**GIT – version control system**

**Система контроля версий**

|  |  |
| --- | --- |
| **msysgit.github.io** | **Ресурс для скачивания GIT’а** |
| **git-scm.com/download/win** | **Ресурс для скачивания GIT’а** |
| **$ git config --global user.name”Slava Bohan”** | **Прописать имя пользователя** |
| **$ git config --global user.email”bohanv@gmail.com”** | **Прописать email пользователя** |
|  |  |
| **$ git config** | **Все настройки GIT’а** |
| **$ git config --list** | **Посмотреть персональные настройки GIT’а** |
|  |  |
| **$ git help commit** | **Помощь о команде “commit”** |
| **$ git commit --help** | **То же самое что и выше** |
|  |  |
| **$ cd c:/My\_Info/InfoForJava** | **Вход в заданную папку** |
| **$ git init** | **Создать GIT репозиторий в папке куда вошел выше** |
| **$ rm -rf .git** | **Удалить git из дирректории** |
| **$ git status** | **Посмотреть статус GIT’а** |
|  |  |
| **$ git add .** | **Команда для добавления untracked файлов в папке в коммит. Точка означает - добавить все файлы.** |
| **$ git add -A** | **Команда для добавления всех untracked файлов в папке в коммит** |
|  |  |
| **$ git rm “file name”** | **Удаление файла из списка проиндексированных** |
| **$ git rm –cached filename.exm** | **Удаление файла из списка проиндексированных** |
| **$ git status –untracked-files=all** | **Показать все неотслеживаемые GIT’ом файлы** |
| **$ git commit -a -m”init”** | **Коммит всех проиндексированных файлов. Пометка «init»** |
| **$ git commit -m”comment” --amend** | **Добавить изменения к комиту. Написать коментарий** |
| **$ git commit --amend --no-edit** | **Добавить изменения к комиту, оставить старый коментарий** |
|  |  |
|  |  |
|  |  |
|  |  |
| **$ git checkout** *filename.exm* | **Откатить состояния файла на его предыдущее закомиченное** |
| **$ git log** | **Показать все коммиты репозитория** |
| **$ git log --since=2.weeks** | **Показать коммиты репозитория сделанные за последние 2 недели** |
| **$ git log --pretty=format:”%h - %an, %ar : %s”** | **Вывод коммитов в сокращенной форме по схеме указанной ключиками** |
| **$ git log -p -2** | **Показывает какие конкретно были сделаны изменения в коммите. -2 – последние два коммита** |
| **$ git config –global core.editor “’C:\ProgramFiles\Notepad++\notepad++.exe’ -multiInst -notabbar -nosession -noPlugin”** | **Вписать изменения в настройки. Вызов редактора при git commit без коментария. В редакторе вписать комент. Редактор закрыть.** |
|  |  |
| **$ git remote** | **Просмотреть какие удалённые репозитории мы имеем. Github даёт имя по умолчанию origin** |
| **$ git remote -v** | **Покажет полный путь до репозитория куда положить и до репозитория откуда взять** |
|  |  |
| **$ git push -u origin master** | **Отправляем в репо “origin” в ветку “master” коммиты. -u – указывает на направление коммитов в дальнейшем по-умолчанию, т.е. впредь имя репо и ветки указывать не надо будет.** |
| **$ git checkout -b new\_f** | **Создаём новую ветку и переходим в нее** |
| **$ git branch -a** | **Показать какие ветки есть в репо.** |
| **$ git branch**  **master**  **\*new\_f** | **Показать какие ветки есть в репо.**  **Ветка master**  **Новая ветка в которой мы находимся** |
| **$ git branch new\_f2** | **Создать новую ветку но не переходим на нее** |
| **$ git branch -v** | **Более детальный просмотр наличия веток** |
| **$ git checkout new\_f2** | **Переключиться на ветку под названием new\_f2** |
| **$ git reset** *file.exm* | **Удалить приготовленный к коммиту файл (проиндексированный) из staging area. Очистка буферной зоны. Сам файл остается в раб.зоне** |
| **$ git reset** | **Удалить все файлы (проиндексированные) приготовленные к коммиту. Очистка буферной зоны** |
| **$ git clone http** | **Клонирует удаленный репо на комп в выбранной дирректории. В папке автоматически инициализируется git** |
| **$ git checkout** *copy/paste commit Nr from github* | **Возврат по номеру коммита** |
|  |  |
| **$ git reset --hard** *copy/paste commit code from github*  **$ git push --force** | **Для удаления коммита из github. Копируем хеш того коммита куда хотим вернуться на локале.**  **Вводим комманды. Одну, затем вторую.** |
| **$ git reset --hard HEAD~2** | **Возврат по репо на два коммита назад. Состояние файлов в проекте меняется на закомиченное состояние** |
| **$ git reset --soft HEAD~2** | **Возврат по репо на два коммита назад. Состояние файлов в проекте не меняется на закомиченное состояние** |
| **$ git commit -c ORIG\_HEAD** | **Объединяем предыдущие откаченные коммиты в единый** |
| **$ git** |  |
|  |  |
|  |  |
| **$ git remote rm origin** | **Отменить регистрацию удаленного репо в git’е** |
| **$ git remote add github** *https://github.com/myuser/proje* | **Связать Git и github** |
|  |  |
|  |  |
| **$ git commit --amend -m”new comment”** | **Изменить предыдущий коммит. Создать новый коммит на основе старого** |
|  |  |
| **$ git pull** | **Для получения обновлений с удалённого репо** |
| **$ git fetch --all**  **$ git reset --hard github/master** | **Если вы изменили ваши локальные файлы, то команда git pull выдаст ошибку. Если вы уверены, что хотите перезаписать локальные файлы, файлами из удаленного репозитория то выполните команды**  **Вместо github подставьте название вашего удаленного репозитория, которое вы зарегистрировали командой git push -u.** |
| **$ git push** | **Выложить изменения на удалённый репо** |
| **$ git push -f** | **В случае, если в удаленном репозитории лежат файлы с версией более новой, чем у вас в локальном, то команда git push выдаст ошибку. Если вы уверены, что хотите перезаписать файлы в удаленном репозитории несмотря на конфликт версий** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |