GIT CHEAT SHEET

Git Config

git config — global user.name <name>	Geçerli kullanıcı tarafından tüm commit'ler için kullanılacak yazar adını tanımlar.
git config — global user.email <email>	Geçerli kullanıcı tarafından tüm commit'ler için kullanılacak yazar e-postasını tanımlar.
git config —global —edit	Kullanıcı adı ve e-posta adresini düzenleme için bir metin düzenleyici açar.

Git Basics (Local Git)

git —version	Kurulan git versiyonu gösterir.
git init <directory>	Belirtilen dizinde boş bir git repository oluşturur.
git add . / git add <directory>	Tüm değişikleri ekler. / Dizindeki tüm değişiklikleri bir sonraki işlem için ekler.
git commit -m "message"	Dizine eklenen değişikleri mesaj ile commit'ler.
git status	Dizindeki hangi dosyaların değişikliğe uğradığını, eklendiğini ya da eklenmediğini gösterir.
git log	Dizinde yapılan geçmiş değişikleri (commit, merge, branch gibi) gösterir.
git diff	İki dizin arasındaki farkları gösterir.

Git Essentials (Filesystem interactions)

git mv <old_file_name> <new_file_name>	Bir dosyayı yeniden adlandırmak veya taşımak için kullanılır.
git rm	Bir dosyayı kaldırmak için kullanılır.
git rm -n (veya —dry-run)	İstenen dosyayı gerçekten kaldırmadan simüle edilmiş bir git rm çalıştırmayı görmek için kullanılır.
git reset / git reset --mixed	Çalışma dizinindeki tüm dosyaların durumunu son commit'e uyacak şekilde sıfırlar ama çalışma dizini değiştirmeden bırakmak için kullanılır.
git reset —hard	Hem dizin hazırlama alanını hem de çalışma dizinini bu daldan önceki commit'in durumuna sıfırlanmasını istediğini belirtir.
git reset --soft	Ne dizin hazırlama alanını ne de çalışma dizinini sıfırlar, sadece HEAD işaretçisini önceki işleme işaret edecek şekilde değiştirir.
git reset <commit_id>	Çalışma dizinindeki tüm dosyaların durumunu <commit_id>'ye kadar sıfırlamak için kullanılır.
git reset —hard <commit_id>	Commit edilmeyen tüm değişikleri siler ve <commit_id>'ye kadar çalışma dizininin durumunu sıfırlar.
git clean —force	İzlenmeyen dosyaları (untracked files) silmek için kullanılır.
git ls-files	Git çalışma dizininde izlenen dosyaları (tracked files) görüntülemek için kullanılır.
git ls-files -others (-o)	Git çalışma dizininde izlenmeyen dosyaları (untracked files) görüntülemek için kullanılır.
git clean —force -X	Git dizininde yok sayılan dosyaları silmek için kullanılır.
git clean -x	Hem yok sayılan hem de izlenmeyen dosyaları kaldırmak için kullanılır.
git clean -xdf	Tüm izlenmeyen, yok sayılan dosyalar (-x) ve dizinleri (-d) kaldırır.
git revert <commit_id>	<commit_id>'de yapılan tüm değişiklikleri geri alır ve geçerli branch'a uygular.
git stash	Mevcut değişiklikleri geçici olarak saklar.
git stash list	Saklanmış dosyaları listeler.
git stash pop	Saklanmış dosyaları geri alır, git dizinine geri getirir.
git stash clear	Saklanmış dosyaları temizler.
git ls-files -v	Varsayılan değiştirilmemiş dosyaları listelemek için kullanılır.

Git Basics (Remote Git)

git remote add <name> <url>	Remote repository'e yeni bir bağlantı oluşturur. Bağlantı eklendikten sonra diğer komutlarda <url> için kısayol olarak <name> (örnek: origin) kullanılır.
git remote —verbose	Remote repository'e oluşturulan bağlantıyı görmek için kullanılır.
git remote show	Remote repository'de bulunan bilgiler hakkında bilgi verir.
git push --set-upstream origin <branch name>	Yerel repository'de yapılan değişikleri remote repository'e göndermek için kullanılır.
git clone <repo_url>	<repo_url> konumunda bulunan repository'i yerel makineye klonlar.
git pull	Başka bir repository'den yeni commitleri indirir ve remote branch'ı yerel branch ile birleştirir.
git fetch	Remote repository'de bulunan yeni commit'leri yerel repository'deki değişikleri değiştirmeden indirmek için kullanılır.
git branch <branch name>	Mevcut branch'tan <branch_name> adında yeni bir branch oluşturur.
git branch	Repository'de bulunan branch'leri listeler.
git checkout <branch name>	Branch'ler arasında geçiş yapar.
git checkout -b <branch name>	Mevcut branch'tan <branch_name> adında yeni bir branch oluşturur ve o branch'e geçiş yapar.
git branch -a	Local branch'ları ve remote tracking branch'ları listeler.
git branch -r	Remote branch'ları listeler.
git branch -vv	Local tracking branch'ları listeler.
git branch --track <track_branch_name> origin/<branch_name>	<track_branch_name> adında origin/<branch_name> için local tracking branch oluşturur.
git merge <branch_name>	<branch_name>'ı master branch ile birleştirir
git rebase <base>	<base>'de bulunan ek commit'leri mevcut branch'ın sonuna ekler ama taşımaz, yeni commit_id ile commit'leri tekrar oluşturur.
git cherry-pick <commit_id>	<commit_id> 'li commit'deki değişikleri alır ve mevcut branch'a taşır.
git push --delete origin <branch name>	Remote repository'de bulunan <branch_name>'i siler.
git branch --delete <branch name>	Yerel repository'de bulunan <branch_name>'i siler.

Git Log (Advanced)

git log —author	Belirli bir yazar için komitleri arar.
git log —grep "<pattern>"	İçinde <pattern> ile eşleşen commit mesajlarını arar.
git log —max-count (or -n or -<limit>)	Commit sayısını sınırlandırır (git log -5).
git log —reverse	En eski commit en önce gelecek şekilde commit'leri sıralar.
git log —after "<date>"	<date>'den sonraki commit'leri gösterir.

git log —before "<date>"	<date>'den önceki commit'leri gösterir.
git show HEAD^	Son commit'i gösterir. git log —max-count=1 —patch HEAD^ komutuna eşittir.
git log —patch (or -p)	Her commit'in tam farkını gösterir.
git log —format <parameter>	<parameter> email, medium, oneline, short, full, fuller ve raw gibi parametreler olabilir.
git log --format="%ar %an did: %s"	(format string), commit başına çeşitli öznitelikleri doldurmak için yer tutucular kullanır.

git shortlog	Her satıra bir commit konusu olacak şekilde yazara göre gruplandırılmış commit'leri gösterir.
git log —graph —decorate	—graph parametresi, commit mesajının sol tarafında metin tabanlı bir commit grafiği çizer. —decorate, gösterilen commit'lerin branch veya tag'lerini ekler.
git log —stat	Hangi dosyaların değiştirildiğini ve her birine eklenen veya silinen ilgili satır sayısını gösterir.
git log — <file>	Sadece belirtilen <file> için commit'leri gösterir.
git blame —date=short <file_name>	Bir dosyanın her satırını en son kimin değiştirdiğini gösterir.
git reflog	Yerel repository'deki değişikliklerin günlüğünü gösterir.