

# Advance git log

```
git log --after="2014-02-12T16:36:00-07:00"  
git log --before="2014-02-12T16:36:00-07:00"  
git log --since="2 weeks 3 days 2 hours 30 minutes 59 seconds ago"  
Wyszukiwanie commitów na podstawie czasu
```

```
git log -4  
Określanie liczby commitów (w tym przypadku 4 commity)
```

```
git log --oneline  
Skompresowanie commitów jedna linijka per commit
```

```
git log --graph  
Wizualizacja commitów w grafie
```

```
git log --decorate  
Dodanie do commitów adnotacji o branchach/tagach itp.
```

```
git log --stat  
Wylistowanie zmodyfikowanych/dodanych/usuniętych plików
```

```
git log --patch  
Ukazanie diffów - zmian wprowadzonych przez dany commit (skrót: -p)
```

```
git log origin/develop --not develop  
Commity z brancha origin/develop, których nie ma na branchu develop („lokalnie“)
```

```
git log develop..origin/develop  
Commity z brancha origin/develop, których nie ma na branchu develop („lokalnie“)
```

```
git log develop...origin/develop  
Commity z brancha origin/develop („serwera“), których nie ma na branchu develop („lokalnie“),  
oraz commity z brancha develop („lokalnie“), których nie ma branch origin/develop („serwer“)
```

```
git log --author="Toma*"  
Commity, których autor treści pasuje do regexa „Toma*“
```

```
git log --committer="Toma*"  
Commity, które zostały stworzone przez osobę pasującą do regexa „Toma*“
```

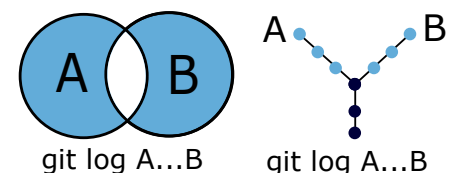
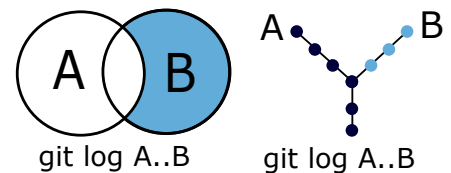
```
git log -grep=".scope*" -i  
Commity których wiadomość pasuje do regexa „.scope*“ (case insensitive: -i)
```

```
git log -S"com.arellomobile.mvp.MvpPresentert" -i  
Commity które dodają/usuwają frazę „com.arellomobile.mvp.MvpPresentert“, tak zwany  
pickaxe (case insensitive: -i)
```

```
git log -all  
Commity z wszystkich branchy (nie tylko z „obecnego“)
```

```
git log --no-merges  
Wykluczenie commitów mergujących z wyników
```

```
git log --pretty="FORMAT"  
Zmiana wyświetlanego formatu commity,  
np. git log --pretty="%an-%H"  
https://devhints.io/git-log-format
```



# Szlifowanie Gita



- Dwa słowa o konfiguracji
- Czym jest HEAD i po co nam to wiedzieć
- Co zrobić gdy zacommitowaliśmy lub wypushowaliśmy coś nie tak
- Jedno słowo o cherry-pick
- Łączenie branchy - rebase vs merge
- BONUS: Jak rebasować, żeby nie bolało - autorskie rozwiązanie
- Push, force with lease, pull with rebase
- 7 zasad dobrego commit message
- Ostrzeżenie konsoli
- Znasz .gitignore, poznaj i .gitattributes

# Git Od Zera



# Docker Od Zera



# Git Efekt Uboczny



# Pipeline CI/CD

