

1.安装

- Windows安装
 - <https://git-scm.com/book/zh/v2>
 - <https://git-scm.com/download/win> 安装.exe程序
 - cmd控制台输入: git --version
 - mac安装
 - <https://zhuanlan.zhihu.com/p/90508170>
 - <http://ping.chinaz.com/github.com>
-

2.最小配置

最小配置：配置user.name和user.email

```
git config --global user.name 'your_name'
git config --global user.email 'your_email@domain.com'
```

config的三个作用域 (缺省等同于local)

- local只对某个仓库有效 `git config --local`
- global对当前用户所有仓库有效 `git config --global`
- system对系统所有的登陆的用户有效(基本不用) `git config --system`

显示config的现有的配置 加--list

- eg: `git config --list ; git config --list --global`
- eg: 显示某一个属性 `git config --local user.name`

优先级: **local > global > system**

清除配置 --unset

- eg: `git config --unset --local user.name`
-

3.建Git仓库的两种场景

- 把已有的项目代码纳入Git管理
 - cd 项目代码所在文件夹
 - git init
- 新建的项目直接用Git管理
 - cd 某个文件夹
 - git init your_project # 会在当前路径下创建和项目名称同名的文件夹
 - cd your_project

这里我们使用第二种场景：

- ☐ 在 Git 命令行方式下，用init创建一个Git仓库
 - `git init git_learning`
- ☐ 进入仓库
 - 此仓库中只有一个默认隐藏的 .git 文件夹
 - 进入后,显示的路径会变成这样：
 - D:\desktop\git\git_learning (master -> origin)
 - `cd git_learning`
- ☐ 拷贝文件到当前路径下
 - `cp ../0-material\readme .`
- ☐ 先执行此命令会报错
 - nothing added to commit but untracked files present (use "git add" to track) 报错: 没有跟踪这个文件,即此文件没有被git管理起来
 - `git commit -m "Add readme"`
- ☐ 对新加入的文件必须先执行 `git add 文件名`
 - `git add readme`
- ☐ 查看git现在的状况如何
 - 在这里会反馈 readme文件已经处于暂存区中
 - `git status`

On branch master

No commits yet

Changes to be committed:
 (use "git rm --cached <file>..." to unstage)
 new file: readme

- ☐ 生成一个正式的commit
 - -m 此次变更的理由
 - 反馈：此master是它的分支 它的root-commit已经创建起来
 - `git commit -m "Add readme"`

```
[master (root-commit) e21d58e] Add readme
1 file changed, 2 insertions(+)
create mode 100644 readme
```

- ☐ 用git的日志命令查看进而验证是否真正创建出来
 - `git log`
 - Author完成这次提交的作者

```
commit e21d58edc9d6bb510a58d19547442b7ebe1db3a9 (HEAD -> master)
Author: localDengChuan <localone_piecedc@163.com>
Date:   Fri Jul 16 04:12:24 2021 +0800

    Add readme
```

ps：当设置好global的参数后,新建了一个仓库。若对此仓库额外设置了user.name和user.emali,那肯定是local这个当前仓库的配置起作用！

综上：

```
git init git_learning
cd git_learning
cp ../0-material/readme .
git add readme
git status
git commit -m "Add readme"
git log
```

4.往仓库里添加文件

良好的习惯：在工作目录上的变更首先要先添加到暂存区

4 次提交，一个有模有样的静态页面生成了



- `cp ../0-material/index.html.01 index.html` `cp -r ../0-material/images .`
- `git add index.html images`
- `git commit -m "Add index + logo"`

```
D:\desktop\git\git_learning (master -> origin)
λ ls -al
total 5
drwxr-xr-x 1 14158 197609 0 7月 16 03:52 ./
drwxr-xr-x 1 14158 197609 0 7月 16 03:51 ../
drwxr-xr-x 1 14158 197609 0 7月 16 12:27 .git/
-rw-r--r-- 1 14158 197609 51 7月 16 03:52 readme

D:\desktop\git\git_learning (master -> origin)
λ cp ../0-material/index.html.01 index.html
```

```
D:\desktop\git\git_learning (master -> origin)
λ cp -r ../0-material\images .

# 在master分支 没有跟踪(git未管理)的文件
D:\desktop\git\git_learning (master -> origin)
λ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    images/
    index.html

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

D:\desktop\git\git_learning (master -> origin)
λ git add index.html images
warning: LF will be replaced by CRLF in index.html.
The file will have its original line endings in your working
directory

# 提交的更改(下面的文件已经放到暂存区)
D:\desktop\git\git_learning (master -> origin)
λ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   images/git-logo.png
    new file:   index.html

D:\desktop\git\git_learning (master -> origin)
λ git commit -m "Add index + logo"
[master dd1595b] Add index + logo
 2 files changed, 49 insertions(+)
 create mode 100644 images/git-logo.png
 create mode 100644 index.html

D:\desktop\git\git_learning (master -> origin)
λ git log
commit dd1595b47a2b15220304caa8d226d225fa290440 (HEAD -> master)
Author: DengChuan <one_piecedc@163.com>
Date:   Fri Jul 16 12:59:48 2021 +0800

    Add index + logo

commit e21d58edc9d6bb510a58d19547442b7ebe1db3a9
Author: localDengChuan <localone_piecedc@163.com>
Date:   Fri Jul 16 04:12:24 2021 +0800
```

Add readme

添加style样式

```
# D:\desktop\git\git_learning (master -> origin)
mkdir styles
cp ../0-material\styles\style.css.01 styles\style.css
cd styles
vi style.css
cd ../
git status
git add styles
git status
git commit -m "Add style"
git log
```

添加js文件

```
cp -r ../0-material\js .
git add js
git commit -m "Add js"
git log
```

修改html文件和css文件

- `git add -u`

```
vi index.html
vi styles\style.css

D:\desktop\git\git_learning (master -> origin)
λ git status
On branch master
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:   index.html
        modified:   styles/style.css

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

# 对git已经跟踪过的文件 我们可以用-u update代表所有工作中已经被git管理的文件一
起提交到暂存区
git add -u
git status
git commit -m "Add refering projects"
git log
```

5.给文件重命名的简单方法

普通笨重方法

- mv readme readme.md
- git add readme.md
- git rm readme

```

D:\desktop\git\git_learning (master -> origin)
λ mv readme readme.md

# 未提交的更改 未跟踪的文件
D:\desktop\git\git_learning (master -> origin)
λ git status
On branch master
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working
directory)
        deleted:    readme

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        readme.md

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

D:\desktop\git\git_learning (master -> origin)
λ git add readme.md

D:\desktop\git\git_learning (master -> origin)
λ git rm readme
rm 'readme'

D:\desktop\git\git_learning (master -> origin)
λ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        renamed:    readme -> readme.md

```

不想保留刚才的一系列操作

将暂存区、工作路径上的所有变更都会被清理掉,且不会影响到git日志

- `git reset --hard` 这命令非常危险!!! 慎用.


```
D:\desktop\git\git_learning (master -> origin)
λ git reset --hard
HEAD is now at b2cc499 Add refering projects

# 我们的工作路径是干净的 没有什么东西需要提交的
D:\desktop\git\git_learning (master -> origin)
λ git status
On branch master
nothing to commit, working tree clean
```

用一条命令替代普通方法的三个步骤

- `git mv readme readme.md`

```
D:\desktop\git\git_learning (master -> origin)
λ git mv readme readme.md

D:\desktop\git\git_learning (master -> origin)
λ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    renamed:    readme -> readme.md

D:\desktop\git\git_learning (master -> origin)
λ git commit -m "Move readme to readme.md"
[master 750db58] Move readme to readme.md
 1 file changed, 0 insertions(+), 0 deletions(-)
 rename readme => readme.md (100%)

D:\desktop\git\git_learning (master -> origin)
λ git status
On branch master
nothing to commit, working tree clean
```

6.通过git log查看版本演变历史

- **git help --web log**
- 简洁的查看日志 `git log --oneline`
- 指定最近n个commit `git log -n4 ; git log -n4 --oneline`
- 查看本地各个分支最后一个提交信息 `git branch -v`
- 创建并切换到分支 `git checkout -b temp 9cf853da7a9`

```
D:\desktop\git\git_learning (master -> origin)
λ git branch -v
* master 750db58 Move readme to readme.md

D:\desktop\git\git_learning (master -> origin)
λ git checkout -b temp 9cf853da7a9
Switched to a new branch 'temp'

D:\desktop\git\git_learning (temp -> origin)
```

- `vi readme` 修改添加几个字符
- `git commit -am "Add test"`
- 查看所有分支 `git branch -av`

```
D:\desktop\git\git_learning (temp -> origin)
λ git commit -am "Add test"
[temp ae598c8] Add test
1 file changed, 1 insertion(+)

D:\desktop\git\git_learning (temp -> origin)
λ git branch -av
  master 750db58 Move readme to readme.md
* temp    ae598c8 Add test
```

- 查看当前分支的演进 (temp) `git log`
- 查看所有分支的演进 图形化 (temp master) `git log --all --graph`

```
# 可以看到temp分支是由9cf853da7a9开头的这个分支演变而来的
D:\desktop\git\git_learning (temp -> origin)
λ git log --all --graph --oneline
* ae598c8 (HEAD -> temp) Add test
| * 750db58 (master) Move readme to readme.md
| * b2cc499 Add refering projects
|/
* 9cf853d Add js
* 58a1440 Add style
* dd1595b Add index + logo
* e21d58e Add readme
```

- 所有分支最近的四次commit(不是每个分支4次) `git log --oneline --all -n4 --graph`
- 如果用了all后面再指定分支,指定的分支是不起作用的 `git log --all temp`

7.通过图形界面工具来查看版本历史

- 终端输入命令: `gitk`
- 每一次变更的提交都有Author(作者)、Committer(提交人)、Branches(归属哪些分支)

```
Author: DengChuan <one_piecedc@163.com> 2021-07-16 13:11:47
Committer: DengChuan <one_piecedc@163.com> 2021-07-16 13:11:47
Parent: dd1595b47a2b15220304caa8d226d225fa290440 (Add index + logo)
Child: 9cf853da7a9fbd7ee53e3ec8fb29a82f28946df0 (Add js)
Branches: master, temp
Follows:
Precedes:

Add style
```

8.探秘'.git'目录 进入.git这个裸仓库文件夹

```
D:\desktop\git\git_learning\.git (temp -> origin)
λ ls -al
total 23
drwxr-xr-x 1 14158 197609 0 7月 19 14:01 ./
drwxr-xr-x 1 14158 197609 0 7月 16 15:56 ../
-rw-r--r-- 1 14158 197609 9 7月 16 15:57 COMMIT_EDITMSG
-rw-r--r-- 1 14158 197609 188 7月 16 12:27 config
-rw-r--r-- 1 14158 197609 73 7月 16 03:27 description
-rw-r--r-- 1 14158 197609 507 7月 19 14:01 gitk.cache
-rw-r--r-- 1 14158 197609 21 7月 16 15:54 HEAD
drwxr-xr-x 1 14158 197609 0 7月 16 03:27 hooks/
-rw-r--r-- 1 14158 197609 554 7月 16 15:57 index
drwxr-xr-x 1 14158 197609 0 7月 16 03:27 info/
drwxr-xr-x 1 14158 197609 0 7月 16 04:12 logs/
drwxr-xr-x 1 14158 197609 0 7月 16 15:57 objects/
-rw-r--r-- 1 14158 197609 41 7月 16 14:36 ORIG_HEAD
drwxr-xr-x 1 14158 197609 0 7月 16 03:27 refs/
```

- 执行切换分支的时,HEAD的内容会发生变化,它告诉我们正工作在哪个分支上。
- HEAD里面是一个引用ref

```

D:\desktop\git\git_learning\.git (temp -> origin)
λ cat HEAD
ref: refs/heads/temp

D:\desktop\git\git_learning\.git (temp -> origin)
λ git branch -av
  master 750db58 Move readme to readme.md
* temp    ae598c8 Add test

D:\desktop\git\git_learning\.git (temp -> origin)
λ git checkout master
fatal: this operation must be run in a work tree

D:\desktop\git\git_learning\.git (temp -> origin)
λ pwd
D:\desktop\git\git_learning\.git

D:\desktop\git\git_learning\.git (temp -> origin)
λ cd ..

D:\desktop\git\git_learning (temp -> origin)
λ git checkout master
Switched to branch 'master'

D:\desktop\git\git_learning (master -> origin)
λ cat .git/HEAD
ref: refs/heads/master

```

- config文件里存放的是跟本地仓库相关的配置信息

```

D:\desktop\git\git_learning (master -> origin)
λ cat .git\config
[core]
    repositoryformatversion = 0
    filemode = false
    bare = false
    logallrefupdates = true
    symlinks = false
    ignorecase = true
[user]
    name = 'DengChuan'
    email = 'one_piecedc@163.com'

```

- tags 代表我们的git仓库可以有很多很多个标签的.也称作'里程碑'!
- heads 对应的就是我们的分支 分支简单理解：独立的开发空间.
- `git cat-file -t 750db58f0` 查看存放对象的类型
- `git cat-file -p 750db58f0` 查看存放对象里的内容

```
D:\desktop\git\git_learning (master -> origin)
λ ls -al .git\refs
total 4
drwxr-xr-x 1 14158 197609 0 7月 16 03:27 ./
drwxr-xr-x 1 14158 197609 0 7月 19 14:22 ../
drwxr-xr-x 1 14158 197609 0 7月 16 15:57 heads/
drwxr-xr-x 1 14158 197609 0 7月 16 03:27 tags/

D:\desktop\git\git_learning (master -> origin)
λ cat .git\refs\heads\master
750db58f0cde14820ec4a45f7202d3ee3dc2d86e

D:\desktop\git\git_learning (master -> origin)
λ git cat-file -t 750db58f0
commit

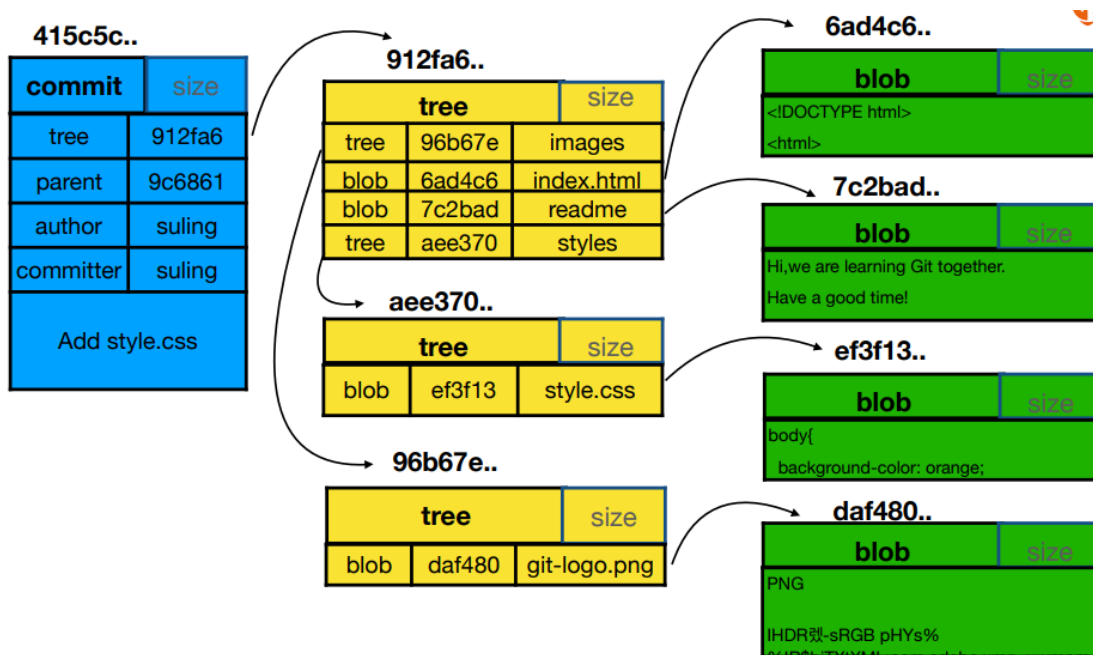
D:\desktop\git\git_learning (master -> origin)
λ git branch -av
* master 750db58 Move readme to readme.md
  temp ae598c8 Add test
```

- objects => 核心的三个对象：commit、tree、blob
- git中 只要任何文件的文件内容相同,那么在git的眼里它就是唯一的一个blob.

9.commit、tree、blob三个对象之间的关系

良好的存储机制对于版本控制系统来说至关重要。

一个commit里面对应着一棵树(tree),里面包含着那个点我们的项目包含着什么文件夹(tree)什么文件(blob)。



- git不理睬不带文件的文件夹 - `mkdir doc` , `git status` , `cd doc` , `echo "hello,World" > readme`
- `find .git/objects -type f` 指定目录下类型是文件的有多少个?

问：新建的Git仓库,有且仅有一个commit,仅仅包含/doc/readme,请问内含多少个tree,多少个blob?

答：1个commit、2个tree、1个blob.(这样理解：tree = commit本身是一个tree + 有多少个文件夹就有多少个tree ; blob = 有多少个文件.)

10.分离头指针下的注意事项

```
git log --oneline --all
git checkout '某一个不是分支的commit 93a8a554'
修改了style/style.css background为blue
git commit -am 'Backgroud to blue'
git checkout master
git branch -av
git branch <新分支的名字> 93a8a554
```