

WRZUCANIE NA GITHUBA

1. Najpierw trzeba przejść do katalogu, w którym jest projekt i katalog ".git" wpisując np.:
`cd /media/sf_php_projekty/prework/Pework_CSS_Coderslab` (tabulator podpowiada)
2. Commitujemy zmiany instrukcją:
`git commit -am "task 1-2"` (w cudzysłowie wpisujemy komentarza do naniesionych zmian)
3. Wrzucamy zmiany na githuba
`git push origin master`

Przydatne:

`git branch` - pokaże nam z której gałęzi korzystamy
`git remote show origin` - pokazuje, gdzie będą wypychane zmiany na githuba
`git clone git@github.com:wojtekzydowski/Pework_CSS_CodersLab.git` - klonujemy repozytorium z Githuba na swój komputer (aby wkleić skopiowany link z githuba należy dać SHIFT + (FN) + Inster)

LINUX (UNBUNTU)

`Ctrl + h` - pokazuje/ukrywa ukryte katalogi
`Ctrl + Alt + t` - wywołanie terminala

`mkdir nazwa_katalogu` - tworzy katalog 'nazwa_katalogu' w bieżącym katalogu
`mkdir nazwa_katalogu/podkatalog` - tworzy katalog 'podkatalog' w katalogu 'nazwa_katalogu'
`touch nazwa_pliku` -> tworzy plik 'nazwa_pliku' w bieżącym katalogu 'nazwa_katalogu'

`ls` -> pokazuje listę plików i podkatalogów w danym katalogu
`ls -l` -> wyświetlenie zawartości katalogu z dodatkowymi informacjami (np. właściciel pliku/katalogu, prawa dostępu)

`cd nazwa_katalogu` -> wejście do katalogu
`cd /` -> wejście do katalogu root
`cd ..` -> wejście do katalogu nadrzędnego

`rm nazwa_pliku` -> usuwanie pliku 'nazwa_pliku'
`rm -r nazwa_katalogu` -> usuwanie katalogu nazwa_katalogu i jego podkatalogów

Problem z uruchomieniem dysku z danymi:
`sudo ntfsfix /dev/sdb1`

KONFIGURACJA GITA

Ustalanie użytkownika i adresu email
`git config --global user.name "Imię i nazwisko"`
`git config --global user.email "adres email"`

ZMIANA EDYTORA TEKSTU NA NANO

`git config --global core.editor nano`

GENEROWANIE PUBLICZNEGO KLUCZA SSH

```
ssh-keygen -t rsa -b 4096 -C "your_email@example.com"
```

Następnie Enter i wybór hasła.

Tak wygenerowany klucz trzeba dodać ssh-agent

Sprawdzenie, czy ssh-agent jest dostępny

```
eval "$(ssh-agent -s)" / powinno pojawić się coś takiego: Agent pid 4805
```

Następnie trzeba dodać klucz ssh do agent-ssh:

```
ssh-add ~/.ssh/id_rsa
```

Ostatnim krokiem jest dodanie klucza ssh do Githuba:

1. Skopiuj klucz ssh do schowka:

```
sudo apt-get install xclip
```

Ściąga i instaluje xclip.

```
xclip -sel clip < ~/.ssh/id_rsa.pub
```

Kopiuj zawartość pliku id_rsa.pub do schowka

2. Dodanie klucza w Githubie

Trzeba na ikonce ze zdjęciem zaznaczyć „Settings” a następnie „SSH and GPG keys” i tam dodać nowy klucz wklejając ze schowka zawartość pliku id_rsa.pub i nazywając komputer z którego ten klucz skopiowano (np. Laptop Asus).

TWORZENIE LOKALNEGO I ZDALNEGO REPOZYTORIUM

```
git init
```

-utworzy ukryty katalog .git w lokalnym katalogu (na razie puste repozytorium), w którym wywołano git init

Przy tworzeniu lokalnego repozytorium najlepiej od razu wpisać:

```
git init
```

```
git add .
```

```
git commit -m 'initial commit'
```

a następnie wskazać zdalne repozytorium na githubie np.

```
git remote add origin https://github.com/wojtekszydowski/prework.git
```

```
git push origin master
```

lub

```
git push -u origin master – wtedy gałąź lokalna będzie śledzić jej zdalny odpowiednik
```

aby sprawdzić jakie repozytorium zdalne jest przyłączone należy wpisać

```
git remote -v
```

GIT - IGNOROWANIE PLIKÓW

Stwórz plik .gitignore

W nim zapisuje się co ma być ignorowane.

Zasady:

- Puste linie lub linie rozpoczynające się od # są ignorowane.
- Działają standardowe wyrażenia glob.
- Możesz zakończyć wyrażenie znakiem ukośnika (/) aby sprecyzować, że chodzi o katalog.
- Możesz negować wyrażenia rozpoczynając je wykrzyknikiem (!).

Przykłady:

```
# komentarz - ta linia jest ignorowana
# żadnych plików .a
*.a
# ale uwzględniaj lib.a, pomimo ignorowania .a w linijce powyżej
!lib.a
# ignoruj plik TODO w katalogu głównym, ale nie podkatalog/TODO
/TODO
# ignoruj wszystkie pliki znajdujące się w katalogu build/
build/
# ignoruj doc/notatki.txt, ale nie doc/server/arch.txt
doc/*.txt
#ignoruj pliki kończące się na .o lub .a
*.[oa]
#ignoruj wszystkie pliki, które nazwy kończą się tyldą (~)
*~
```

GIT - PODSTAWOWE KOMENDY

git add -A

-obejmuje wszystkie pliki i katalogi kontrolą wersji - oznacza je jako śledzone

git add nazwa_pliku

-obejmuje kontrolą wersji plik nazwa_pliku - oznacza plik do śledzenia

git add *.html - wszystkie pliki html

git add ?la.txt - wszystkie pliki zaczynające się dowolną literą, a następnie mające w nazwie la.txt

git commit -m "Krótki opis rewizji np. co zmieniono w tej rewizji"

-umieszczenie zmian w repozytorium git

WAŻNE (tak powinniśmy kończyć pracę - zapisać wszystkie zmiany do repozytorium):

git add -A

git commit -m "Krótki opis rewizji np. co zmieniono w tej rewizji"

GIT - SPRAWDZANIE HISTORII REWIZJI

git log

-sprawdzanie historii rewizji

git log -p -2

-pokazuje różnice wprowadzone z każdą rewizją ograniczając do ostatnich 2 rewizji

git log --pretty=oneline

-sprawdza historię rewizji i zapisuje w skróconej wersji w jednej linii

git log --abbrev-commit --pretty=oneline

-pokazuje skrócone nazwy rewizji

git log --pretty=format:"%h, %ar, %s" (ważne - po format jest : i później bez spacji reszta)

-pozwala określić własny wygląd i format informacji wyświetlanych poleceniem log - zmienne:

%H Suma kontrolna zmiany

%h Skrócona suma kontrolna zmiany

%T Suma kontrolna drzewa

%t Skrócona suma kontrolna drzewa

%P Sumy kontrolne rodziców

%p Skrócone sumy kontrolne rodziców

%an Nazwisko autora

%ae Adres e-mail autora

%ad Data autora (format respektuje opcję -date=)

%ar Względna data autora

%cn Nazwisko zatwierdzającego zmiany

%ce Adres e-mail zatwierdzającego zmiany

%cd Data zatwierdzającego zmiany

%cr Data zatwierdzającego zmiany, względna

%s Temat

Pozostałe opcje git log:

-p Pokaż pod każdą zmianą powiązaną łątkę

--stat Pokaż pod każdą zmianą statystyki zmodyfikowanych plików

--shortstat Pokaż wyłącznie zmienione/wstawione/usunięte linie z polecenia --stat

--name-only Pokaż pod każdą zmianą listę zmodyfikowanych plików

--name-status Pokaż listę plików o dodanych/zmodyfikowanych/usuniętych informacjach.

--abbrev-commit Pokaż tylko pierwsze kilka znaków (zamiast 40-tu) sumy kontrolnej SHA-1.

--relative-date Pokaż datę w formacie względnym (np. 2 tygodnie temu)

--graph Pokaż graf ASCII gałęzi oraz historię scaleń obok wyniku.

--pretty Pokaż zatwierdzone zmiany w poprawionym formacie. Dostępne opcje obejmują oneline, short, full, fuller oraz format (gdzie określa własny format)

GIT - POPRAWKI I COFANIE ZMIAN

git commit --amend

-poprawka do ostatniej rewizji

git reset HEAD nazwa-pliku

-usuwa plik z poczekalni (np. po tym jak się dodało go do poczekalni git add -A)

GIT - GAŁĘZIE

git branch

-pokazuje jakie są dostępne gałęzie i oznacza * gałąź na której obecnie jesteś (aktywną)

git branch nazwa_galezi

-tworzy nową gałąź o nazwie nazwa_galezi

git checkout nazwa_galezi

-przełączenie się (przejście) na gałąź o nazwie nazwa_galezi

git checkout master -przełączenie się na gałąź główną (lokalnie)

git merge nazwa_galezi - scalenie dwóch gałęzi w jedną (trzeba najpierw przełączyć się na gałąź, do której chcesz zmiany scalić - do tej wyższej/główniejszej gałęzi)

git branch -d nazwa_galezi - usunięcie gałęzi

git branch --merged

-pokazuje, które gałęzie już zostały scalone do gałęzi aktywnej (można wtedy spokojnie użyć git branch -d nazwa_galezi, aby usunąć wtedy już zbędną gałąź)

git branch --no-merged

-pokazuje, które gałęzie nie zostały jeszcze scalone do gałęzi aktywnej (aby usunąć taką niescaloną jeszcze gałąź to trzeba użyć git branch -D nazwa_galezi - duża litera D zamiast małej)

GIT - POZOSTAŁE KOMENDY

git status

-sprawdzenie stanu plików - informuje czy pliki są aktualne

git diff

-pokazuje co zmienione w poszczególnych plikach - które linie zostały dodane, a które usunięte (zmiany spoza poczekalni)

git diff --cached

-pokazuje zmiany wysłane dotąd do poczekalni

git rm nazwa_pliku

-usuwa z repozytorium plik nazwa_pliku, a także kasuje go z katalogu roboczego (plik musiał być wcześniej dodany do repozytorium)

<https://git-scm.com/book/pl/v1/Ga%C5%82%C4%99zie-Gita-Ga%C5%82%C4%99zie-zdalne>

Do przerobienia (pominięte):

<https://git-scm.com/book/pl/v1/Podstawy-Gita-Praca-ze-zdalnym-repozytorium>