

Методичка по работе с Git

Терминология

Термин	Ссылка	Пояснение
Git	https://git-scm.com/downloads	Git - это очень популярная технология контроля версий, используемая при разработке программного обеспечения и веб-приложений. Онлайн инструкция на русском языке - https://git-scm.com/book/ru/v2/
Github	https://github.com/	Популярный сервис для хранения Git-репозиториях. С 2018 года принадлежит Microsoft
Bitbucket	https://bitbucket.org/	Популярный сервис для хранения Git-репозиториях. Принадлежит Atlassian и очень тесно интегрирован с популярными IT-приложениями этой фирмы как Trello или Jira
Gitlab	https://gitlab.com/	Альтернативный сервис для хранения Git-репозиториях.

Очень важно понимать, что Git и Github не одно и то же. Дополнительные материалы, книги и ссылки следует обязательно прочитать самому. И прежде всего стоит начать с <https://git-scm.com/book/ru/v2/>

Полезная информация

Для работы новичку достаточно знать несколько приказов и установить саму программу <https://git-scm.com/downloads>. Все команды ниже надо вводить в окно терминала/консоли. Я использую **Git Bash**

Создание SSH ключа для работы с git сервисами

Для полноценной работы с git-репозиториями требуется SSH-ключ. У Google можно спросить как его получить - "**how to generate a new ssh key**". Или просто ввести команду ниже, если подробности создания уже известны:

```
ssh-keygen -t rsa -b 4096 -C "your_email@example.com"
```

Инициализация проекта

Часто новичкам удобнее создавать проект в пустой папке:

```
mkdir newProjectname  
cd newProjectname  
git init
```

Добавление в репозиторий файлов

Добавление в репозиторий файла index.html:

```
git add index.html
```

Добавление в репозиторий всех файлов в папке:

```
git add .
```

Часто эта команда приводит к плохим последствиям. Потому что добавляет не только файлы разработки, но и временные файлы программы и инструментов, в которых работает программист. Рекомендуется добавлять в репозиторий файлы осознанно и поимённо.

Коммит

Фиксирование изменений, ещё несколько лет назад именно так переводили этот термин в литературе. Сегодня этот англицизм плотно вошёл во всю техническую литературу мира без перевода.

```
git commit -m "комментарий к коммиту, который объясняет изменения"
```

Ветки

По умолчанию работа ведётся в ветке мастер. Выйти из ветки мастер и войти в новосоздаваемую ветку одним приказом:

```
git checkout -b newBranchName
```

Выйти/вернуться в ветку, которая на проекте уже была создана

```
git checkout master
```

Удалённый репозиторий и работа с ним

Origin - это адрес вашего репозитория на одном из git-сервисов, которые вы выбрали. К примеру - Gitlab, Bitbucket или Github. Теоретически можно изменить имя удалённого репозитория на любое другое. Но у новичков достаточно путаницы в терминах и командах, чтобы дополнительно усложнять себе жизнь.

Синхронизировать ветку мастер на локальном компьютере и origin

Отправить изменения в интернет:

```
git push origin master
```

Обновить локальный репозиторий:

```
git pull origin master
```

Проверить состояние репозитория или лог изменений в ветке:

```
git status  
git log
```

Подробнее

Ask Google - "**git book online**"

Берлин. 2019. Андрей Подлубный. <https://t.me/andron13>