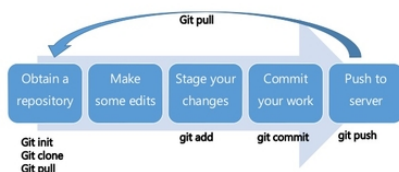


## Git - основы работы с системой контроля версий

### Git usages : Understanding Git Workflow



### Push / Pull

- `git push <remote> <branch>`
- `git pull <remote> <branch>`



## Терминология при работе с Git

**Репозиторий** - это папка-хранилище, где хранятся и поддерживаются какие-либо данные (абстрактная, главная папка на сервере для нашего проекта, исходного кода)

**GitHub** - облачное хранилище

Команда **clone** - скачивание репозитория (набора директорий с файлами) с сервера GitHub

Команда **add** - индексирование файлов и нахождение изменений в них

Команда **commit** - сохранение изменений

Команда **push** - сохранение в репозитории изменений на сервере GitHub

### Git - основные команды

```

php-код
1  -----
2  # Инициализация, клонирование, настройка
3  -----
4  $ cd                               // Переход в папку
5  $ clear                             // Очистить
6  -----
7  $ git init                         // Создание репозитория
8  $ git clone ССЫЛКА                 // Сказать по HTTPS://
9  -----
10 $ git config --global user.name = "" // Первичная настройка репозитория глобально
11 $ git config --global user.email = ""
12 $ git config --global core.editor "'C:\Program Files (x86)\Notepad++\notepad++.exe' -mu
13 $ git config --list                 // Список настроек
14 -----
15 - ФАЙЛ НАСТРОЕК: C:/users/.gitconfig (файл настроек)
16 - ФАЙЛ ИГНОРИРОВАНИЯ В КОРНЕ ПРОЕКТА: /.gitignore
17 - CTRL + INSERT (это CTRL + V)
18 -----
19 # Логи, справка
20 -----
21 $ git log                           // Если пришли в новую контору и решили посмотреть что там
22 $ git log --since=2.weeks           // Показать коммиты за последние две недели
23 $ git help log                       // Справка
24 -----
25 # Добавление в индекс
26 -----
27 # Зеленым отмечено состояние файла то что проиндексировано
28 # Красным отмечаются то что еще не проиндексировано и не пойдет в коммит
29 $ git add .                          // Добавить все
30 $ git add *.php                      // Добавить только PHP
31 $ git rm --cached ИМЯ-ФАЙЛА          // Если файл был добавлен по ошибке, или добави
32 $ git rm -r --cached ИМЯ-ПАПКИ      // Если папка была добавлена по ошибке - удалит
33 -----
34 # Отмена до состояния репозитория
35 -----
36 # Если мы поэкспериментировали и хотим откатиться до того состояния что уже было проинде
37 # и что бы не делать это руками - до того состояния как в репозитории
38 $ git checkout -- ИМЯ-ФАЙЛА // Откатывает до того состояния как в репозитории
39 -----
40 # Коммит
41 -----
42 $ git commit -m "Commit"
43 $ git commit                         // Откроется окно с редактором, но у меня он по
44                                     // Выход: 1) нажимаем ESC, 2) вводим :x (сохран
45 -----
46 # Ветки
47 -----
48 # По умолчанию в Git есть только ветка Master
49
50
51
52

```

```

53 -----
54 $ git branch           // Список веток
55 $ git branch -v        // Список веток с их различием
56 $ git branch new_f     // Создать ветку
57 $ git branch -D new_f  // Удалить ветку
58
59 $ git checkout new_f2   // Переключится на ветку
60 $ git checkout -b new_f // Создать ветку и переключится на нее
61
62 -----
63 # Ветки слияние
64 # Объединение веток
65 -----
66
67 // Для мерджа нужно поставить прогу
68 // https://sourceforge.net/projects/kdiff3/files/
69 $ git config --global merge.tool kdiff3
70 $ git config --global mergetool.kdiff3.cmd "C:\\Program Files\\KDiff3\\kdiff3" $BASE $LOCAL $REMOTE -o $MERGED
71
72 $ git merge master      // Соединить с веткой такой-то
73 $ git mergetool         // Если есть конфликт, запустить утилиту
74
75 -----
76 # Отправка изменений на хостинг,
77 # создание удаленных репозитория
78 -----
79 $ git remote           // Список удаленных репозитория
80 $ git remote -v        // Список удаленных репозитория (подробно)
81
82 // !!!!! Не понял как это работает
83 $ git remote add master https://github.com/litovchenko1/site.git // Создать удаленный репозиторий
84
85 $ git push             // Протолкнуть изменения
86 $ git push origin master // Протолкнуть (загрузить) изменения
87 // $ git push -u origin master // Протолкнуть изменения и установить ветку origin/master
88
89 $ git config --global push.default matching // Настройка для push
90 $ git config --global push.default simple // Настройка для push
91
92 -----
93 # Забор изменений с хостинг (с удаленных репозитория)
94 -----
95 $ git fetch            // получение изменений из интернета из репозитория и слияние веток
96 $ git pull             // забор изменений из удаленного репозитория (стягивание изменений)
97
98 -----
99 # Пример настроек
100 # C:/Users/Александр/.gitconfig
101 -----
102 [user]
103     name = ivan litovchenko
104     email = i-litovan@yandex.ru
105 [core]
106     editor = 'C:\\Program Files (x86)\\Notepad++\\notepad++.exe' -multiInst -notabbar -nosession
107 [mergetool "kdiff3"]
108     cmd = "C:\\\\Program Files\\\\KDiff3\\\\kdiff3\\" $BASE $LOCAL $REMOTE -o $MERGED
109 [merge]
110     tool = kdiff3

```

php-код

```

1 git init
2 git remote add origin ssh://<ваш логин>@<сервер с репозиторием>/<путь до репозитория>/<репозиторий>
3 #Например, если проект расположен на моем сайте, и он доступен по ssh с моим логином,
4 а репозиторий хранится в корневой папке, то получится адрес вроде
5 #ssh://m_zhurov@develop.zhurov.me/home/m_zhurov/zhurov.me.git
6 git pull

```

## Автоматическая авторизация на Github и Bitbucket

php-код

```

1 # Добавляем в файл ~/.ssh/config путь к приватному ключу:
2 Host bitbucket.org
3     IdentityFile /home/user/.ssh/id_rsa
4 Host github.com
5     IdentityFile /home/user/.ssh/id_rsa
6
7 # Проверяем работу (именно git@github.com, иначе получим ошибку Permission denied (publickey))
8 ssh -T git@github.com
9 ssh -T git@bitbucket.org
10
11 # Исправляем конфиг репозитория
12 # Исправляем .git/config, где user/репо.git - пользователь и название репозитория:
13 [remote "github"]
14     url = git@github.com:user/repo.git
15     fetch = +refs/heads/*:refs/remotes/github/*
16 [remote "bucket"]
17     url = git@bitbucket.org:user/repo.git
18     fetch = +refs/heads/*:refs/remotes/bucket/*
19
20 # Пушим - Если все в порядке, пуш пройдет без запроса логина и пароля.
21 git push bucket master
22 git push github master

```

## Основные команды в командной строке (как посмотреть размер папки на сервере по SSH)

php-код

```

1 du -sh [путь]
2 du -sh *

```

