به نام خدا

آموزش Git

منبع : Lynda

فهرست مطالب

فیلم ۱

tick

فیلم ۲

tick

فیلم ۳

مقدمه

Git :

- keeps track of changes
- version 1,2,…
- Version Control System **VCS** - distributed version control
- use VCS for management -Source code management SCM
- open source
- created by Linus Torvalds
- Cross platform
- faster than other source controls before it

فیلم ۴

تاریخچه VCS

- SCCS
  - source code control system
  - close source
  - **save snapshot of changes**
- RCS
  - cross-platform
  - open source

- faster than SCCS
- **نحوه کارکردش را نفهمیدم**

- CVS
  - concurrent version system
  - open source
  - **more than 1 user can work on the same file in the sam time**
- SVN
  - open source
  - faster than CVS
  - **take snapshot of directory not just file**
  - **نحوه کارکردش را دقیقا نفهمیدم**
- BitKeeper SCM
  - close source
  - در ابتدا برای لینوکس استفاده میشده اما بعد از چند وقت از رایگان بودن (بخشی اش رایگان بوده) در می آید و لینوکس هم دیگر از ان استفاده نمیکند و به سراغ گیت میرود .
  - **نحوه کارکردش را دقیقا نفهمیدم**
- Git
  - replaced instead of BitKeeper to manage linux kernel source code
  - ....

فیلم ۵

درباره distributed version control

- هر کدام از vcs هایی که جلسه قبل صحبت شد ، یک central code repository model دارند و وظیفه ما است که آنرا آپدیت نگه داریم با توجه به submit های جدید و ... اما گیت این طور نیست
- each of team users maintain their own repositories instead of working from a cntral repository
- changes are stored as “change sets”
  - track changes not versions
  - دقیقا نفهمیدم
- no single master repository
- no need to communicate with a cntral server
  - no network needed
  - faster
  - کمی توضیحات دیگر داده شد که مهم نبود ولی واضح هم نبود و نفهمیدم

فیلم ۶

چه کسی از Git استفاده میکند ؟

- anyone who tracks edits
  - review history log of changes
  - view differnces between versions
  - retrieve old versions

- anyone who wants to share changes with collaborators
- programmers

چه کسی از گیت استفاده نمیکند ؟

- anyone who wants to track non-text file
  - images,videos,musics و ....

فیلم ۷ و ۸ و ۹

نصب گیت – tick

کارکردن با git bash – tick

با **`which git`** میشه فهمید که گیت کجا ذخیره شده است

همجنین با **`git --version`** هم میشه فهمید که ورژنی که از گیت نصب کردیم چی هست

فیلم ۱۰

Git configuration

commands :

- System
  - git config –system
- User
  - git config –global
- Project
  - git config

تمرین ها :

ابتدا کامند های کانفیگ را میزنیم :

```
> git config --global user.name “mahdi niknejad”
> git config --global user.email “m.niknejad@aut.ac.ir”
> git config --list
user.name=mahdi niknejad
user.email=m.niknejad@aut.ac.ir
> git config user.name
mahdi niknejad
> git config user.email
m.niknejad@aut.ac.ir
```

و بعد میرویم تا فایل **gitconfig.** را پیدا کنیم :

```
> cd ~
> ls -la (you can see .gitconfig in the list)
> cat .gitconfig
[user]
        name = mahdi niknejad
        email = m.niknejad@aut.ac.ir
        signingKey = 5FFDE666E0CE2889
```

ست کردن ادیتور دیفالت :

```
> git config --global core.editor “nano” (or “vim” , “emacs” , “notepad” , “mate” , ... )
> git config --global color.ui true
> cat .gitconfig
```

فیلم ۱۱

ظاهرا در لینوکس ، **git completion** نداریم (گرچه در ویندوز به صورت دیفالت هست !) و باید ست کرد که اینجا آموزش داده شده است .

فیلم ۱۲

در مورد **git help** :

```
> git help
> git help add
```

در اینجا میخواهیم ببینیم که add چه میکند . در لینوکس ، یک صفحه در همان command-line باز میشود (صفحه Git manual) که توضیحاتی نوشته و در ویندوز ، در browser می آید .

برای جلو رفتن در صفحه Git manual ، از f استفاده میکنیم

برای عقب رفتن در صفحه Git manual ، از b استفاده میکنیم

برای خارج شدن هم از q استفاده میکنیم

این صفحه همان man هست . مثلا میزدیم :

```
man ls
```

یا

```
man git-add (= git help add)
```

فیلم ۱۳

initializing a repository

کامندی که برای initialize کردن نیاز داریم تا بتوانیم از گیت برای یک فایل استفاده کنیم ، git init است .

فیلم ۱۴

میتوانیم در آن پروژه که ساختیم بزنیم ls -la و میبینیم :

```
> ls -la
total 4
drwxr-xr-x 1 AVAJANG 197121 0 Aug  3 07:59 ./
drwxr-xr-x 1 AVAJANG 197121 0 Aug  3 07:59 ../
drwxr-xr-x 1 AVAJANG 197121 0 Aug  3 09:43 .git/
```

(توجه : اگر ls خالی بزنیم چیزی نشان نمیدهد چون hidden هستند)

همانطور که میبینید دایرکتوری .git ساخته شده که تمام track ها در آن ذخیره میشود

برای دیدن داخل دایرکتوری .git میزنیم :

```
> ls -la .git
total 11
drwxr-xr-x 1 AVAJANG 197121   0 Aug  3 09:43 ./
```

```
drwxr-xr-x 1 AVAJANG 197121   0 Aug  3 07:59 ../
-rw-r--r-- 1 AVAJANG 197121 130 Aug  3 09:43 config
-rw-r--r-- 1 AVAJANG 197121  73 Aug  3 07:59 description
-rw-r--r-- 1 AVAJANG 197121  23 Aug  3 07:59 HEAD
drwxr-xr-x 1 AVAJANG 197121   0 Aug  3 07:59 hooks/
drwxr-xr-x 1 AVAJANG 197121   0 Aug  3 07:59 info/
drwxr-xr-x 1 AVAJANG 197121   0 Aug  3 07:59 objects/
drwxr-xr-x 1 AVAJANG 197121   0 Aug  3 07:59 refs/
```

فیلم ۱۵

یک فایل متنی first_file.txt در فولدر پروژه میسازیم (با همان touch) و بعد میزنیم git status و میبینیم که untracked file داریم و حالا باید git add . بزنیم و بعدش git commit –m “message” بزنیم. حالا دیگه این فایل tracked شده است .

پس کار هایی که کردیم :

- make changes
- add changes
- commit changes

```
AVAJANG@Mahdi MINGW64 /d/        1            /                /Git/Git-ex
ercise/my-project (master)
$ git status
On branch master

No commits yet

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

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

AVAJANG@Mahdi MINGW64 /d/        1            /                /Git/Git-ex
ercise/my-project (master)
$ git add .
warning: LF will be replaced by CRLF in first_file.txt.
The file will have its original line endings in your working directory

AVAJANG@Mahdi MINGW64 /d/        1            /                /Git/Git-ex
ercise/my-project (master)
$ git status
On branch master

No commits yet

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


AVAJANG@Mahdi MINGW64 /d/        1            /                /Git/Git-ex
ercise/my-project (master)
$ git commit -m "initial commit"
[master (root-commit) 3dc1910] initial commit
 1 file changed, 2 insertions(+)
 create mode 100644 first_file.txt

AVAJANG@Mahdi MINGW64 /d/        1            /                /Git/Git-ex
ercise/my-project (master)
$
```

فیلم ۱۶

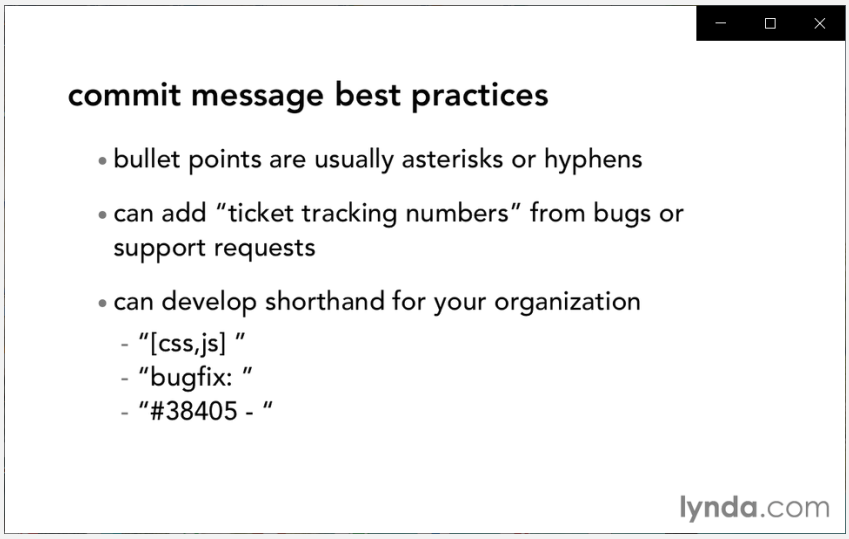
در مورد نوشتن commit message

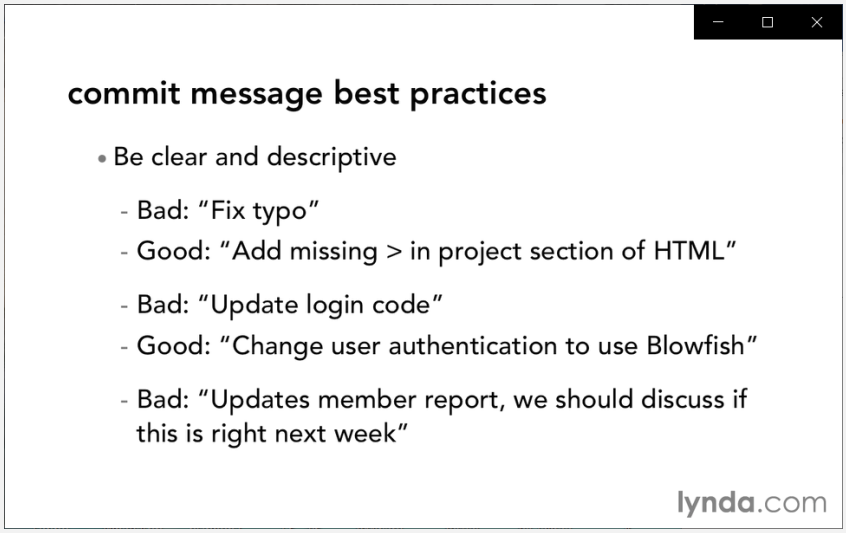
best practices :

- یک خطی و کوتاه (کمتر از 50 کاراکتر) ولی کامل
- نوشتن جملات به **زمان حال** ، نه گذشته !
  - مثلا :

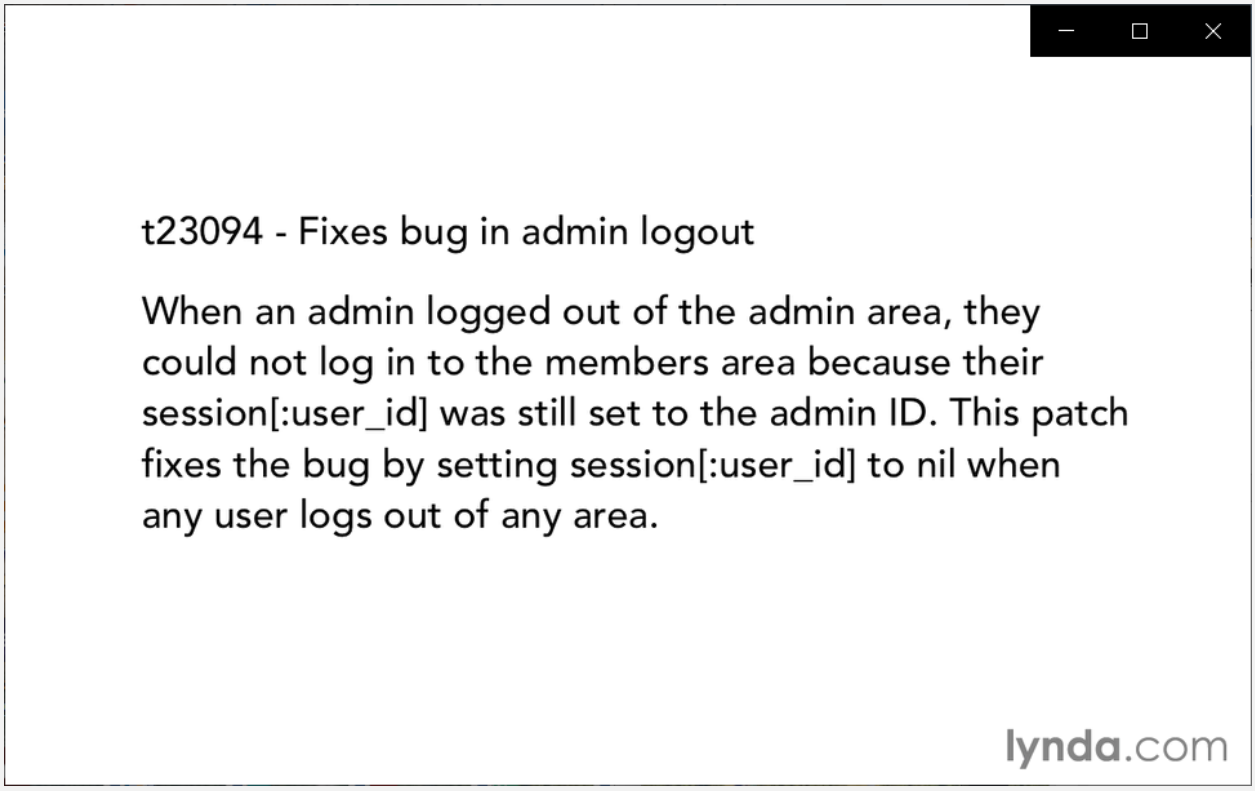
fix bug	correct
fixes bug	correct

~~fixed bug~~ incorrect





مثالی از یک `commit` خوب :



فیلم ۱۷

برای دیدن `commit` ها(ی اخیر) باید از `git log` استفاده کنیم .

```
> git log
commit 3dc191018c913956fe2f419cff442ec128b96abd (HEAD -> master)
Author: mahdi niknejad <m.niknejad@aut.ac.ir>
Date:   Mon Aug 3 10:11:20 2020 +0430

    initial commit
```

برای اینکه فقط `n` تا(محدود تا) `commit` را ببینیم از `git log -n 3` مثلا استفاده میکنیم.

برای اینکه از یک تاریخی به بعد ببینیم که این پروژه چه تغییراتی داشته و کامیت هایش چه بوده ، می آییم

و از `git log --since=2019-08-12` مثلا استفاده میکنیم .

و همچنین برای اینکه ببینیم تا چه تاریخی ، چه تغییرات و کامیت هایی داشته ایم در پروژه می آییم

و از `git log --until=2019-08-12` مثلا استفاده کنیم.

برای اینکه ببینیم کی کامیت ها را زده است و مثلا سرچ کنیم نویسنده کامیت را از `git log --author="Ali"` مثلا استفاده میکنیم.

برای پیدا کردن یک کلمه یا حرف در متن کامیت از `git log --grep="com"` مثلا استفاده میکنیم .

فیلم ۱۸

اضافه کردن یک پروژه آماده به ریپوزیتوری موجود در Github :

برای اینکه یک پروژه آماده را روی گیت ببریم :

1. Create a new repository on GitHub. You can also add a gitignore file, a readme and a licence if you want
2. Open Git Bash
3. Change the current working directory to your local project.
4. Initialize the local directory as a Git repository.
   `git init`
5. Add the files in your new local repository. This stages them for the first commit.
   `git add .`
6. Commit the files that you've staged in your local repository.
   `git commit -m "initial commit"`
7. Copy the https url of your newly created repo
8. In the Command prompt, add the URL for the remote repository where your local repository will be pushed.

   `git remote add origin remote repository URL`

   `git remote -v`
9. Push the changes in your local repository to GitHub.

   `git push -f origin master`