

什么是Git?

Git是一套程序源代码的分布式版本管理系统，用于最初用于管理Linux核心代码的开发，后来被多个开源工程采用，现在已经成为互联网协作开发的标准的源代码管理软件。

官方网站

<https://git-scm.com/>

Git安装

<https://git-scm.com/downloads>

Git检查安装版本

```
$ git version
```

Git在线体验

<https://try.github.io/>

采用Git的云服务

- GitHub
- gitee(oschina)
- 等等...

理解版本管理的流程

• 代码开发流程

- 1.系统开发，编写代码
- 2.提交代码给Git本地库
- 3.将代码提交到Git远程库，分享给团队其他人
- 4.从远程库获取最新代码
- 5.继续修改编写代码
- 6.重复第二步及以后的操作

• Git基础概念

- 1.本地工作文件夹
- 2.Git索引区 (Stage)
- 3.Git库 (Repository)

local: 本地库

remote: 远程库 (服务器端)

Git初始设定

• 建立一个Git库

```
$ git init
```

初始化仓库

• 设置基础信息

```
$ git config -l #查看配置信息  
$ git config --global user.name "" #全局设置你的用户名  
$ git config --global user.email "" #全局设置你的邮箱
```

配置信息

• 方便的命令

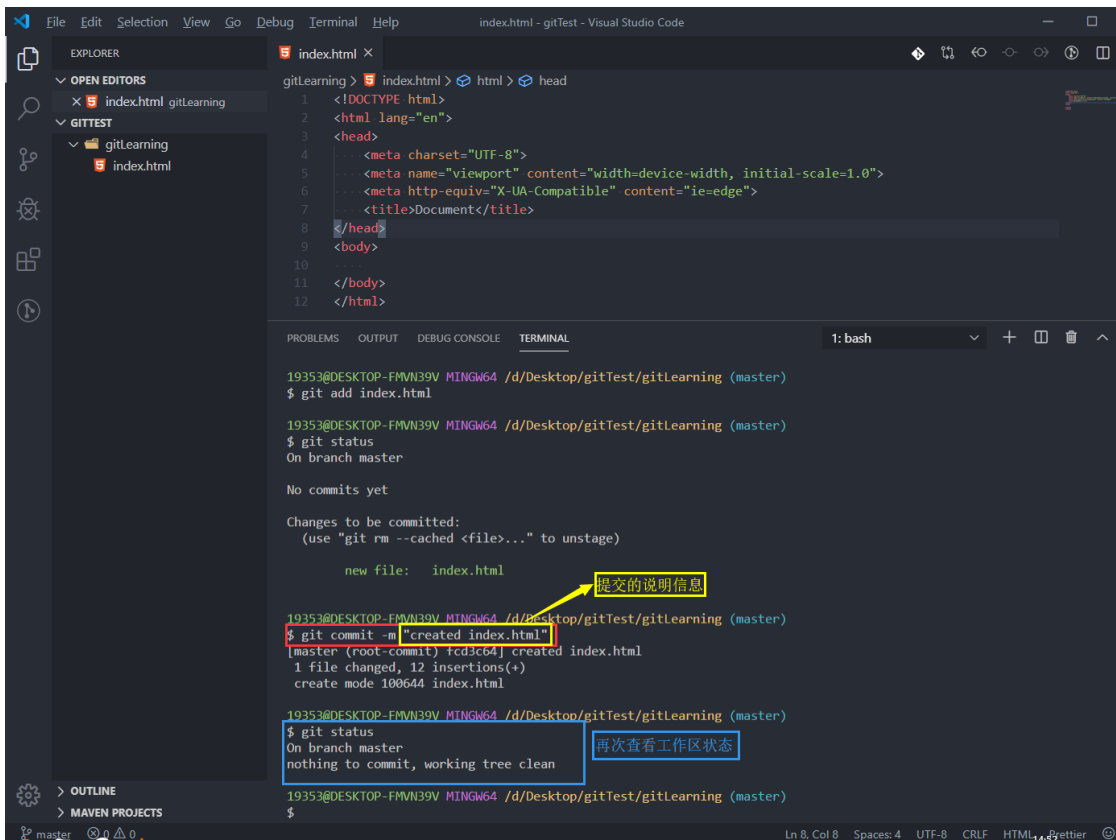
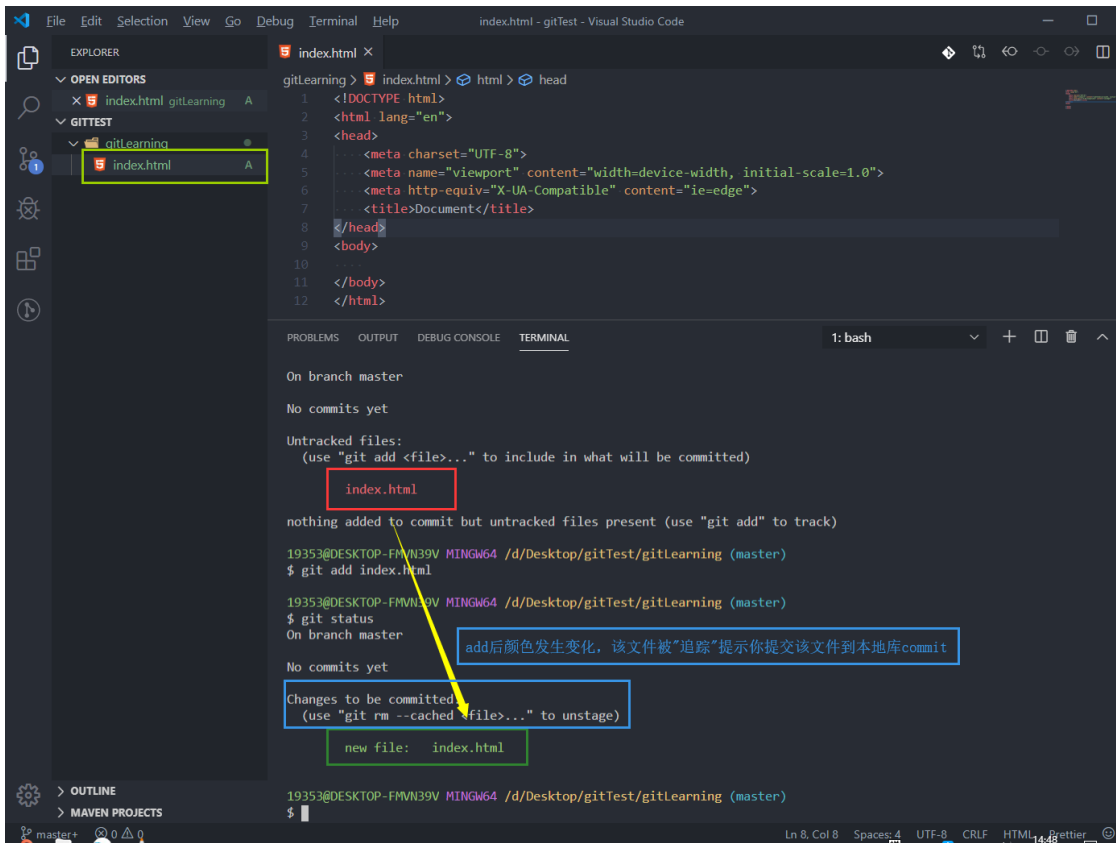
```
$ git config --help
```

Windows用户浏览器弹出

第一次提交

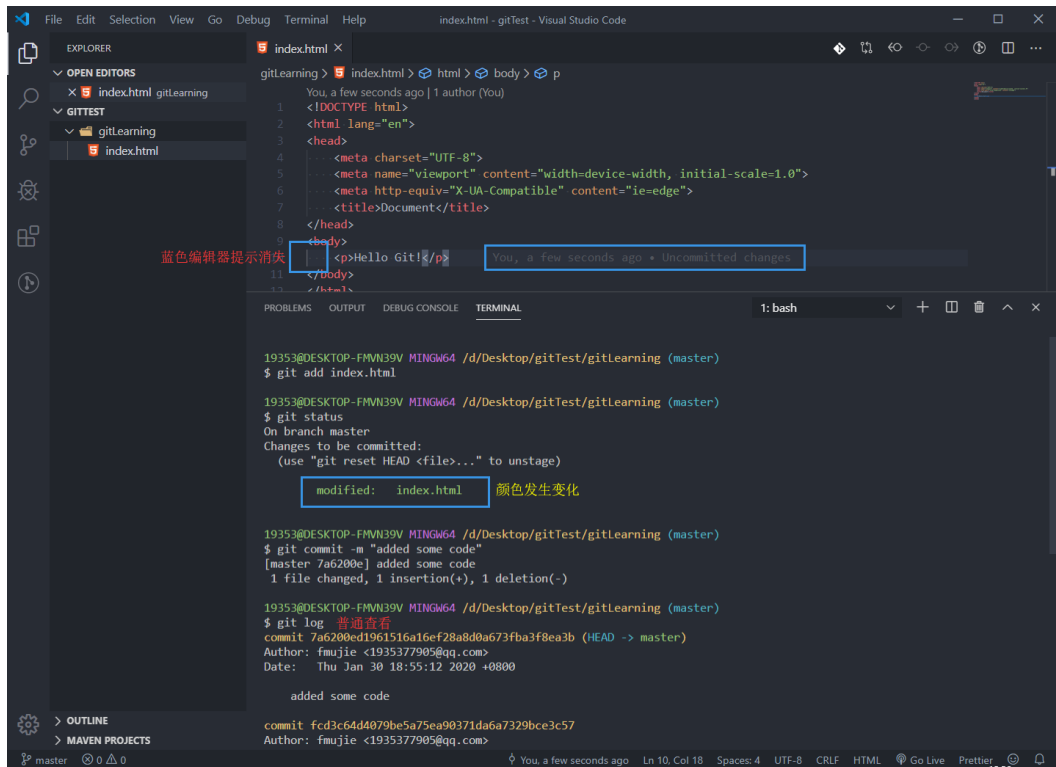
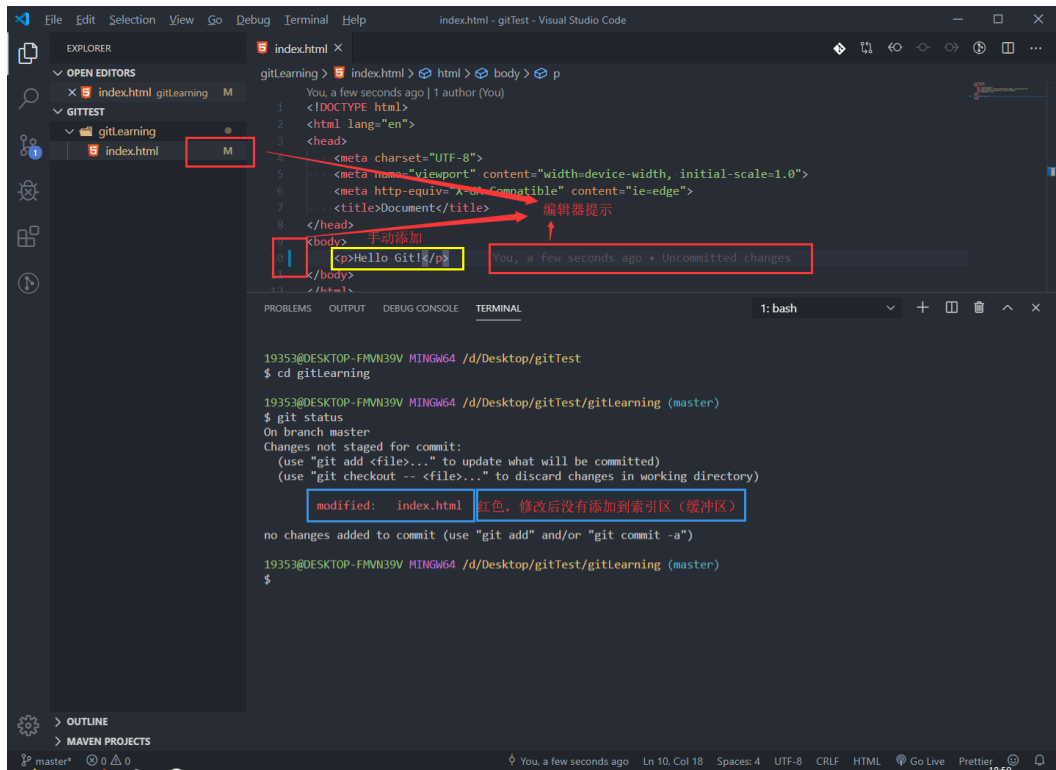
- 建立文件（本地工作文件夹）
- 追加文件（索引区）
- 提交文件（本地库）

```
index.html - gitTest - Visual Studio Code  
EXPLORER  
gitLearning > index.html  
index.html  
GITTEST  
gitLearning  
index.html 手动创建index.html  
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL  
1: bash  
19353@DESKTOP-FMWN39V MINGW64 /d/Desktop/gitTest  
$ mkdir gitLearning 在当前文件夹内创建一个名为gitLearning的文件夹  
19353@DESKTOP-FMWN39V MINGW64 /d/Desktop/gitTest  
$ cd gitLearning cd 进入gitLearning文件夹  
19353@DESKTOP-FMWN39V MINGW64 /d/Desktop/gitTest/gitLearning  
$ git init 初始化仓库  
Initialized empty Git repository in D:/Desktop/gitTest/gitLearning/.git/  
19353@DESKTOP-FMWN39V MINGW64 /d/Desktop/gitTest/gitLearning (master)  
$ git status 查看状态  
On branch master  
  
No commits yet  
  
Untracked files:  
(use "git add <file>.." to include in what will be committed)  
index.html 该文件未被“追踪”使用git add添加至缓冲区（索引区）  
  
nothing added to commit but untracked files present (use "git add" to track)  
19353@DESKTOP-FMWN39V MINGW64 /d/Desktop/gitTest/gitLearning (master)  
$
```



查看提交履历

```
$ git log
$ git log --oneline
$ git log -n
```



```
19353@DESKTOP-FMVN39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$ git log --oneline 一行显示, 简洁
7a6200e (HEAD -> master) added some code
fcd3c64 created index.html
```

```
19353@DESKTOP-FMVN39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$ git log -2 查看最近提交的几(2)次纪律
commit 7a6200ed1961516a16ef28a8d0a673fba3f8ea3b (HEAD -> master)
Author: fmujie <1935377905@qq.com>
Date: Thu Jan 30 18:55:12 2020 +0800

    added some code

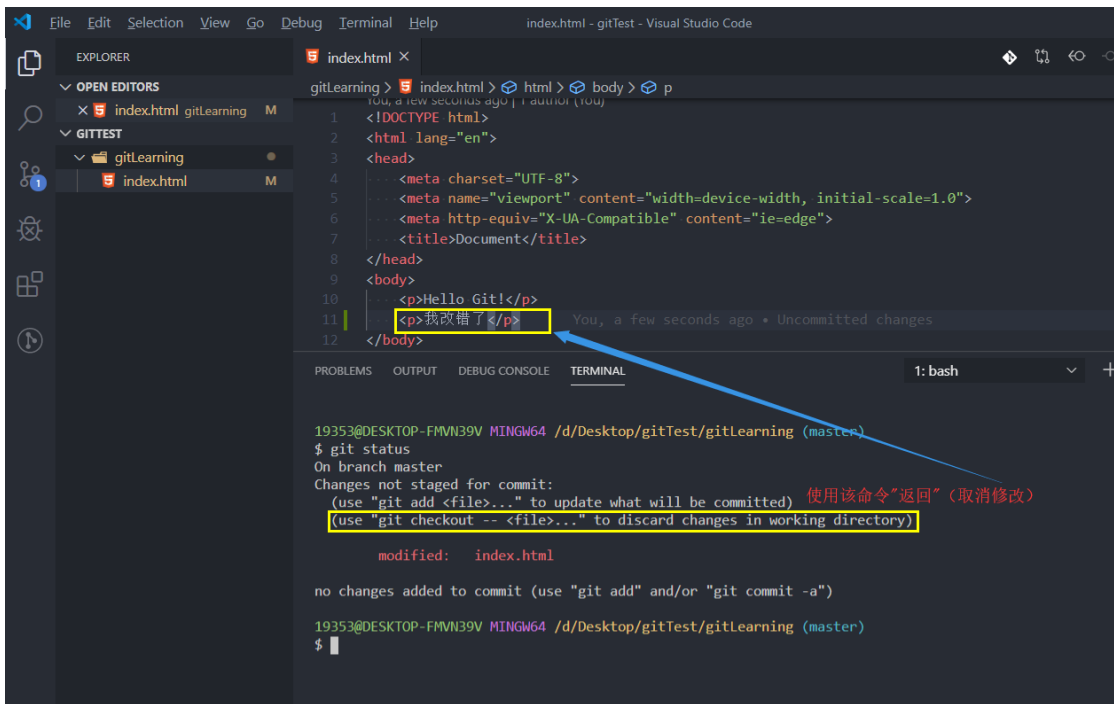
commit fcd3c64d4079be5a75ea90371da6a7329bce3c57
Author: fmujie <1935377905@qq.com>
Date: Thu Jan 30 14:52:05 2020 +0800

    created index.html
```

把握Git状态

```
$ git status
```

```
$ git checkout -- [file] #切换版本
```



The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left shows the project structure with 'index.html' selected. The main editor window displays the content of 'index.html', which includes a paragraph: `<p>Hello Git!</p>` and an uncommitted change: `<p>我改错了</p>`. The Terminal window at the bottom shows the output of the `git status` command, indicating that there are uncommitted changes in 'index.html'. A blue arrow points from the uncommitted change in the code editor to the corresponding line in the terminal output.

```
19353@DESKTOP-FMVN39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed) 使用该命令"返回" (取消修改)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   index.html

no changes added to commit (use "git add" and/or "git commit -a")

19353@DESKTOP-FMVN39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$
```

```
File Edit Selection View Go Debug Terminal Help index.html - gitTest - Visual Studio Code
EXPLORER
index.html X
gitLearning > index.html > html > body
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <meta http-equiv="X-UA-Compatible" content="ie=edge">
7   <title>Document</title>
8 </head>
9 <body>
10   <p>Hello Git!</p>
11 </body>
12 </html>
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: bash
19353@DESKTOP-FMWN39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

   modified:   index.html

no changes added to commit (use "git add" and/or "git commit -a")
19353@DESKTOP-FMWN39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$ git checkout -- index.html 该命令运行后可以看见刚才的更改已经移除
19353@DESKTOP-FMWN39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$ git status
On branch master
nothing to commit, working tree clean 查看工作区状态, clean!
```

```
File Edit Selection View Go Debug Terminal Help index.html - gitTest - Visual Studio Code
EXPLORER
index.html X
gitLearning > index.html > html > body > p
1 You, a few seconds ago | 1 author (You)
2 <!DOCTYPE html>
3 <html lang="en">
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <meta http-equiv="X-UA-Compatible" content="ie=edge">
8   <title>Document</title>
9 </head>
10 <body>
11   <p>Hello Git!</p>
12   <p>我改错了, 而且加到索引区(缓冲区)</p>
13 </body>
14 </html>
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: bash
19353@DESKTOP-FMWN39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

   modified:   index.html

no changes added to commit (use "git add" and/or "git commit -a")
19353@DESKTOP-FMWN39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$ git add index.html 添加到索引区(缓冲区)
19353@DESKTOP-FMWN39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage) 这时候撤销修改需要先将文件撤出索引区(缓冲区), 再撤销修改

   modified:   index.html

19353@DESKTOP-FMWN39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$
```

```
gitLearning > index.html > html > body > p
1 You, a few seconds ago | 1 author (You)
2 <!DOCTYPE html>
3 <html lang="en">
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <meta http-equiv="X-UA-Compatible" content="ie=edge">
8   <title>Document</title>
9 </head>
10 <body>
11   <p>Hello Git!</p>
12 </body>
13 </html>

19353@DESKTOP-FMVN39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

   modified:   index.html

no changes added to commit (use "git add" and/or "git commit -a")
19353@DESKTOP-FMVN39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$ git add index.html
添加到索引区(缓冲区)
19353@DESKTOP-FMVN39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)
   modified:   index.html
这时候撤销修改需要先将文件撤出索引区(缓冲区), 再撤销修改
19353@DESKTOP-FMVN39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$
```

```
gitLearning > index.html > html > body
1 You, 29 minutes ago | 1 author (You)
2 <!DOCTYPE html>
3 <html lang="en">
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <meta http-equiv="X-UA-Compatible" content="ie=edge">
8   <title>Document</title>
9 </head>
10 <body>
11   <p>Hello Git!</p>
12 </body>
13 </html>

19353@DESKTOP-FMVN39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$ git checkout -- index.html
可以看到更改已经消失
再次查看工作区状态, clean!
19353@DESKTOP-FMVN39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$ git status
On branch master
nothing to commit, working tree clean
19353@DESKTOP-FMVN39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$
```

比较修改内容

```
$ git diff [--cached] #缓冲区
```

```

19353@DESKTOP-FMWN39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$ git status 查看状态，显示文件已经发生更改
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   index.html

no changes added to commit (use "git add" and/or "git commit -a")

19353@DESKTOP-FMWN39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$ git diff 使用git diff查看更改的地方
diff --git a/index.html b/index.html
index 2c41c9e..9f8961b 100644
--- a/index.html
+++ b/index.html
@@ -8,5 +8,6 @@
 </head>
 <body>
+ <p>Hello Git!</p>
+ <p>我修改了此文件</p>
 </body>
 </html>
\ No newline at end of file

```

```

19353@DESKTOP-FMWN39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$ git add index.html 从工作区添加到索引区(缓冲区)

19353@DESKTOP-FMWN39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   index.html 确定已经添加

19353@DESKTOP-FMWN39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$ git diff git diff是将工作区的进行比较，要比较索引区(缓冲区)的则需要加参数，指定索引区(缓冲区)和数据库的进行比较
diff --git a/index.html b/index.html
index 2c41c9e..9f8961b 100644
--- a/index.html
+++ b/index.html
@@ -8,5 +8,6 @@
 </head>
 <body>
+ <p>Hello Git!</p>
+ <p>我修改了此文件</p>
 </body>
 </html>
\ No newline at end of file

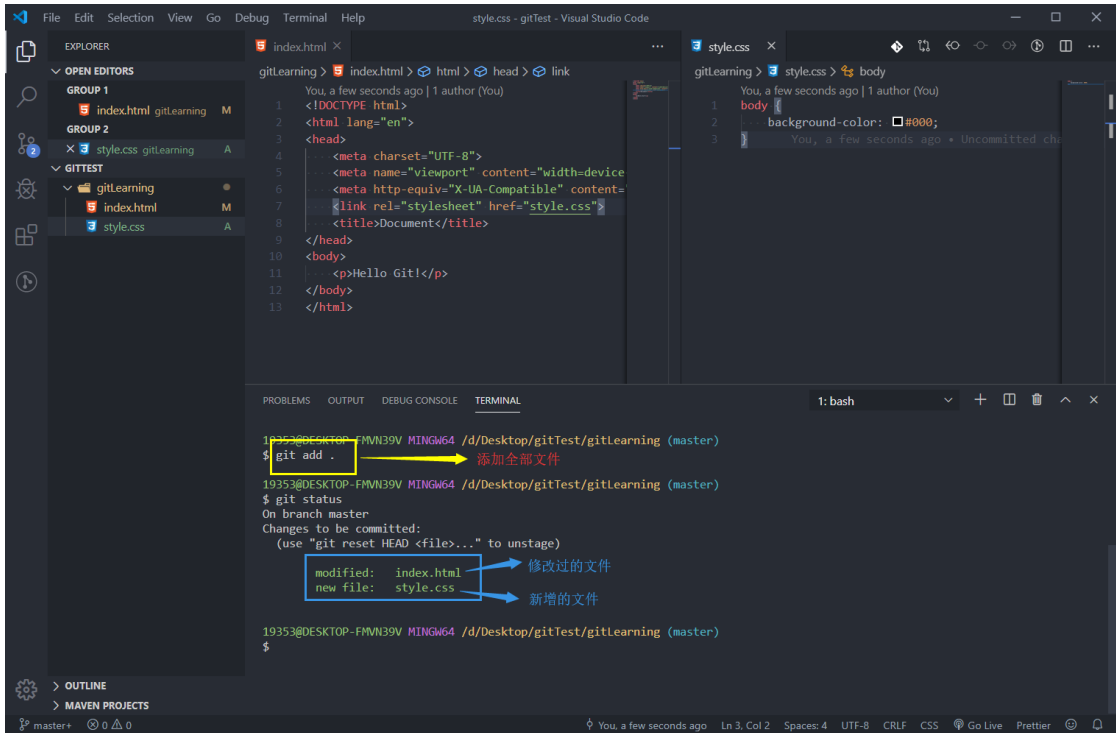
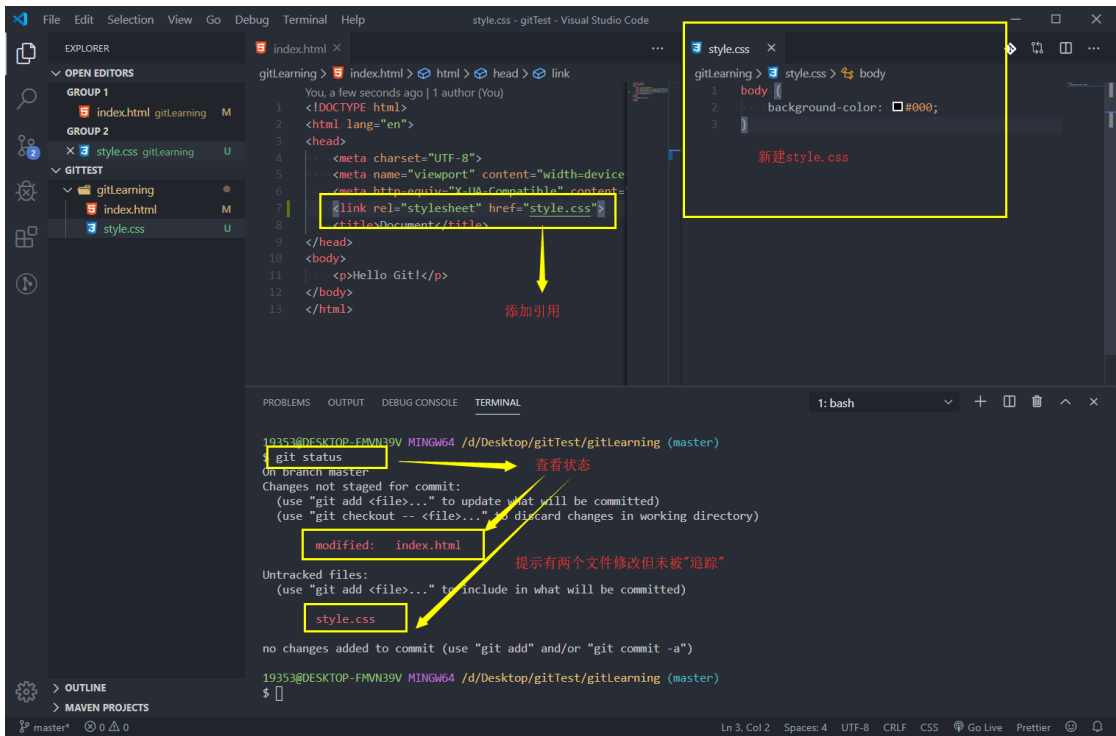
```

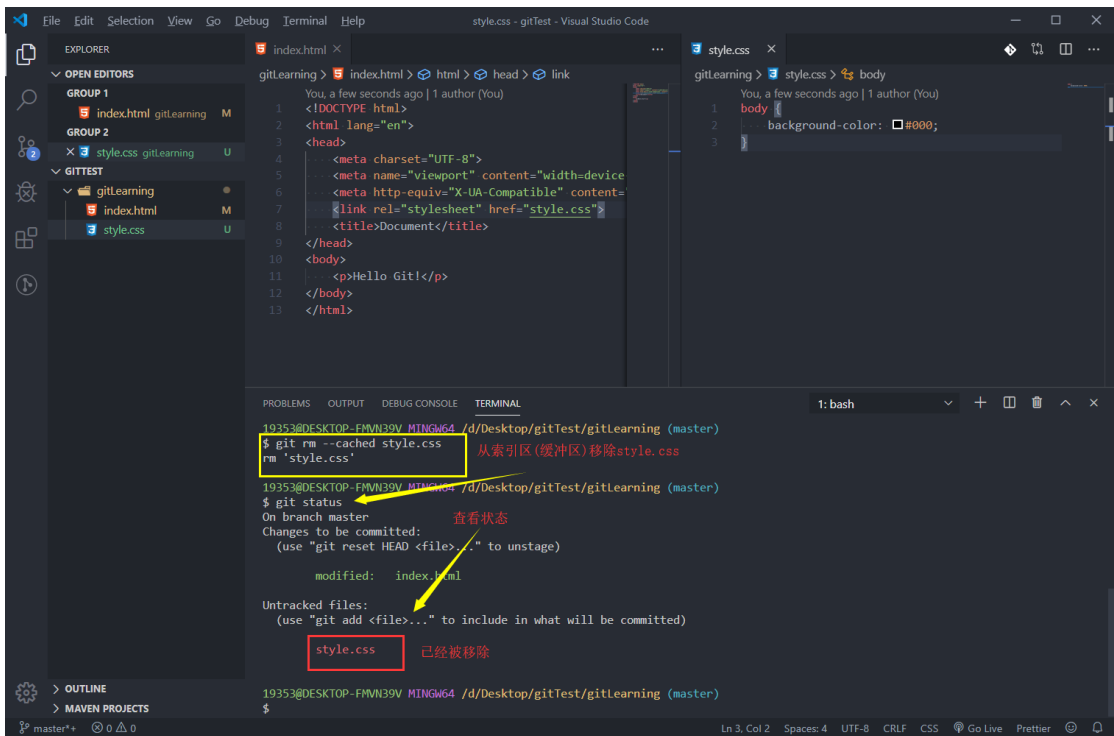
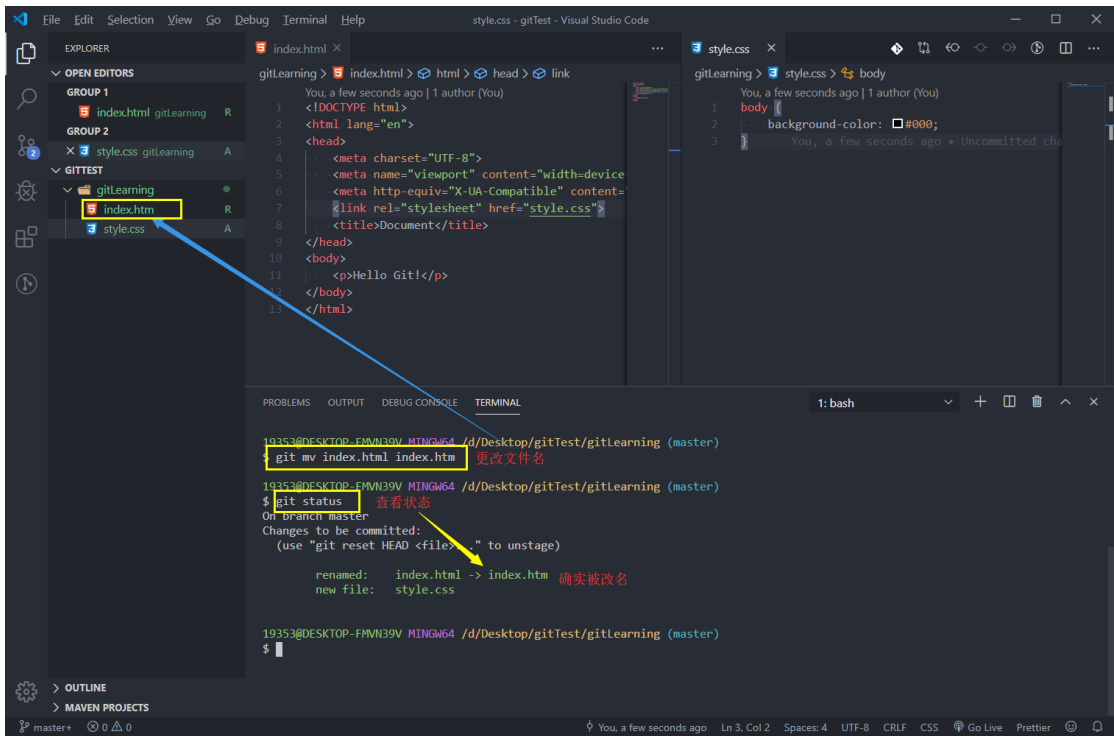
Git文件操作

```

$ git add [file1 file2 ...]
$ git add . #一次性添加全部文件包括当前目录
$ git rm #删除
$ git mv #文件改名(文件移动)

```



Git忽略管理

设置Git忽略的文件，这些文件不参与Git库的提交和管理。（例如 [node_modules]文件夹）

- gitignore

新建 test.tmp 临时文件

新建 .gitignore 文件

```

19353@DESKTOP-FMWN39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$ git status
On branch master
Untracked files:
(use "git add <file>..." to include in what will be committed)

.gitignore
test.tmp
查看状态, 这两个文件未被追踪

nothing added to commit but untracked files present (use "git add" to track)
19353@DESKTOP-FMWN39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$

```

添加一行 *.tmp (所有后缀为 .tmp 的文件)

再次查看状态

```

19353@DESKTOP-FMWN39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$ git status
On branch master
Untracked files:
(use "git add <file>..." to include in what will be committed)

.gitignore
test.tmp
仅仅显示 .gitignore 未被追踪
test.tmp 未显示, 则说明 git 版本管理忽略了后缀为 .tmp 文件的管理

nothing added to commit but untracked files present (use "git add" to track)

19353@DESKTOP-FMWN39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$ git add .
添加

19353@DESKTOP-FMWN39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$ git commit -m "added .gitignore"
提交
[master 5a01108] added .gitignore
1 file changed, 2 insertions(+)
create mode 100644 .gitignore

19353@DESKTOP-FMWN39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$

```

更新最近的提交

- git commit -m "commit message"
- git commit --amend
- git commit -am "commit message" [--amend]

```
gitLearning > index.html > html > body > p
You, a few seconds ago | 1 author (You)
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 <meta http-equiv="X-UA-Compatible" content="ie=edge">
7 <link rel="stylesheet" href="style.css">
8 <title>Document</title>
9 </head>
10 <body>
11 <p>Hello Git!</p>
12 <p>我修改了一些东西</p> You, a few seconds ago • Uncommitted changes
13 </body>
14 </html>

19353@DESKTOP-FMVN39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$ git add .

19353@DESKTOP-FMVN39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$ git commit -m "modified"
[master b7e0c9a] modified
1 file changed, 1 insertion(+)

19353@DESKTOP-FMVN39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$ git log -1
commit b7e0c9a505a507444ec58d4a36ef09d56971feaa (HEAD -> master)
Author: fmujiie <1935377905@qq.com>
Date: Fri Jan 31 15:10:23 2020 +0800

    modified

19353@DESKTOP-FMVN39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$
```

平常的修改平常的提交

```
gitLearning > index.html > html > body > p
6 <meta http-equiv="X-UA-Compatible" content="ie=edge">
7 <link rel="stylesheet" href="style.css">
8 <title>Document</title>
9 </head>
10 <body>
11 <p>Hello Git!</p>
12 <p>我修改了一些东西 我漏加了一个感叹号"!"</p> You, a few seconds ago • Uncommitted changes
13 </body>
14 </html>

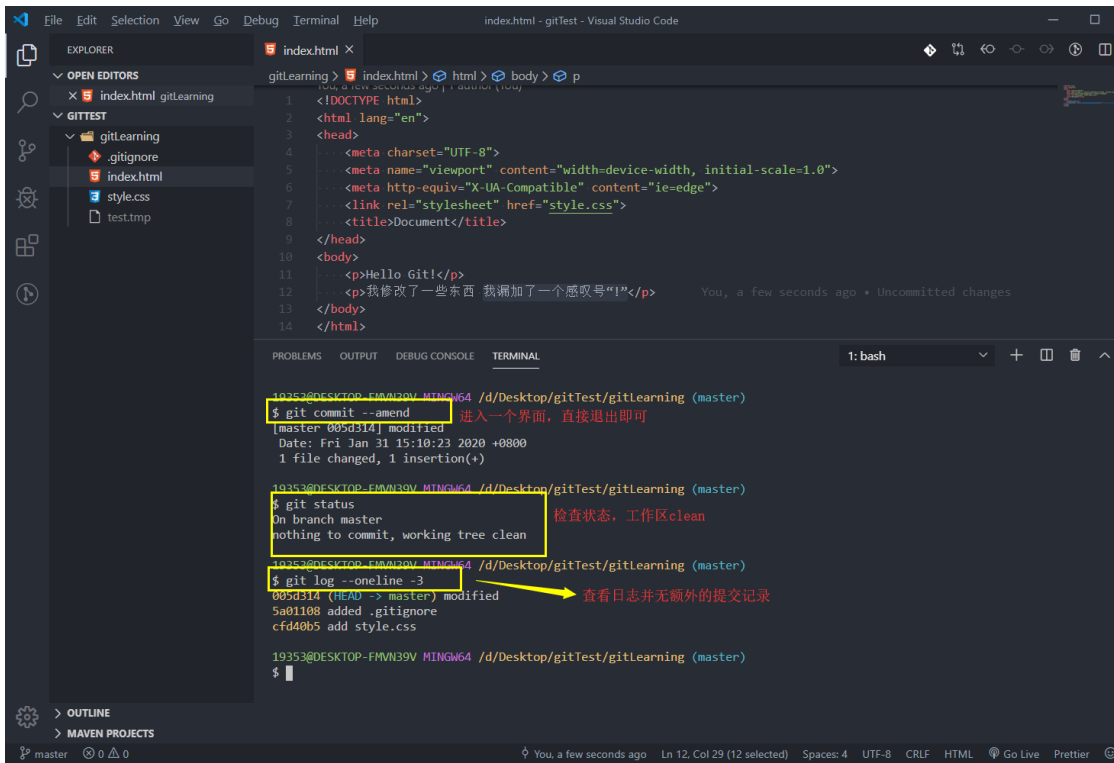
19353@DESKTOP-FMVN39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$ git add .

19353@DESKTOP-FMVN39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$ git log --oneline -3
b7e0c9a (HEAD -> master) modified
5a01108 added .gitignore
cfd40b5 add style.css

19353@DESKTOP-FMVN39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$
```

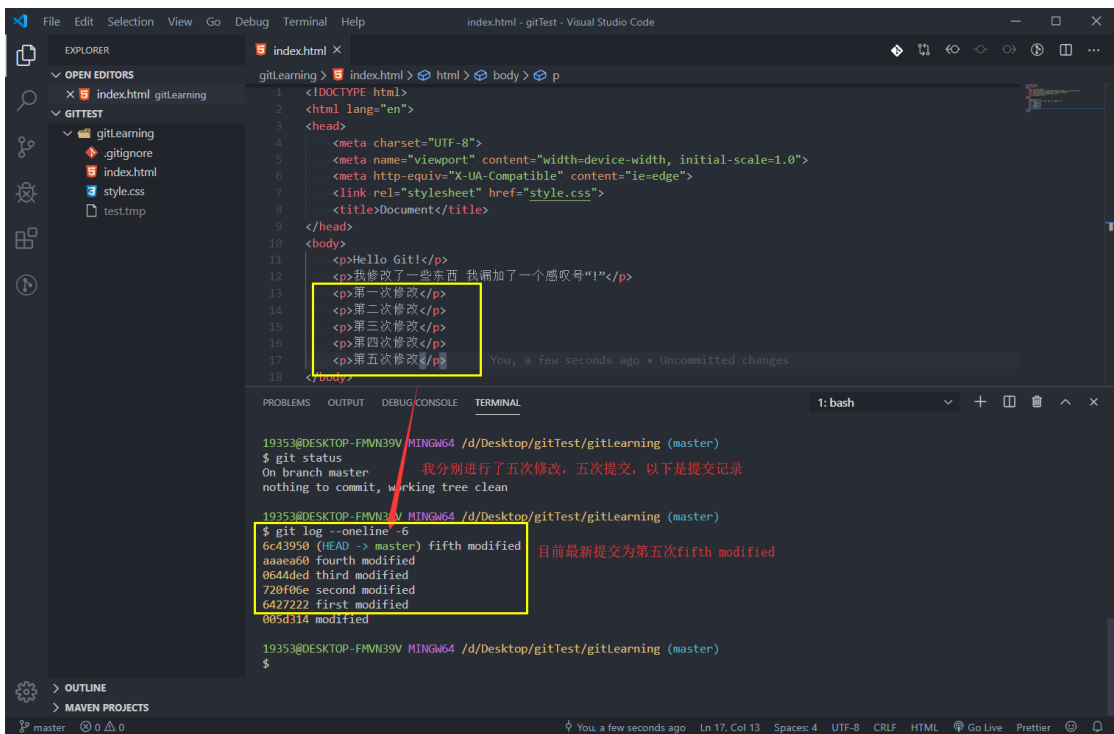
我漏加了一个感叹号"!"

现在我想将修改追加到上次的提交里，
不用再额外提交一次



返回过去1

- git reset --hard HEAD 回到数据库中最新的版本
- git reset --hard HEAD~
- git reset --hard HEAD~n



The screenshot shows the Visual Studio Code interface with a terminal window open. The terminal displays the following commands and output:

```
19353@DESKTOP-FMWN39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   index.html

no changes added to commit (use "git add" and/or "git commit -a")

19353@DESKTOP-FMWN39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$ git reset --hard HEAD
HEAD is now at 6c43950 fifth modified
使用命令退回到数据库中最新版本第五次修改

19353@DESKTOP-FMWN39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$
```

In the editor, the `index.html` file is open, showing HTML content. A yellow box highlights the sixth paragraph: `<p>在这里我进行了第六次修改</p>`. A red arrow points from this box to the `modified: index.html` line in the terminal output.

The screenshot shows the Visual Studio Code interface with a terminal window open. The terminal displays the following commands and output:

```
19353@DESKTOP-FMWN39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$ git reset --hard HEAD~3
HEAD is now at 720f06e second modified

19353@DESKTOP-FMWN39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$ git status
On branch master
nothing to commit, working tree clean

19353@DESKTOP-FMWN39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$
```

In the editor, the `index.html` file is open, showing HTML content. A yellow box highlights the second paragraph: `<p>我修改了一些东西 我添加了一个感叹号"!"</p>`. A red arrow points from this box to the `HEAD is now at 720f06e second modified` line in the terminal output.

```

19353@DESKTOP-FMVM39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$ git log --oneline
720f06e (HEAD -> master) second modified
6427222 first modified
005d314 modified
5a01108 added .gitignore
cfd40b5 add style.css
7a6200e added some code
fcd3c64 created index.html

19353@DESKTOP-FMVM39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$ git reset --hard HEAD~2
HEAD is now at 005d314 modified
19353@DESKTOP-FMVM39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$

```

```

19353@DESKTOP-FMVM39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$ git log --oneline -5
005d314 (HEAD -> master) modified
5a01108 added .gitignore
cfd40b5 add style.css
7a6200e added some code
fcd3c64 created index.html

19353@DESKTOP-FMVM39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$ git reset --hard 720f06e
HEAD is now at 720f06e second modified

19353@DESKTOP-FMVM39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$ git log --oneline
720f06e (HEAD -> master) second modified
6427222 first modified
005d314 modified
5a01108 added .gitignore
cfd40b5 add style.css
7a6200e added some code
fcd3c64 created index.html

19353@DESKTOP-FMVM39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$

```

回到过去 2

- git reflog [-n num]
- git reset --hard [commit_id]

返回过去之后, 通过git reflog命令找到现在的位置(commit_id), 再从过去返回回来

```

19353@DESKTOP-FMWN39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$ git status
On branch master
nothing to commit, working tree clean

19353@DESKTOP-FMWN39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$ git log --oneline
720f06e (HEAD -> master) second modified
6427222 first modified
005d314 modified
5a01108 added .gitignore
cfd40b5 add style.css
7a6200e added some code
fcd3c64 created index.html
19353@DESKTOP-FMWN39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$

```

现在看记录是在第二次修改，但是我想回到第五次修改的那一个版本，但是版本id我又忘了咋办？见下图

```

19353@DESKTOP-FMWN39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$ git reflog
720f06e (HEAD -> master) HEAD@{0}: reset: moving to 720f06e
005d314 HEAD@{1}: reset: moving to HEAD-2
720f06e (HEAD -> master) HEAD@{2}: reset: moving to HEAD-3
6c43950 HEAD@{3}: reset: moving to HEAD
6c43950 HEAD@{4}: commit: fifth modified
aaaea60 HEAD@{5}: commit: fourth modified
0644ded HEAD@{6}: commit: third modified
720f06e (HEAD -> master) HEAD@{7}: commit: second modified
6427222 HEAD@{8}: commit: first modified
005d314 HEAD@{9}: commit (amend): modified
b7e0c9a HEAD@{10}: commit: modified
5a01108 HEAD@{11}: commit: added .gitignore
cfd40b5 HEAD@{12}: commit: add style.css
7a6200e HEAD@{13}: commit: added some code
fcd3c64 HEAD@{14}: commit (initial): created index.html
19353@DESKTOP-FMWN39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$ git reset --hard 6c43950
HEAD is now at 6c43950 fifth modified
19353@DESKTOP-FMWN39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$

```

成功返回

可以说明，你提交的代码并没有消失，提交记录仅仅是HEAD指针发生了变化，你提交的每次源代码都在数据库里保存着!!!

该命令查看全部记录，后可加-n控制条数

这里可以看出第五次修改的版本id

使用分支

Git分支功能对于项目开发中的团队合作有着非常重要的作用，同时对于生产环境的更新管理也起着不可替代的作用，是Git最重要的功能。

在项目开始前，应该首先对Git分支的管理有一个明确的规划，明确每个分支的功能和担当者，这样才会保证项目正常推进，不至于陷入混乱。

- git branch [name]
- git checkout branch_name


```
gitLearning > index.html > html > body > p
1 <meta charset="utf-8">
2 <meta name="viewport" content="width=device-width, initial-scale=1.0">
3 <meta http-equiv="X-UA-Compatible" content="ie=edge">
4 <link rel="stylesheet" href="style.css">
5 <title>Document</title>
6 </head>
7 <body>
8 <p>我是主分支/p</p>
9 </body>
10 </html>
```

```
19353@DESKTOP-FMVB39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$ git add .
19353@DESKTOP-FMVB39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$ git commit -m "master branch"
[master 7e8581a] master branch
1 file changed, 1 insertion(+), 7 deletions(-)
19353@DESKTOP-FMVB39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$ git branch
* master
19353@DESKTOP-FMVB39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$ git branch dev
19353@DESKTOP-FMVB39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$ git branch
* master
  dev
19353@DESKTOP-FMVB39V MINGW64 /d/Desktop/gitTest/gitLearning (dev)
$ git checkout dev
Switched to branch 'dev'
19353@DESKTOP-FMVB39V MINGW64 /d/Desktop/gitTest/gitLearning (dev)
$ git branch
* dev
  master
```

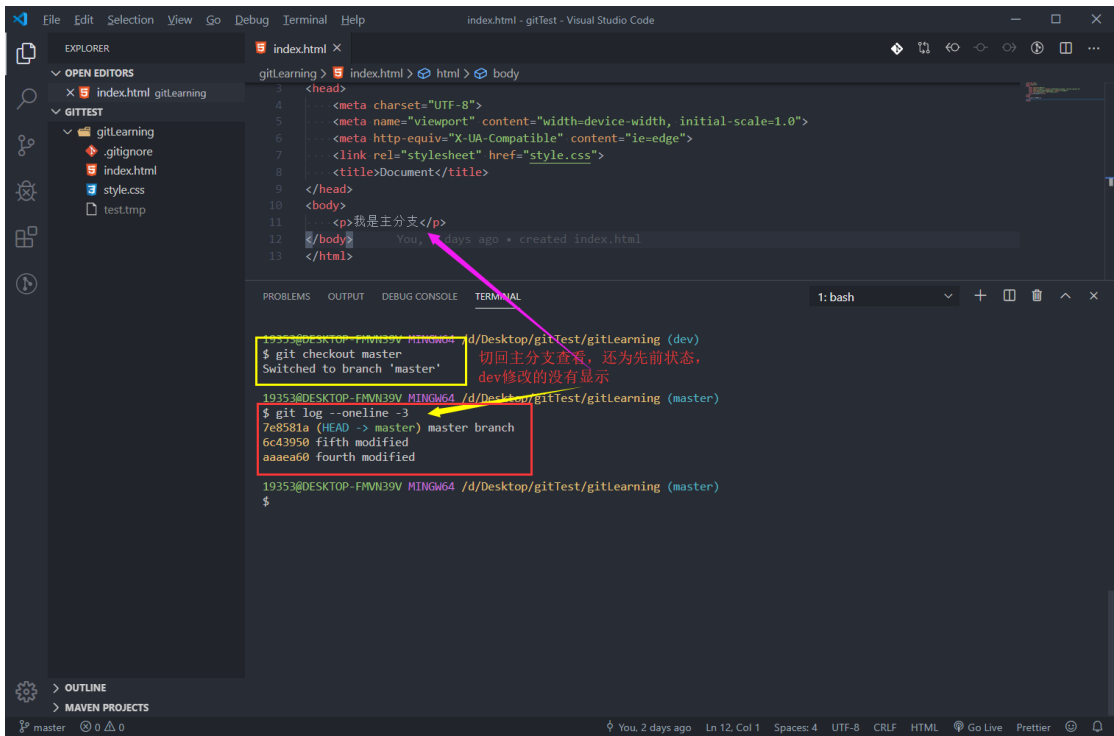
在分支上修改了内容，正常提交到主分支
查看分支确定当前是主分支
建立新分支dev
查看分支dev建立成功与否
切换到dev分支，与当前主分支同步
查看是否切换成功 看*标

```
gitLearning > index.html > html > body > p
1 <head>
2 <meta charset="UTF-8">
3 <meta name="viewport" content="width=device-width, initial-scale=1.0">
4 <meta http-equiv="X-UA-Compatible" content="ie=edge">
5 <link rel="stylesheet" href="style.css">
6 <title>Document</title>
7 </head>
8 <body>
9 <p>我是主分支/p</p>
10 <p>新增一条在dev分支上的记录/p</p>
11 </body>
12 </html>
```

```
19353@DESKTOP-FMVB39V MINGW64 /d/Desktop/gitTest/gitLearning (dev)
$ git add .
19353@DESKTOP-FMVB39V MINGW64 /d/Desktop/gitTest/gitLearning (dev)
$ git commit -m "dev branch"
[dev 66de09c] dev branch
1 file changed, 1 insertion(+)
```

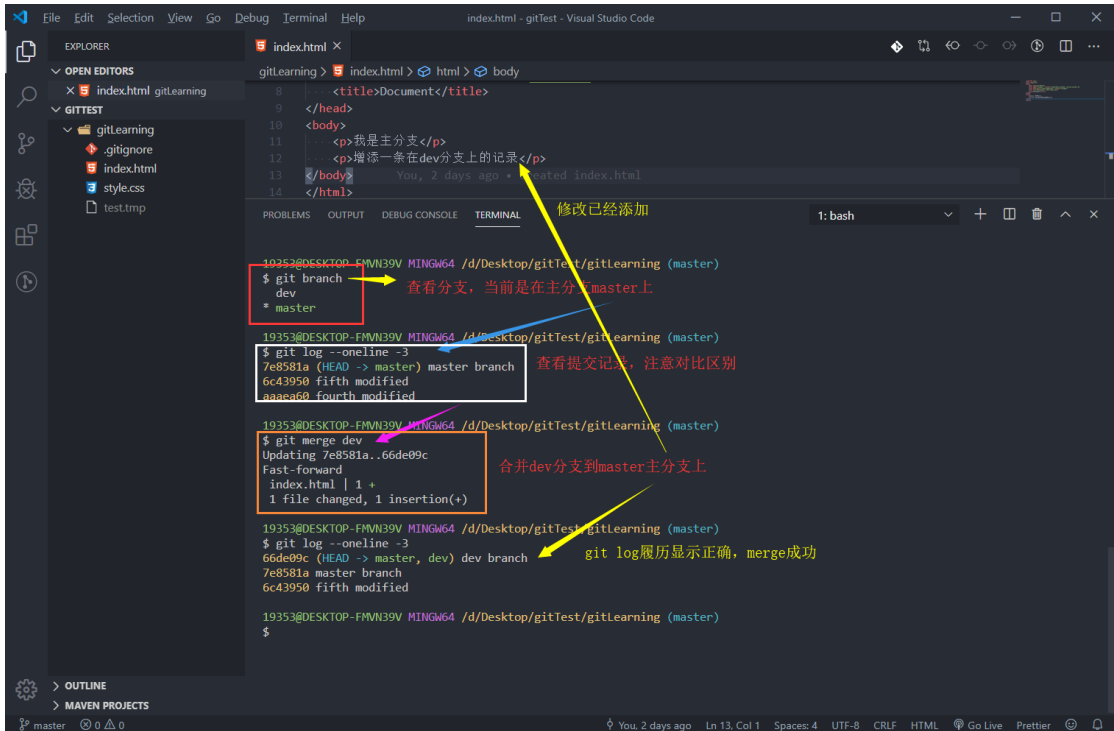
```
19353@DESKTOP-FMVB39V MINGW64 /d/Desktop/gitTest/gitLearning (dev)
$ git log --oneline -3
66de09c (HEAD -> dev) dev branch
7e8581a (master) master branch
6c43950 fifth modified
```

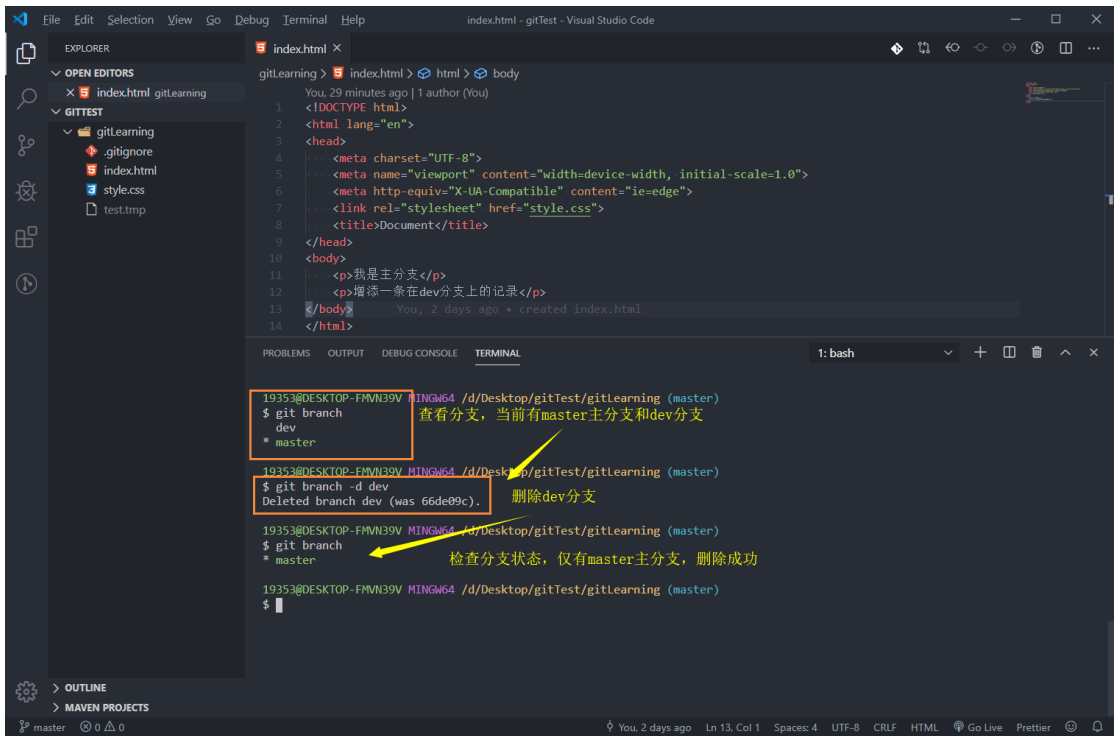
正常提交
查看提交记录，当前显示在dev分支上
提交



合并分支

- git merge 合并分支
- git branch -d [name] 删掉分支

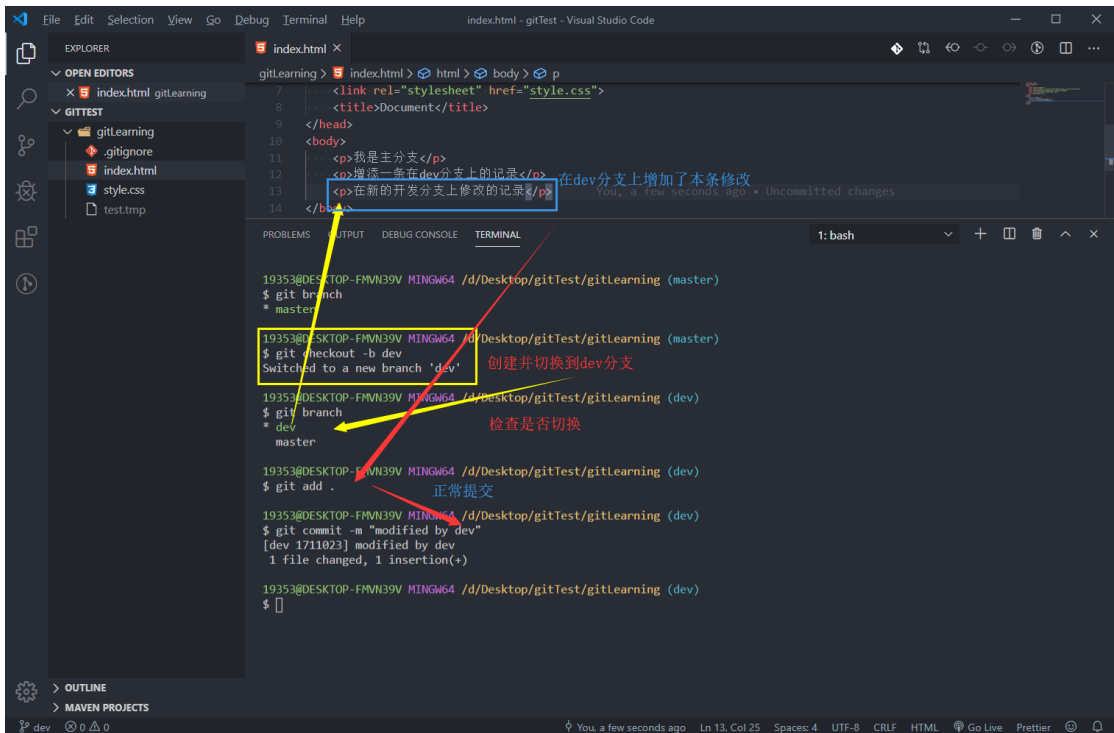


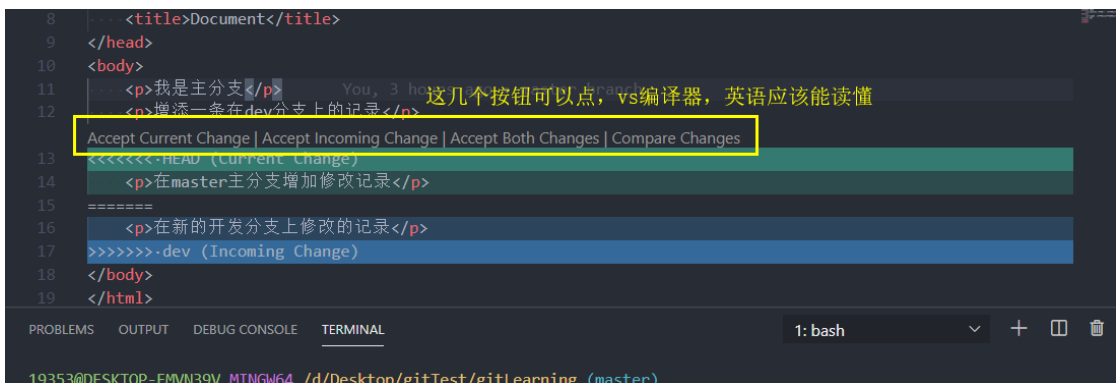
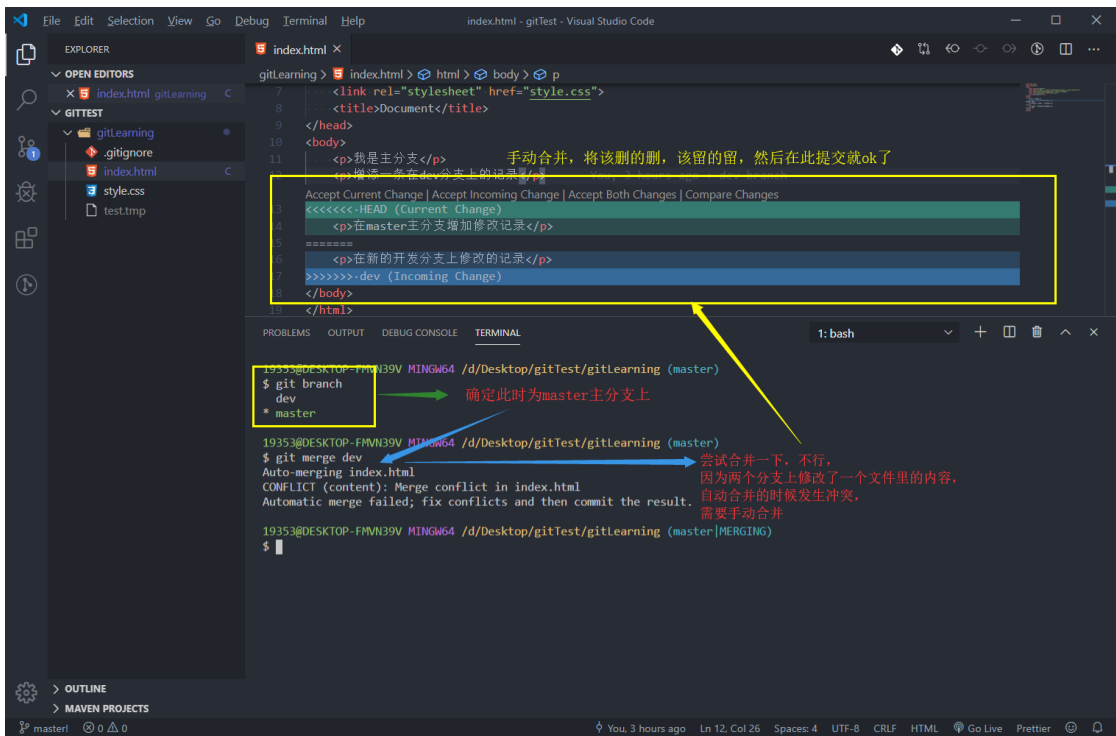
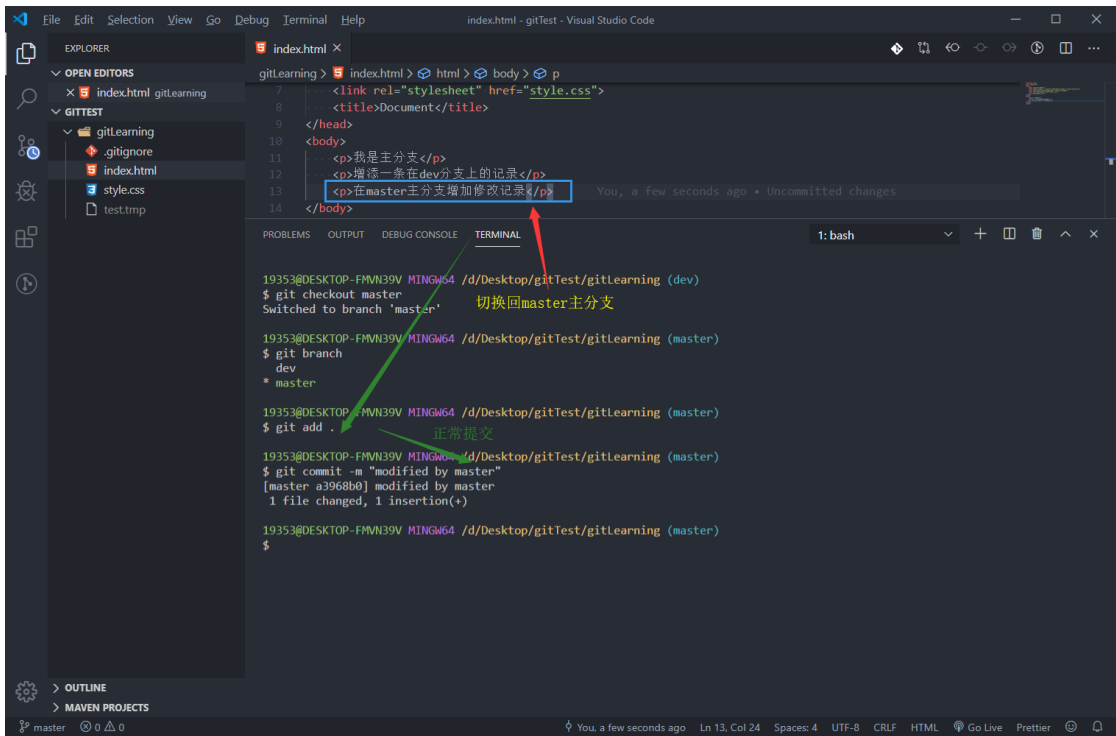


分支冲突

当团队中多人同时编辑一个文件的时候，难免出现源代码的编辑合并出现冲突的问题，那么我们应该如何解决呢？

- 源代码冲突场景
- git checkout -b [branch_name]





The image shows a Visual Studio Code window with a file explorer on the left and a terminal at the bottom. The editor displays the content of `index.html`, which has a merge conflict in the `body` section. The conflicting lines are highlighted in yellow:

```
11 <p>我是主分支</p>
12 <p>增添一个在dev分支上的记录</p>
13 <p>在master主分支增加修改记录</p>
14 <p>在新的开发分支上修改的记录</p>
```

The terminal window shows the following sequence of commands and outputs:

```
19353@DESKTOP-FMWN39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$ git branch
dev
* master

19353@DESKTOP-FMWN39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$ git merge dev
Auto-merging index.html
CONFLICT (content): Merge conflict in index.html
Automatic merge failed; fix conflicts and then commit the result.

19353@DESKTOP-FMWN39V MINGW64 /d/Desktop/gitTest/gitLearning (master|MERGING)
$ git add .

19353@DESKTOP-FMWN39V MINGW64 /d/Desktop/gitTest/gitLearning (master|MERGING)
$ git commit -m "merged by yourself"
[master 159408d] merged by yourself

19353@DESKTOP-FMWN39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$ git log --oneline -3
159408d (HEAD -> master) merged by yourself
a3968b0 modified by master
1711023 (dev) modified by dev

19353@DESKTOP-FMWN39V MINGW64 /d/Desktop/gitTest/gitLearning (master)
$
```

A yellow box highlights the terminal output from the `git add .` command onwards. A yellow arrow points from the text "手动合并之后再提交一下就ok了" (Manually merge and then submit, it's OK) to the `git add .` command.