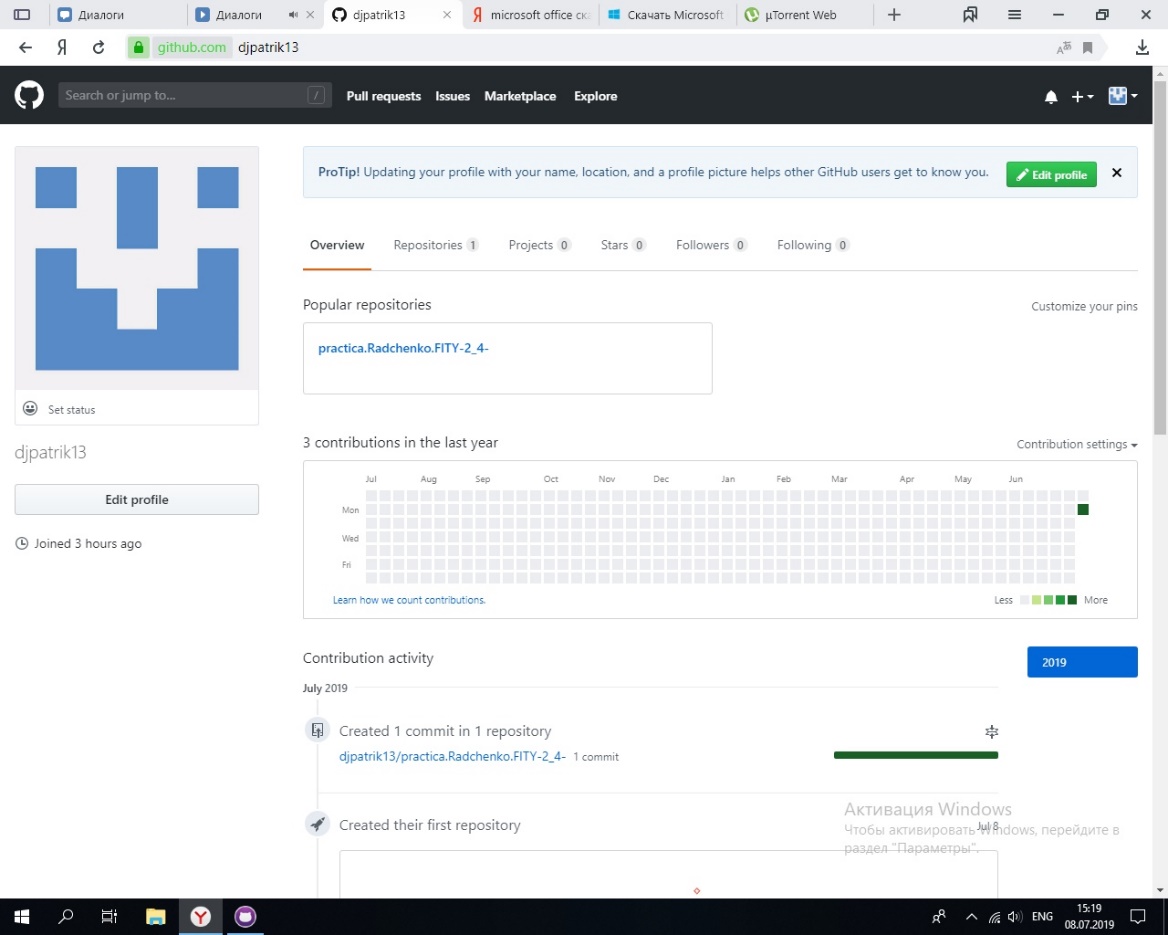
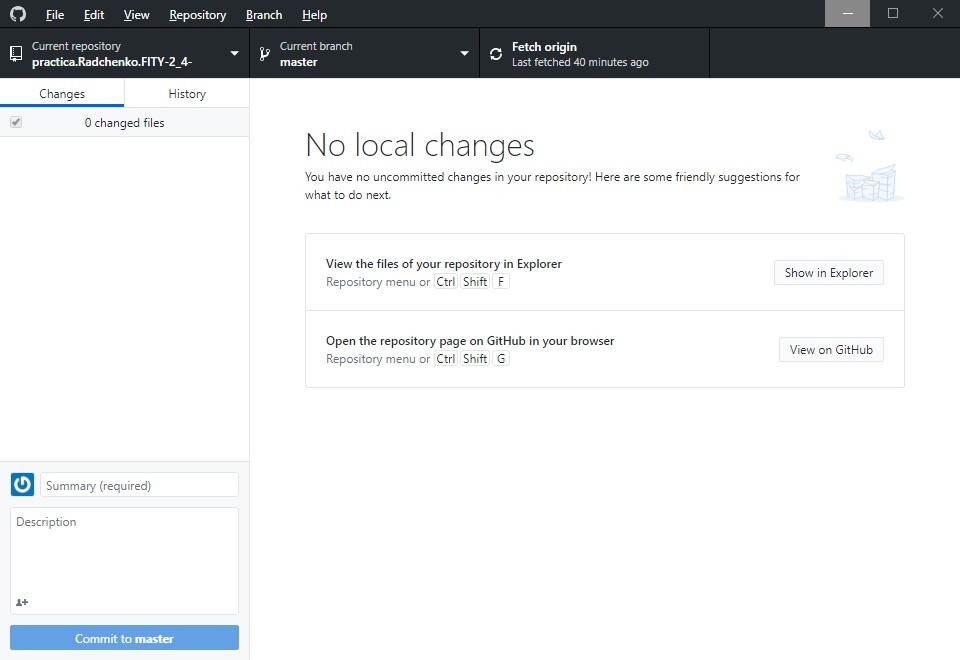
1.Регистрация на <https://github.com>



2.Установка клиента GitHub Deskop



3.Краткая теория

Git — это набор консольных утилит, которые отслеживают и фиксируют изменения в файлах (чаще всего речь идет об исходном коде программ, но вы можете использовать его для любых файлов на ваш вкус). С его помощью вы можете откатиться на более старую версию вашего проекта, сравнивать, анализировать, сливать изменения и многое другое. Этот процесс называется контролем версий. Существуют различные системы для контроля версий.

Git является распределенным, то есть не зависит от одного центрального сервера, на котором хранятся файлы. Вместо этого он работает полностью локально, сохраняя данные в папках на жестком диске, которые называются репозиторием. Тем не менее, вы можете хранить копию репозитория онлайн, это сильно облегчает работу над одним проектом для нескольких людей. Для этого используются сайты вроде github и bitbucket.

Создание нового репозитория

Как мы отметили ранее, git хранит свои файлы и историю прямо в папке проекта. Чтобы создать новый репозиторий, нам нужно открыть терминал, зайти в папку нашего проекта и выполнить команду init. Это включит приложение в этой конкретной папке и создаст скрытую директорию .git, где будет храниться история репозитория и настройки.  
Создайте на рабочем столе папку под названием git\_exercise. Для этого в окне терминала введите:



|  |  |
| --- | --- |
| 1  2  3 | $ mkdir Desktop/git\_exercise/  $ cd Desktop/git\_exercise/  $ git init |

Командная строка должна вернуть что-то вроде:



|  |  |
| --- | --- |
| 1 | Initialized empty Git repository in /home/user/Desktop/git\_exercise/.git/ |

Это значит, что наш репозиторий был успешно создан, но пока что пуст. Теперь создайте текстовый файл под названием hello.txt и сохраните его в директории git\_exercise.

Подготовка файлов

В git есть концепция области подготовленных файлов. Можно представить ее как холст, на который наносят изменения, которые нужны в коммите. Сперва он пустой, но затем мы добавляем на него файлы (или части файлов, или даже одиночные строчки) командой add и, наконец, коммитим все нужное в репозиторий (создаем слепок нужного нам состояния) командой commit.  
В нашем случае у нас только один файл, так что добавим его:



|  |  |
| --- | --- |
| 1 | $ git add hello.txt |

Если нам нужно добавить все, что находится в директории, мы можем использовать



|  |  |
| --- | --- |
| 1 | $ git add -A |

Проверим статус снова, на этот раз мы должны получить другой ответ:



|  |  |
| --- | --- |
| 1  2  3  4  5  6 | $ git status  On branch master  Initial commit  Changes to be committed:  (use "git rm --cached ..." to unstage)  new file: hello.txt |

Файл готов к коммиту. Сообщение о состоянии также говорит нам о том, какие изменения относительно файла были проведены в области подготовки — в данном случае это новый файл, но файлы могут быть модифицированы или удалены.

Коммит(фиксация изменений)

Коммит представляет собой состояние репозитория в определенный момент времени. Это похоже на снапшот, к которому мы можем вернуться и увидеть состояние объектов на определенный момент времени.  
Чтобы зафиксировать изменения, нам нужно хотя бы одно изменение в области подготовки (мы только что создали его при помощи git add), после которого мы может коммитить:



|  |  |
| --- | --- |
| 1 | $ git commit -m "Initial commit." |

Эта команда создаст новый коммит со всеми изменениями из области подготовки (добавление файла hello.txt). Ключ -m и сообщение «Initial commit.» — это созданное пользователем описание всех изменений, включенных в коммит. Считается хорошей практикой делать коммиты часто и всегда писать содержательные комментарии.