



IT INNOVATIONS ACADEMY

Git

Git nədir?

Git yazdığımız program təminatlarını internet üzərində saxlamağımızı və idarə etməyimizi təmin edən bir versiya nəzarət sistemidir.

Nədir bu version control system? Version control system sənədlərdə edilən bütün dəyişiklikləri kim tərəfindən, nə vaxt edildiyi və sənədin yenilənmədən əvvəlki halına qayıtmasına imkan verən bir saxlama sistemidir.

Nə üçün gitdən istifadə etməliyik ?

Böyük projectlər üzərində işlədiyimiz zaman bir neçə nəfər eyni zamanda çalışdığı üçün kodların birləşdirilməsi və ən son versiyanın müəyyənləşdirilməsi prosesi çox qarışıq və olduqca vaxt tələb edən prosesdir. Lakin **Git** vasitəsilə isə buna ehtiyac qalmır. **Git** əmirləri vasitəsi ilə rahatlıqla hamı öz üzərinə düşən işi yerinə yetirir. Daha sonra edilmiş işlər birləşdirilir.

Git proyektlərimizi saxlamaq üçün ən məşhur servis GitHub-dır.

Gitdəki əsas anlayışlar



Repository projectin fayıllarını uzaq serverdə saxlayır . Qısaldılmış şəkildə " Repo" kimi də adlanır.

Branch Biz hər hansı bir projecti versionlamaq istədikdə git avtomatik olaraq master branch yaradır və biz bu branch üzərində işləmiş oluruq. Tutaq ki sizin bir projectiniz var və başqa biridə o project üzərində başqa bir iş görmək istəyir həmin anda eyni branch üzərində ikinizdə işləsəz məsələ çox qarışıq və anlaşılmaz hall alacaqdır. Elə bu səbəbdən iki ayrı branch ilə işləyərək məsələ daha aydın olur. Nəticə etibarlı ilə edilmiş bütün dəyişikliklər olduğunuz brancha aid olur. Ümumi desək git ağac məntiqi ilə işləyir.

Main Repository 'in ana branchıdır. Biz əgər git üzərində dəyişiklik etməsək master branch üzərində davam edəcəkdir.

Commit projectdə hər hansı dəyişiklik olduqda onun anlıq görüntüsünü alır və yaddaşda saxlayır.

Checkout mövcud branch üzərindəki commitlərin üzərində keçid etməyimizə kömək edir.

Fork Repository'nin bir kopyasının alınmasıdır.

Pull Request Pull request olaraq göndərmək demək mən layihədə dəyişikliklər etdim və bu dəyişiklikləri səndə təsdiqlə və projecti merge et deməkdir.

Merge Branch üzərində etdiyimiz dəyişiklikləri master branch'i üzərində birləşdirmək prosesidir.

· İlk öncə bu **Git** bash proqramını yükləmək üçün <https://git-scm.com/downloads> linkə

daxil olub download knopkasını klikləyirik.

Git Komandaları:

- **Git config:** bu komanda vasitəsilə git istifadəçi adı, və mail əlavə edə bilərik, bunları əlavə etmədən hər hansı bir iş gödükdə default olaraq sistemdəki username təyin olunacaqdır.
- **Git init:** Biz projectimizi versionlamaq istədikdə terminalımızı açaraq həmin qovluğa gəlirik. Daha sonra "git init" əmrini yazırıq. Beləliklə projectimizdə .git qovluğu yaranır və burada projectimizin kopyası saxlanılır. init komandası, qovluğu bir Git Repository halına gətirir və .git sonluqlu bir qovluq yaradır. Git init komandası qovluğumuzu uzaqdakı bir serverə göndərmək üçün hazır vəziyyətə gətirir.
- **Git add:** add komandası hər-hansı bir yeni faylı və ya qovluqdakı bütün faylları əlavə edir." Git add ." git add boşluq oyaraq nöqtə qoymağınız vacibdir.
- **Git clone:** bu komanda local -dakı və ya serverdəki istənilən projectin kopyasını hazırda olduğumuz qovluğa kopyalayır.
- **Git commit:** Commit ilə etdiyimiz dəyişikliklər haqqında açıqlamaları qeyd edirik. Commit proyektimizi staging area -dan (Hazırlıq sahəsi) local repository bölməsinə keçirir. git commit -m " bu ilk commitdir".
- **Git status** Projectdəki faylların həmin anki vəziyyətini göstərir və olunmuş dəyişiklikləri göstərir.
- **Git remote "git remote -v"** əmri ilə projectə bağlı olan uzaq serveri listələyir. Mövcud projecti uzaq serverə əlavə etmək üçün isə "git remote add" əmrindən istifadə edirik.
- **Git rm:** Seçilmiş faylı silir. məsələn biz git add komandası ilə əlavə etdiyimiz hansısa faylı sonradan silmək istədikdə istifadə edirik.
- **Git push:** Projectimizi server -ə (GitHub, GitLab) göndərmək üçün istifadə olunur.
- **Git pull** Uzaqdakı serverdəki layihə fayllarına bir dəyişiklik və ya əlavə varsa, onları yerli layihəmizdəki fayllarla birləşdirir.