عنوان: **#6 آموزش سیستم مدیریت کد Git : استفاده به صورت محلی (بخش دوم)**
نویسنده: حسام امامی
تاریخ: ۱۳۹۱/۰۶/۱۹ ۱:۲۴
آدرس: www.dotnettips.info
گروه‌ها: Git

در قسمت قبل برخی از دستورات مورد نیاز برای کار با git به صورت محلی گفته شد. در اینجا به بخشی دیگر از این دستورات خواهیم پرداخت:

مشاهده تغییرات فایل ها:

در بسیاری از موارد نیاز است تا بتوانیم تفاوت فایل‌های موجود در working tree و فایل‌های موجود در stage و repository را دریابیم. بدین منظور می‌توان از دستورات زیر استفاده کرد:

```
git log
```

برای مشاهده تغییرات فایل‌ها بین دو commit دلخواه از کد زیر استفاده می‌کنیم:

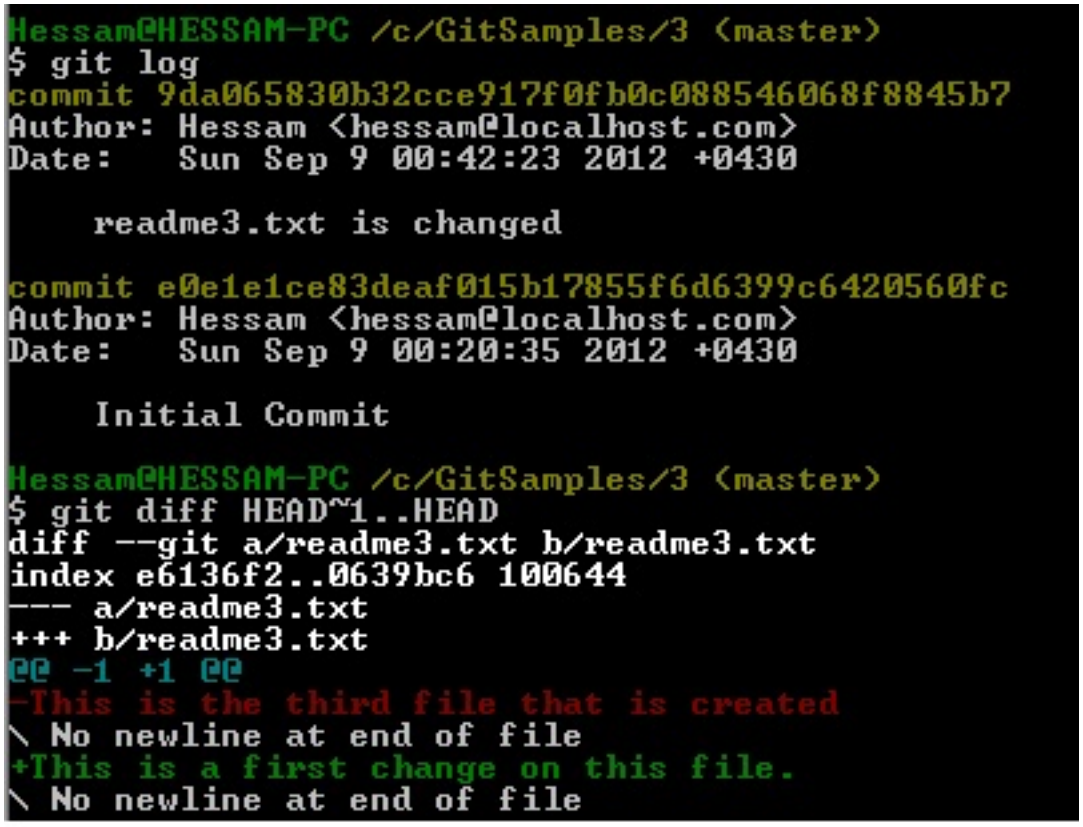
```
git diff
```

تذکر: در اغلب موارد می‌توانید تنها از چند مقدار اول SHA-1 برای آدرس‌دهی استفاده نمود. چون معمولا این کد به اندازه کافی دارای تغییرات است.البته کار کردن با کدهای SHA-1 ممکن است مشکل باشد؛ به همین جهت می‌توان از دستور زیر نیز برای مشاهده تغییرات استفاده نمود:

```
git diff HEAD~[number]..HEAD~[number]
```

توجه کنید که کلمه HEAD اشاره به وضعیت جاری head دارد و عدد number اختلاف آن را با وضعیت جاری مشخص می‌نماید. به عنوان مثال در شکل زیر ما می‌خواهیم اختلاف فایل‌ها را بین ۲ دستور commit با مقادیر 9da و e0e را مشخص نماییم. همانطور که ملاحظه می‌کنید اولی اشاره به وضعیت جاری head و دومی وضعیت قبلی head است. بنابراین ما از دستور زیر استفاده می‌کنیم:

```
git diff HEAD~1..HEAD
```



همچنین اگر بخواهیم اختلاف فایلی را در working tree و stage ببینیم، کافی است که از دستور زیر استفاده کنیم:

```
git diff --staged [filename]
```

در صورتی‌که در تنظیمات git، نرم افزار پیش فرضی را برای نمایش اختلاف فایل‌ها تعیین نکرده باشید، git اختلاف فایل‌ها را خود نمایش می‌دهد. اما از آنجاییکه این نمایش چندان مطلوب نیست، بهتر است از دستور زیر برای تنظیم نمایش اختلاف فایل‌ها در نرم افزار دیگری استفاده کنید:

```
git config --global diff.external <path_to_wrapper_script>
```

تنظیمات مورد نیاز برای این کار در اینجا گفته شده است.
تذکر: راه حل ساده برای این منظور نصب git extension است که در آموزش نصب گفته شد.

تنظیم git برای صرفنظر کردن از برخی فایل‌ها:

اگراز دستوراتی نظیر . add استفاده کنید متوجه خواهید شد در بعضی موارد نیازی ندارید که تمامی فایل‌های موجود در working tree به repository اضافه شوند. فایل‌ها در git به دو دسته تقسیم می‌شوند؛ برخی که در حال حاضر دنبال شده و برخی که git تغییرات آنها را دنبال نمی‌کند. در صورتیکه بخواهید فایلی که تغییرات آن دنبال نمی‌شود را به طور کلی حذف کنید، می‌توانید از دستور clean استفاده کنید. دو اصلاح کننده معروف این دستور n- برای نمایش آنکه چه فایل هایی حذف خواهند شد و f- برای اجبار در حذف آنها:

```
git clean -n [filename]
git clean -f [filename]
```

اما در برخی موارد نیاز است که فایل‌ها وجود داشته باشند، اما تنها git تغییرات آن‌ها را دنبال نکند، نه آنکه مانند دستور بالا آنها را از working tree نیز حذف نماید. بدین منظور git از فایل بی‌نامی با پسوند gitignore. استفاده می‌کند این فایل از عبارات منظم به شکل بسیار محدودی پشتیبانی می‌کند. در ادامه برخی از دستوراتی را که می‌توان برای حذف برخی فایل‌ها در این فایل نوشت را مشاهده خواهید کرد:

۱ مجموعه: مثال [adgJHn]

۲ بازه: [0-9] یا [a-z]

۳ حذف یک دایرکتوری با نوشتن آدرس آن و قرار دادن / (البته توجه کنید که با این کار sub directoryها هنوز هم track خواهند شد)

می‌توان با استفاده از علامت ! برخی از فایل‌ها و یا دایرکتوری‌ها را مستثنی کرد

می‌توان این تنظیمات را در فایلی با نام دلخواه ذخیره کرد و سپس با استفاده از دستور زیر آن‌ها را به صورت global یا سراسری اعمال نمود:

```
git config global core.excludesfile [path and filename]
```

توجه کنید که git تغییرات پوشه‌های خالی را دنبال نمی‌کند بنابراین اگر قصد دارید پوشه‌ای در repository ذخیره شود یک فایل temp در آن ایجاد کنید

چند مثال:

اگر بخواهید فایل‌های باینری داخل فولدر bin در repository ذخیره نشوند این خط را در این فایل اضافه می‌کنیم:

```
bin/
```

هیچ فایلی با پسوند txt را در نظر نگیر:

```
*.txt
```

هیچ فایلی را با پسوند txt در فولدر bin در نظر نگیر

```
/bin/*.txt
```

هیچ فایلی با پسوند txt را در نظر نگیر به جز readme1.txt

```
*.txt
!readme1.txt
```

توجه کنید که هر آنچه بین دو علامت # قرار گیرد به عنوان توضیح در نظر گرفته می‌شود

نظرات خوانندگان

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۶/۱۹ ۹:۱۳

یک gitignore مفید برای VS.NET:

```
#OS junk files
[Tt]humbs.db
*.DS_Store

#Visual Studio files
*.[Oo]bj
*.user
*.aps
*.pch
*.vspscc
*.vssscc
*_i.c
*_p.c
*.ncb
*.suo
*.tlb
*.tlh
*.bak
*.[Cc]ache
*.ilk
*.log
*.lib
*.sbr
*.sdf
*.opensdf
*.unsuccessfulbuild
ipch/
obj/
[Bb]in
[Dd]ebug*/
[Rr]elease*/
Ankh.NoLoad

#MonoDevelop
*.pidb
*.userprefs

#Tooling
_ReSharper*/
*.resharper
[Tt]est[Rr]esult*
*.sass-cache

#Project files
[Bb]uild/

#Subversion files
.svn

# Office Temp Files
~$*

#NuGet
packages/

#ncrunch
*ncrunch*
*crunch*.local.xml

# visual studio database projects
*.dbmdl

#Test files
*.testsettings
```

نویسنده: ناصر فرجی

تاریخ: ۱۳۹۳/۰۸/۱۳ ۰:۳۲

دو تا - اول global در این دستور رو یادتون رفته بزارید. لطفا اصلاح کنید

```
git config --global core.excludesfile [path and filename]
```