Cheatography

Git (Russian) Cheat Sheet

by makhov via cheatography.com/34496/cs/10801/

Конфигурация

```
git config --global user.name "Your Name"
```

Устанавливает имя, которое будет указано в коммитах и тэгах

```
git config --global user.email "you@example.com"
```

Устанавливает email, который будет указано в коммитах и тэгах

```
git config --global color.ui auto
```

Добавляет немного цвета в вывод команды git

Начало работы

```
git init [project name]
```

Создает новый локальный репозиторий. Если указан**[project name]**, Git создаст новую директорию **[project name]** и создаст репозиторий внутри нее. Иначе репозиторий будет создан в текущей директории.

```
git clone <project url>
```

Скачивает проект со всей историей с удаленного репозитория.

Игнорирование файлов

```
$ cat .gitignore
/logs/
!/logs/,gitkeep
/tmp
*.swp
```

Согласно этому файлу Git будет игнорировать все файлы в папке **logs**, исключая файл **.gitkeep**, всю папку**/tmp** и все файлы формата ***.swp**. Описание файла работает для директории и поддиректорий, в которой расположен файл **.gitignore**

Популярные команды

```
git status
```

Показывает статус работы. Новые, зафиксированные, измененные файлы. Текущую ветку.

```
git diff [file]
```

Показывает изменения между **рабочей директорией** и **зафиксированными изменениями**

```
git diff --staged [file]
```

Показывает изменения между **зафиксированными изменениями** и **индексом** (закомиченными изменениями)

```
git checkout [file]
```

Отменяет изменения в **рабочей директории**. Операция **необратимая**.

```
git add [file]
```

Фиксирует изменения файла. Используйте **.**, чтобы добавить все изменения в директории и поддиректориях.

```
git reset [file]
```

Отменяет **фиксацию** файла.

```
git commit [-m "message here"]
```

Создает новый коммит из **зафиксированных** изменений. Сообщение обязательно. Если не указать его через **-m**, откроется текстовый редактор.

```
git rm [file]
```

Удаляет файл из директории и **фиксирует** удаление файла.

```
git stash
```

Прячет все незафиксированные изменения в "тайник" (stash).

```
git stash pop
```

Применяет изменения из "тайника" в рабочей директории и очищает его.

```
git stash drop
```

Очищает "тайник".

Cheatography

Git (Russian) Cheat Sheet

by makhov via cheatography.com/34496/cs/10801/

Работа с ветками

```
git branch [-a]
```

Список всех локальных веток. **-a** покажет список всех веток (не только локальные)

```
git branch [name]
```

Создает новую ветку, ссылающуюся на текущий **HEAD**

```
git checkout [-b] [name]
```

Переключает **рабочую директорию** на указанную ветку. С опцией **-b** создает ветку, если она не существует.

```
git merge [from name]
```

Присоединяет указанную ветку к текущей.

```
git branch -d [name]
```

Удаляет ветку, если она уже вмержена в другую. Использование **-D** вместо **-d** форсирует изменения.

Просмотр изменений

```
git log [-n count]
```

Показывает историю коммитов текущей ветки. **-n count** ограничивает кол-во коммитов.

```
git log --oneline --graph --decorate
```

Обзор истории ветки. Одна строка на коммит.

```
git log ref..
```

Список коммитов ветки, которые не вмерженны в **ref**. **ref** может быть имя ветки или имя тэга.

```
git log ..ref
```

Список коммитов ветки, которые есть в **ref**, но нет в текущей ветке.

```
git reflog
```

Список действий (мержи, коммиты и т.д.), сделанных в локальном репозитории.