VERSION CONTROL WITH GIT

GIT IS YOUR FRIEND, NOT YOUR FOE.

WHAT IS VERSION CONTROL

- A system that track your changes
- Allows you to go back to a previous state or version
- Collaboration between developers/team
- Information on who changed what and when

HISTORICAL INFORMATION

Created by Linus Torvald in 2005 when is was doing Linux development

THE "REPO"

A collection of all the files (and their history)

The repo exists on your machine (and a server – like github)

Cloning from a git repo make a copy of that project on your computer

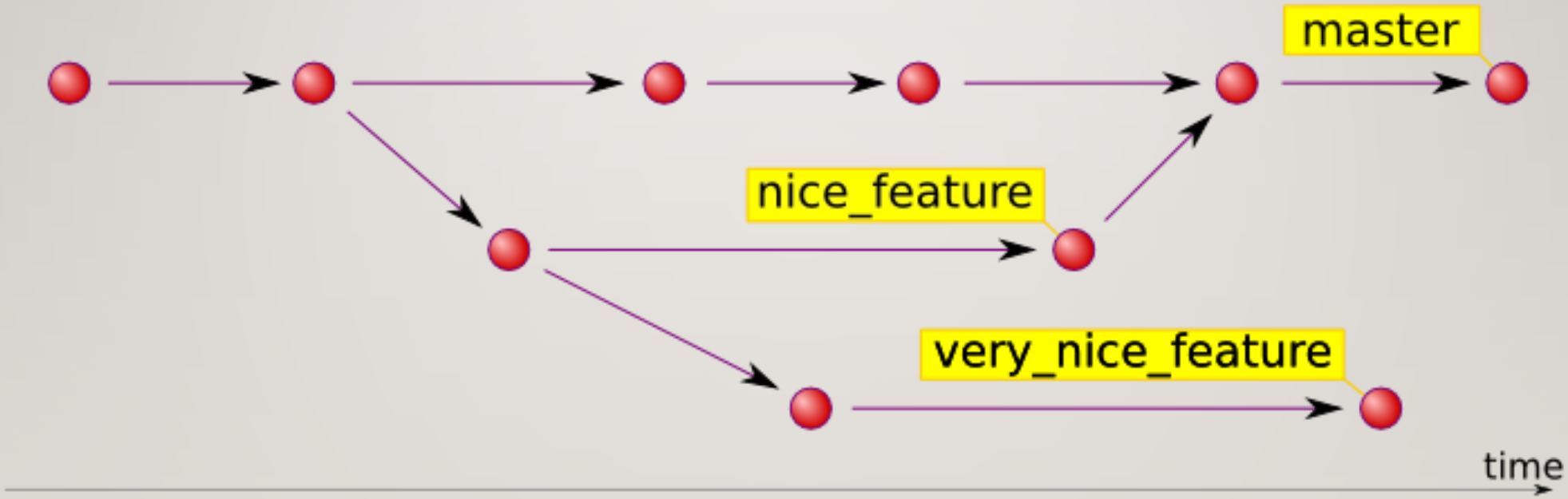
BRANCHES

All commits exist on a branch

There can be many (many!) branches

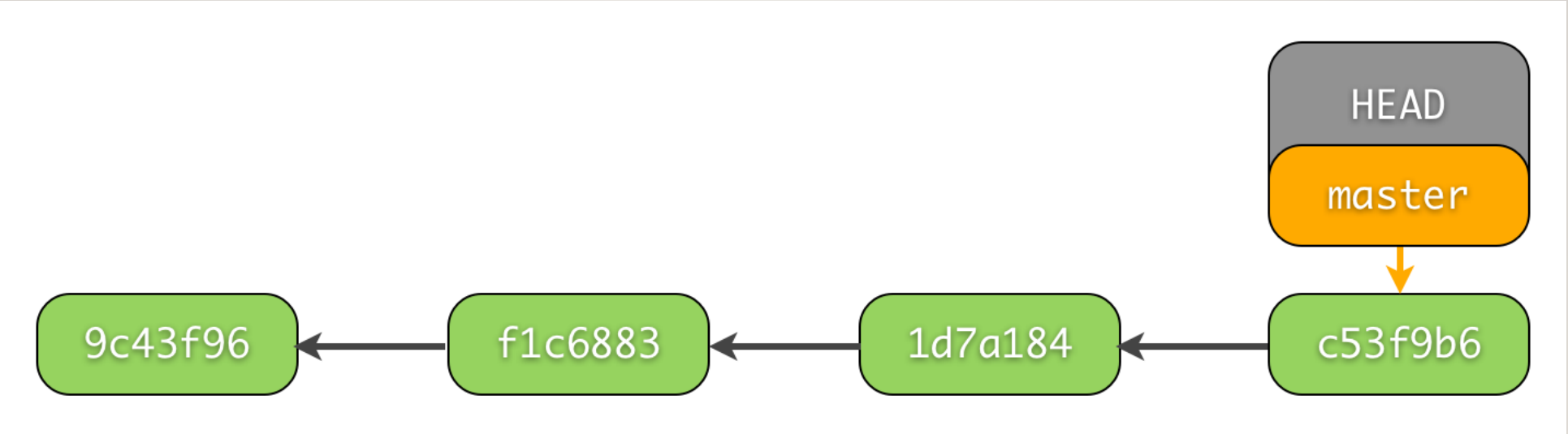
The main branch is the “master”

BRANCH EXAMPLE



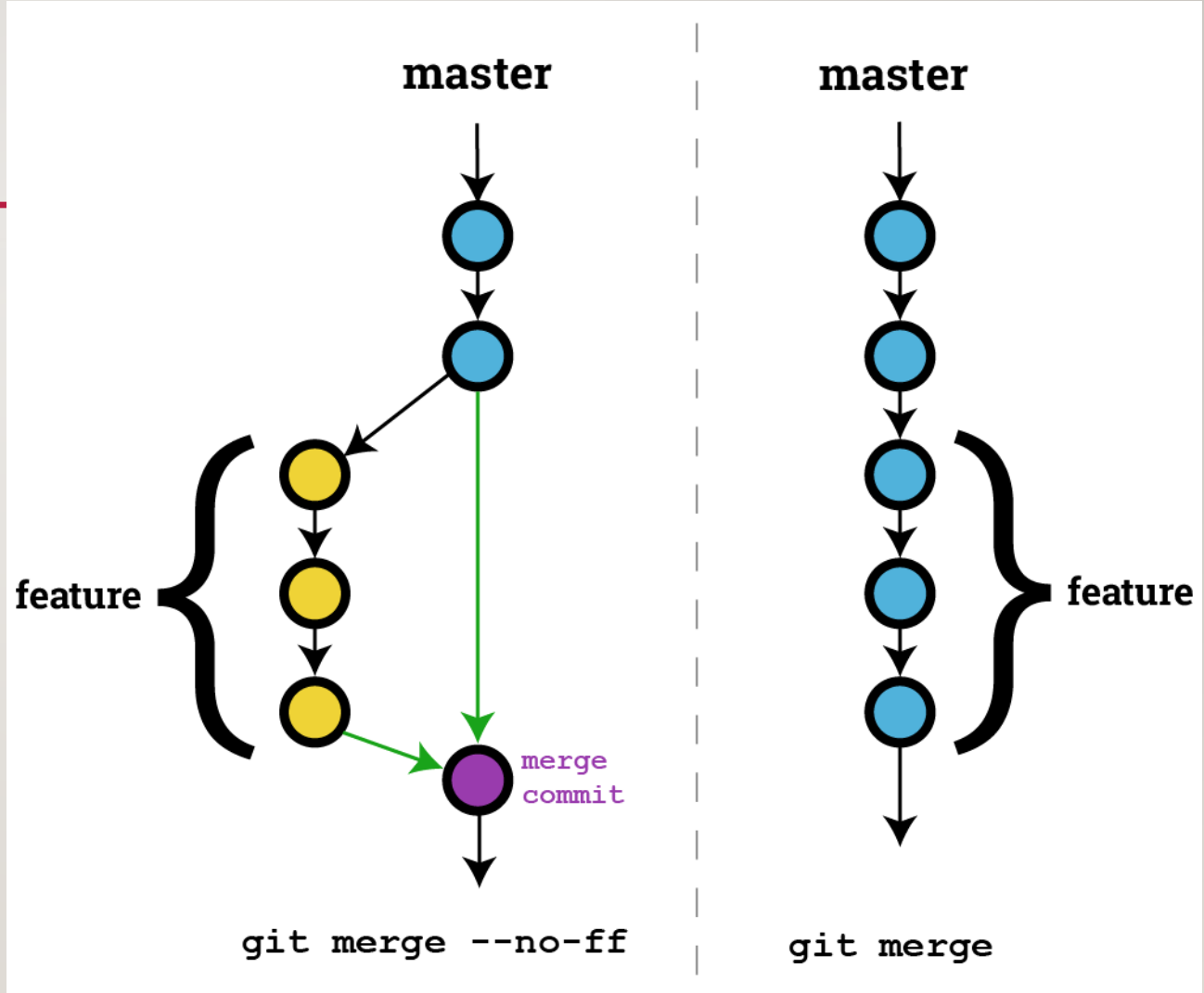
WHAT IS "HEAD"

A reference to the most recent commit.



WHAT IS "MERGING"

Putting your changes into a branch



INITIALIZING THE PROJECT

```
26      * @Template()
27      */
28     public function helloAction($name)
29     {
30         return array('name' => $name);
31     }
32
33     /**
34      * @Route("/contact", name="_demo_contact")
35      * @Template()
36      */
37     public function contactAction()
38     {
39         $form = $this->get('form.factory')->create
40
41         $request = $this->get('request');
42         if ('POST' == $request->getMethod()) {
```

Create your project with your source, configuration files,
etc…

INITIALIZING GIT FOR YOUR PROJECT

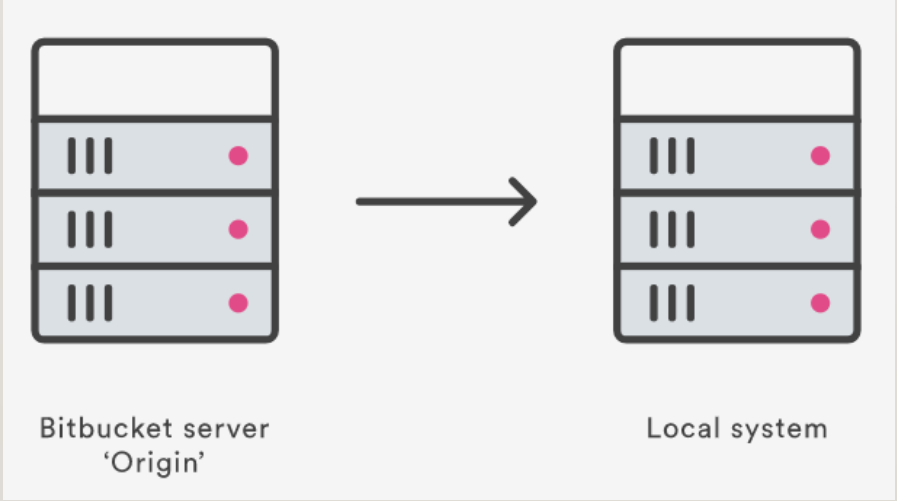
C:/proj_dir/git **init or clone**

```
These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
   clone      Clone a repository into a new directory
   init       Create an empty Git repository or reinitialize an existing one
```

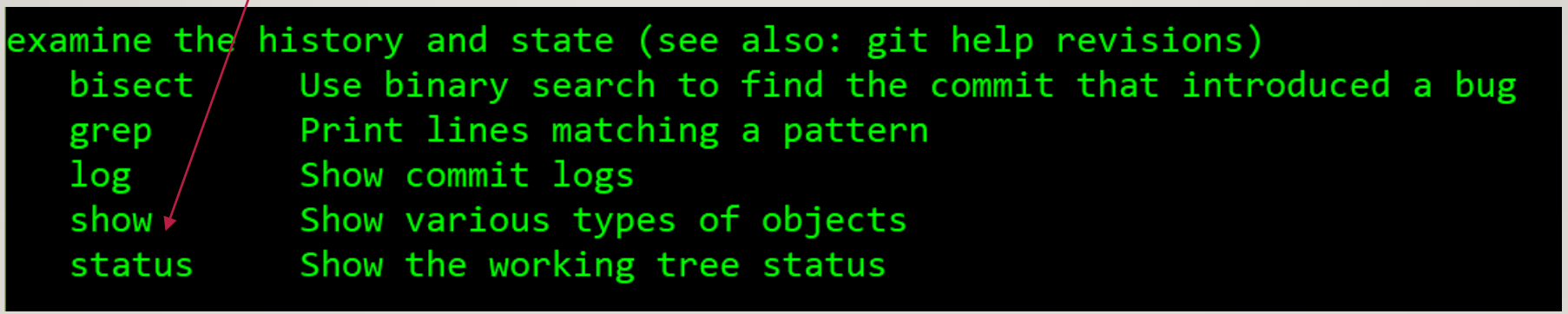
CLONING A REPOSITORY (“REPO”)

Git refers to copying a repository as "**cloning**" it. When you clone a repository, you create a connection between the server (which Git knows as **origin**) and your **local** system.



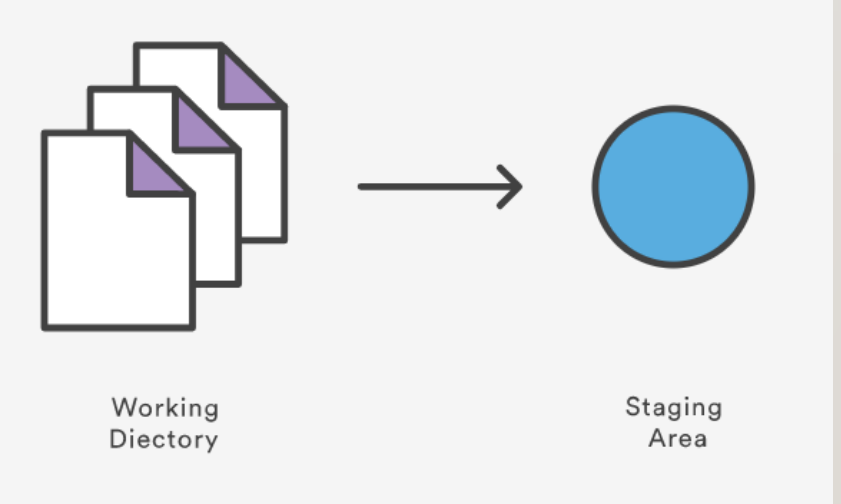
GIT STATUS

Git status indicates what has changed in your local repository against the origin



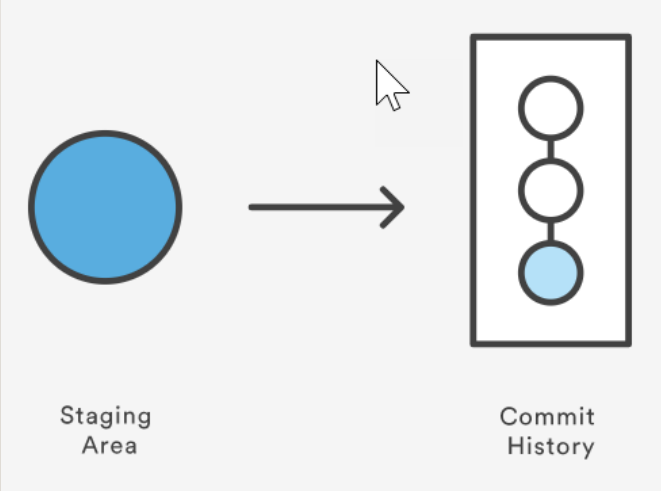
GIT ADD

Git add moves changes from your **local** repo to Git's **staging** area. The stage is just before **committing** to the **origin**.



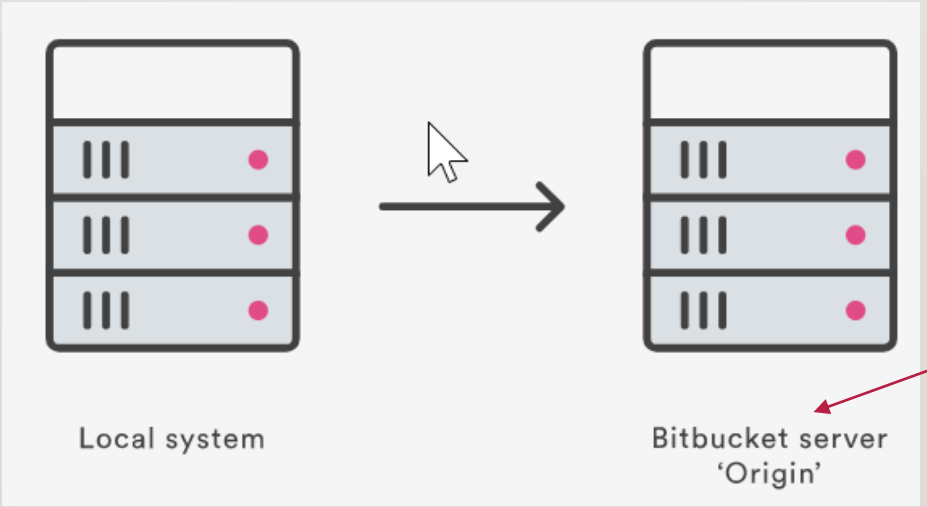
COMMIT (LOCAL)

The git commit takes the staged snapshot and commits it to the project history. Combined with git add, this process defines the basic workflow.



PUSH

You manage connections with other repositories and publish local history by "pushing" branches to other repositories.



Bitbucket, for example (or github)

PULL

When you do a pull command, you are asking for any changes that other team members have pushed to the origin.

Bitbucket.com (or github.com)