* 1. Что такое ветвление и зачем оно нужно?

Ветвление в Git - это механизм создания копии репозитория, которая позволяет рабоать над разными версиями своего проекта одновременно. Это позволяет разработчикам работать независимо друг от друга, не мешая друг другу и не затрагивая основную ветвь (обычно это ветвь master). Ветвления также позволяют экспериментировать с различными функциями, которые не готовы еще включать в основную ветвь. Если какая-то функция не работает или вызывает ошибки, вернуться к значимому моменту разработки можно, путем переключения на нужную ветвь. Ветвление и слияние в Git являются мощными инструментами, которые позволяют эффективно работать в команде и избежать ошибочных изменений в основной ветви.

* 1. Что такое Ветка?

Ветка (branch) в git — это линия разработки, которая происходит от определенной точки в истории проекта. Ветки помогают упростить и организовать процесс разработки, позволяя создавать и работать с разными версиями проекта одновременно. Каждая ветка имеет свое название и хранит уникальный набор коммитов. Ветки могут быть созданы как для разработки новых функций и исправлений, так и для тестирования и отладки кода без влияния на основную ветку разработки. В git существует специальная ветка master, которая обычно является основной веткой проекта.

* 1. Как создать новую ветку через командную строку?

Допустим вы хотите создать новую ветку с именем testing. Вы можете это сделать командой git branch :

$ git branch testing

В результате создаётся новый указатель на текущий коммит.

* 1. Как переключиться на новую ветку через командную строку?

Вы можете легко это увидеть при помощи простой команды git log, которая покажет вам куда указывают указатели веток. Эта опция называется --decorate.

$ git log --oneline --decorate

f30ab (HEAD -> master, testing) Add feature #32 - ability to add new formats to the central interface

34ac2 Fix bug #1328 - stack overflow under certain conditions

98ca9 Initial commit

Для переключения на существующую ветку выполните команду git checkout. Давайте переключимся на ветку testing:

$ git checkout testing

В результате указатель HEAD переместится на ветку testing.

* 1. Как происходит ветвление и слияние? Какие бывают основные конфликты слияния?

**git log не показывает все ветки по умолчанию**

Если выполнить команду git log прямо сейчас, то в её выводе только что созданная ветка «testing» фигурировать не будет.

Ветка никуда не исчезла; просто Git не знает, что именно она вас интересует, и выводит наиболее полезную по его мнению информацию. Другими словами, по умолчанию git log отобразит историю коммитов только для текущей ветки.

Для просмотра истории коммитов другой ветки необходимо явно указать её имя: git log testing Чтобы посмотреть историю по всем веткам — выполните команду с дополнительным флагом: git log --all.

**Переключение веток меняет файлы в рабочем каталоге**

Важно запомнить, что при переключении веток в Git происходит изменение файлов в рабочем каталоге. Если вы переключаетесь на старую ветку, то рабочий каталог будет выглядеть так же, как выглядел на момент последнего коммита в ту ветку. Если Git по каким-то причинам не может этого сделать — он не позволит вам переключиться вообще.

**Одновременное создание новой ветки и переключение на неё**

Как правило, при создании новой ветки вы хотите сразу на неё переключиться — это можно сделать используя команду git checkout -b <newbranchname>.

* 1. Как происходит создание новой ветки в Pycharm 1. Откройте Pycharm и выберите проект, в котором вы хотите создать новую ветку.  
     2. Нажмите на кнопку "VCS" (версионный контроль) в верхней панели инструментов.  
     3. Выберите "Git" из выпадающего меню.  
     4. Выберите "Branches" (ветви) из контекстного меню.  
     5. Нажмите на кнопку "New Branch" (новая ветвь) в правом верхнем углу окна.  
     6. Введите название новой ветки и выберите базовую ветку (обычно это ветка "master").  
     7. Нажмите на кнопку "Create" (создать).  
     8. После создания новой ветки, вы можете переключиться на нее, выбрав ее в списке веток.  
     9. Теперь вы можете внести изменения в новую ветку и сохранить их в репозитории.