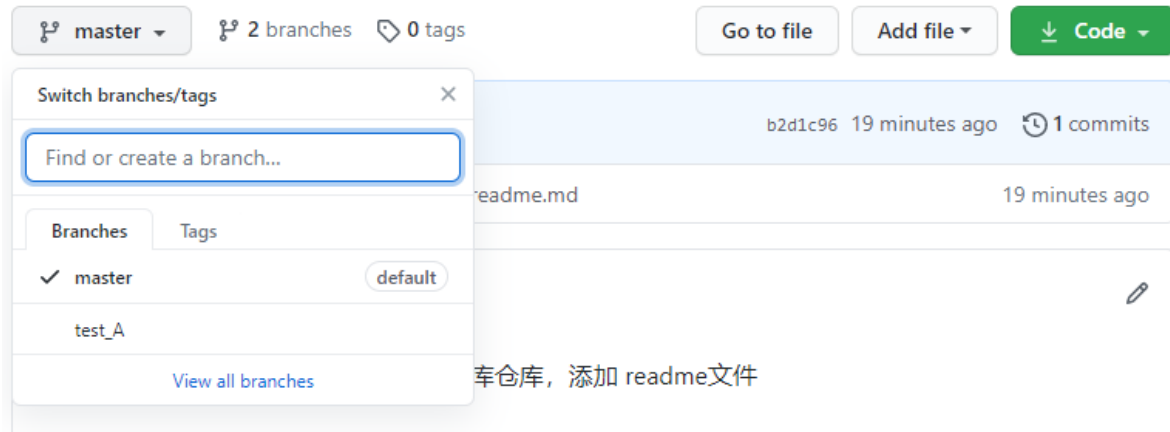
**第三步: 本地创建一个分支，并推送至远端仓库**

1. 本地创建一个分支，并推送至远端仓库

```
# 本地常见 test_A 分支
13919@WIN-892N3OKOR35 MINGW64 ~/Desktop/gitlearning (master)
$ git branch test_A
# 查看当前分支,发现 git branch test_A 只是创建分支，但未切换分支
13919@WIN-892N3OKOR35 MINGW64 ~/Desktop/gitlearning (master)
$ git branch
* master
  test_A
# 从master分支切换至test_A分支
13919@WIN-892N3OKOR35 MINGW64 ~/Desktop/gitlearning (master)
$ git checkout test_A
Switched to branch 'test_A'

# 本地创建分支并推送至远端
```

```
13919@WIN-892N3OKOR35 MINGW64 ~/Desktop/gitlearning (test_A)
$ git push origin test_A
```



## 2. 删除本地分支，并推送至远端仓库

```
# 删除本地分支 test_A
13919@WIN-892N3OKOR35 MINGW64 ~/Desktop/gitlearning (master)
$ git branch -d test_A
Deleted branch test_A (was b2d1c96)
# 删除远端分支 test_A
13919@WIN-892N3OKOR35 MINGW64 ~/Desktop/gitlearning (master)
$ git push origin :test_A
To github.com:xixihaha123-xjh/gitlearning.git
- [deleted]          test_A
```

## 第四步：本地仓库：test\_A分支上添加文件，修改文件内容，之后再回退

### 1. 本地创建test\_A.c 文件，内容如下：

```
# 本地创建test_A.c 文件
13919@WIN-892N3OKOR35 MINGW64 ~/Desktop/gitlearning (test_A)
$ git add test_A.c
# git status查看状态
13919@WIN-892N3OKOR35 MINGW64 ~/Desktop/gitlearning (test_A)
$ git status
On branch test_A
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   test_A.c
```

```
// test_A
// gcc hello.c -o hello
#include <stdio.h>

int main()
{
    printf("hello world\n");
    return 0;
}
```

2. 此时 `git checkout .` 恢复暂存区当前目录的所有文件到工作区,

```
# 恢复暂存区当前目录的所有文件到工作区
$ git checkout .
Updated 1 path from the index
# git status查看状态
$ git status
On branch test_A
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   test_A.c
```

3. 修改 test\_A.c 文件, 此时 test\_A.c 文件, 未执行过 add和commit操作

```
// test_A
// gcc hello.c -o hello
// test git checkout .
#include <stdio.h>

int main()
{
    printf("hello world\n");
    return 0;
}

$ git status
On branch test_A
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   test_A.c

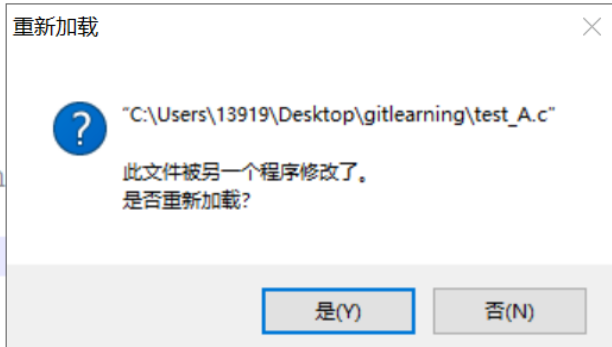
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   test_A.c
```

4. 执行 git checkout, 恢复暂存区当前目录的所有文件, 发现修改的内容回退至修改前

```
13919@WIN-892N3OKOR35 MINGW64 ~/Desktop/gitlearning (test_A)
$ git checkout .
Updated 1 path from the index
```

```
// test_A
// gcc hello.c -o hello
// test git checkout .
#include <stdio.h>

int main()
{
    printf("hello world\n");
    return 0;
}
```



## 第五步：本地仓库：test\_A分支上添加文件，add添加至暂存区后，再修改文件内容，之后再回退

1. 修改 test\_A.c 文件，git add添加至暂存区，之后 git reset，重置暂存区文件

```
13919@WIN-892N3OKOR35 MINGW64 ~/Desktop/gitlearning (test_A)
$ git add test_A.c

13919@WIN-892N3OKOR35 MINGW64 ~/Desktop/gitlearning (test_A)
$ git reset test_A
```

2. git reset test\_A 之后，发现test\_A.c 修改的内容并没有回退，只是重置了暂存区

```
13919@WIN-892N3OKOR35 MINGW64 ~/Desktop/gitlearning (test_A)
$ git status
On branch test_A
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    test_A.c
```

3. git reset --hard 重置暂存区与工作区，与上一次 commit 保持一致，发现未commit提交的文件 test\_A.c 文件不见了

```
13919@WIN-892N3OKOR35 MINGW64 ~/Desktop/gitlearning (test_A)
$ git add test_A.c

13919@WIN-892N3OKOR35 MINGW64 ~/Desktop/gitlearning (test_A)
$ git reset --hard test_A
HEAD is now at b2d1c96 Create readme.md
```

## 第六步：从develop分支拉取feature branch, 从feature branch拉person branch,那么特性分支改变，如何同步到个人分支上

1. 基于master分支创建develop分支

```
# 创建develop分支，并切换至develop分支
13919@WIN-892N3OKOR35 MINGW64 ~/Desktop/gitlearning (master)
$ git checkout -b develop
Switched to a new branch 'develop'
# develop分支添加develop.c文件，并推送至远端
```

- master分支上并没有develop分支添加develop.c文件


🔗 develop had recent pushes 6 minutes ago [Compare & pull request](#)

🔗 feature had recent pushes 3 minutes ago [Compare & pull request](#)

🔗 master ▾ 🔗 3 branches 🔒 0 tags [Go to file](#) [Add file ▾](#) [↓ Code ▾](#)

🌱 xixihaha123-xjh Create readme.md b2d1c96 1 hour ago 🕒 1 commits

📄 readme.md	Create readme.md	1 hour ago
-------------	------------------	------------

readme.md 

第一步: 创建远端仓库, 并且在远端仓库仓库, 添加 readme文件

- develop分支添加 develop.c文件

🔗 develop had recent pushes 7 minutes ago [Compare & pull request](#)


🔗 feature had recent pushes 5 minutes ago [Compare & pull request](#)

🔗 develop ▾ 🔗 3 branches 🔒 0 tags [Go to file](#) [Add file ▾](#) [↓ Code ▾](#)

This branch is 1 commit ahead of master. [🔗 Pull request](#) [± Compare](#)

🌱 xixihaha123-xjh add develop.c da20386 8 minutes ago 🕒 2 commits

📄 develop.c	add develop.c	8 minutes ago
📄 readme.md	Create readme.md	1 hour ago

readme.md 

第一步: 创建远端仓库, 并且在远端仓库仓库, 添加 readme文件

## 2. 基于develop分支创建feature 分支

```
13919@WIN-892N3OKOR35 MINGW64 ~/Desktop/gitlearning (develop)
$ git checkout -b feature develop # 基于develop分支创建feature 分支
Switched to a new branch 'feature'
```

## 3. 修改develop分支上的develop.c文件, 并且推送至远端

## 4. 同步develop分支内容至feature分支

```
13919@WIN-892N3OKOR35 MINGW64 ~/Desktop/gitlearning (develop)
$ git branch # 查看当前分支
```

```
checkout
* develop
feature
master

13919@WIN-892N3OKOR35 MINGW64 ~/Desktop/gitlearning (develop)
$ git checkout feature # 切换至feature分支
Switched to branch 'feature'
# 同步develop分支至feature分支
13919@WIN-892N3OKOR35 MINGW64 ~/Desktop/gitlearning (feature)
$ git pull origin develop # 拉取父分支的文件
From github.com:xixihaha123-xjh/gitlearning
* branch          develop    -> FETCH_HEAD
```

##### 5. 自动同步develop分支内容至feature分支

