Git не только для IT

Написать это меня побудил следующий комментарий в ЖЖ:  
*Книга писалась и редактировалась на 3 компах и еще на АТ и хоть я старался перетаскивать текущий вариант везде, получалось не везде.*

На этом моменте у меня в голове что-то отчётливо щёлкнуло и я подумал, -- а ведь системы контроля версий в целом (и git в частности) написаны как раз для таких случаев!

Начнём с простого вопроса – а о чём я, собственно, говорю?  
Кому и зачем это нужно?

Ответ простой. Довольно давно программисты столкнулись с тем, что программы пишутся и, что важно, *переписываются*. Что образуются разные версии и с ними нужно как-то управляться. Что иногда нужно посмотреть, а как оно было раньше. Что иногда одновременно нужны две версии. Что необходима совместная работа, наконец.

В общем, у *писателей программ* возникли те же сложности, что и у писателей любых текстов.

Поэтому разработчики создали такой класс программ, как *системы контроля версий*.  
Несмотря на несколько устрашающее название, суть таких программ проста – они хранят историю изменений файла (чаще проекта. Для простоты будем считать, что папки с файлами).

Они умеют много чего ещё, но нас это пока не интересует.  
Систем контроля версий довольно много, но де-факто почти везде сейчас используется git.  
О нём и пойдёт речь.

Прежде чем мы перейдём к практике, попробую описать как выглядит работа с git.  
Git работает с репозиториями. Собственно, если просто, то репозиторий – это просто папка с файлами, в которой есть каталог .git, в котором хранится история изменений, версии, ветки, ~~сучья и чорт лысый~~.

Для нас сейчас важно то, что это просто папка.  
Т.е. храниться она может на компьютере, на флэшке, на Яндекс.Диске, в Dropbox или в веб-сервисах для хранения кода, таких как Github или Gitlab.  
Неважно где, важно чтобы это была одна папка, доступная отовсюду, где мы планируем работать с текстом.

Давайте для простоты сначала поговорим о флэшке, а затем я расскажу как работать с удалённым репозиторием (спойлер: так же).

Ну-с! Пора приступать. Я постарался выбрать клиент для Git поудобнее (выбирал методом пристального взгляда на [скриншоты на официальном сайте Git](https://git-scm.com/downloads/guis). Можете взять любой другой, суть у всех одна) и выбрал [Git Extensions](https://gitextensions.github.io).

Нам потребуются Git Extensions и, собственно, Git (я не очень понимаю, почему его нет в составе Git Extensions, но пусть так).

Поскольку по условиям задачи с текстом мы планируем работать на разных компьютерах, то я буду использовать Portable версии, не требующие установки.

Скачиваем Git отсюда: <https://git-scm.com/download/win> (напоминаю, я скачал Portable версию. Она же "thumbdrive edition").

Скачиваем Git Extensions отсюда: <https://github.com/gitextensions/gitextensions/releases/latest> (также Portable).

Распаковываем скачанные файлы на флэшку, например в d:\utils (название папки абсолютно произвольное, я так называю по старой привычке со времён DOS) и запускаем GitExtensions.exe.  
При первом запуске GitExtensions попросит у вас указать путь к git.exe, а также заполнить имя и электронную почту. Остальные настройки можно не трогать.

Жмём «ок», открывается основное окно программы, нажимаем в нём «создать новый репозиторий» и указываем где (например: d:\Статьи).  
Собственно всё. Теперь либо начинаем писать новую статью и сохраняем в этот каталог, либо просто копируем в него нужные файлы.

И вот тут начинается работа непосредственно с git.  
Для начала нас будет интересовать ровно одна кнопка – Коммит.  
Она нужна, чтобы указать git какие файлы нам нужно отслеживать, либо она же нужна, чтобы зафиксировать изменения в уже отслеживаемом файле.

Мне кажется, здесь нужен скриншот для наглядности).  
Вот, прямо сейчас я нажал «сохранить» в Word и нажимаю кнопку Коммит в Git Extensions.