

# GIT workflow

**Nigdy nie  
commitujemy  
do mastera!**

## Feature branch

- `git checkout -b <feature branch>`
- pracujemy na feature branches (konwencja: te gałęzie są „prywatne”)

## Wystanie gałęzi na serwer

- `git push origin -u <feature branch>`
- nie bać się PUSH! (bo „konwencja prywatności”...)
- 3 cele:
  - cały zespół widzi co jest „in progress” (`git branch -a`)
  - potencjalnie code review
  - backup

## Praca (repeat)

- `git add -p`
- `git commit`
- regularny `git push`

## Porządki w historii

- `git rebase -i master`
- squash / edit / fixup / reword
- regularny `git push --force-with-lease` (jeden z niewielu przypadków gdy jest to dozwolone!)

## Integracja

- `git checkout master`
- `git pull`
- `git checkout -`
- `git rebase master -p`
- uwagi:
  - robić często, regularnie!
  - tutaj konflikty (nie boimy się ich, bo mamy RERERE)

## Merge

- `git checkout master`
- `git merge --no-ff --edit -`
- w „commit message” pełna treść zadania i uwagi

## Zmiany na serwer

- `git push`
- błąd?
  - wycofujemy merge: `git reset --hard ORIG_HEAD`
  - powtarzamy operację integracji

## Porządki

- `git push origin :<feature branch>`
- `git branch -d <feature branch>`
- `git remote update --prune`