HABR

Переходим в проект

git ?

git init - созд гит репозиторий (обьявить)

git config --global user.name "Deedoc" - имя изменения (кто сделал)

git config --global user.email mishasavitskiy2006@gmail.com - Обьявить имейл

cat .git/config - Посмотреть номер и почту самому

РЕШЕНИЕ git config --global --add safe.directory C:/Users/Root/Desktop/Git1

'C:/Users/Root/Desktop/Git1' is owned by:

BUILTIN/▒▒▒▒▒▒▒▒▒▒▒▒▒▒ (S-1-5-32-544)

but the current user is:

DEEDOCPC/Root (S-1-5-21-145950746-3346160413-1486257710-1001)

To add an exception for this directory, call:

git config --global --add safe.directory C:/Users/Root/Desktop/Git1

{ Сделать пост имя и почту: git config --global user.name "Deedoc" - имя изменения (кто сделал) git config --global user.email mishasavitskiy2006@gmail.com }

Просмотреть изменения global: git config --list --global Or cat ~/.gitconfig

Алиасы:

git config --global alias.c '!config --global;echo "Hello"' -На c выполн config

git config --global alias.c '!git ...;git ...'

git c --list Содержание c алиаса

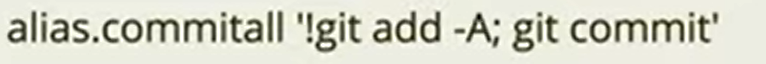
git help ? -Подсказки Or git help

После : слева внизу: / эт поиск п слову и т.д

ДОБАВЛЕНИЕ ЭЛЕМЕНТОВ:

- если в индексе есть файлы и надо обновить без индекса

-для определенного файла (gitignore) в репозиторий

-алиас чтоб все комитить

ls- список файлов

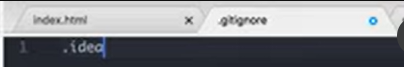
git status- файлы кот не добавлены

git add ?.? --> index

git reset HEAD .файл без расширения - delete file from index

git commit

GITIGNORE in the directory



C WIN на UNIX:

Git update-index –chmod=+x . (.=все файлы)

Инф о коммите: git show (Или можно еще идентификатор добавить)

При коммите указ автора и время с почтой: git commit –author=’? <Почта>’ –data=’…’



 -Когда мало текста



Файлы:

Git rm -r src = rm -r src + git add src

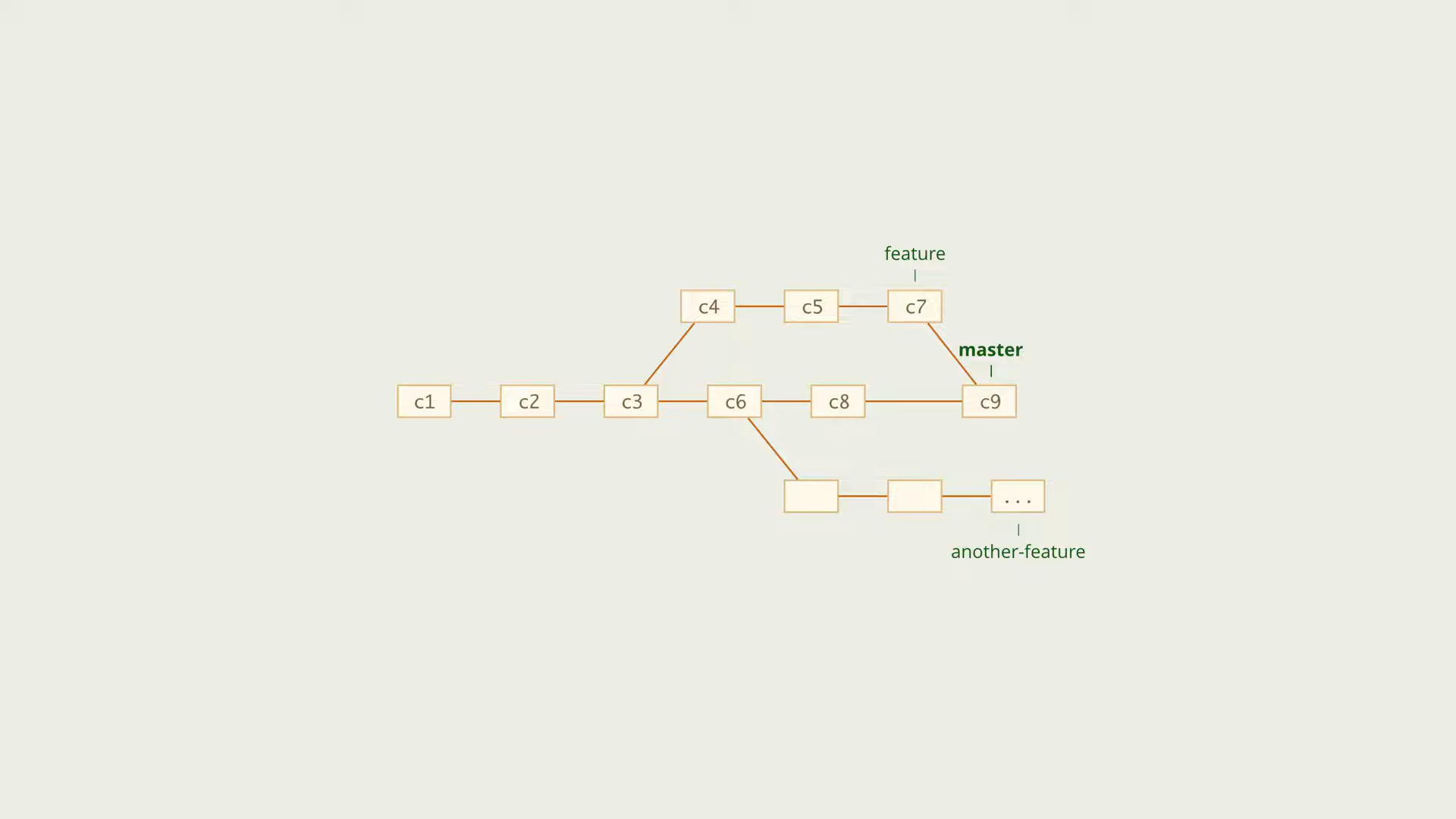
Cashed для операций с индексом вместо рабочей директории

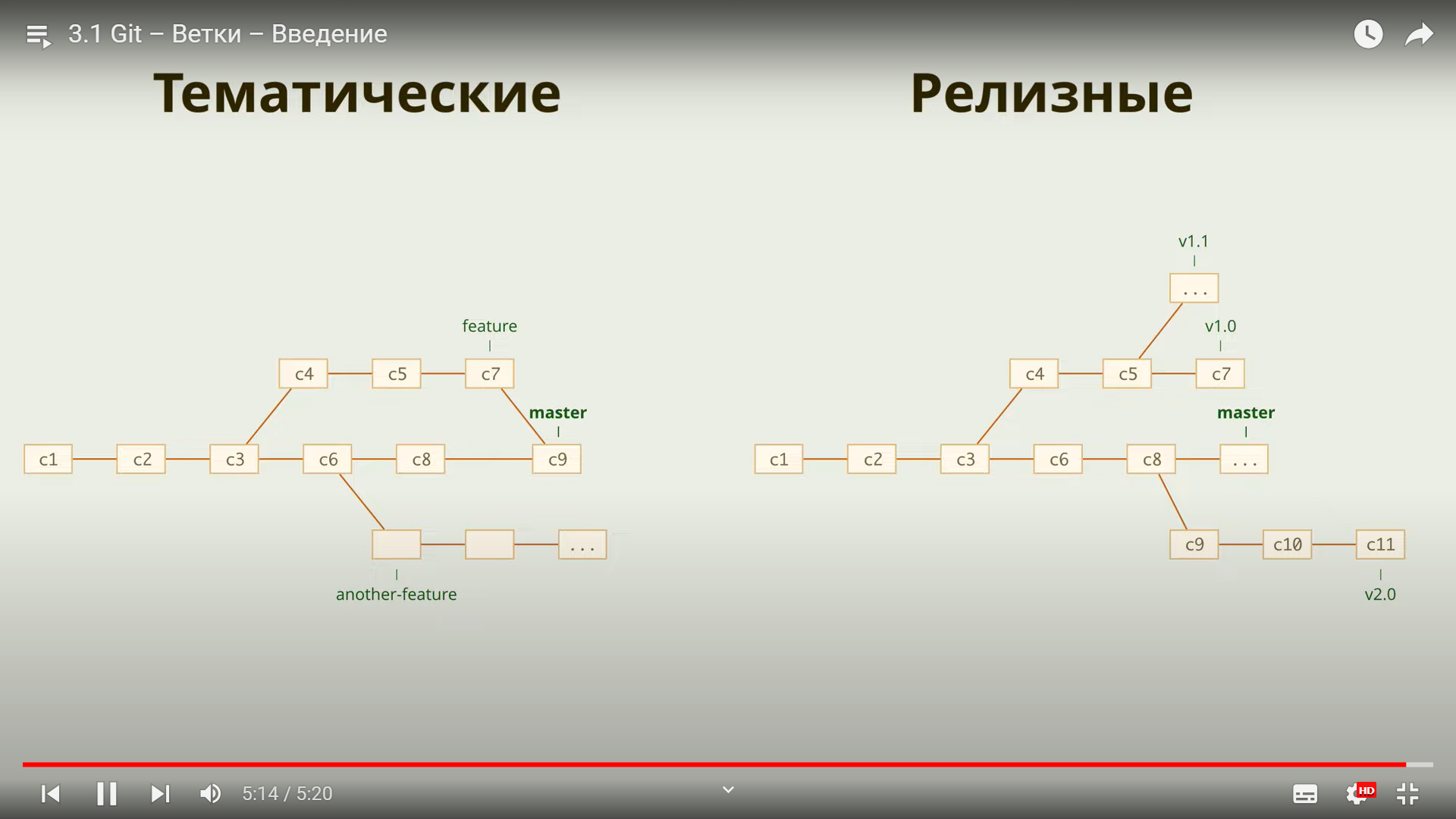
Git rm -f файл

Переминование:

Git mv index.html hello.html

ВЕТВИ:





Просмотр веток: Git branch -v

Файл Head хрант ссылку на тек ветку

1 ветка: Git commit -m “?????”

2: git branch имя – Созд новой ветки (|| master) Git checkout -b имя –

Созд ветки и переключени

Git checkout имя ветки – Переключение на другую ветку

ОСОБЕННОСТИ CHECKOUT

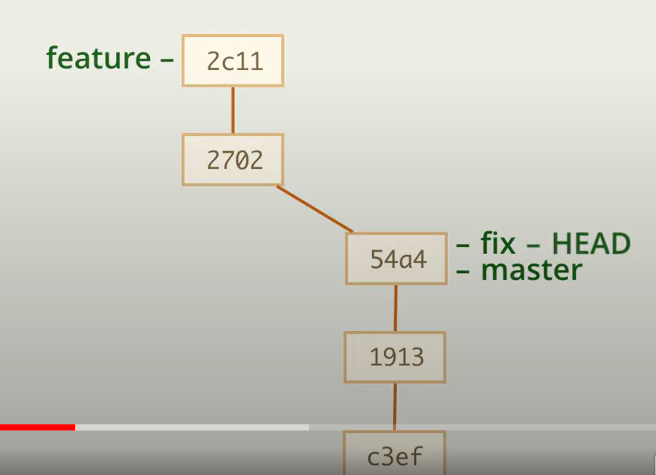
git checkout ? (master или др.) Если ошибка (Она только при создании новых файлов):

1)Отмен все изменения (не закомиченные) git checkout -f ? (master,HEAD)

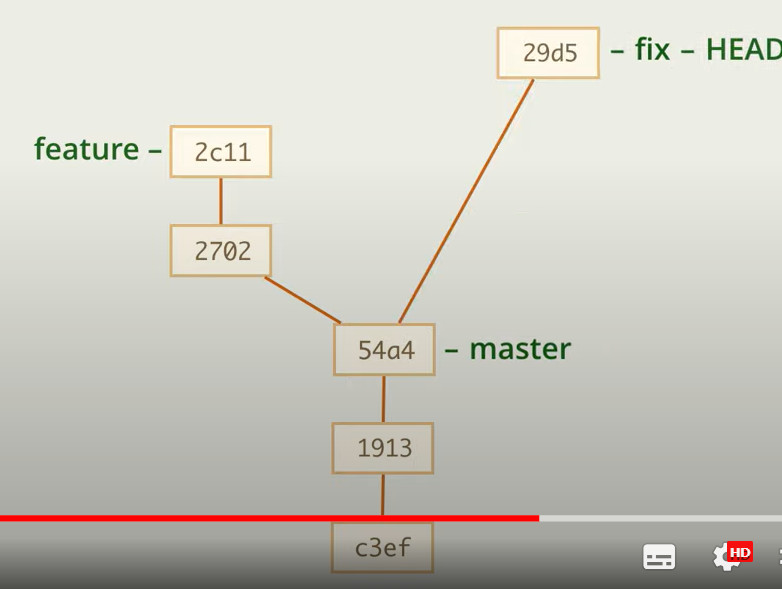
2)Сохран изменен: git stash – Архивация и удаление изменений в спец виде.

Если при переходе на ветку ЛЮБУЮ хотит вернуть все изменения: git stash pop

В мастер созд новую ветку



Коммит



Перенос Master вниз по ветке:

1. Созд фикс
2. Уходим с ветки мастер
3. Мастер назад (git branch -f master цифры коммита )
4. Фикс естественно остается в нач положении master

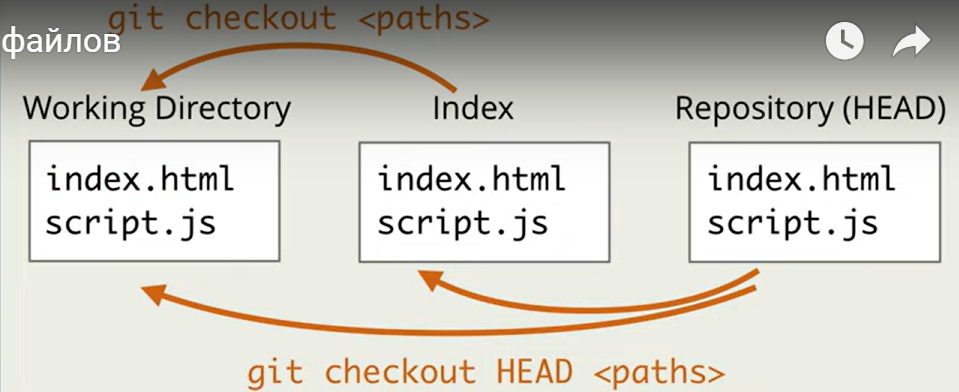
Переход на коммиты:

Git checkout <<id коммита>>

Git cherry-pick <<id комм>> - Скопировать на взятую ветку id комм.

Вернуть версию файла несколько коммитов назад –

Git checkout путь файл



Убрать файл из индекса: git reset файл

ИНФ О КОММИТАХ- git log / git log “\*” –online

Git log master

git Show master - id и другое

git show “\*”:index.html – what was in the file

git show :index.html- что в файле в ближ коммите

Пути



Head=@

~~ ⬄ ~2

ПОИСК ID ПО СЛОВУ

GIT SHOW :/”\*”

Git merge <<на какой коммит применить текущий коммит>>

Прошлый идентификатор коммита master: cat .git/ORIG\_HEAD

Переместить обратно Master: git branch -f master ORIG\_HEAD

**Удаление веток**

Git branch -d “\*” (Удаляет чисто ссылку) !!!!!!!Если head на этой ветку происх удаление

/D

Удалить изменения: опять по Id присвоить имя ветке(git branch Имя ID)

Удаление директориб от незаком изменений: git checkout -f ?

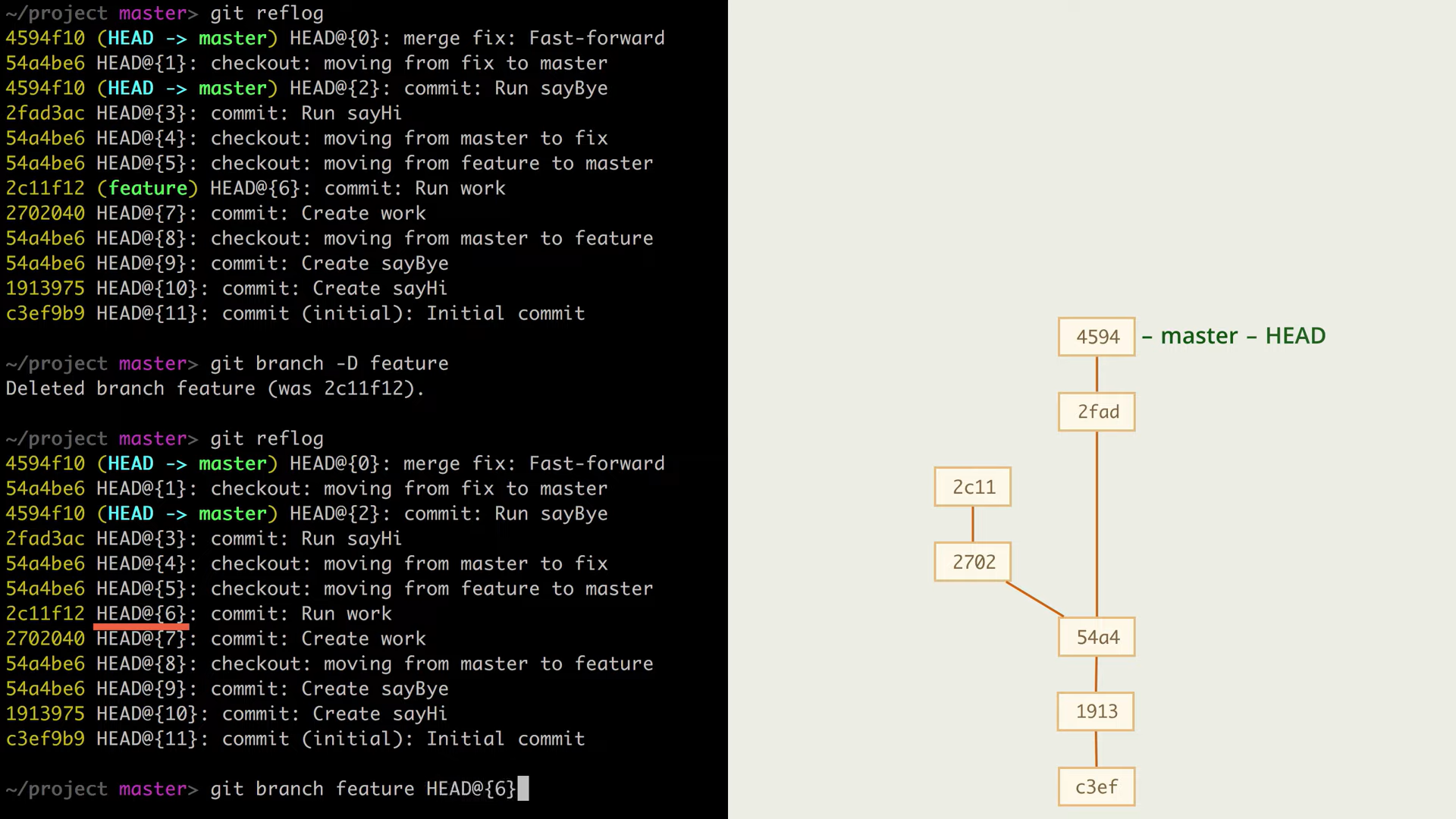
Git reset –hard

Если не удал изменения от неотсл файла : git clead -dxf

**ПРОСМОТРЕТЬ ID**

Git reflog – Все id HEAD (что с ним происходило) (СНИЗУ ВВЕРХ)

Воссстанлвление старых удал веток



ПЕРЕЙТИ В КОММИТ:

1)Git reset –hard id – переход на коммит по id перенос master

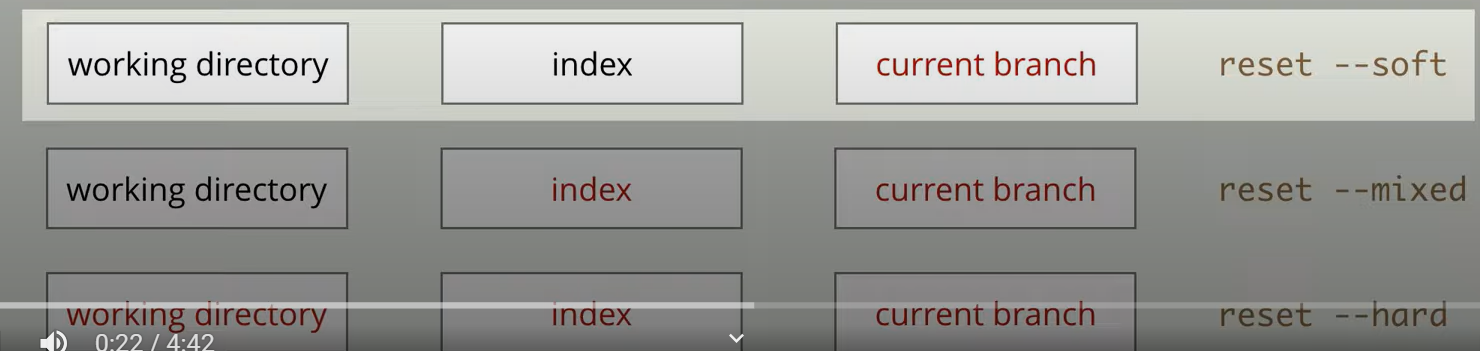
ОТКАТ 

2)git reset -–soft ? (Если не трогать раб директорию и индекс)

Git commit --amend (Откат назад на 1 коммит и создание нового паралельно старому(Предыдузий в orig\_head))

3) mixed ⬄ По умолчанию

git reset ?



Посмотре

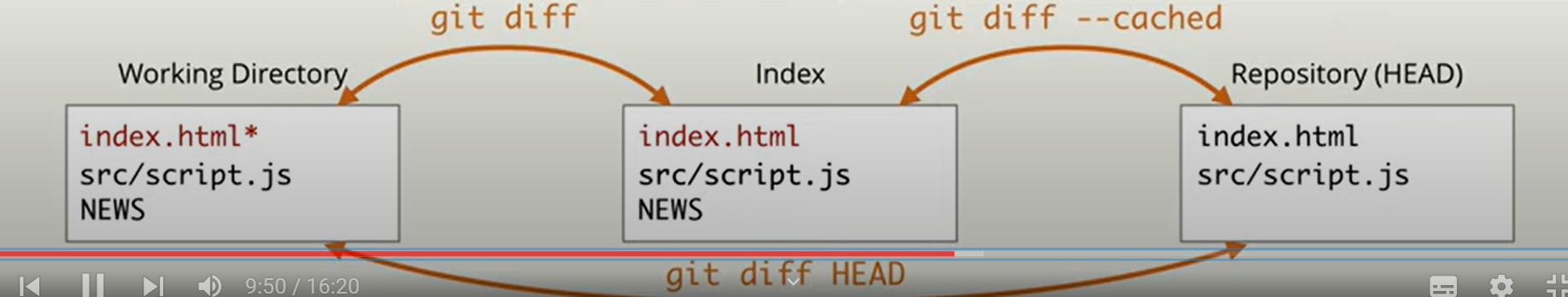
Ть что по сравнению с 1 коммитон находится в другом

Git diff ? ? (Что в 1 вопр посен по сравнен со 2-м) <=/> git diff master … feature(Что с fiture по сравнению c master)

Что я изменил по сравнению с коммитом: git diff HEAD

Что я изменил по сравнению с индексом: git diff

Что я изменил по сранению коммита с индексом: git diff cached (c head)



Git diff – значит путь пишем есл –

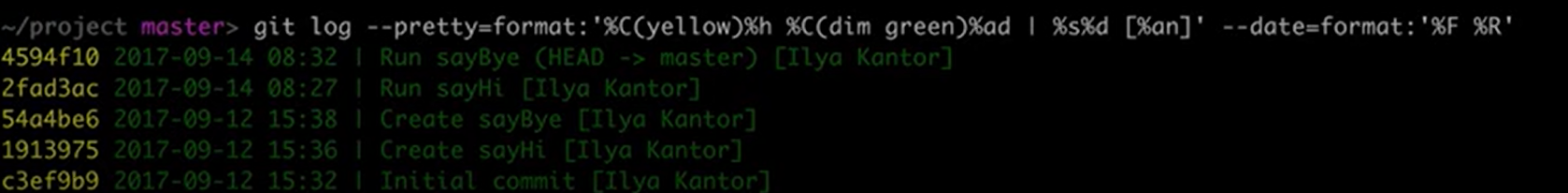
ПРОСМОТР ИСТОРИИ РАЗРАБОТКИ СВЕРХУ НОВ КОММИТЫ

Git log

Git log --oneline

**Сборка что выведет git log**: git log –pretty=format:’%h(сокр идент коммита) %cd(date) <<cr вместо cd короткая дата>> !(Верт черта) %s(заголовок) %d(ссылки) [%an](имя авт)’

Подробнее: git help log / git help config



Постоавить пост цвета:

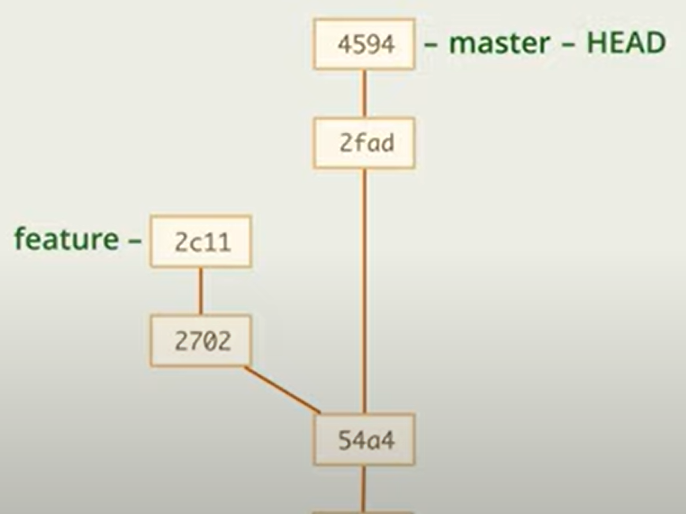




GIT LOG для нескольких Графов

Git log ? ? –graph

Git log –all –graph

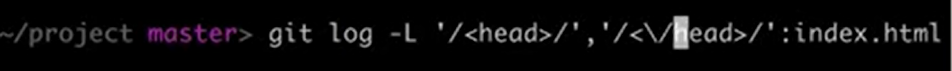
Git log master..fiture (Id и т.д fiture с момента отхождения от master ) 

Cравнить 2 ыетки без вывода хвоста: git log master…?

Git log где менялся файл git log ?.html И т.д

ПОИСК КОММИТОВ ПО СЛОВУ В ИХ ЗАГОДЛВКЕ git log –grep ‘? ’ -i(не обр внимание на регистр)

Сравнение измененного кода в разных коммитак по слову Git log -g ? -p

Сравнить файлы в разных коммитах по строкам или тегам 

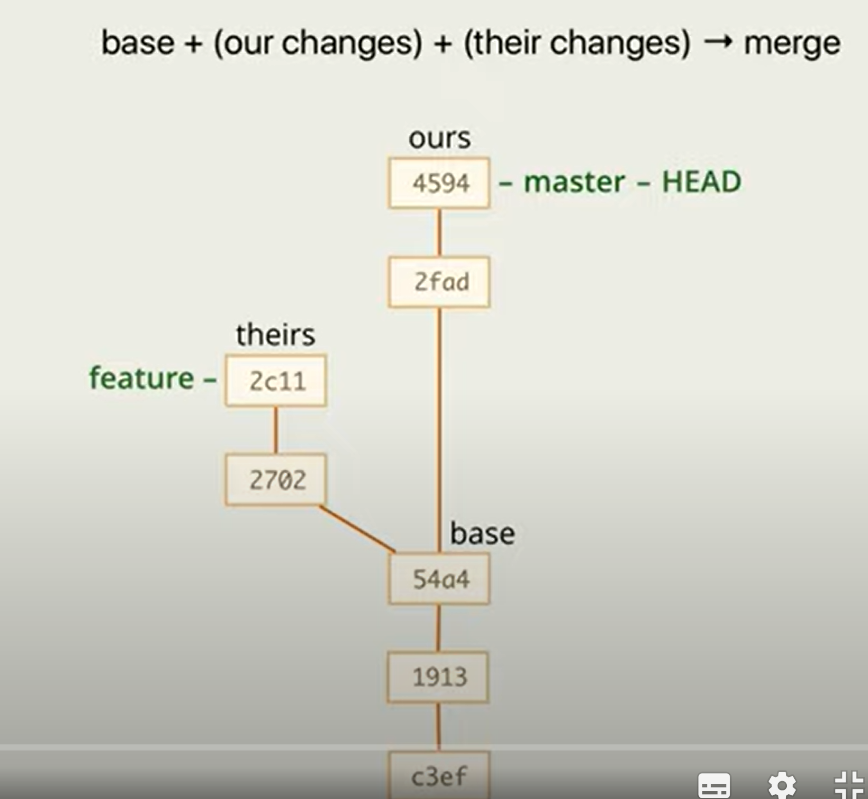
Закрыв слеш надо экранировать \

КТО НАПИСАЛ КОД:

Git blame ?d

СЛИЯНИЕ ВЕТОК:

Git merge

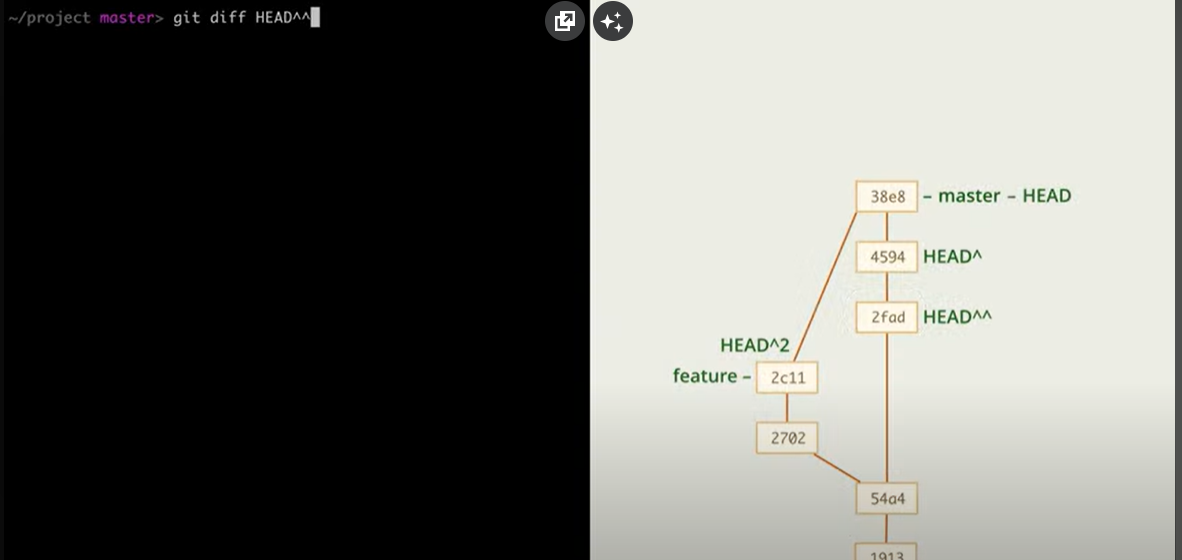


<https://www.youtube.com/watch?v=PXK1hIifpWU&list=PLDyvV36pndZFHXjXuwA_NywNrVQO0aQqb&index=34&t=4s>

Показывать в конфликте стар верс ф-ла новую и среднюю



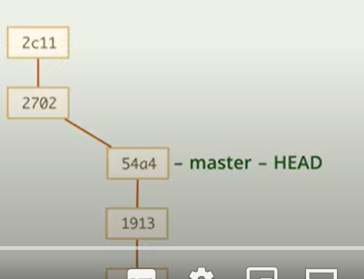
Просмотреть изменения по сравнению с пред верс git diff head^2^ (2 родитель 1родит)

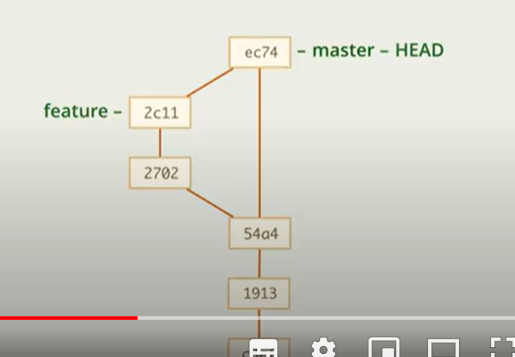


ОСТАНОВИТЬ ГИД ПЕРЕД КОММИТОМ ДЛЯ ИСПР ОШИБОК:

GIT MERGE ? –NO-COMMIT

ВСЕГДА ВМЕСТО ПЕРЕМОТКИ ДЕЛ КОММИТ СЛИЯНИЯ

 =>



КОПИРОВАНИЕ КОММИТОВ:

Git cherry-pick ? (b211 dwhw /wdww / master..fit)

При ошибках

 - верн все как было

- прод вып чери пиг



ПЕРЕНОС ВЕТОК:

<https://www.youtube.com/watch?v=jxwPgfmutjs&list=PLDyvV36pndZFHXjXuwA_NywNrVQO0aQqb&index=41>

Переходим на ветку и пишем git rebase ветка куда перенести

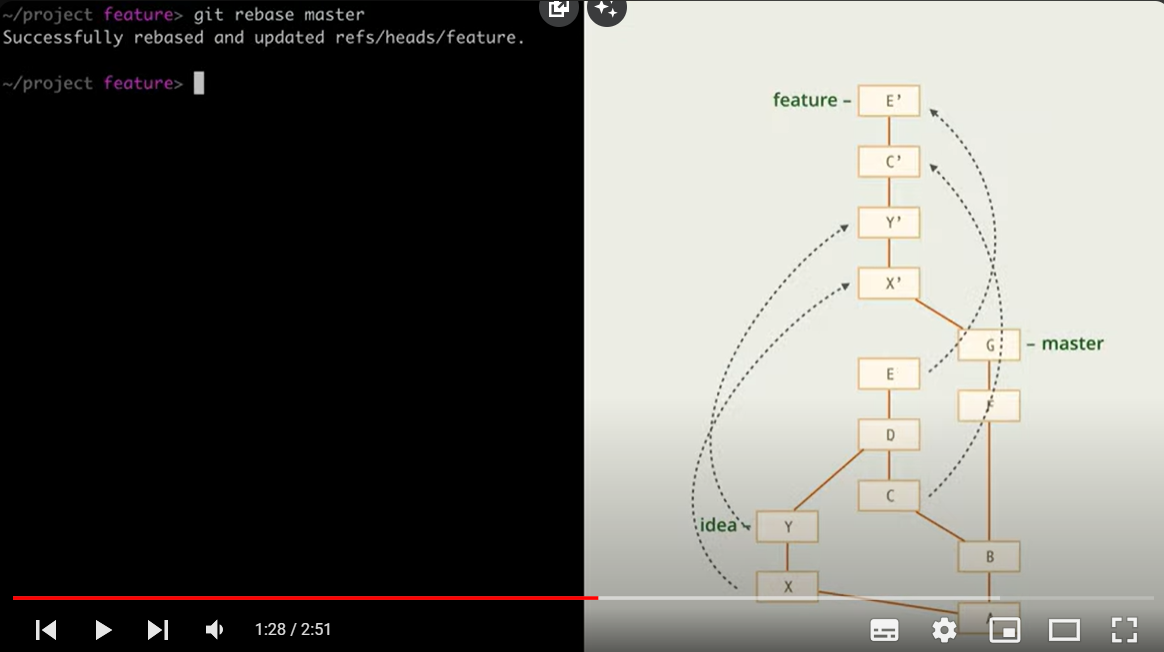
ТЕСТИРОВАНИЕ КОММИТА НА ОШИБКИ ПЕРЕМЕЩ ВЕТКИ ЗАРЕНЕЕ

<https://www.youtube.com/watch?v=dnT8YzpqEYA&list=PLDyvV36pndZFHXjXuwA_NywNrVQO0aQqb&index=43>

ПЕРЕНОС КОММИТОВ НА ВЕТКИ

Git rebase –-onto ? с какого момента(коммит)

ТЯЖЕЛЫЙ СЛУЧАЙ ПЕРЕНОСА ВЕТОК



ПЕРЕБАЗИРОВАНИЕ ИЛИ ИЗМЕНЕНИЯ КОМИТОВ

Git -rebase -I master

<https://www.youtube.com/watch?v=4K9X1Aa1nvA&list=PLDyvV36pndZFHXjXuwA_NywNrVQO0aQqb&index=46&pp=iAQB>

Отмена коммитов у др людей которые уже сделали изменения моих коммитов у себя

<https://www.youtube.com/watch?v=aRbOsagYs8w&list=PLDyvV36pndZFHXjXuwA_NywNrVQO0aQqb&index=49>

<https://www.youtube.com/watch?v=3HDSLArx3qw&list=PLDyvV36pndZFHXjXuwA_NywNrVQO0aQqb&index=50>