Віддалені репозиторії

HTTPS://GIT-SCM.COM/

ПІДРУЧНИК

HANDBOOK

Віддалені репозиторії

Щоб побачити, які віддалені сервера ви налаштували, ви можете виконати команду

```
$ git remote
```

Якщо ви отримали своє сховище клонуванням, то ви побачите origin — таке ім'я Git дає серверу, з якого ви зробили клон. Якщо додати до наведеної вище команди параметр –v, то будуть виведені посилання на віддалені сервери, які Git зберігає та використовує при читанні та записі до цього сховища

```
$ git remote -v
```

Якщо у вас більш ніж одне віддалене сховище, ця команда описує їх усі.

Детальніше...

Додавання віддалених сховищ

Команда clone неявно додає віддалене сховище origin. Щоб додати нове віддалене Git сховище під заданим ім’ям, на яке ви можете легко посилатись, виконайте:

```
$ git remote add <ім’я> <посилання>
```

наприклад

```
$ git remote add second https://github.com/krenevych/oop.git
```

Детальніше...

Перейменування віддалених сховищ

Щоб перейменувати віддалене сховище треба скористатися командою

```
$ git remote rename <ім’я> <нове ім’я>
```

Наприклад, команда

```
$ git remote rename origin new_origin
```

Перейменує віддалений репозиторій `origin` на `new_origin`.

Видалення віддалених сховищ

Щоб видалити віддалене сховище треба скористатися командою

```
$ git remote remove <ім’я>
```

Наприклад, команда

```
$ git remote remove origin
```

видалить віддалений репозиторій `origin`.

Віддалені гілки

Віддалені посилання (гілки) — це посилання (вказівники) у ваших віддалених сховищах.
Локально ці вказівники неможливо змінити, але їх змінює Git, коли ви виконуєте мережеві операції, щоб вони точно відповідали стану віддаленого сховища
Щоб вивести список віддалених гілок треба виконати команду

```
$ git remote show <репозиторій>
```

або

```
$ git ls-remote <репозиторій>
```

де `<репозиторій>` ім'я віддаленого репозиторія. Наприклад для стандартного репозиторія буде

```
$ git remote show origin
```

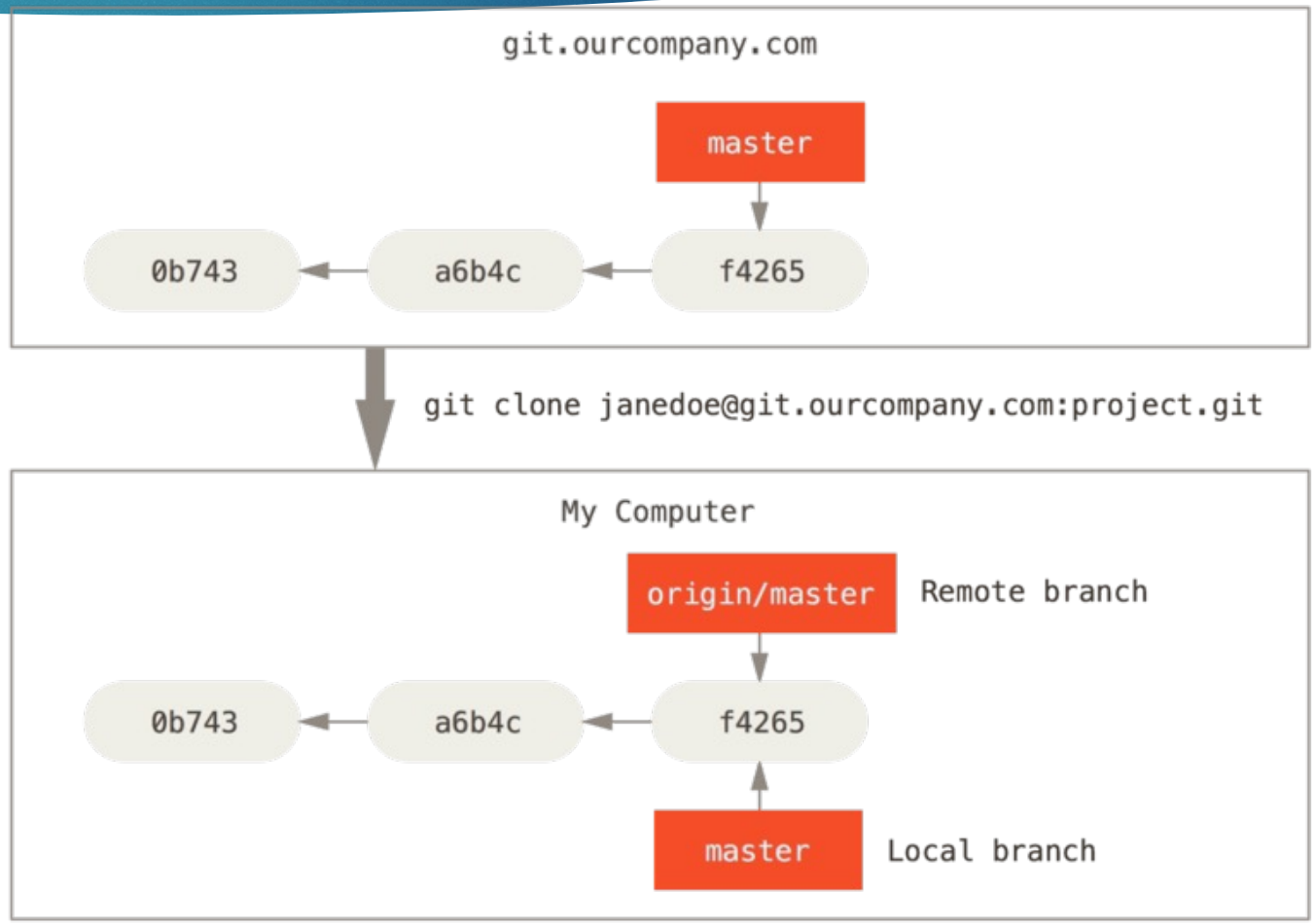
Віддалені гілки мають такий запис: <віддалене сховище>/<гілка>. Наприклад, гілка master з віддаленого сховища origin буде мати вигляд для нашого локального репозиторію

```
origin/master
```

Детальніше...

Віддалені гілки

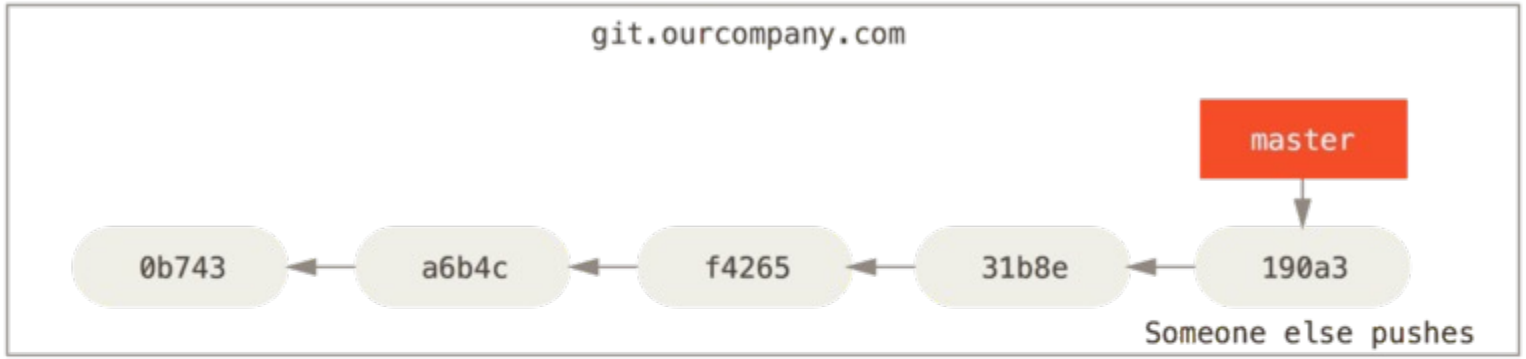
Коли ви склонуєте з віддаленого сервера репозиторій, команда clone автоматично іменує його origin, стягує всі дані, створює вказівник на те місце, де зараз знаходиться master і локально іменує це посилання origin/master (якщо основна гілка віддаленого репозиторія називається master)

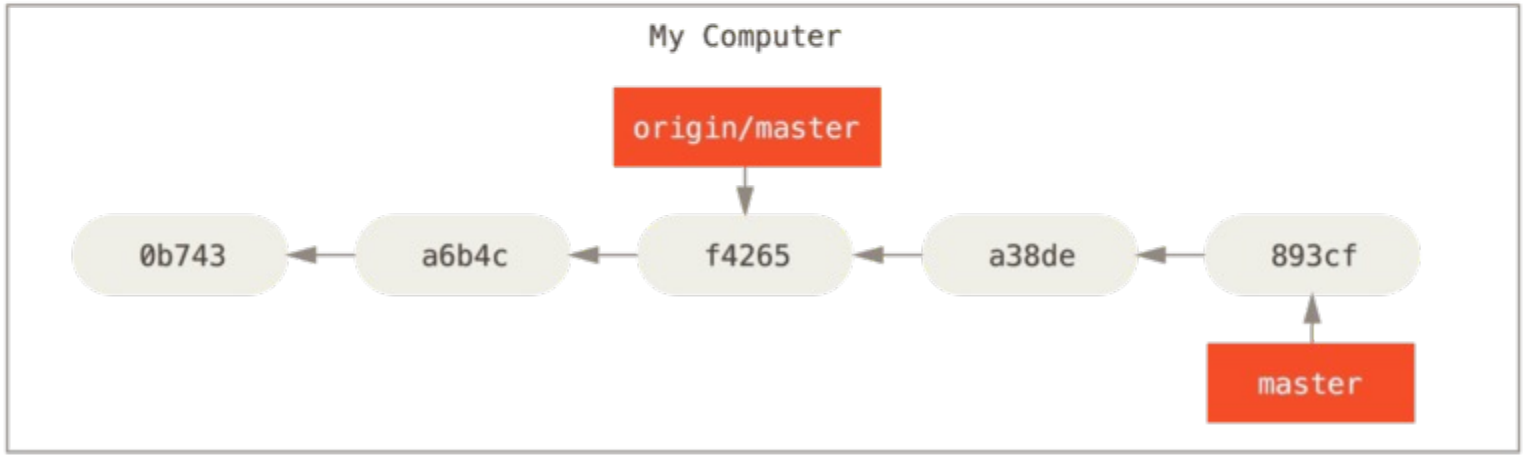


Детальніше...

Віддалені гілки

Якщо ви виконали якусь роботу на локальній гілці master, і водночас, хтось виклав зміни на git.ourcompany.com в master, тоді ваші історії прогресують по-різному. Доки ви не синхронізуетесь з сервером, вказівник origin/master не буде рухатись





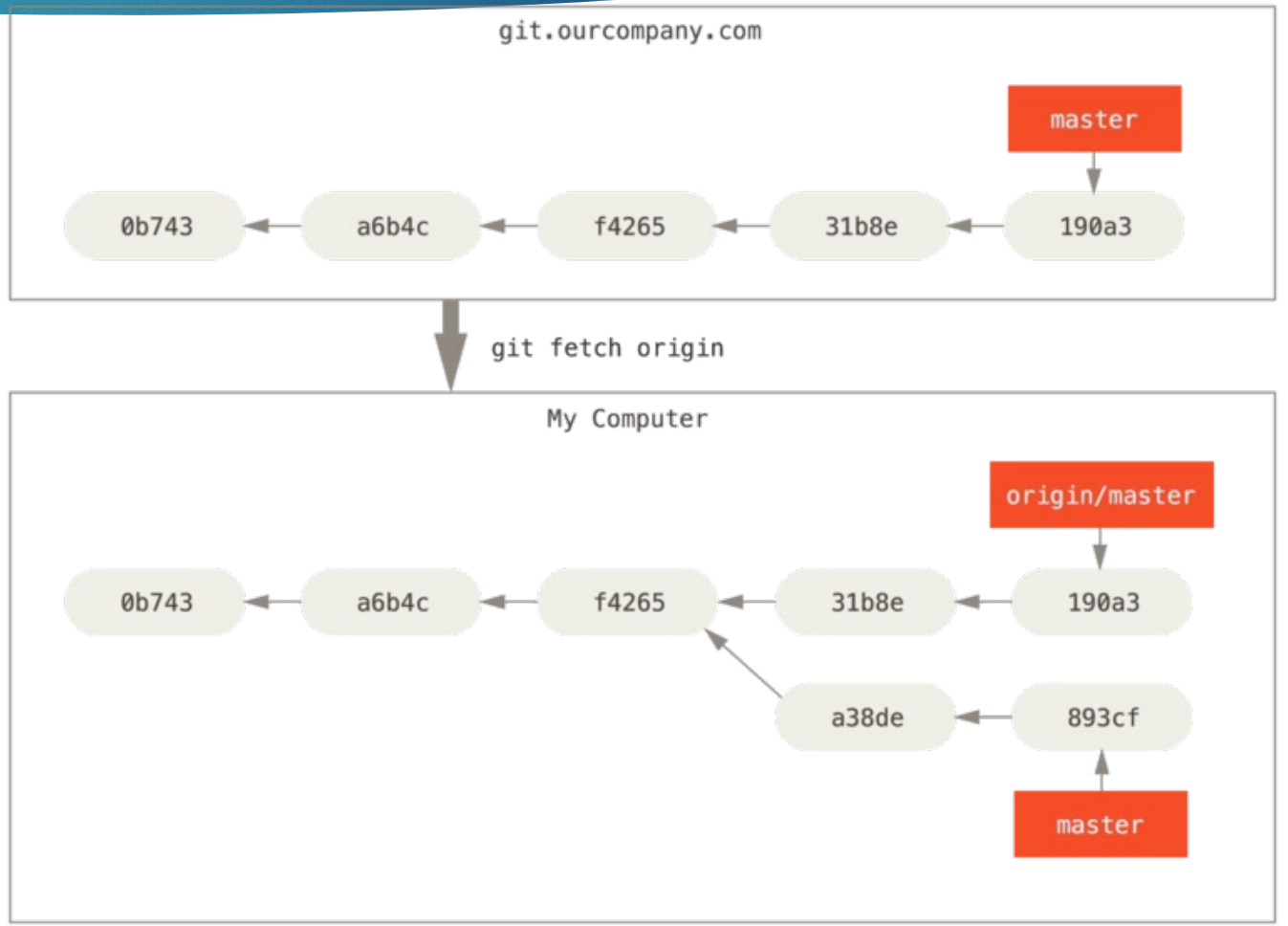
Детальніше...

Синхронізація з віддаленим репозиторієм

Щоб відновити синхронність, виконайте команду

```
$ git fetch origin
```

Ця команда шукає який сервер відповідає імені origin, отримує дані, яких ви ще не маєте і оновлює вашу локальну базу даних, переміщаючи вказівник origin/master на нову, більш актуальну, позицію.



Детальніше...

Особливості команди fetch

Під час виклику, команда fetch не створює жодних локальних гілок та не проводить злиття віддалених та локальних гілок – вона проводить оновлення стану віддалено-відслідковуваних гілок віддаленого сервера не проводячи жодних маніпуляцій з локальними гілками.

Щоб злити отримані зміни в вашу поточну робочу гілку виконайте

```
$ git merge <віддалена гілка>
```

наприклад

```
$ git merge origin/main
```

вмержить зміни з віддаленої гілки `origin/main` у поточну гілку.

Стягування з віддаленого репозиторія

Команда

```
$ git pull <репозиторій> <гілка>
```

де `<репозиторій>` - віддалений репозиторій, `<гілка>` - гілка віддаленого репозиторія здійснює синхронізацію з віддаленим репозиторієм, та об'єднує у поточній гілці зміни, отримані з віддаленої гілки. Такими чином, для прикладу, фактично команда

```
$ git pull origin main
```

заміняє виклик двох команд

```
$ git fetch origin
$ git merge origin/main
```

Надсилання на віддалений репозиторій

Ваші локальні гілки не синхронізуються з віддаленими автоматично — потрібно явно надсилати гілки, якими хочете поділитися

```
$ git push <репозиторій> <гілка>
```

тут `<репозиторій>` - віддалений репозиторій і `<гілка>` - гілка яка надсилається на віддалений репозиторій, при умові, що назва локальної та віддаленої гілки однакові, наприклад

```
$ git push origin main
```

надішле зміни з локальної гілки main на гілку `main` віддаленого репозиторія origin.

Детальніше...

Надсилання на віддалений репозиторій

Якщо локальна гілка, та гілка на віддаленому репозиторії називаються по різному, то команда буде виглядати таким чином

```
$ git push <репозиторій> <локальна гілка>:<гілка на віддаленому репозиторії>
```

Наприклад, команда

```
$ git push origin main:develop
```

надішле зміни з локальної гілки main на гілку develop віддаленого репозиторія origin.

Описаний тут підхід використовують, коли внесення зміни у гілку віддаленого репозиторія заборонений напряму, наприклад, для реалізації процедури рецензування коду відповідно до політик компанії. На GitHub це реалізується через підхід, що називається Pull request.

Детальніше...

Видаляння віддалених гілок

Для видалення віддаленої гілки користуйтеся командою

```
$ git push <репозиторій> --delete <гілка>
```

тут <репозиторій> - віддалений репозиторій і <гілка> - гілка на віддаленому репозиторії, наприклад

```
$ git push origin --delete develop
```

Відслідковувані гілки

Відслідковувані гілки — це локальні гілки, що мають безпосередній зв’язок з віддаленою гілкою. Якщо ви знаходитеся на відслідковуваній гілці і потягнете зміни, виконуючи команду

```
$ git pull
```

Git знатиме з якого сервера брати та з якої гілки зливати зміни. Аналогічно для команди

```
$ git push
```

Git знатиме на який віддалений сервер та на яку його віддалену гілку зливати зміни.

Така відслідковувана локальна гілка називається **tracking branch**. А віддалена гілка за якою вона стежить називається **upstream branch**.

Детальніше...

Перегляд відслідковуваних гілок

Коли ви клонуєте репозиторій, Git автоматично створює гілку main, яка слідкує за віддаленою гілкою origin/main. Також, операції переключення на локальну гілку з віддаленої створює такий зв’язок.
Команда

```
$ git branch -vv
```

дозволяє дізнатися, які у вас налаштовані відслідковувані гілки.

Детальніше...

Налаштування відслідковування

Щоб налаштувати відслідковувані гілки явно можна скористатися командою

```
$ git branch –u <віддалена гілка>
```

де `<віддалена гілка>` - гілка на віддаленому репозиторії. Наприклад команда

```
$ git branch –u origin/develop
```

встановлює зв'язок між поточною локальною гілкою та гілкою `develop` віддаленого репозиторія `origin`.
Також, такий зв'язок можна встановити при засиланні змін на відалений сервер командою push з параметром –u

```
$ git push -u origin main
```

встановлює зв'язок між поточною локальною гілкою та гілкою `main` віддаленого репозиторія `origin`.

Детальніше...