Podstawy GIT

Hello!

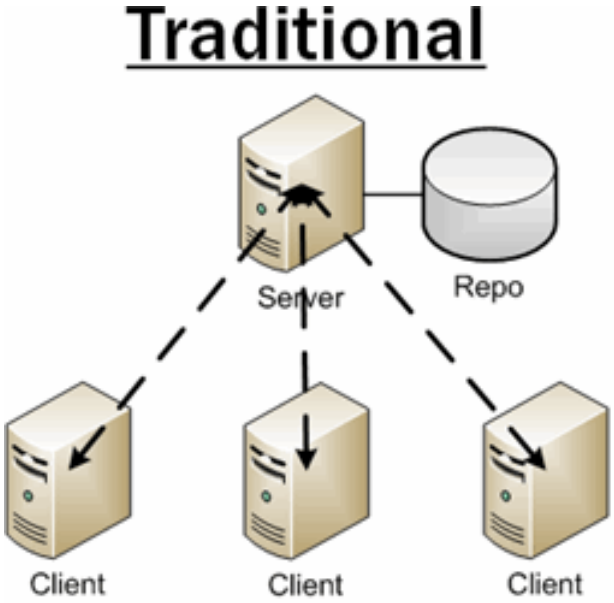
Tomasz Lisowski

Software developer, JIT Solutions
IT trainer

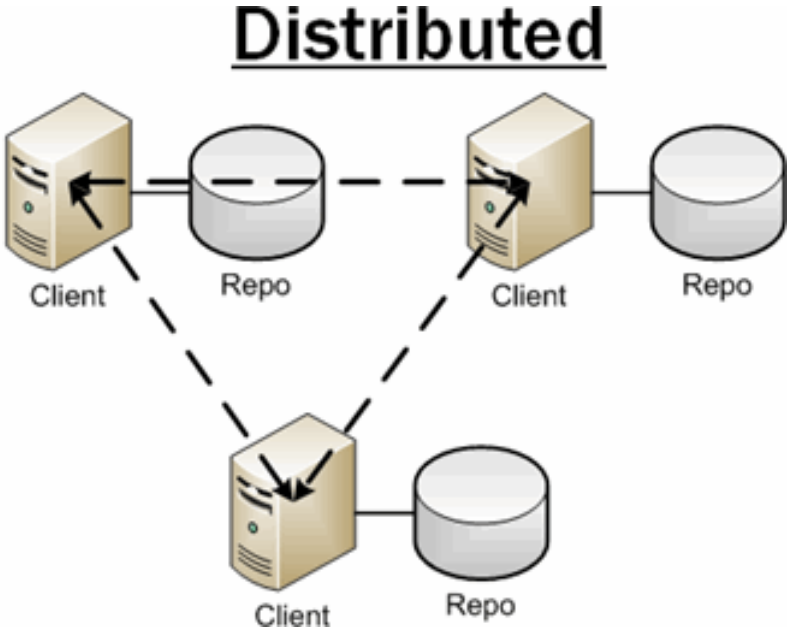
Agenda

- system kontroli wersji
- podstawowe operacje
- branche
- push/pull

System kontroli wersji



Istnieje tylko jedno centralne repozytorium



Każdy ma swoje lokalne repozytorium. Serwer też ma swoje repo.

GIT
lokalne repozytorium

- klonowanie pobiera na dysk całe repozytorium
- ..razem z historią commitów
- ..i wszystkimi branchami
- tworzy lokalną bazę danych projektu

GIT
zalety

- rozproszony
- bardzo wydajny
- bezpieczny
- elastyczny (nieliniowe flow pracy)
- darmowy (open source)
- aktualny standard

GIT
narzędzia

- polecenia bezpośrednio z konsoli
- konsola “Git for windows”
- lub z IDE
- ..ale niektóre operacje mogą się inaczej nazywać

Podstawowe operacje

Operacje
clone

- kopiuje istniejące repozytorium
- pobiera prawie wszystkie dane z serwera
- dostępna cała historia projektu
- możliwość odwtorzenia repozytorium

- ćwiczenie:
  sklonowanie projektu

Operacje
add

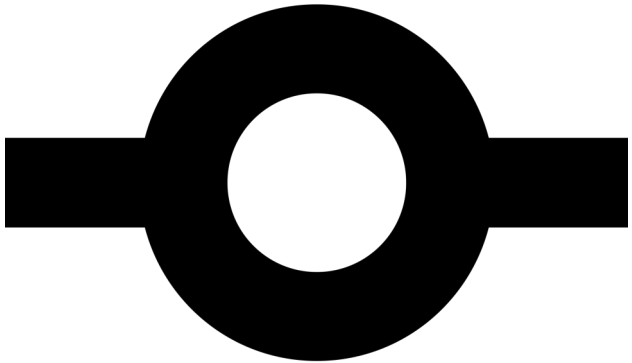
- dodaje pliki do śledzenia
- pliki w poczekalni
  “changes to be commited”
- zapisana wersja pliku z momentu operacji add
- git add <parametr>
  tutaj nazwa pliku lub .

- ćwiczenie:
  dodaj plik w projekcie i wykonaj operacje add

Operacje
commit

- zatwierdza zmiany
- zapisuje różnicę zawartości plików
- stały i niezmienny
- muszą posiadać wiadomość
- każdy commit jest unikalny (klucz SHA1)

- ćwiczenie:
  wykonaj commit swoich zmian



Operacje
push

- “wypycha” zmiany na serwer zewnętrzny
- git push [nazwa_repo] [nazwa_gałęzi]
  np. git push origin develop
- wymagane uprawnienia do zapisu
- ..oraz brak konfliktów

Operacje
pull

- pobranie i włączenie zmian
- wszystkie dane, których brak lokalnie
- próba automatycznego scalenia
- możliwe problemy
  - tutaj potrzebne ręcznie scalenie kodu

Jaki mamy status?

GIT
Status

- sprawdza stan plików
- pokazuje aktualną gałąź
- “*nothing to commit, working directory clean*”
- “*untracked files:*”
- *“Changes to be commited:”*

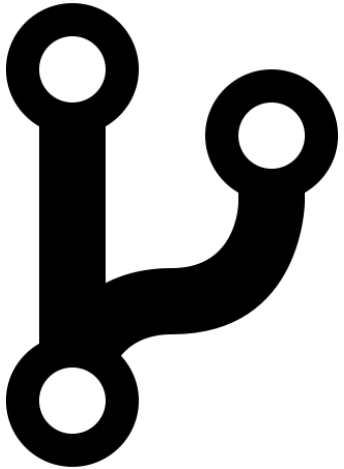
```
$ git status
On branch feature/JZ3W-22
Your branch is up-to-date with 'origin/feature/JZ3W-22'.
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   com.infoshareacademy.wave-soft-web/src/main/resources/configuration.properties
        modified:   com.infoshareacademy.wave-soft-web/src/main/resources/scripts/setup-datasource.cli

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   com.infoshareacademy.wave-soft-web/src/main/java/Hellojava.java
        modified:   com.infoshareacademy.wave-soft-web/src/main/java/UserListServlet.java
        modified:   com.infoshareacademy.wave-soft-web/src/main/resources/META-INF/persistence.xml
        modified:   com.infoshareacademy.wave-soft-web/src/main/webapp/findPart.jsp
        modified:   com.infoshareacademy.wave-soft-web/src/main/webapp/userList.jsp
```

GIT
Status - ćwiczenie

- sprawdź aktualny status
- wprowadź zmianę
- status?
- dodaj plik do śledzenia
- wprowadź zmianę

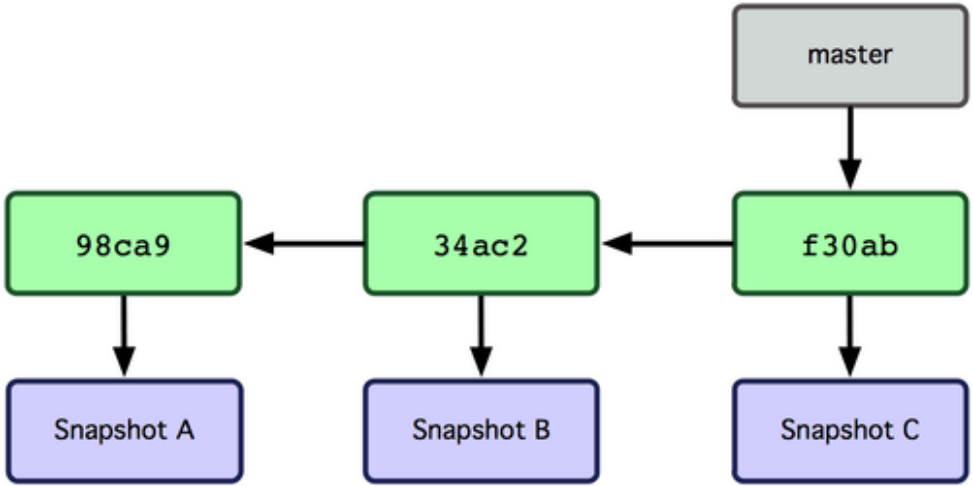
- wynik?

Branch / gałąź



GIT
Branch

- git branch [nazwa]
- wskaźnik na commit (!)
- można go zawsze dodać/usunąć/zmienić
- pierwszy i domyślny
  – master



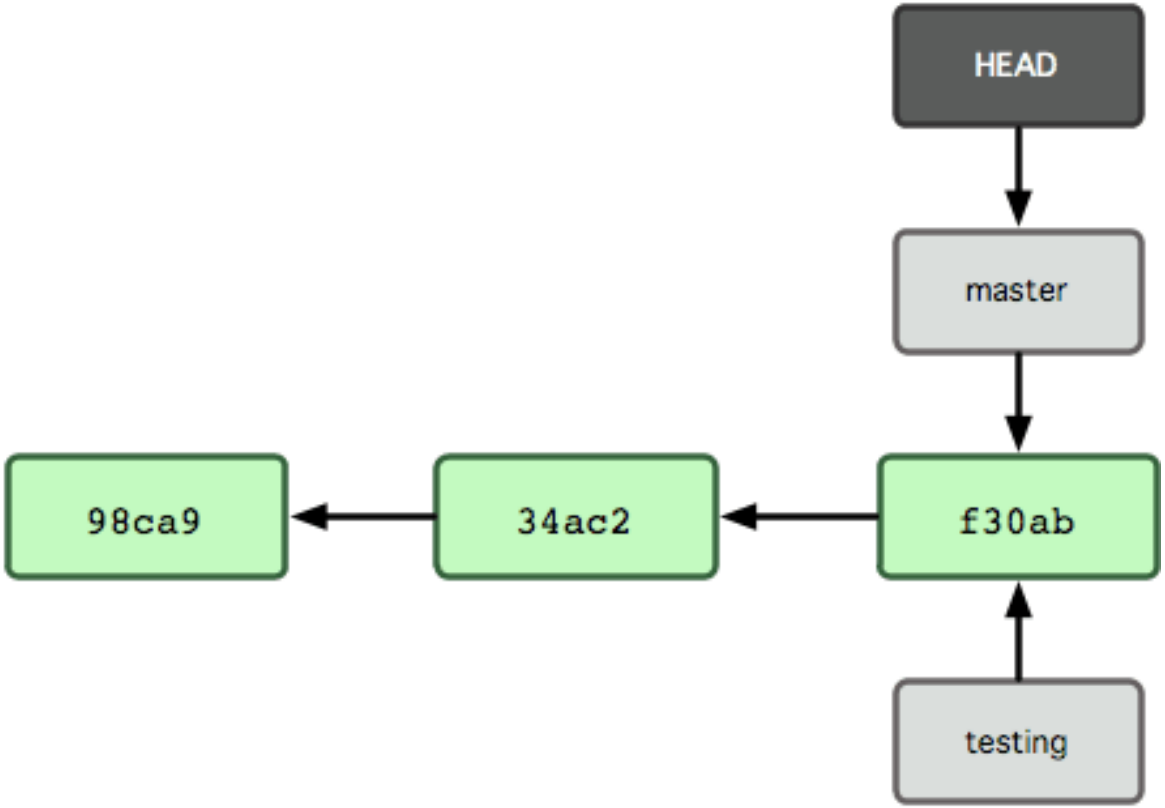
GIT
Branch - HEAD

- specjalny wskaźnik gałęzi
- wskazuje aktualną, lokalną gałąź
- *git branch* tworzy nową, ale nie zmienia HEADa
- *git checkout* – przełączanie gałęzi

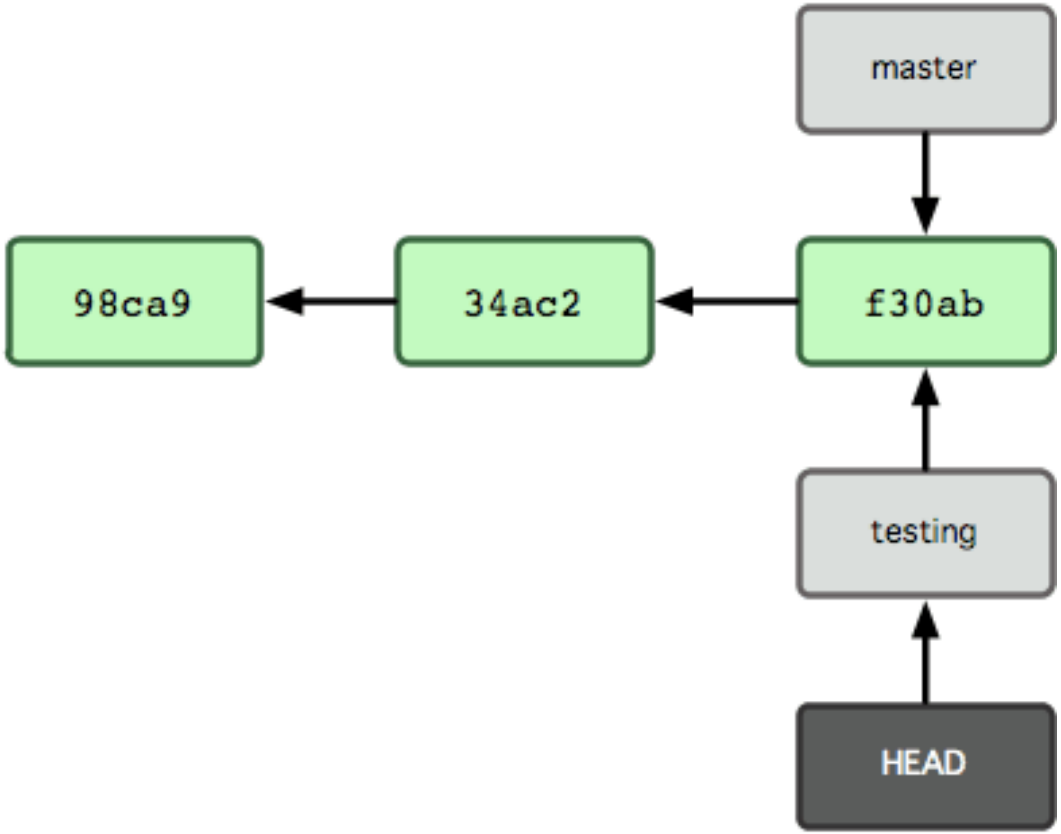
git branch newBranch
– tworzy branch *newBranch*

git checkout – b newBranch
– przełącza na branch *newBranch* (-b tworzy gdy brak)

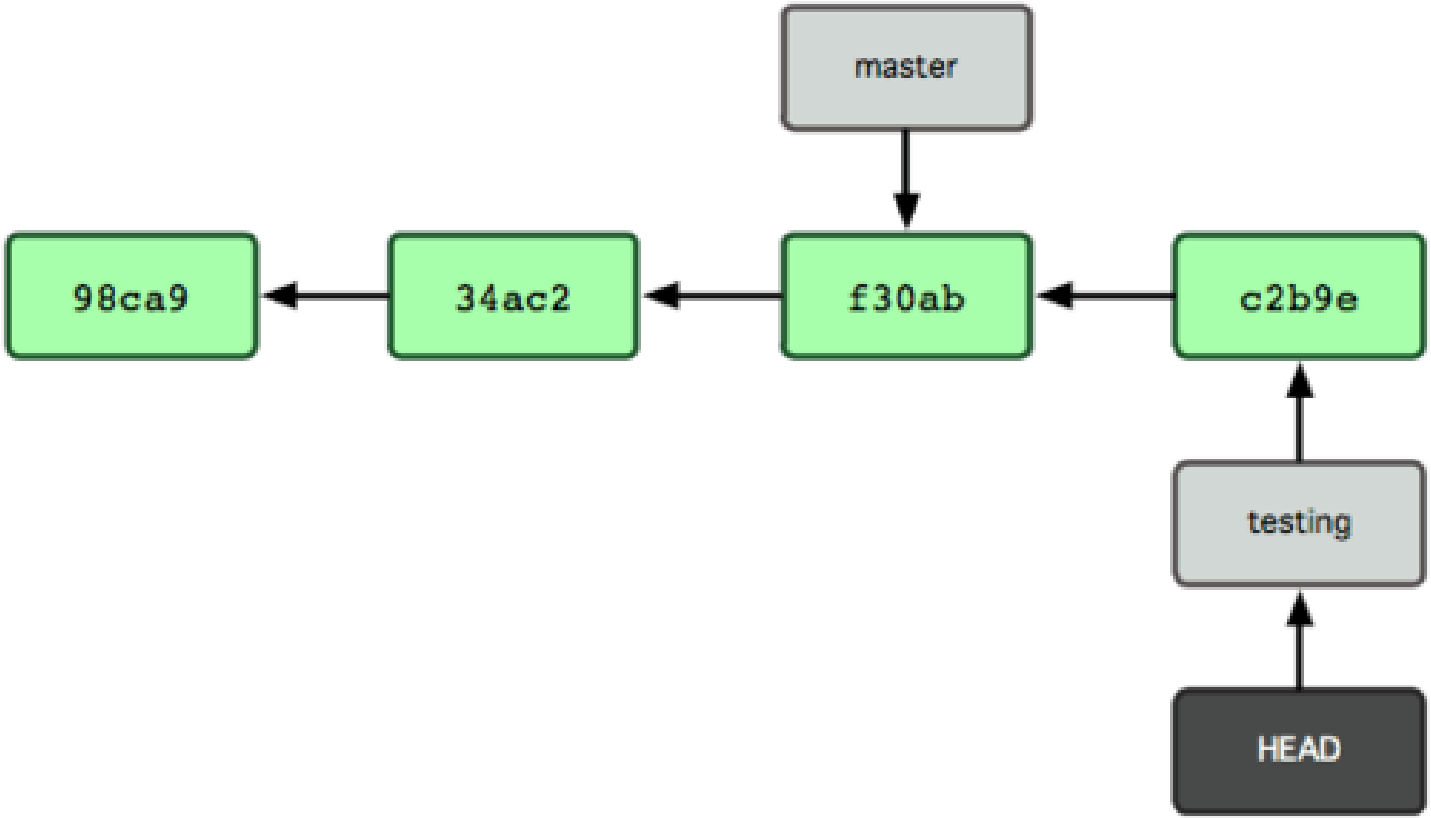
GIT
Branch - HEAD



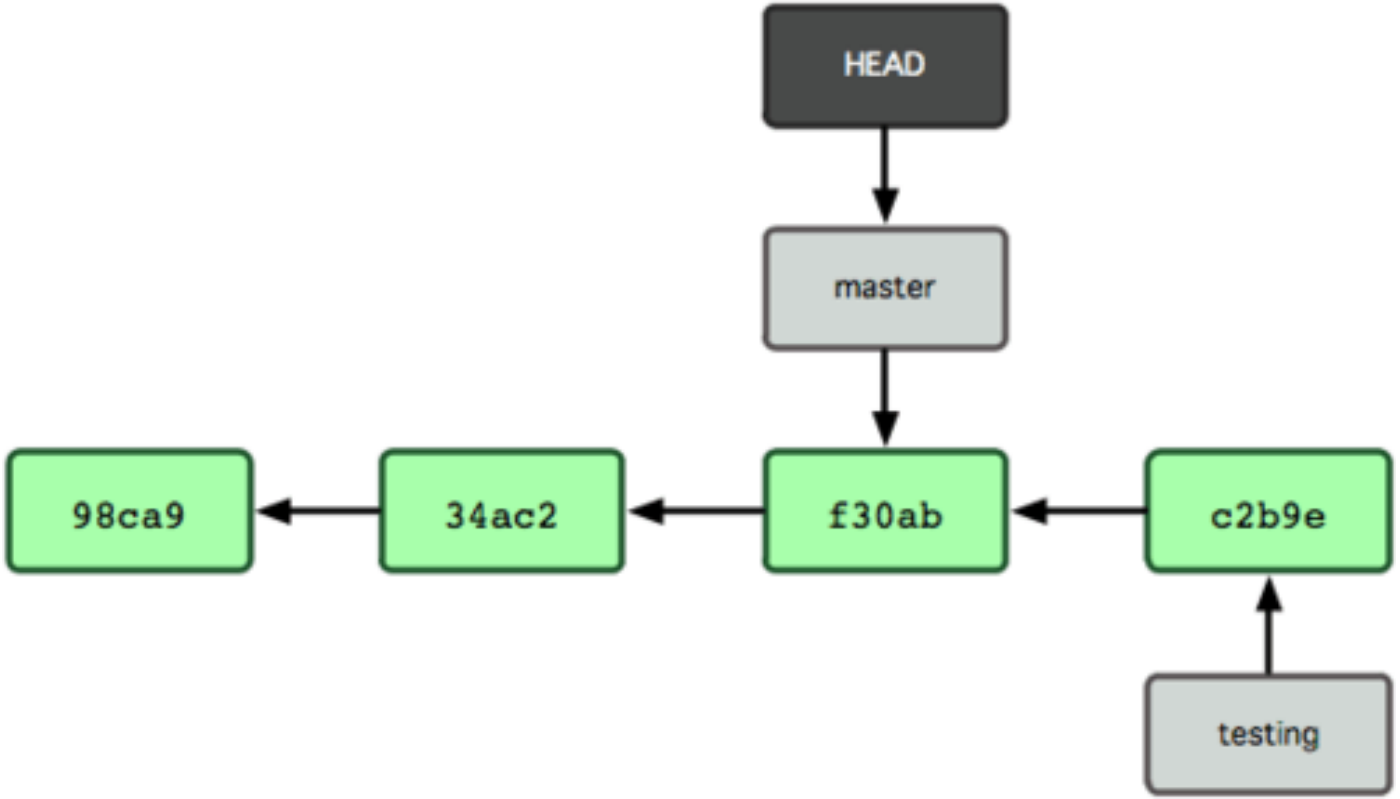
GIT
Branch - HEAD



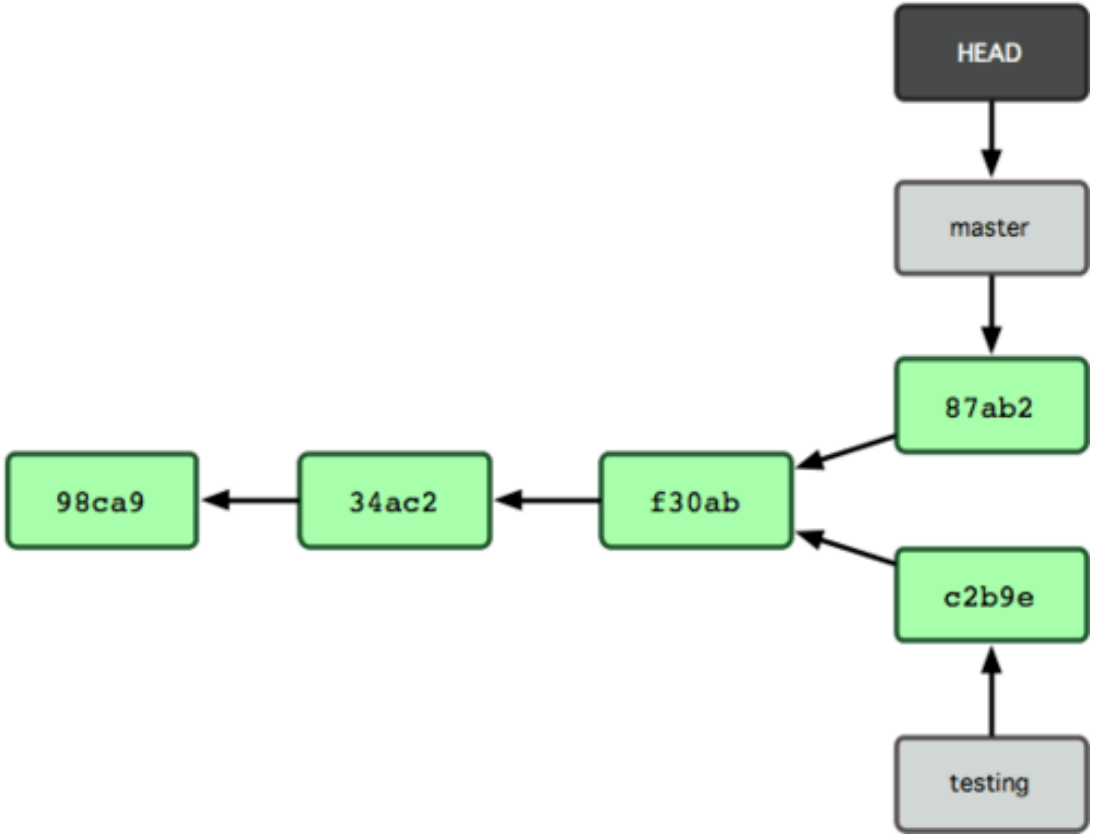
GIT
Branch - HEAD



GIT
Branch - HEAD

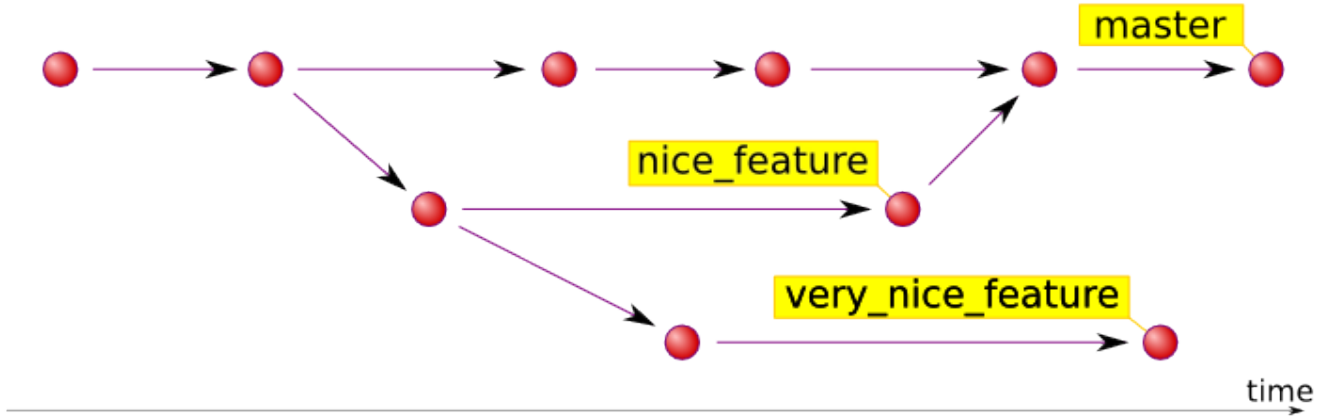


GIT
Branch - HEAD



GIT
Branch

- fizycznie branch jest tylko plikiem
- tani w tworzeniu I usuwaniu
- szybkie przełączanie
- w odróżnieniu od commitów są płynne i zmienne



GIT
Branch - ćwiczenie

- stwórz swój branch o nazwie *NazwiskoImie*
- wprowadź zmiany
- dodaj i wykonaj commit
- wykonaj PUSH swojej gałęzi
  git push origin NazwiskoImie
- sprawdź status
- sprawdź w przeglądarce czy zmiana znalazła się na serwerze githuba

GIT
Branch – ćwiczenie 2

- wykonaj operację PULL
  git pull origin
- przełącz się na branch drugiej osoby
  git checkout NazwaBrancha
- zobacz jej/jego zmiany lokalnie u siebie

Thanks!

Q&A

tomasz.lisowski@protonmail.ch