**Ćwiczenia z systemu kontroli wersji git**

# **Podstawowe operacja na repozytorium**

1. Załóż konto na [https://github.com](https://github.com/)
2. Prowadzący dokonuje wprowadzenia na temat Git’a. Powinien wyjaśnić w jakim celu stosujemy narzędzia kontroli wersji, jak ułatwiają pracę, dla kogo. Wprowadzenie ma dać pogląd na filozofię pracy z git’em
3. Zapoznaj się z materiałami na <https://www.atlassian.com/git>
4. Warto także zapoznać się z książką <https://git-scm.com/book/pl/v1/>
5. Dobierz się w parę z sąsiadem z ławki lub z grupy.
6. Stwórz repozytorium kodu dla projektu pod nazwą ProgUzytkowe
7. Podczas tworzenia wybierz licencje MIT, zainicjalizuj repozytorium plikiem README.md
8. Załóż folder na dysku „D” w nim będziesz przechowywać swoje projekty i repozytoria
9. Uruchom z menu start ‘git Bash’, w konsoli przejdź do twojego folderu

cd /d  
cd /mojfolder

1. Sklonuj repozytorium na dysk komputera, użyj polecenia

git clone

adres swoje repozytorium weź z strony z github

1. Zapoznaj się z formatem markdown <http://commonmark.org/help/>, jest on powszechnie wykorzystywany w plikach README.md
2. Otwórz plik README.md w notepad++
   1. Ustaw nagłówek h1, aby był czytelny dla ludzi (bez podkreśleń itp).
   2. Dodaj autora projektu
   3. Dodpisz nagłówek h2 „Opis projektu”
   4. Napisz krótkie podsumowanie projektu.
3. Użyj polecenia   
     
   git status

aby przejrzeć zmiany jak są do zakomitowania

1. Użyj polecenia

git diff

aby zobaczyć jakie linie zostały zmienione

1. Przedstaw się git’owi, w konsoli wpisz polecenia:

$ git config --global user.name "Jan Nowak"

$ git config --global user.email jannowak@example.com

1. Zrób ‘commit’ twoich zmian, użyj polecenia

git commit -am ‘opis zmian’

1. Za pierwszym razem git zwróci komunikat błędu, że ciebie nie zna i musisz się przedstawić, użyj poleceń które podpowiada git i ustaw user.name i user.emal na takie jakie masz w serwisie github
2. Stwórz nowy plik index.html
3. Dodaj go do plików monitorowanych przez git

git add

1. Dodaj podstawową strukturę html w pliku index.html i zakomituj zmiany (pamiętaj o poprawnym opisie zmian).
2. Użyj polecenia   
   git log

w celu przejrzenia listy dotychczasowych zmian

1. Wypchnij zmiany do github (git push), zobacz na stronie github’a jak to wygląda

# **Tworzenie gałęzi kodu, przechodzenie po hisorii komitów, cofanie zmian**

1. Przerób interaktywny tutorial na <https://try.github.io/levels/1/challenges/1> . Zwróć szczególną uwagę na to jak kolejne komity tworzą pewne punkty w historii projektu. Ważne polecenia: git branch, git checkout, git reset, git revert, git merge, git fetch, git pull.
2. Pytania kontrolne:
   1. Czym się różni branch od checkout?
   2. Czym się różni reset od revert?
   3. W jaki sposób mogę przenieść się do danego commita?
   4. W jaki sposób mogę przełączyć się na inną gałąź kodu?
   5. W jaki sposób mogę scalić zmiany?
   6. Czym się różni fetch od pull?
3. W swoim repozytorium stwrórz nową gałąź ‘dev’, przejdź do niej. Wydaj odpowiednie polecenia z konsoli.
4. Dodaj dwie zmiany w plikach w gałęzi ‘dev’
   1. Zmień linię 1 w pliku README.md, dodaj jakiś wyraz na końcu
   2. Dokonaj dowolnych zmian w pliku index.html
   3. Zakomituj zmiany i wypchnij je do repozytorium na github
   4. Otwórz stronę projektu na github i przyjrzyj się zmianom, odszukaj swoją nową gałąź ‘dev’ i wyświetl listę commit’ów
5. W konsoli scal zmiany do gałęzi master

# **Praca zespołowa**

1. W ustawieniach projektu (Settings) na stronie github.com dodaj nowego kolaboratora(współpracownika) niech to będzie twój kolega/koleżnaka z ławki. Musisz podać jego login z github
2. Odszukaj w powiadomieniach github informację o udostępnieniu tobie projektu. Wejdź do tego projektu.
3. Ściągnij (clone) udostępniony tobie projekt na swój komputer.
4. Dokonaj lokalnych zmian w udostępnionym tobie projekcie w pliku README.md, zrób commit i push na serwer.
5. Zobacz na stronie github, co zostało zmienione w twoim projekcie.
6. Ściągnij zmiany poczynione przez twojego współpracownika, do siebie na lokalny komputer (git pull)
7. Rozpocznijcie pracę nad jednym ustalonym projektem.
   1. Niech każdy z was doda dwa pliki
   2. Niech każdy z was dokona zmian w plikach projektu
   3. Niech każdy z was zrobi min. 4 commity
   4. Ściągnij zmiany zrobione przez współpracownika,   
      Uwaga! Mogą nastąpić konfilikty, które będziecie musieli rozwiązać poprzez edycję plików, następnie należy wykonać git commit
   5. Wypchnijcie swoje zmiany na serwer
8. Niech każdy z was utworzy swoją gałąź
   1. Dokona w niej zmian (wyedytujcie pliki)
   2. Zakomituje zmiany i wypchnie na serwer (commit i push)
   3. Ściągnij gałąź swojego kolegi
   4. Scal się z zmianami z gałęzi kolegi, wypchnij zmiany na serwer do swojej gałęzi