02.11.2024

Тема:  Использование *Git* для контроля версий.

Цель: Научится использовать функции гита

Ход работы:

Во время регистрации вам будет предложено проверить адрес электронной почты. Без проверенного адреса электронной почты вы не сможете выполнить некоторые основные задачи *GitHub*, например создание репозитория.

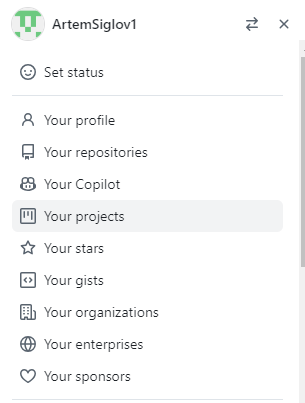


Рисунок 12 – ваш профиль

Для начала создадим репозиторий для этого в правом верхнем углу открываем меню пользователя и выбираем пункт *«Your repositories».* На странице с репозиториями нажимаем на кнопку *«New».* Откроется страница с созданием репозитория. Задаем название репозитория *TravelAgency* как показано на рисунке 13. Остальные поля и пункты можно оставить в исходном положении.

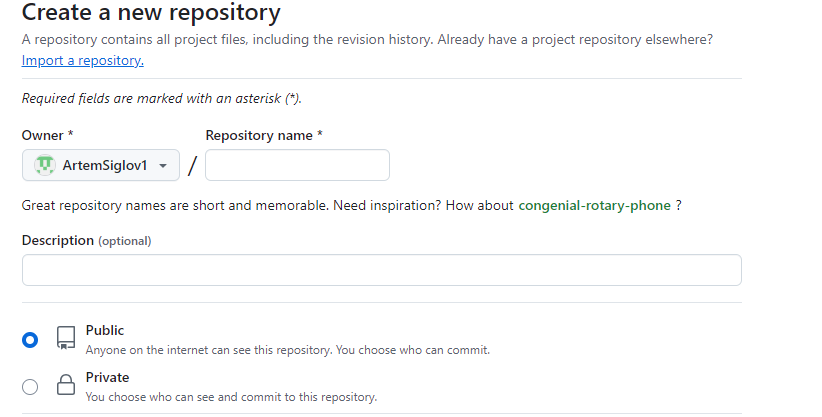
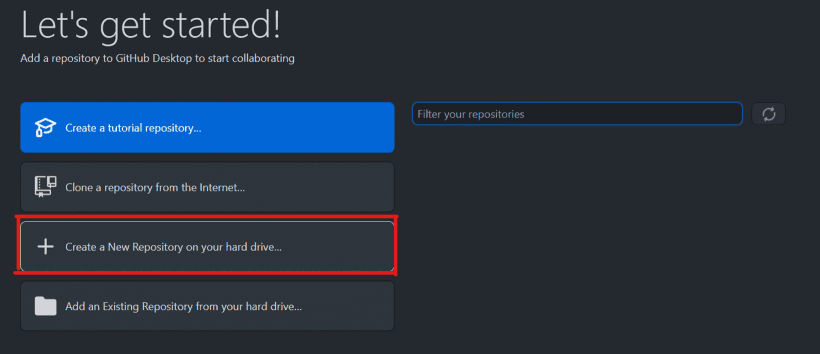


Рисунок 13 – создание удаленного репозитория

После создания удаленного репозитория необходимо создать локальный репозиторий. Для того что бы разобраться как работать с *GitHub Desktop* можно перейти на сайт [*https://htmlacademy.ru/blog/git/github-desktop*](https://htmlacademy.ru/blog/git/github-desktop)либо просмотреть информацию ниже, она взята с этого сайта.

После первого входа в *GitHub Desktop* вас попросят ввести ваши [логин и пароль от *GitHub.com*](https://htmlacademy.ru/blog/git/github-account). После этого у вас появится доступ ко всем репозиториям, сохранённым в профиле.

Создание репозитория: если вы никогда не пользовались гитхабом, нужно будет создать репозиторий для работы над проектом. На главном экране *GitHub Desktop* выберите пункт *«Create a New Repository on your hard drive».*

 Рисунок 14 – начало работы

Нужно будет ввести название репозитория, его описание и выбрать папку на компьютере, куда будут сохраняться файлы.

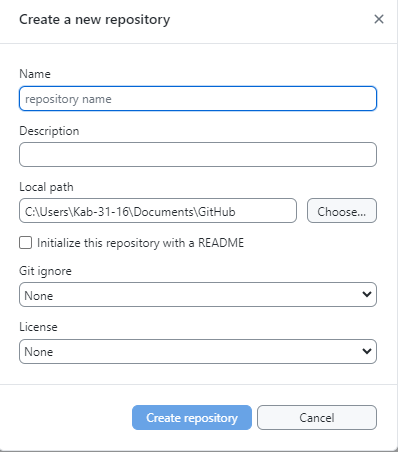


Рисунок 15 – создание локального репозитория

После этого нажимаем на *Create repository*, ждём несколько секунд и готово – на компьютере появилась папка, которой можно пользоваться для разработки вашего проекта.

Клонирование репозитория: если у вас уже какой-нибудь репозиторий на Гитхабе, его можно клонировать. Клонировать – это скачать все файлы к себе на компьютер, чтобы можно было их изменять и потом загружать обратно.

Выбираем *Add -> Clone Repository...*

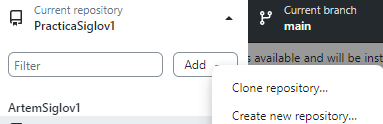


Рисунок 16 – клонирование репозитория

В открывшемся окне выбираем один из имеющихся репозиториев. В данном случае он называется *zaverstai,* но у вас может быть любой другой.

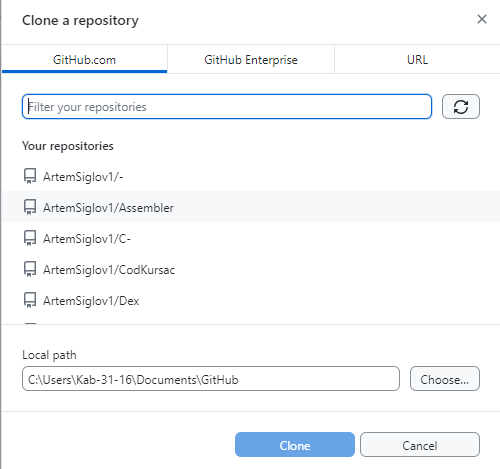


Рисунок 17 – выбор репозитория

После этого файлы репозитория начнут скачиваться – если их много, то это займет некоторое время.

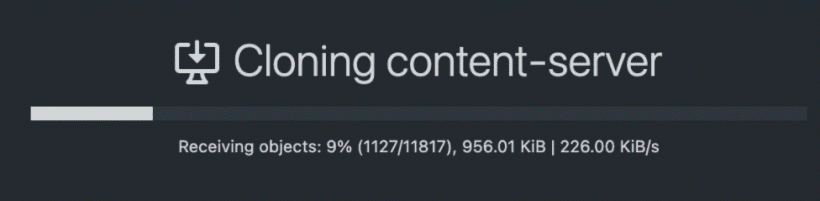


Рисунок 18 – скачивание файлов

Работа с репозиторием. Меняем файлы и сохраняем обратно: вне зависимости от того, создали вы репозиторий или клонировали его, так выглядит *GitHub Desktop* с открытым репозиторием, в котором мы пока ничего не меняли.

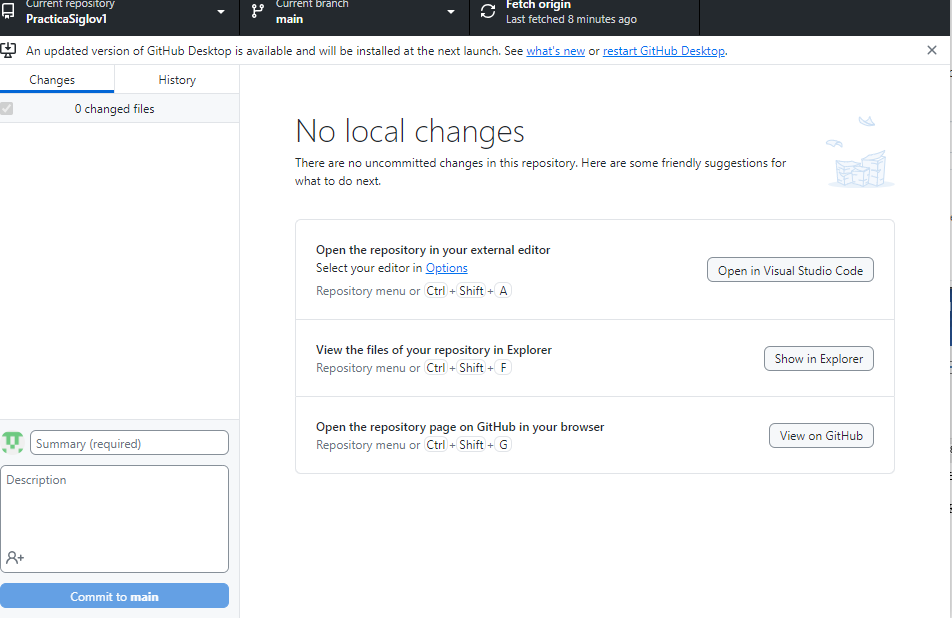


Рисунок 19 – вид приложения с открыты репозиторием

*Слева* – *поле для измененных файлов, справа* – *служебная информация. Слева снизу* – *поле для коммитов.*

Давайте добавим какой-нибудь файл. Например, я добавил в локальный репозиторий (скопировал в папку) файл index.html. Вы можете добавить текстовый файл.

Сразу после добавления или изменения файла в окне GitHub Desktop будет видно, что изменилось – если мы добавили целый новый файл, то все строчки будут с плюсиками и зелёные. Это значит, что они были добавлены в файл и GitHub Desktop раньше их никогда не видел.

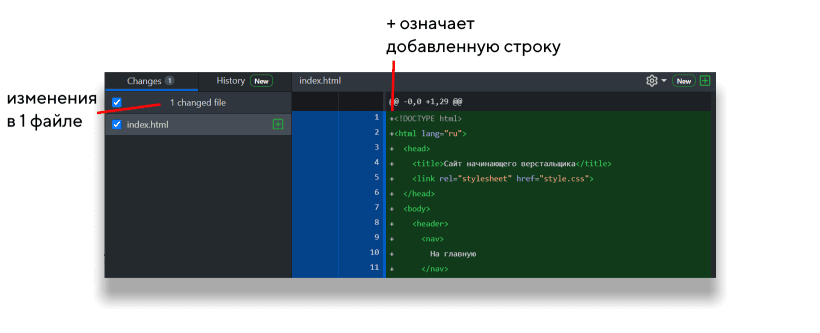


Рисунок 20 – вид при добавлении файла

Загружаем новый репозиторий на *GitHub*: после того, как мы добавили какой-то код в свежесозданный репозиторий, нужно сделать коммит, то есть зафиксировать все сохранённые изменения и дать им название. Текст должен быть лаконичным и в то же время сообщать о том, что делает коммит. Например, «добавляет имя наставника в *Readme*», «вводит функцию сортировки изображений», «правит ошибку в поиске городов на карте». Вводим имя жмём большую синюю кнопку *«Commit to main»*

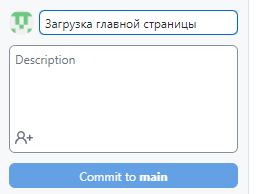


Рисунок 21 – создание коммита

Изменения, которые мы внесли и сохранили, пока локальны. Их нужно послать на *GitHub.* Чтобы опубликовать свежесозданный репозиторий на *GitHub*, нажмите *Publish repository*.

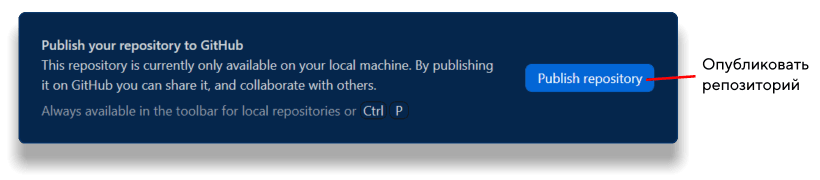


Рисунок 22 – опубликовать репозиторий

Появится подтверждение о публикации репозитория – проверяем название и описание, если нужно, ставим галочку о том, что код приватный, и публикуем.

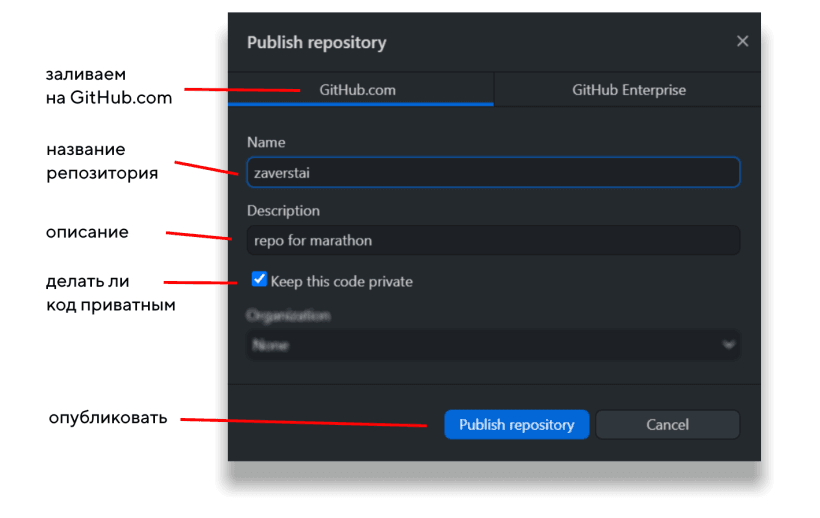


Рисунок 23 – подтверждение о публикации репозитория

Готово – после этого репозиторий появится в вашем профиле на *GitHub. com.*

Добавляем код и коммитим изменения: репозиторий создан и загружен на GitHub, теперь нужно добавить немного кода.

Когда вы допишете код в файлы, которые находятся в репозитории, вы сможете просмотреть все их изменения в окне *GitHub Desktop*. Вот здесь, например, мы изменили «второй» на «третий» в тексте страницы – и изменения сразу видны, можно проверить, что всё исправленное будет загружено.

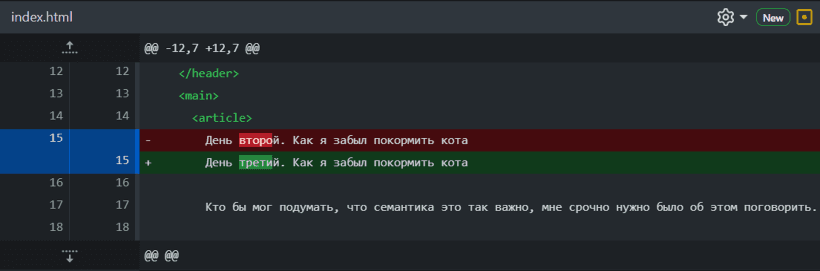


Рисунок 24 – просмотр исправлений

Дальше действуем по проверенной схеме – коммитим изменения.

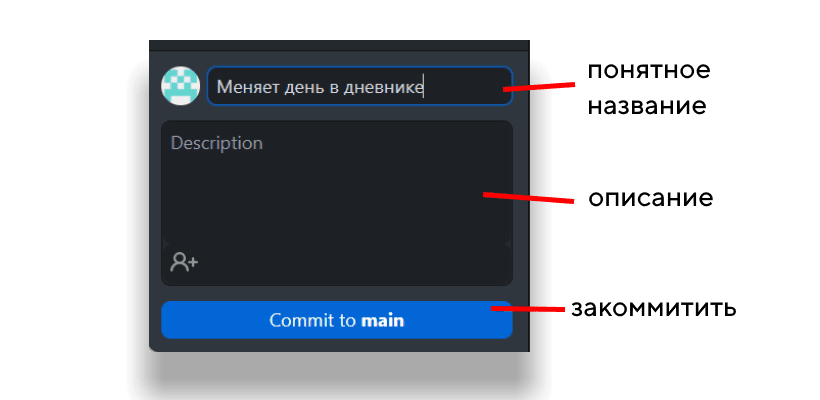


Рисунок 25 – коммит изменений

В центре главного экрана появится предложение запушить коммит в удалённый репозиторий. Соглашаемся и жмём *Push origin*.

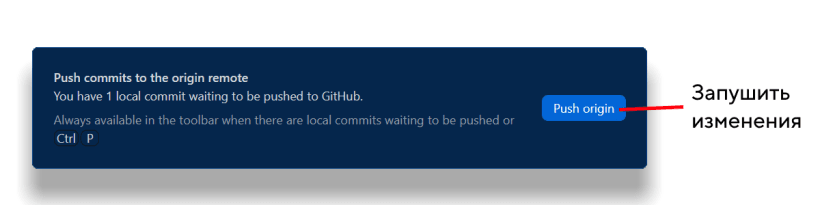


Рисунок 26 – пуш изменений

Готово! Теперь, если зайти на [*GitHub.com*](http://github.com/), в наш репозиторий, увидим изменённый файл, который мы только что отправили.

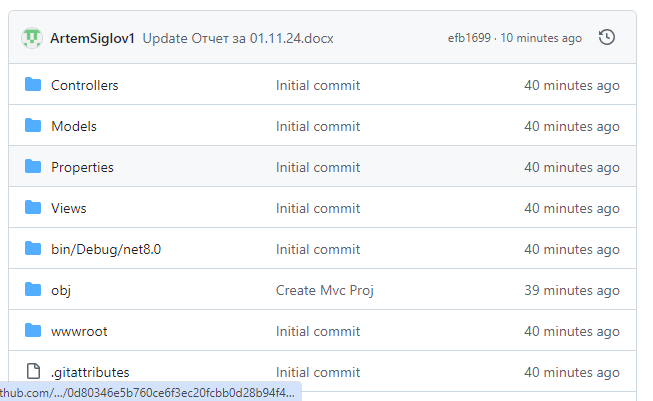


Рисунок 27 – проверка отправки изменений

Вывод: теперь я могу создать или склонировать репозиторий, добавить туда файлы, опубликовать всё это на [*GitHub.com*](http://github.com/), не прикасаясь к консоли.