PROGRAM SZKOLENIA NARZĘDZIA I DEVOPS

# Szkolenie Git

Zaczniesz swobodnie pracować na gałęziach, rozwiązywać konflikty i odzyskiwać zgubione zmiany.

| CZAS TRWANIA | POZIOM | FORMA |
| --- | --- | --- |
| **2 dni** | **od podstaw** | **warsztat dla zespołu, zdalnie lub stacjonarnie** |

Zgubiony commit, konflikt przy merge'u albo gałąź, której nikt nie rozumie, zjadają zespołowi godziny. Wiele osób zna pięć komend na pamięć, a przy każdej nietypowej sytuacji kasuje katalog i klonuje repozytorium od nowa.

Na szkoleniu pracujesz na prawdziwym repozytorium i celowo psujesz historię, żeby nauczyć się ją naprawiać. Wychodzisz z pewnością, że w Gicie prawie nic nie ginie na zawsze, i z nawykami, dzięki którym historię projektu da się czytać.

## PROGRAM

**2 dni**

- Instalacja i konfiguracja
  - `git config`: tożsamość, edytor i aliasy
  - Klucze SSH (ed25519) do uwierzytelniania na GitHubie
- Jak Git przechowuje historię
  - Katalog roboczy, poczekalnia (staging) i commit
  - Gałąź jako wskaźnik na commit
  - Ignorowanie plików: `.gitignore`
  - Końce linii, gdy zespół pracuje na Windowsie i macOS: `.gitattributes`
- Commity
  - Codzienny cykl: `git status`, `git diff`, `git add -p`, `git commit`
  - Czytanie historii: `git log --graph`, `git show`, `git blame`
  - Poprawianie ostatniego commita (`--amend`)
  - Cofanie zmian w plikach: `git restore`
  - Cofanie commitów: `git revert` i `git reset`
- Gałęzie i tagi
  - Tworzenie i przełączanie gałęzi: `git switch`
  - Tagi i wersjonowanie wydań
  - Strategie pracy z gałęziami: GitHub Flow, trunk-based, GitFlow

- Łączenie zmian
  - `git merge` i `git rebase`: kiedy które
  - Rozwiązywanie konfliktów przy merge'u i rebase'ie
  - Interaktywny rebase: porządkowanie historii przed pull requestem
- Praca w zespole ze zdalnym repozytorium
  - `git fetch`, `git pull` i `git push`
  - Fork i pull request na GitHubie
  - `gh`, czyli GitHub CLI: pull requesty i issues z terminala
- Ratunek i szukanie błędów
  - Odkładanie niedokończonej pracy: `git stash`
  - Przenoszenie commitów: `git cherry-pick`
  - Odzyskiwanie zgubionych zmian: `git reflog`
  - Szukanie commita, który wprowadził błąd: `git bisect`
- Narzędzia i dalsze kroki
  - Historia w edytorze (VS Code, JetBrains) i lazygit w terminalu
  - `git worktree`: kilka gałęzi naraz w osobnych katalogach
  - Podpisywanie commitów kluczem SSH i oznaczenie „Verified"
  - Nadchodzący Git 3.0: `main`, SHA-256 i reftable jako domyślne w nowych repozytoriach, choć GitHub nie przyjmuje jeszcze repozytoriów SHA-256

Jak przeglądać cudze zmiany w pull requeście, uczysz się na szkoleniu Code review.

## DLA KOGO?

- Dla osób, które zaczynają pracę z Gitem.
- Dla programistów, którzy znają kilka komend i boją się reszty.
- Dla zespołów, które przechodzą na pracę z pull requestami.

## CO MUSISZ UMIEĆ?

- Pracować w terminalu na poziomie podstawowym: poruszać się po katalogach i uruchamiać komendy.
- Edytować pliki tekstowe w dowolnym edytorze kodu.
- Mieć konto na GitHubie i zainstalowanego Gita. Gita nie musisz znać.

## CZEGO SIĘ NAUCZYSZ?

- Narysujesz, co dzieje się z gałęziami przy merge'u i rebase'ie.
- Cofniesz zmianę w pliku albo cały commit bez utraty pracy innych osób.
- Rozwiążesz konflikt przy merge'u i rebase'ie.
- Uporządkujesz historię gałęzi interaktywnym rebase'em przed code review.
- Odzyskasz utracone zmiany przez `git reflog`.
- Znajdziesz commit, który wprowadził błąd, przez `git bisect`.
- Będziesz pracować na dwóch gałęziach naraz dzięki `git worktree`.

- Podpiszesz commity kluczem SSH.
- Dobierzesz strategię gałęzi (GitHub Flow, trunk-based, GitFlow) do swojego zespołu.
- Przygotujesz pull request gotowy do code review.

## OPINIE UCZESTNIKÓW

> *„Oprócz samego Angulara Piotr zdążył nam przekazać mnóstwo wiedzy na temat Git, Visual Studio Code, TypeScript i dobrych praktyk programistycznych - czego oczywiście nie było w programie."*
>
> Sebastian, Full-Stack Developer (szkolenie z Angulara)

> *„The whole team was able to learn from the mini-workshops and presentations he regularly shared with us, covering advanced Git tricks, React, JavaScript and more."*
>
> Paul Ngei, współpracownik z zespołu w Box (rekomendacja na LinkedIn)

## TRENER

### Piotr Kowalski

W branży IT od 2008 roku. Od 2014 jest mentorem dla wielu developerów. Jako trener spędził na sali szkoleniowej ponad 1000 godzin, a na LinkedIn ma 52 rekomendacje. Każde jego szkolenie to dawka solidnej wiedzy oraz humoru.

Więcej: piecioshka.pl/piotr-kowalski

## ZORGANIZUJMY TO SZKOLENIE W TWOIM ZESPOLE

Program dopasuję do Waszego projektu i poziomu grupy. Napisz, z czym pracujecie, a odpowiem z propozycją terminu i zakresu.

**KONTAKT** **piecioshka.pl/kontakt**
**E-MAIL** **piecioshka+blog [małpka] gmail.com**
**SZKOLENIE** **piecioshka.pl/szkolenia/git/**