### git checkout

این دستور برای جابجایی بین شاخه‌ها و تغییر وضعیت HEAD به نسخه‌ای خاص از پروژه استفاده می‌شود. همچنین می‌توان از آن برای بررسی یک commit یا فایل خاص استفاده کرد.

#### کاربردهای اصلی

1. جابجایی به شاخه دیگر:

`git checkout branch-name`

2. بررسی فایل یا commit خاص:

`git checkout commit-hash -- file-path`

3. ساخت و جابجایی به شاخه جدید (در نسخه‌های قدیمی‌تر گیت):

`git checkout -b new-branch-name`

#### فلگ‌های پرکاربرد

1. `-b`: ایجاد یک شاخه جدید و جابجایی به آن

مثال: `git checkout -b feature-branch`

2. `--track`: تنظیم شاخه محلی برای دنبال کردن یک شاخه ریموت

مثال: `git checkout --track origin/feature-branch`

3. `--`: مشخص کردن فایل‌های خاص برای بازگرداندن به حالت خاصی

مثال: `git checkout HEAD -- file.txt`

### git stash

این دستور برای ذخیره موقت تغییراتی که هنوز commit نشده‌اند استفاده می‌شود. با استفاده از این دستور می‌توانید بدون از دست دادن تغییرات، روی مسائل دیگر کار کنید.

#### کاربردهای اصلی

1. ذخیره تغییرات فعلی:

`git stash`

2. برگرداندن تغییرات ذخیره‌شده:

`git stash apply`

3. مشاهده لیست stash‌ها:

`git stash list`

#### فلگ‌های پرکاربرد

1. `-u` یا `--include-untracked`: شامل فایل‌های untracked در stash

مثال: `git stash -u`

2. `--keep-index`: تغییرات stage شده را نگه داشته و فقط تغییرات دیگر را stash می‌کند

مثال: `git stash --keep-index`

3. `pop`: برگرداندن تغییرات stash‌شده و حذف آن‌ها از لیست stash

مثال: `git stash pop`