

# M05 PRACA ZDALNA

## L04 Repo "hostowane"

(Bo kodem warto się dzielić)

---

Maciej Aniserowicz

# Do "normalnego" repozytorium...

... nie wysyłamy zmian

Szczególnie jeśli ktoś na nim pracuje!

# BARE repository

"Bare": gołe, nagie, ... , **proste**

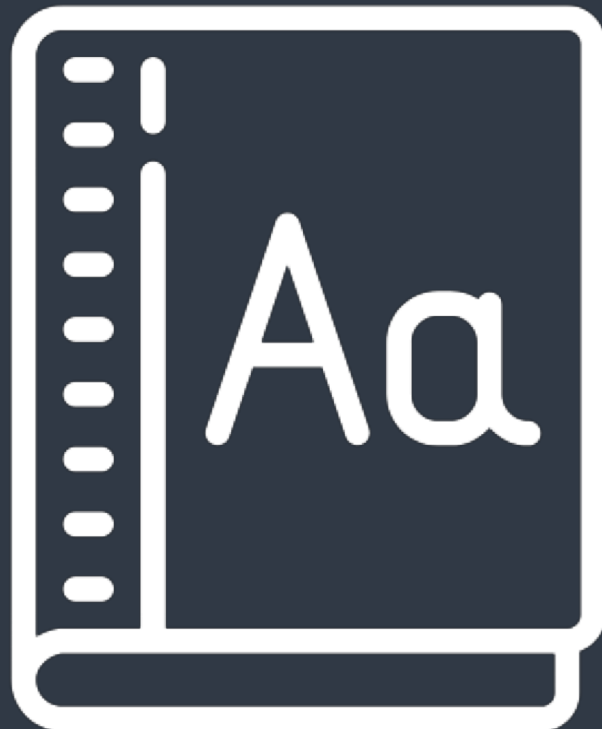
## Bare repository

---


Repozytorium **bez working copy**.

Serwerowe.

Służy do interakcji między innymi repozytoriami (klienckimi).



## Bare demo

- 
1. `mkdir server_repo.git`
  2. `cd server_repo.git`
  3. `git init --bare`

## BEST PRACTICE

---

Nazwa katalogu z bare repo  
kończy się na **.git**

(mimo, że to katalog, a nie plik!)



# Bez Working Copy

Nie ma katalogu ".git".

Wszystko jest "na wierzchu"!

... "nagie"

## Po co?

Do takiego repo można wysyłać swoje zmiany.

Do tego ono służy! Do dzielenia się commitami.

Jest to klasyczne "repozytorium zdalne".

Jego rolą jest wymiana informacji między programistami



# Połączenie między repozytoriami

[repo 1]

1. `git remote add origin ../server_repo.git`

dodaje połączenie do naszego nowego *bare repo*

2. `git remote show origin`

# Połączenie między repozytoriami

[repo 2]

1. `git remote`
2. `git remote remove origin`  
usuwamy bezpośredni "link" do pierwszego
3. `git remote add origin ../server_repo.git`  
i dodajemy link do nowego
4. `git remote show origin`

# Uwaga

MOŻNA (**technicznie**) bez bare repo wymieniać się kodem!

MOŻNA **wysłać komuś** inną gałąź niż tę, na której pracuje.

(komunikacja **peer-to-peer**)

Ale tego używa się w **bardzo wyjątkowych** przypadkach.

## Podsumowanie

---

- `git init --bare`
- `git remote add`
- `git remote remove`



## ZADANIE

---

- ✓ Stwórz **bare** repo
- ✓ Dodaj je jako **remote** w **obu** dotychczasowych repozytoriach

