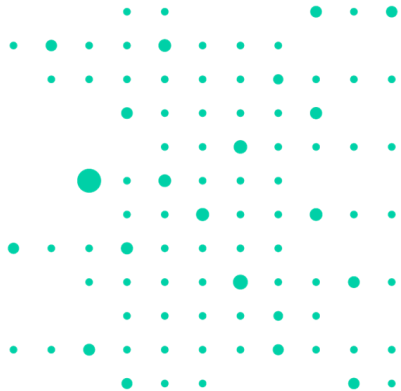




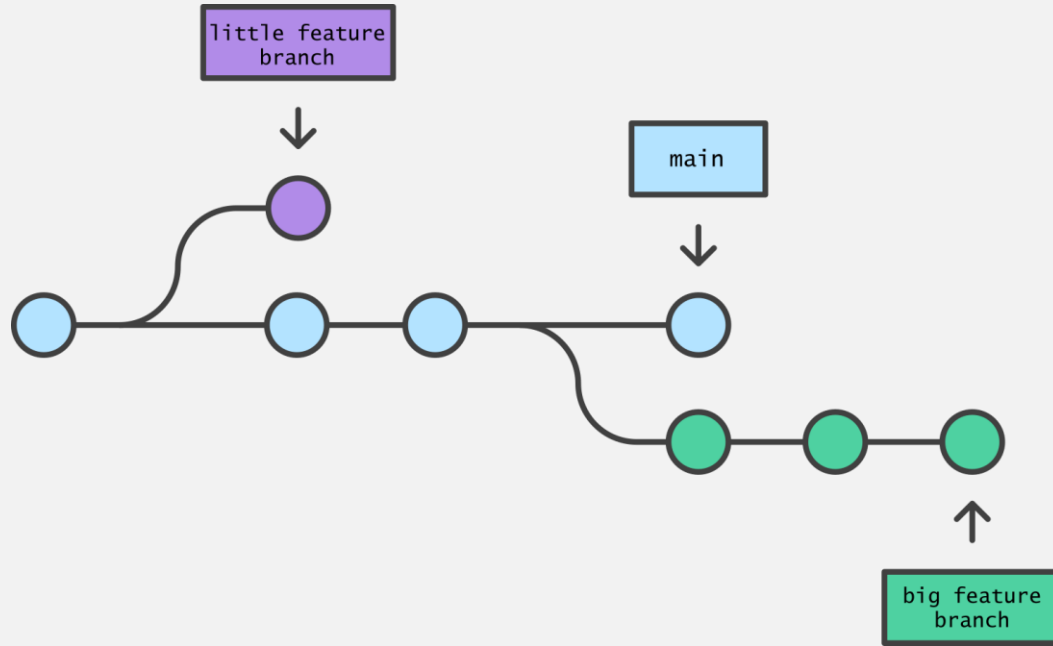
Advanced git usage



Git Branch

В Git ветки — это элемент повседневного процесса разработки. По сути ветки в Git представляют собой указатель на снимок изменений. Если нужно добавить новую возможность или исправить ошибку (незначительную или серьезную), вы создаете новую ветку, в которой будут размещаться эти изменения.

Git Branch



Git Branch

Распространенные опции

git branch(**git branch --list**) - Отображение списка веток в репозитории

git branch <branch> - Создание новой ветки с именем < branch >, но без перехода на нее.

git branch -d <branch> - Удаление указанной ветки, «безопасная» операция, поскольку Git не позволит удалить ветку, если в ней есть неслитые изменения.

Git Merge

Команда **git merge** выполняет слияние отдельных направлений разработки, созданных с помощью команды **git branch**, в единую ветку.

Проще всего слияние ветки ``main`` в функциональную ветку выполняется с помощью следующей команды:

```
git checkout feature
```

```
git merge main
```

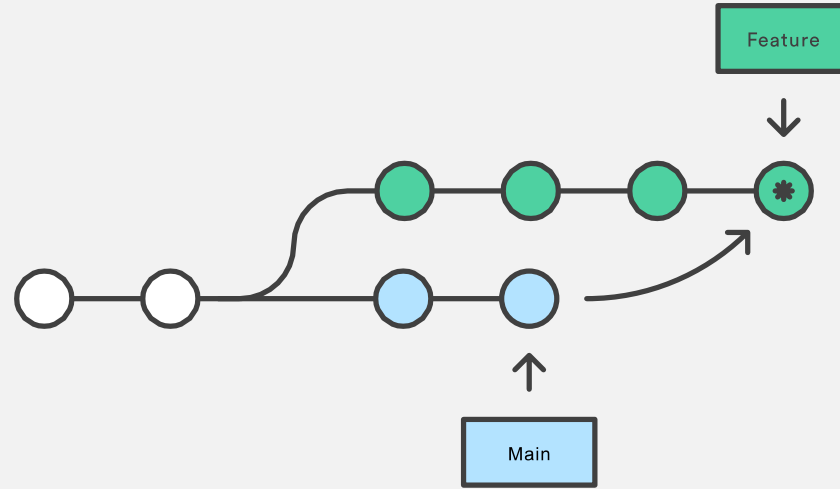
При желании этот код можно записать в одну строку:

```
git merge <feature> main
```

Эта операция создает в ветке ``<feature>`` новый «коммит слияния», связывающий истории обеих веток.

Git Merge

Merging main into the feature branch



* Merge Commit

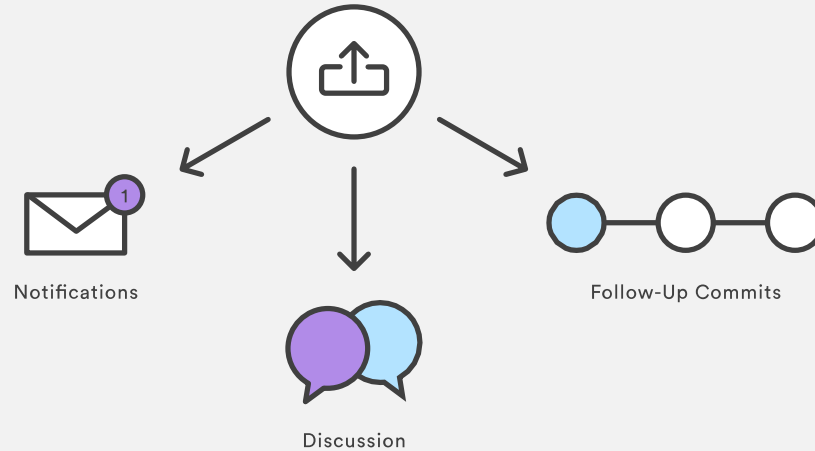
Git Pull request

В упрощенном виде запросы pull — это механизм, с помощью которого разработчик уведомляет участников команды о том, что он подготовил некий функционал. Закончив работу над функциональной веткой, разработчик создает запрос pull с помощью аккаунта Git системы которой используете. Так все участники процесса узнают, что требуется проверить код и выполнить слияние с главной веткой (main).

Git Pull request

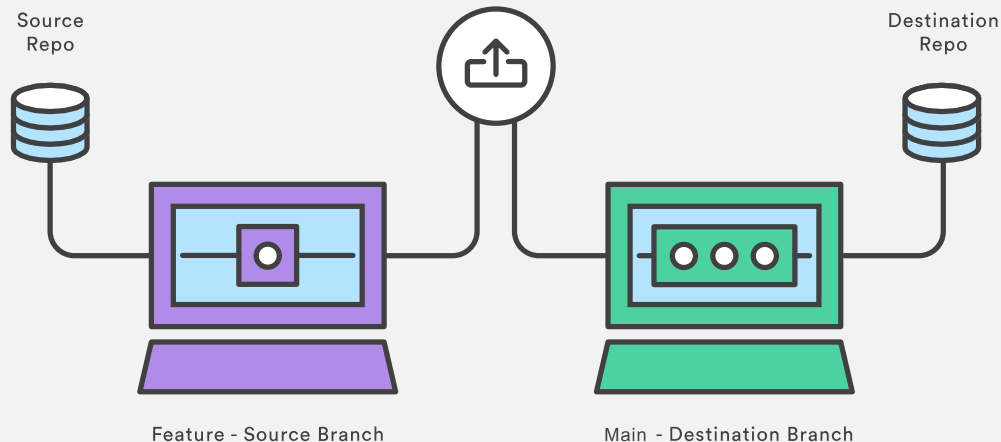
Однако запрос pull — это не просто уведомление, а специальный форум для обсуждения предлагаемой функции. Если с изменениями возникли какие-либо проблемы, участники команды могут публиковать в запросе pull отзывы и даже изменять функцию с помощью дополнительных коммитов. Все эти действия отслеживаются непосредственно внутри запроса pull.

Git Pull request



Git Pull request

Создавая пул-реквест, вы всего лишь *просите* другого разработчика (например, человека, занимающегося поддержкой проекта) *забрать* ветку из вашего репозитория в его репозиторий.



Git Конфликты

Слияние и конфликты являются неотъемлемой частью работы с Git.

В большинстве случаев Git самостоятельно решает, как автоматически интегрировать новые изменения.

Обычно конфликты возникают, когда два человека изменяют одни и те же строки в файле или один разработчик удаляет файл, который в это время изменяет другой разработчик.

Git Конфликты

Типы конфликтов слияния

Конфликт во время слияния может произойти в двух отдельных точках — при запуске и во время процесса слияния.

Далее рассмотрим, как разрешать каждый из этих конфликтных сценариев.

Git Конфликты

Git прерывает работу в самом начале слияния

Выполнение команды слияния прерывается в самом начале, если Git обнаруживает изменения в рабочем каталоге или разделе проиндексированных файлов текущего проекта.

Локальное состояние необходимо стабилизировать с помощью команд `git stash`, `git checkout`, `git commit` или `git reset`.

Git Конфликты

Git прерывает работу во время слияния

Сбой В ПРОЦЕССЕ слияния говорит о наличии конфликта между текущей локальной веткой и веткой, с которой выполняется слияние. Это свидетельствует о конфликте с кодом другого разработчика.

Git Конфликты

Разрешение конфликтов слияния

Самый простой способ разрешить конфликт — отредактировать конфликтующий файл. Откройте файл `merge.txt` в привычном редакторе. И найти подобные записи и исправить.

```
<<<<<< HEAD
```

```
=====
```

```
>>>>>> new_branch_to_merge_later
```

После редактирования файла выполните команду `git add merge.txt`, чтобы добавить новое объединенное содержимое в раздел проиндексированных файлов. И затем как обычно `commit` и потом `push` в ветку.