Czym jest repozytorium?

* 1. To rozwiązanie informatyczne służące przede wszystkim do magazynowania, składowania i upowszechniania różnego rodzaju plików. Różnią się od bibliotek cyfrowych i stron które ujawniają najnowsze wersje plików tym, że przechowują one oprócz aktualnych kopii ich archiwalne wersje.
  2. Obowiązki repozytorium to segregacja i kategoryzowanie danych oddanych do przetrzymania
  3. Istnieją różne typy repozytoriów tj.
     + **Repozytorium instytucjonalne** – zakładane przez instytucje. Stosowane do przechowywania i promowania prac, których twórcami są jednostki należące do danej organizacji. Poza tym takie repozytorium pozwala na zarządzanie instytucją w pewien sposób oraz na przygotowywanie bardzo detalicznych statystyk. **Przykłady**: DASH (Digital Access to Scholarship at Harvard), DSpace@MIT, repo.agh.edu.pl
     + **Repozytorium dziedzinowe** – osoby zajmujące się daną specjalizacją mogą wrzucać wyniki swoich doświadczeń, publikacje i efekty pracy. **Przykład**: arXiv
     + **Repozytorium projektowe** – Repozytorium projektowe odnosi się do centralnego składowiska, w którym przechowywane są różnorodne pliki, dokumenty, dane i inne zasoby związane z danym projektem. Służy do organizacji, przechowywania i udostępniania materiałów projektowych w celu ułatwienia współpracy i zarządzania projektem przez zespół projektowy. Repozytoria projektowe często służą do przechowywania plików takich jak dokumenty projektowe, specyfikacje, plany, grafiki, modele 3D i inne powiązane z projektem zasoby. **Przykłady**: Github, Gitlab, OneDrive for business

1. Zarządzanie repozytorium
   1. Inicjowanie – git init
      * Git init – inicjalizuje repozytorium GIT w lokalnym katalogu
   2. Klonowanie – git clone
      * Git clone {adres repozytorium} – klonujemy repozytorium do obecnego katalogu
   3. Dodawanie –
      * git add {path} – dodaje plik do repozytorium (komenda/ zdjecie z githuba)
      * git add -A – dodaje wszystkie pliki do repozytorium które w nim nie są
   4. Zatwierdzanie zmian
      * git commit – tworzy commita z aktualnie zmienionych plików
      * Git commit -m „wiadomość” – tworzy commita z wiadomością
   5. Zarządzanie branchami - \*szybki opis co to branche\*;
      * git branch – pokazuje listę gałęzi w repozytorium
      * git checkout (-b) {nazwa gałęzi} – przełącza na podaną gałąź (tworzy nową gałąż o danej nazwie
      * git merge {nazwa gałęzi} – dołączenie zmian ze wskazanej gałęzi
      * git rebase {nazwa gałęzi} – dołączenie zmian ze wskazanej gałęzi z zachowaniem kolejności wprowadzania zmian
   6. Zarządzanie zdalnymi repo
      * git remote add
      * git remote remove {repo} – usuwa wskazane repozytorium
      * git push – wypycha zmiany na wskazaną gałąź
      * git pull – pobierz zmiany z aktualnego brancha
      * git fetch – kasuje branche już nie istniejące na głównym repo
   7. Rozwiązywanie konfliktów scalania

Czym jest commit – zatwierdzenie zmian zapisanych w „staging area” (tam się dodają pliki z git add)