

GIT

Ćwiczenie 1 – Powtórzenie

1. Tworzymy puste repozytorium o nazwie lab03.
2. Tworzymy plik main.sh, dodajemy do repozytorium i zatwierdzamy zmiany (komitujemy).
3. Tworzymy nową gałąź o nazwie hello i przełączamy się do niej.
4. Sprawdzamy listę dostępnych gałęzi i historię zatwierdzeń z pokazywaniem etykiet (log --decorate).
5. W pliku main.sh piszemy program „Hello World”. Tworzymy plik README z naszym imieniem i nazwiskiem. Wszystkie zmiany dodajemy i zatwierdzamy.
6. Przełączamy się na gałąź master, scalamy zmiany wykonane w gałęzi hello do gałęzi master.
7. Usuwamy gałąź hello. Tworzymy nową gałąź develop.
8. Będąc w gałęzi master dodajemy w pierwszej linijce pliku main.sh informacje o prawach autorskich: „# Copyright by ...”. Zmiany zatwierdzamy.
9. Przechodzimy do gałęzi develop i w pierwszej linii dopisujemy komentarz „# Program wyświetlający tekst w konsoli”.
10. Zmiany dodajemy do repo i zatwierdzamy.
11. Przechodzimy z powrotem do gałęzi master. Scalamy master z develop.
12. Sprawdzamy poleceniem git status w jakim pliku jest konflikt, ręcznie usuwamy konflikt w pliku
13. Zatwierdzamy konflikt z domyślną wiadomością
14. Publikujemy repo na githubie i dodajemy prowadzącego (piotrekgie) jako kontrybutora

Ćwiczenie 2 – Aliasy

Stwórz aliasy dla

- polecenia graficznego wyświetlania commitów w repozytorium
“log --graph --decorate --pretty=oneline - -abbrev-commit”
pod nazwą “logl”
- polecenia wyświetlającego listę zdefiniowanych aliasów “config
--get-regexp ^alias\.” pod nazwą “alias”

Ćwiczenie 3 – Stash

1. Wracamy na gałąź develop
2. Zmodyfikuj plik main.sh by wyświetlał w konsoli "Hello \$nazwa_uzytkownika"
3. Sprawdzamy status plików
4. Musimy zrobić hotfixa, ale nie chcemy jeszcze commitować zmian na developie - umieszczamy więc zmiany w schowku
5. Sprawdzamy ponownie status plików
6. Wracamy na mastera, tworzymy gałąź "hotfix/fix_hello_world"
7. Zmieniamy plik main.sh by wyświetlał tekst "Hello guys"
8. Dodajemy i zatwierdzamy zmiany zmiany (z komentarzem "Fixed hello text"), przechodzimy na mastera i zaciągamy gałąź z hotfixem.
9. Wracamy na gałąź develop, przywracamy pliki ze schowka i edytujemy plik tak aby skrypt tworzył plik o nazwie "lista_plikow.txt" i wypisywał w nim listę wszystkich plików wraz z plikami ukrytymi z katalogu domowego zalogowanego użytkownika
10. Zatwierdzamy zmiany, przełączamy się na mastera, scalamy go z developem i wypuszczamy zmiany na github.

Ćwiczenie 4 – Tagi

1. Sprawdzamy znacznik pierwszej rewizji (commita) i dodajemy do niego znacznik v1.0 z komentarzem “Initial project”
2. Dodajemy znacznik v1.1 do rewizji “Fixed hello text” z komentarzem “Hello text release”
3. Dodajemy znacznik v.2 do HEAD z komentarzem “List all files and username hello text”
4. Przełączamy się na każdą wersję i sprawdzamy działanie skryptu