Шпаргалка по Git

**Створення репозиторію**

* Створити новий репозиторій: git init
* Клонувати репозиторій з віддаленого сервера: git clone https://webdev-sv@bitbucket.org/webdev-sv/project.git

**Додавання та видалення файлів**

* Внести в індекс всі зміни, включаючи нові файли: git add .
* Добавити файл: git add file.txt
* Видалити файл: git rm file.txt
* Видалити невідслідковувані файли: git clean -f
* Видалити файл з індексу: git rm --cached file.txt
* Не слідкувати за змінами у файлі: git update-index --assume-unchanged file.txt
* Відміна попередньої команди: git update-index --no-assume-unchanged file.txt

**Комміт**

* Зробити комміт: git commit -m "Текст комміту"
* Зробити комміт і ввести його опис за допомогою $EDITOR: git commit

**Робота з віддаленими гілками**

* Завантажити всі гілки локального репозиторію на віддалений репозиторій і зробити злиття: git push origin
* Завантажити гілку master локального репозиторію на віддалений і зробити злиття: git push origin master
* Завантажити поточну гілку локального репозиторію на віддалений і зробити злиття: git push origin HEAD
* Завантажити гілку з віддаленого репозиторію у локальний і зробити злиття: git pull origin master

**Робота з гілками**

* Створити нову гілку: git branch new-branch
* Створити нову гілку і перемкнутися на неї: git checkout -b new-branch
* Вивести всі гілки: git branch
* Перейти на іншу гілку: checkout some-other-branch
* Злити поточну гілку з вказаною: git merge new-branch
* Переіменувати гілку: git branch -m new-branch
* Видалити гілку після злиття: git branch -d some\_branch
* Видалити гілку у будь-якому випадку: git branch -D some\_branch
* Видалити гілку на віддаленому репозиторії: git push origin: some-branch

**​Відкат змін**

* Зміна тексту останього комміту: git commit --amend -m 'новий текст'
* Відкат до останього комміту з збереженням змін: git reset --soft HEAD
* Повернутися до конкретного комміту без збереження змін: git reset --hard f654321edf8458ce01235bb76a58c5ca4a58c1c2c
* Повернутися до конкретного комміту з збереженням змін: git reset --soft f654321edf8458ce01235bb76a58c5ca4a58c1c2c
* Повернути версію файлу з поточного репозиторію (відміна змін файла в робочій дерикторії): git checkout --file.name
* Видалити файл з індексу з збереженням змін: git reset HEAD file.name
* Повернути версію файлу з вказаного комміту: git checkout 20abc58f84cd --file.name
* Відмінити останній комміт: git revert abc7dc6d5cd4d4d4d4d4d4d

**Перегляд змін**

* Поточний стан репозиторію: git status
* Останні зміни в репозиторії: git log
* Останні зміни в репозиторії в один рядок: git log --oneline
* Останні зміни вказаного файлу: git log file.txt
* Зміни зроблені у вказаному комміті: git show f12356485458ce06fbc5bb76a58c5ca4a58c675g
* Зміни внесені в індекс: git diff
* Зміни не внесені в індекс: git diff --cached
* Зміни порівняно з останнім коммітом: git diff HEAD
* Показати різницю між коммітом с1223df1 і поточним: git diff с1223df1
* Показати різницю між гілками: git diff master..new\_branch
* Показати різницю між гілками (зміни в однмоу рядку): git diff --color-word master..new\_branch

**Налаштування**

* Не коммітити зміну прав у файлів: git config core.filemode false
* Встановити Sublime Text редактором за замовчуванням:git config --global core.editor "subl -n -w"
* Кольорове відображення інформації: git config --global color.ui true​