Mikołajkowe laborki

Patryk Jar
Karol Pisarek
Kamil Pakur

06/12/2019

Wyłączenie plików z wersjonowania
plik: .gitignore

Pliki które nie powinny być wersjonowane:

- ✓ pliki generowane w ramach budowania projektu (np. katalog build)
- ✓ pliki tworzone przez IDE
- ✓ pliki zewnętrznych bibliotek (gdy są pobierane z innego źródła)
- ✓ pliki logów
- ✓ pliki tymczasowe

Aby celowo wyłączyć śledzenie tych plików przez GITa należy skonfigurować plik zawierający ignorowane wzorce plików/katalogów

.gitignore – plik wersjonowany (wzorce współdzielone)

.git/info/exclude – plik nie podlegający wersjonowaniu (wzorce prywatne)

kainos®

Wyłączenie plików z wersjonowania

plik: .gitignore

Zasady tworzenia wzorców

- ✓ puste linie i zaczynające się od # *są ignorowane*
- ✓ * *oznacza dowolny ciąg znaków a ? pojedynczy znak poza /*
- ✓ można korzystać z operatora zakresów znaków [a-zA-Z]
- ✓ gdy wzorzec zawiera / na początku bądź w środku dopasowywanie jest relatywne względem katalogu zawierającego .gitignore
- ✓ gdy wzorzec zawiera / na końcu dopasowane zostaną katalogi w całym drzewie katalogów
- ✓ gdy wzorzec nie zawiera / dopasowane mogą zostać zarówno pliki jak i katalogi w całym drzewie katalogów
- ✓ ! umożliwia zanegowanie wzorca poprzedzającego

kainos®

Wyłączenie plików z wersjonowania
plik: .gitignore

Przykłady Wzorców	
Wzorzec	Pasujące ścieżki
project/build lub /project/build	wyłącznie katalog lub plik project/build
build/	wszystkie podkatalogi o nazwie build np. build, project/build, project/module/build
build	wszystkie podkatalogi lub pliki o nazwie build
*.log	wszystkie pliki o rozszerzeniu log

Dodatkowe informacje: https://git-scm.com/docs/gitignore
Generator .gitignore: https://www.gitignore.io

Wyłączenie plików z wersjonowania

plik: .gitignore

```
joannar@JANNI ~/GitCourse/HelloWorld (master) $ cat .gitignore
*.log
ignoreThisFile.txt
ignoreThisFolder/
joannar@JANNI ~/GitCourse/HelloWorld (master) $ touch ignoreThisFile.txt file.log HelloWorld.txt
joannar@JANNI ~/GitCourse/HelloWorld (master) $ mkdir ignoreThisFolder
joannar@JANNI ~/GitCourse/HelloWorld (master) $ touch ignoreThisFolder/file.txt
joannar@JANNI ~/GitCourse/HelloWorld (master) $ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        .gitignore
        HelloWorld.txt

nothing added to commit but untracked files present (use "git add" to track)
```

kainos®

Zadanie 1

1. Stwórz własny branch z najświeższego mastera
2. Dodaj folder na swoje prezenty (np. “Prezenty Jana Kowalskiego”)
3. Stwórz plik 2019.txt. Powinien znajdować się “obok” folderu (**nie ”w”**) ”Prezenty…”

Ściągawka

$ git checkout –b – *twórzenie brancha*

Zadanie 1

1. Wpisz coś do pliku 2019.txt
2. Upewnij się, że plik 2019.txt nie trafi do św. Mikołaja!
    1. (stwórz .gitignore i dodaj regułę pasującą do pliku 2019.txt)

Ściągawka

$ nano 2019.txt
$ touch .gitignore – *stworzenie pliku .gitignore*

Zadanie 1

1. Stwórz merge request (pamiętaj o folderze!)
    1. .gitkeep

Ściągawka

Gitlab.com

kainos®

Zadanie 1

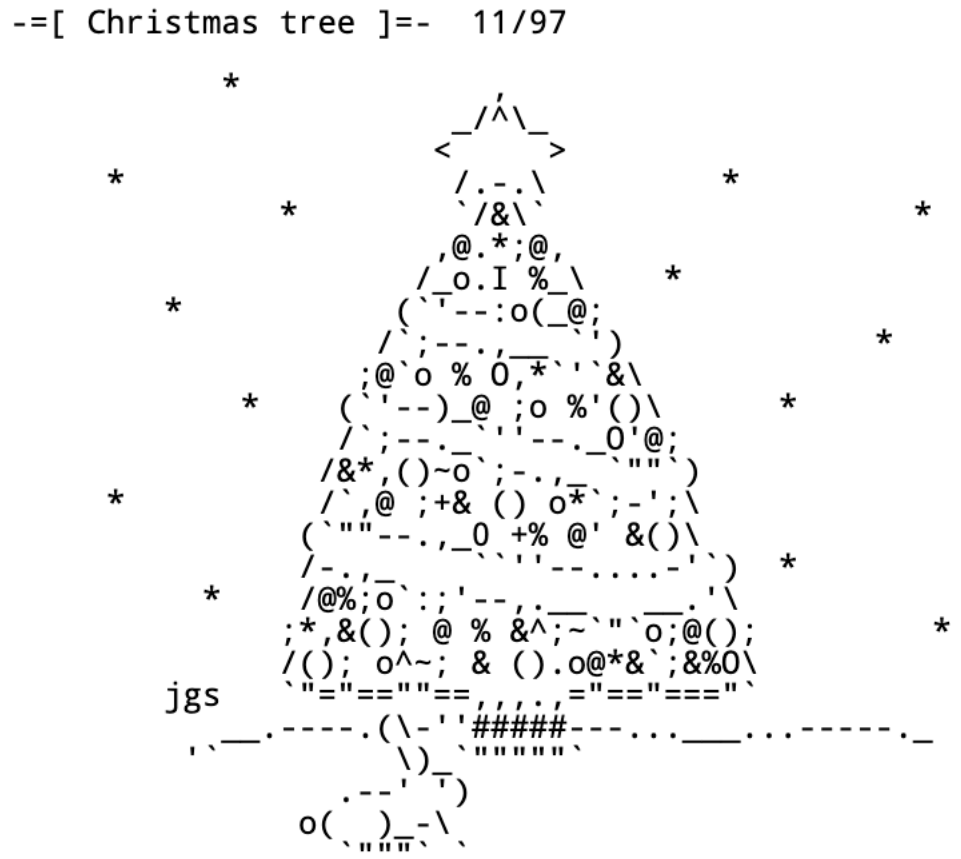
1. Podzielcie się na grupy 2-3 osobowe.
2. Umieście sobie nawzajem prezenty w katalogach z prezentami.
3. Dodajcie komentarze i poprawnie listy prezentów.

Podpowiedź

Gdy nie podoba Ci się prezent możesz dać:
+ komentarz w merge request z opisem, co byś chciał
+ dać „łapkę w dół”

kainos®

Choinka – popraw ją!



Komendy, by zacząć

```
$ git checkout master
$ git pull
```

kainos®

Cofanie niedodanych zmian

polecenie: git checkout

Polecenie cofa wszystkie zmiany do ostatniego „adda”.

git checkout .

git checkout ścieżka/do/pliku

Odrzucone prezenty!

git add . [TYLKO add! Bez commit!!!]

Cofanie dodania

polecenie: git *unadd*

Nie ma takiego polecenia ☹

Cofanie dodania

polecenie: git reset

Dodaję:

git add .

O nie, nie chciałem wcale!

git reset .

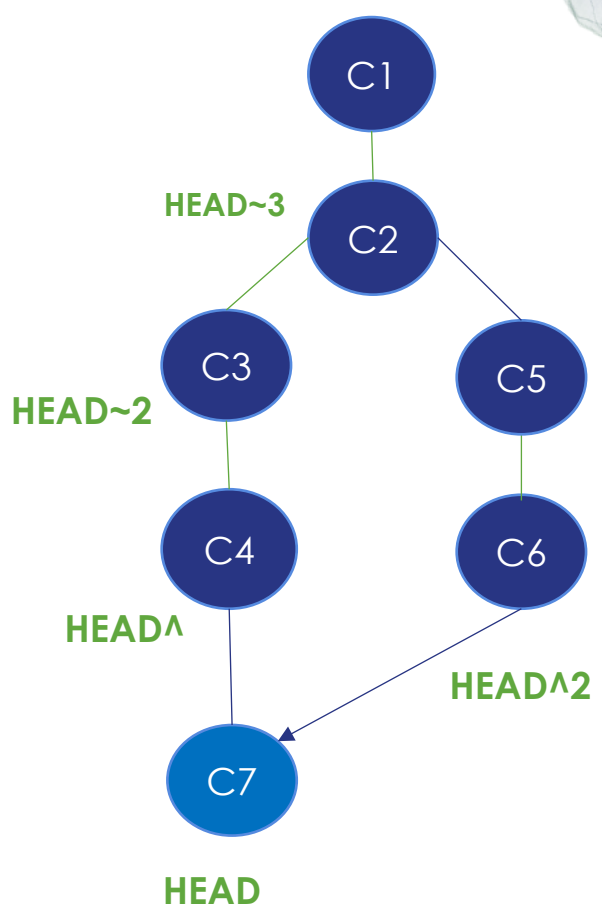
Albo, gdy tylko wybrany plik/folder:

git reset ścieżka/do/pliku

Historia commitów

Sposoby, które umożliwiają wskazanie konkretnego commit:

- ✓ **commit-id** (suma kontrolna SHA-1)
  - ✓ c6b5451ab6df449fed1a40ccf6b18a82ab7c493c
  - ✓ c6b5451
- ✓ **branch**
- ✓ **HEAD** – aktualny commit
- ✓ **HEAD^** - rodzic aktualnego commita (inaczej z **HEAD~**)
- ✓ **HEAD^2** – rodzic rodzica wybranego commita
- ✓ **HEAD~3** – trzeci przodek wybranego commita, itd.



kainos®

Cofanie lokalnego commita

polecenie: git reset *HEAD*

Stwórz 3 commity lokalnie (nie pushuj).

Go, go, go! ☺

Wykonaj polecenie:

git status

Cofnij się do stanu kodu sprzed 2 commitów

Cofanie - powtórzenie

polecenia: git checkout

Cofnięcie wszystkiego, co nie jest dodane

Uwaga!

To nie usunie nowych plików, a jedynie to, co zostało zmienione w plikach zarządzanych przez git.

kainos®

Cofanie - powtórzenie

polecenia: git reset *plik*

„Oddodaje” plik do gita. (jakby `git unadd plik`)

Można wskazać cały folder.

Uwaga!

Nie cofa zmian. Jedynie sprawia, że git „zapomina” o nich.

Cofanie - powtórzenie

polecenia: git reset *HEAD~N*

Cofa lokalnie o **N** commitów w historii gita.

Uwaga!

Jeśli cofnie się dalej niż tylko lokalna historia może nie udać się zrobić push.
Git uzna, że lokalna wersja jest nieaktualna.

kainos®