**СИСТЕМА КОНТРОЛЯ ВЕРСИЙ GIT**

***Система контроля версий*** - программа, которая позволяет хранить разные версии одного и того же документа, легко переключаться между ранними и поздними вариантами, вносить и отслеживать изменения.

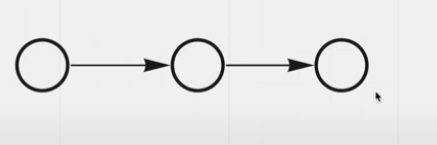
***Github*** – своего рода соцсеть для разработчиков, в которой можно делиться проектами, комментить их, ставить лайки и так далее.

***Репозиторий*** - это рабочая директория с вашим проектом. По сути, это та же папка с HTML, CSS, JavaScript и прочими файлами, что хранится у вас на компьютере, но находится на сервере GitHub. Поэтому вы можете работать с проектом удалённо на любой машине, не переживая, что какие-то из ваших файлов потеряются — все данные будут в репозитории при условии, что вы их туда отправите.

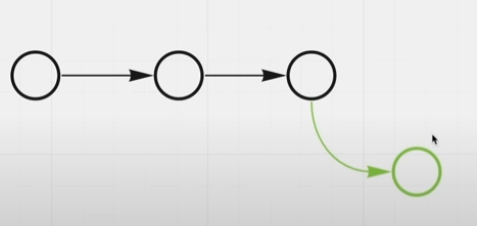
Если над проектом трудится команда разработчиков, как правило, создаётся общий репозиторий, в котором находится рабочая версия проекта (назовём его *мастер-репозиторий*). При этом каждый пользователь клонирует себе в профиль оригинальный репозиторий и работает именно с копией. Такая копия называется ***форком***. Так как **форк** — ваша персональная версия мастер-репозитория, в нём вы можете пробовать разные решения, менять код и не бояться что-то сломать в основной версии проекта.

***Создать коммит (commit)*** значит зафиксировать изменения любых файлов, входящих в репозиторий. По сути, ***коммит – это сохранение, к которому можно откатиться в случае ошибок при дальнейшей работе.***

***Ветка в Github*** – это копия репозитория. Как минимум, репозиторий должен содержать основную ветку, идя по которой разработчик движется в определённом векторе разработки, составляя попутно коммиты. Представим, что сохранения – это кружки. Сохранения основной ветки – чёрные кружки:

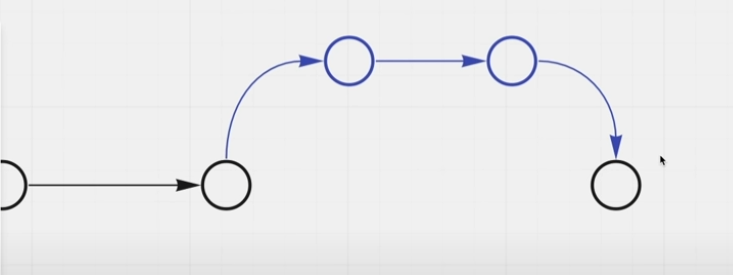


Что если разработчик придумал какую-нибудь фичу, которая не входила в планы или, допустим, сильно отличается от задуманного? Для этого он может создать новую ветку, по которой будет вести работу и не мешать разработке в основной ветке. Кстати, новая ветка при создании является копией последнего сохранения (т.е. третьего чёрного кружка):



***Запрос на извлечение (pull request)*** означает, что вы сообщаете другим, что вы передали изменения, внесённые вами в ветке, в главный репозиторий. Соавторы хранилища могут принять или отклонить запрос на извлечение. После его открытия вы можете обсудить и проанализировать свою работу с соавторами.

Допустим, мы дали возможность поработать над какой-либо частью проекта другу. Для этого он создаёт новую ветку и работает в ней, а потом создаёт пулл-реквест, чтобы внедрить эти изменения в основную ветку:



**ОСНОВНЫЕ КОМАНДЫ**

***git status*** – отслеживание изменений в репозитории

***git log*** – показ логов репозитория

***git add имя\_файла.формат*** - команда, которая говорит Git, на какие файлы нужно смотреть и записывать их изменения, а все остальные файлы будут игнорироваться (после этой команды можно ввести git status, чтобы посмотреть, какие произошли изменения)

git add \* - добавить все файлы в список отслеживаемых

***git commit -m "тут ваше сообщение о коммите"*** – сделать коммит внесённых изменений (подтвердить изменения)

***git push origin main*** – пуш коммита на main ветку (можно просто **git push**)

***git pull*** – обновить локальный репозиторий с сервера

Удаление файла/папки из репозитория:

