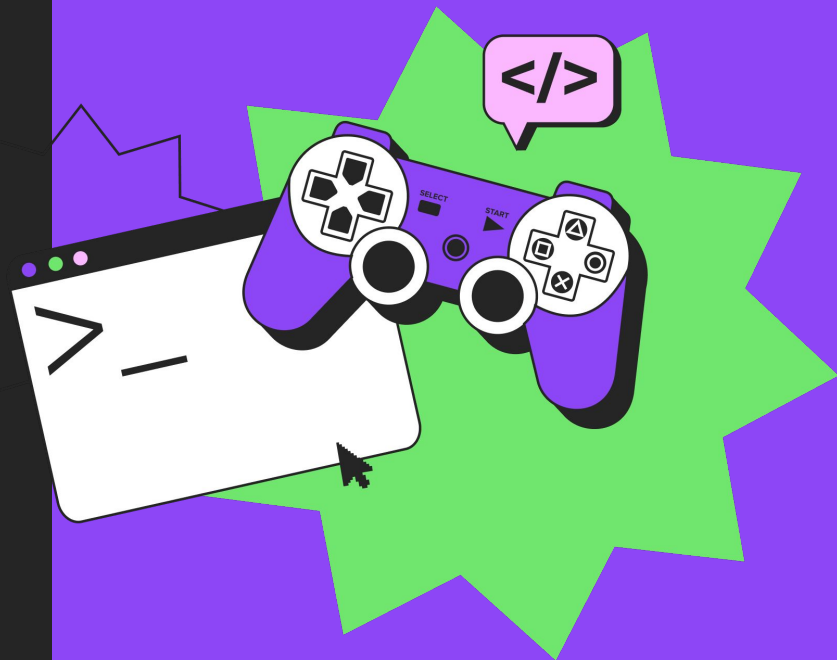
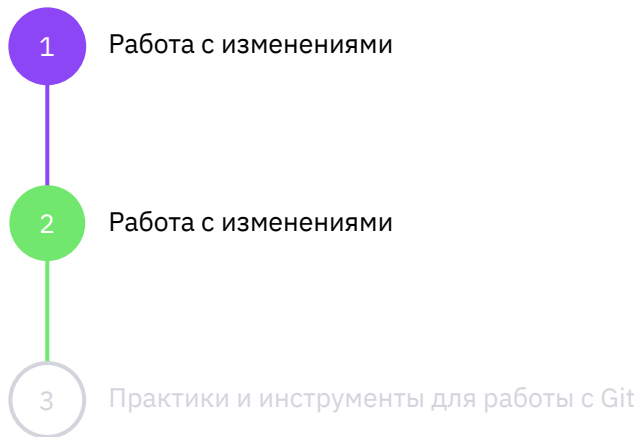


Система контроля версий Git — advanced





План курса



Работа с изменениями

Урок 2

Жизненный цикл изменений в Git, сравнение версий и просмотр изменений, игнорирование изменений, gitignore, отмена сохранённых и несохранённых изменений и слияния веток, откладывание, слияние и перемещение изменений





Что будет на уроке сегодня



Что будет на уроке сегодня



Жизненный цикл изменений в Git



Что будет на уроке сегодня






Жизненный цикл изменений в Git



Сравнение версий и просмотр изменений







Что будет на уроке сегодня

-  Жизненный цикл изменений в Git
-  Сравнение версий и просмотр изменений
-  Игнорирование изменений, gitignore








Что будет на уроке сегодня

-  Жизненный цикл изменений в Git
-  Сравнение версий и просмотр изменений
-  Игнорирование изменений, gitignore
-  Отмена несохранённых изменений









Что будет на уроке сегодня

-  Жизненный цикл изменений в Git
-  Сравнение версий и просмотр изменений
-  Игнорирование изменений, gitignore
-  Отмена несохранённых изменений
-  Отмена сохранённых изменений










Что будет на уроке сегодня

-  Жизненный цикл изменений в Git
-  Сравнение версий и просмотр изменений
-  Игнорирование изменений, gitignore
-  Отмена несохранённых изменений
-  Отмена сохранённых изменений
-  Отмена слияний веток











Что будет на уроке сегодня

-  Жизненный цикл изменений в Git
-  Сравнение версий и просмотр изменений
-  Игнорирование изменений, gitignore
-  Отмена несохранённых изменений
-  Отмена сохранённых изменений
-  Отмена слияний веток
-  Откладывание изменений



Что будет на уроке сегодня

-  Жизненный цикл изменений в Git
-  Сравнение версий и просмотр изменений
-  Игнорирование изменений, gitignore
-  Отмена несохранённых изменений
-  Отмена сохранённых изменений
-  Отмена слияний веток
-  Откладывание изменений
-  Слияние и перемещение изменений



Жизненный цикл изменений в Git



Состояния файлов в Git-репозитории



Состояния файлов в Git-репозитории

Закоммичен
committed



Состояния файлов в Git-репозитории

Закоммичен
committed



Изменён
not staged



Состояния файлов в Git-репозитории

Закоммичен
committed



Изменён
not staged



ГОТОВ К КОММИТУ
staged



Состояния файлов в Git-репозитории

Закоммичен
committed



Изменён
not staged



Готов к коммиту
staged





Состояния файлов в Git-репозитории

Закоммичен
committed



Изменён
not staged



Готов к коммиту
staged



Создание коммита

```
git commit -m 'some changes'
```



Состояния файлов в Git-репозитории

Просмотр состояния

```
git status
```

Закоммичен
committed



Изменён
not staged



Готов к коммиту
staged



Создание коммита

```
git commit -m 'some changes'
```



Состояния файлов в Git-репозитории

Просмотр состояния

```
git status
```

Добавление в индекс

```
git add <file>
```

Закоммичен
committed



Изменён
not staged



Готов к коммиту
staged



Создание коммита

```
git commit -m 'some changes'
```



Индекс в системе контроля версий Git



Индекс в системе контроля версий Git

Специальная область, в которой хранятся изменения файлов, готовые к коммиту


Команды управления состояниями файлов

Команды управления состояниями файлов



```
git status
```

Команды управления состояниями файлов

 `git status`

 `git add <file(s)>`

Команды управления состояниями файлов



```
git status
```



```
git add <file(s)>
```



```
git commit -m 'some changes'
```

Команды управления состояниями файлов



```
git status
```



```
git add <file(s)>
```



```
git commit -m 'some changes'
```



```
git show
```



Сравнение версий и просмотр изменений

Изменения в рабочей директории

Изменения в рабочей директории



В коде одного файла перед коммитом

Изменения в рабочей директории



В коде одного файла перед коммитом



В проекте перед коммитом

Изменения в рабочей директории



В коде одного файла перед коммитом







В проекте перед коммитом








В файле между коммитами







Изменения в рабочей директории

-  В коде одного файла перед коммитом
-  В проекте перед коммитом
-  В файле между коммитами
-  В проекте между коммитами

Изменения в рабочей директории


-  В коде одного файла перед коммитом
-  В проекте перед коммитом
-  В файле между коммитами
-  В проекте между коммитами
-  В разных ветках


Изменения в рабочей директории

-  В коде одного файла перед коммитом
-  В проекте перед коммитом
-  В файле между коммитами
-  В проекте между коммитами
-  В разных ветках
-  В коде файла вне системы Git

Команды просмотра изменений

Команды просмотра изменений

 `git log`

 `git status`

Команды просмотра изменений



`git log`



`git status`



`git diff <file>`



`git diff`

Команды просмотра изменений



```
git log
```



```
git status
```



```
git diff <file>
```



```
git diff
```



```
git diff <commit>
```



```
git diff <commit1> <commit2>
```



```
git diff <commit1> <commit2> <file>
```



```
git diff <file1> <file2>
```


Команды просмотра изменений



`git log`



`git status`



`git diff <file>`



`git diff`



`git diff <commit>`



`git diff <commit1> <commit2>`



`git diff <commit1> <commit2> <file>`



`git diff <file1> <file2>`



`git blame <file>`



Игнорирование изменений, файл .gitignore

Не рекомендуется хранить в Git-репозиториях

Не рекомендуется хранить в Git-репозиториях



Лог-файлы (логи)

Не рекомендуется хранить в Git-репозиториях






Лог-файлы (логи)







Файлы, загруженные пользователями






Не рекомендуется хранить в Git-репозиториях

-  Лог-файлы (логи)
-  Файлы, загруженные пользователями
-  Служебные файлы сред разработки







Не рекомендуется хранить в Git-репозиториях

-  Лог-файлы (логи)
-  Файлы, загруженные пользователями
-  Служебные файлы сред разработки
-  Внешние библиотеки








Не рекомендуется хранить в Git-репозиториях

-  Лог-файлы (логи)
-  Файлы, загруженные пользователями
-  Служебные файлы сред разработки
-  Внешние библиотеки
-  Файлы локальной конфигурации

Не рекомендуется хранить в Git-репозиториях

-  Лог-файлы (логи)
-  Файлы, загруженные пользователями
-  Служебные файлы сред разработки
-  Внешние библиотеки
-  Файлы локальной конфигурации
-  Файлы операционной системы

Не рекомендуется хранить в Git-репозиториях

-  Лог-файлы (логи)
-  Файлы, загруженные пользователями
-  Служебные файлы сред разработки
-  Внешние библиотеки
-  Файлы локальной конфигурации
-  Файлы операционной системы
-  Очень большие файлы


Синтаксис файла .gitignore

Синтаксис файла .gitignore




`error.log`

Синтаксис файла .gitignore


 `error.log`

 `*.log`


Синтаксис файла .gitignore

 `error.log`


 `*.log`

 `logs/`

Синтаксис файла .gitignore

 `error.log`

 `*.log`

 `logs/`

 `logs/*.log`

Синтаксис файла .gitignore



`error.log`



`*.log`



`logs/`



`logs/*.log`



`!logs/special.log`



Отмена несохранённых изменений

Удаление несохранённых изменений

Удаление несохранённых изменений



```
git restore <file>
```

Удаление несохранённых изменений



```
git restore <file>
```



```
git restore --staged <file>
```

Удаление несохранённых изменений



```
git restore <file>
```



```
git restore --staged <file>
```



```
git reset --hard
```

Удаление несохранённых изменений



```
git restore <file>
```



```
git restore --staged <file>
```



```
git reset --hard
```



```
git clean -f
```

Удаление несохранённых изменений



```
git restore <file>
```



```
git restore --staged <file>
```



```
git reset --hard
```



```
git clean -f
```



```
git rm --cached
```



Отмена сохранённых изменений


Отмена сохранённых изменений

Отмена сохранённых изменений



```
git checkout <commit> <file>
```

Отмена сохранённых изменений

 `git checkout <commit> <file>`

 `git revert <commit>`

Отмена сохранённых изменений



```
git checkout <commit> <file>
```



```
git revert <commit>
```



```
git revert --no-commit <commit>
```

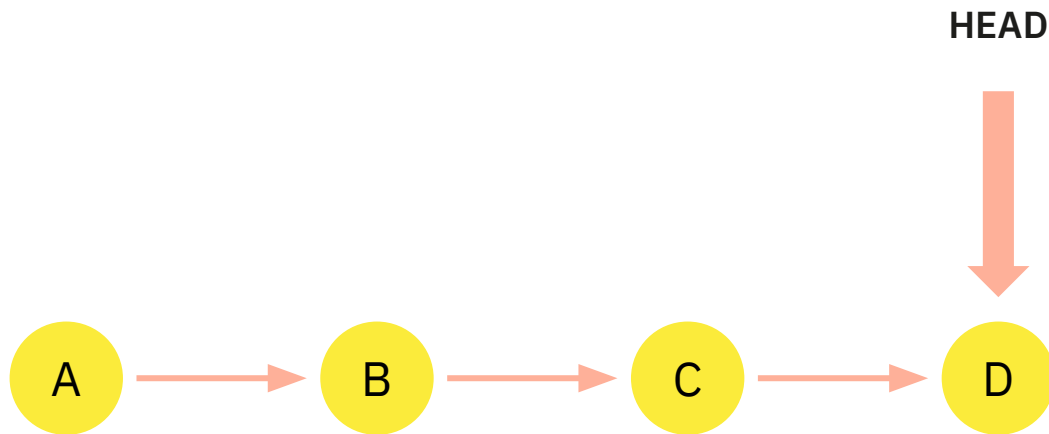
Удаление (сброс) коммитов

Указатель HEAD

Указатель HEAD

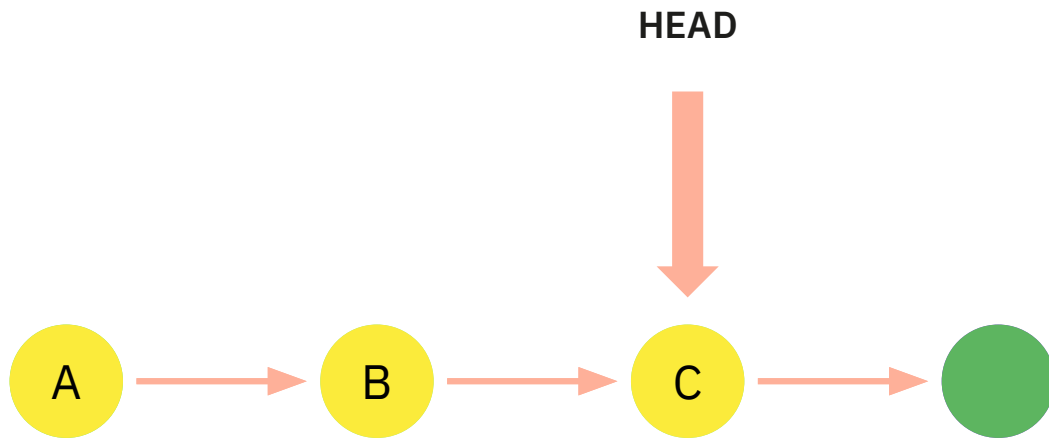


Указатель HEAD

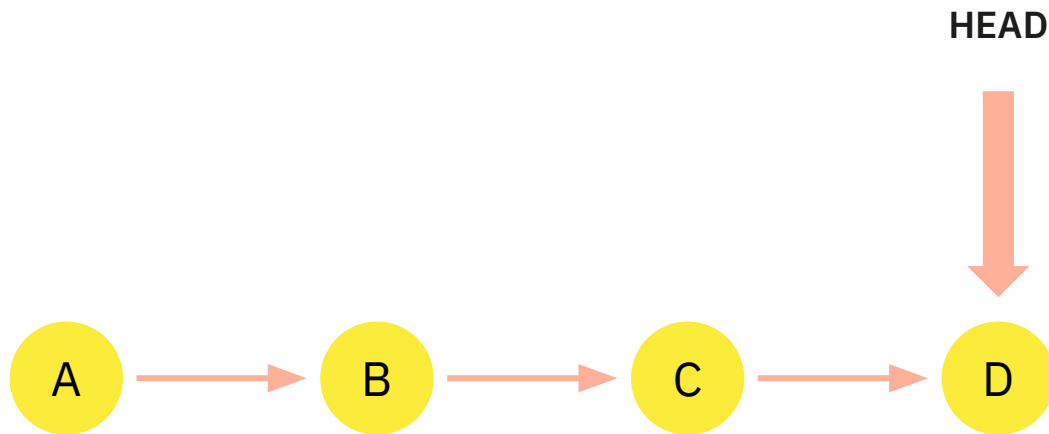


Указатель HEAD

```
git reset --soft C
```

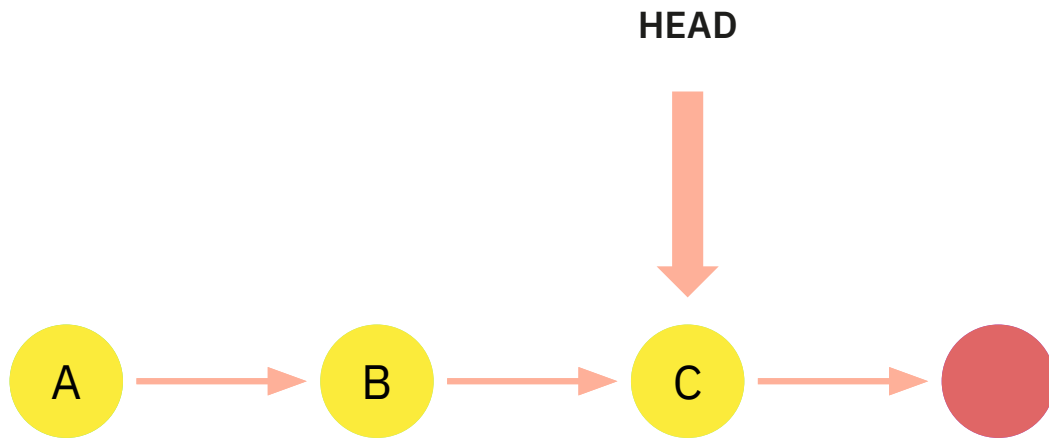


Указатель HEAD

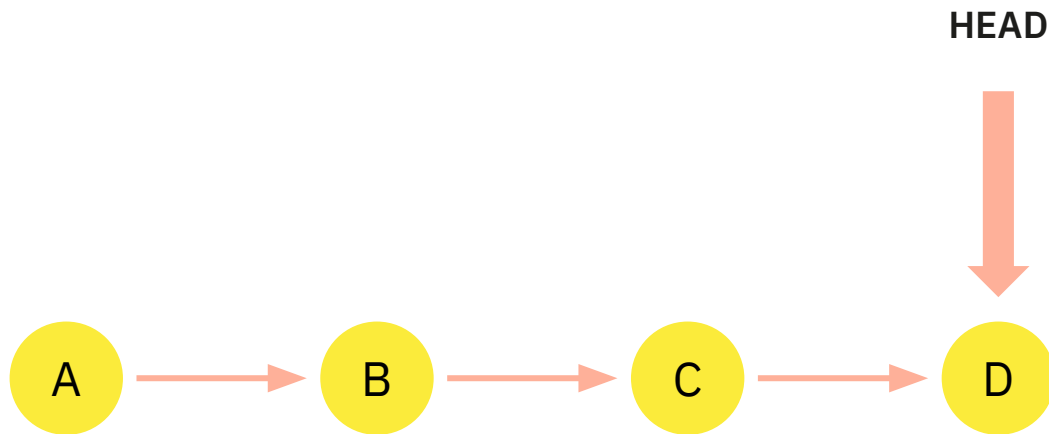


Указатель HEAD

```
git reset --mixed C
```

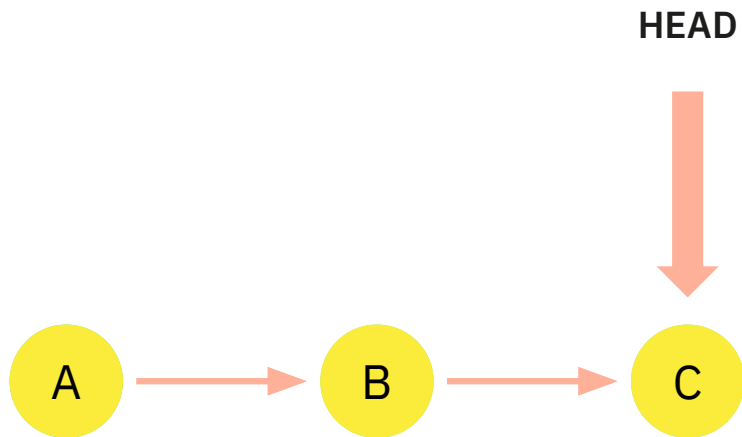


Указатель HEAD





Указатель HEAD


```
git reset --hard C
```




Удаление (сброс) коммитов


 `git reset --soft <commit>`


 `git reset --mixed <commit>`


 `git reset --hard <commit>`


Удаление (сброс) коммитов

 `git reset --soft <commit>`

 `git reset --mixed <commit>`

 `git reset --hard <commit>`

 `git commit --amend -m <new>`


 `git commit --amend --no-edit`




Отмена слияния веток

Отмена слияния веток

Отмена слияния веток

 `git reset --merge <commit>`

 `git merge --abort`



Откладывание изменений

Принцип работы команд `git stash` и `git stash pop`

Принцип работы команд `git stash` и `git stash pop`

 `git stash`  0

 `git stash`  1

 `git stash`  2

Принцип работы команд `git stash` и `git stash pop`

 `git stash`  0

 `git stash`  1

 `git stash`  2

 `git stash pop`  2

Принцип работы команд `git stash` и `git stash pop`

 `git stash`  0

 `git stash`  1

 `git stash`  2

 `git stash pop`  2

 `git stash pop`  1

Принцип работы команд `git stash` и `git stash pop`

 `git stash`  0

 `git stash`  1

 `git stash`  2

 `git stash pop`  2

 `git stash pop`  1

 `git stash pop`  0

Команды откладывания изменений

Команды откладывания изменений



`git stash`

Команды откладывания изменений



`git stash`



`git stash list`

Команды откладывания изменений



`git stash`



`git stash list`



`git stash pop`

Команды откладывания изменений



`git stash`



`git stash list`



`git stash pop`



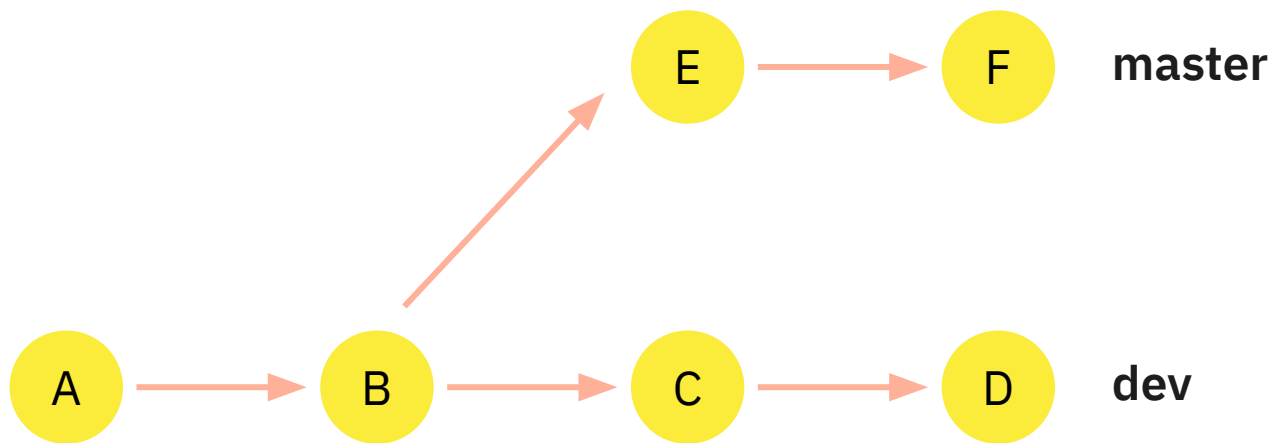
`git stash drop`



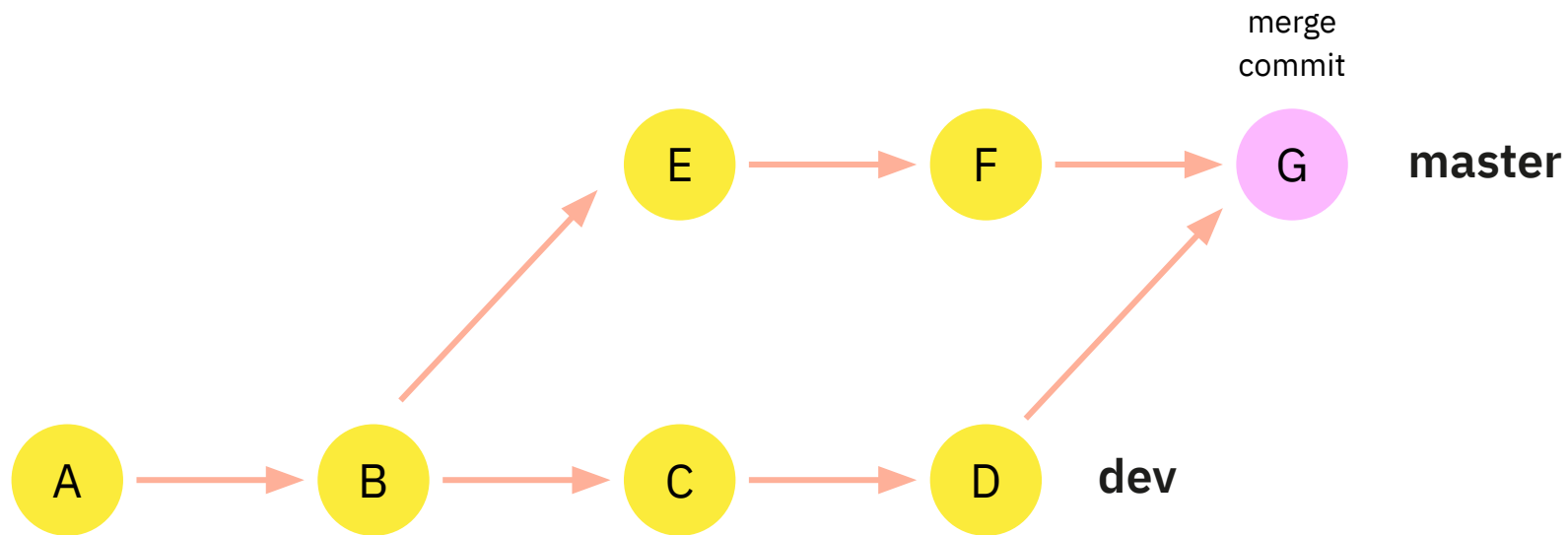
Перемещение изменений

Слияние, операция merge

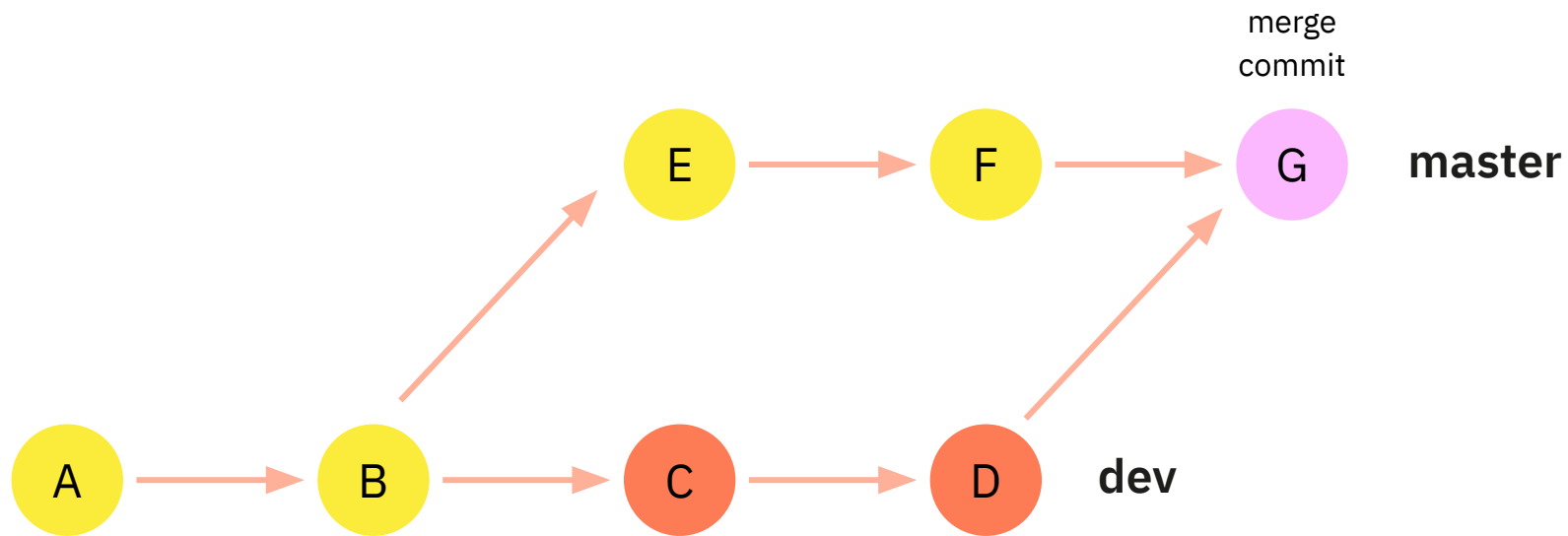
Слияние, операция merge



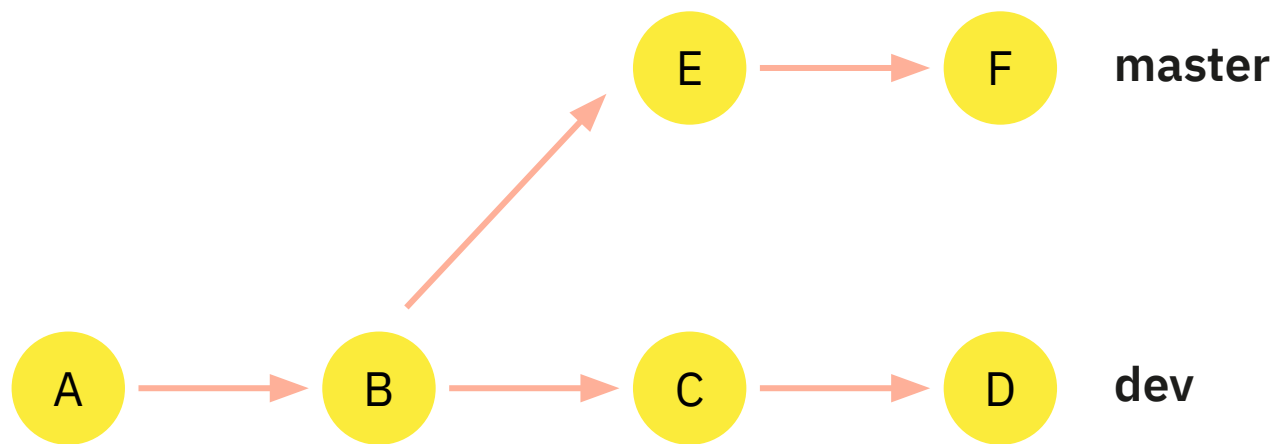
Слияние, операция merge



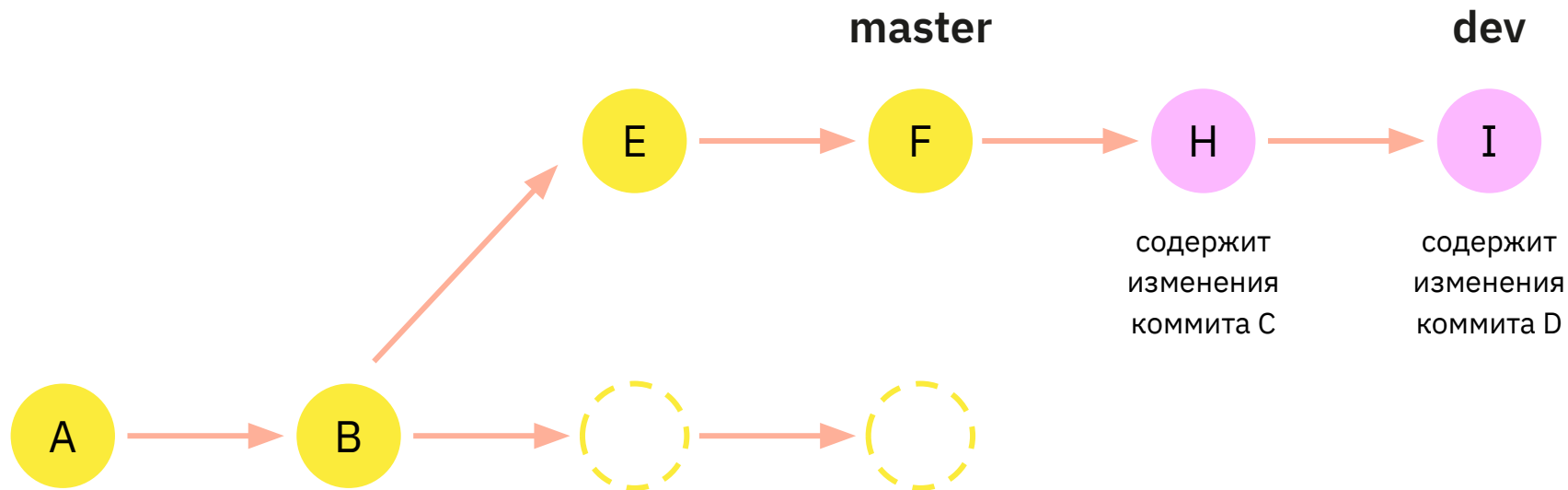
Слияние, операция merge



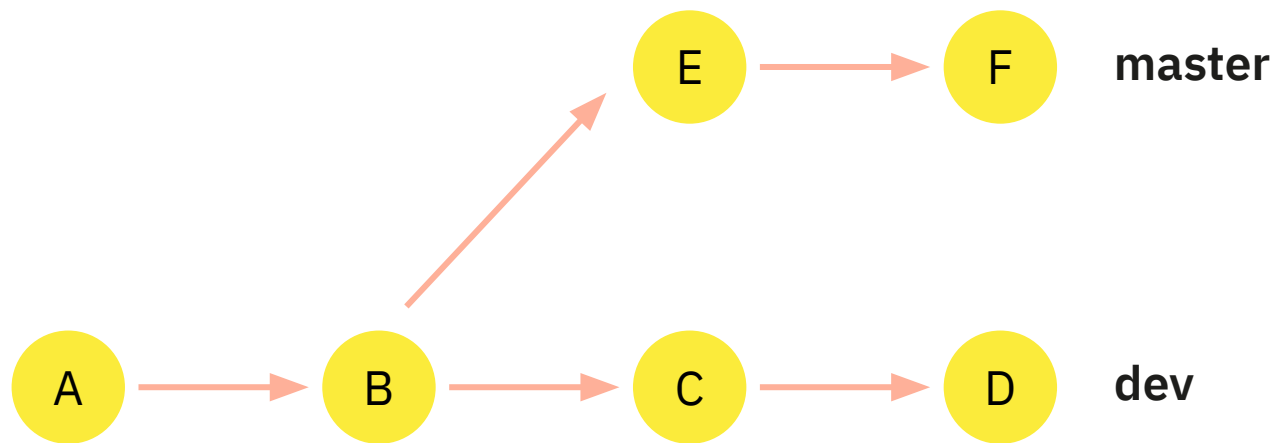
Слияние, операция rebase



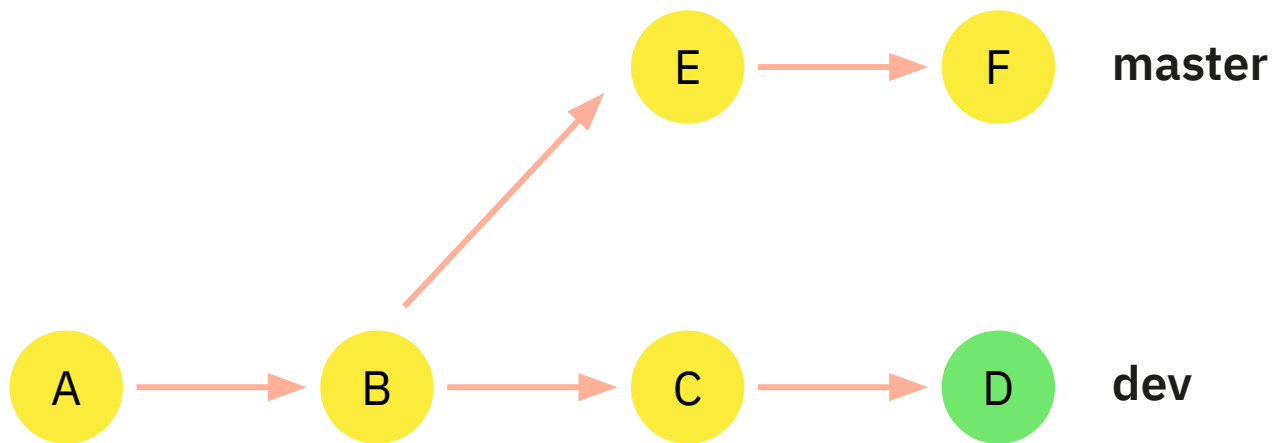
Слияние, операция rebase



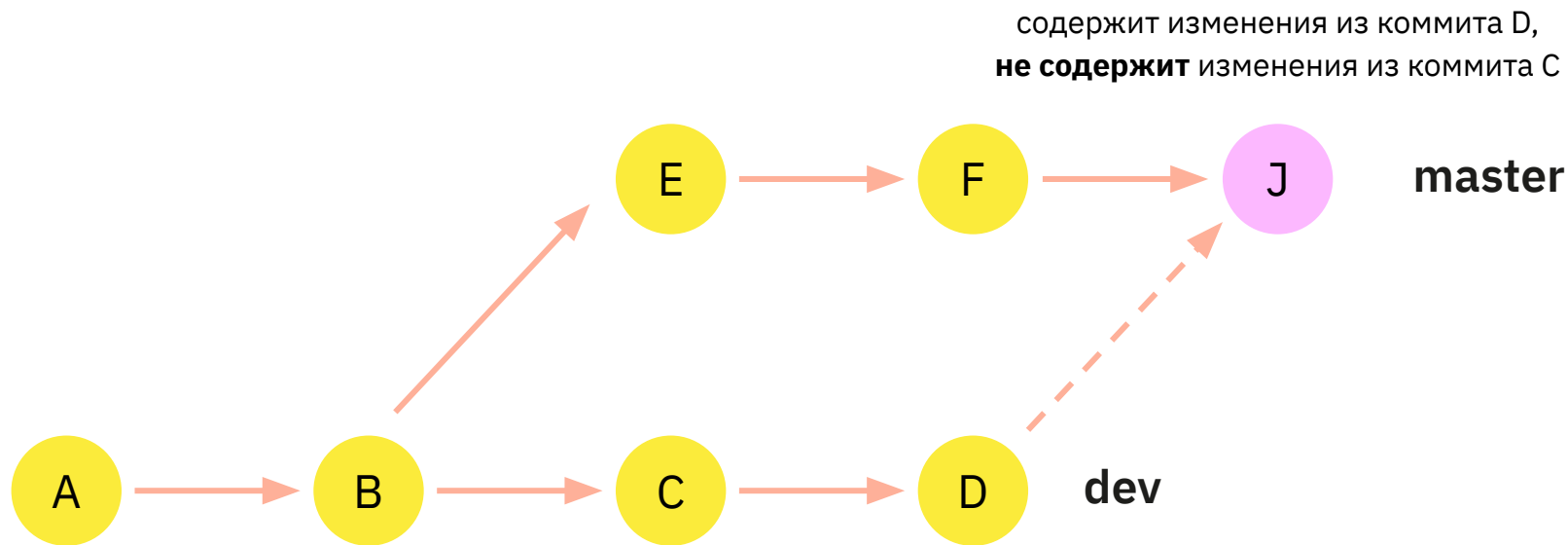
Слияние, операция cherry-pick



Слияние, операция cherry-pick



Слияние, операция cherry-pick



Команды перемещения изменений

Команды перемещения изменений

 `git merge new_branch`

Команды перемещения изменений



```
git merge new_branch
```



```
git rebase new_branch
```

Команды перемещения изменений



```
git merge new_branch
```











```
git rebase new_branch
```



```
git cherry-pick commit
```



Что вы сегодня узнали

-  Жизненный цикл изменений в Git
-  Сравнение версий и просмотр изменений
-  Игнорирование изменений, gitignore
-  Отмена несохранённых изменений
-  Отмена сохранённых изменений
-  Отмена слияний веток
-  Откладывание изменений
-  Слияние и перемещение изменений



Спасибо за внимание!