

# Git for sysadmins

# Git for sysadmins

-or-

I was an SCM engineer for a year  
and lived to tell about it

This talk will  
teach you to use git

Git tutorials suck

Why use git, then?

- Configuration basics
  - Frontend basics
  - Backend basics

Install & configure git

Install & configure git  
<http://git-scm.com>



Install & configure git  
<http://git-scm.com>  
Add name & email

# Install & configure git

<http://git-scm.com>

## Add name & email

```
git config --global user.name "Your Name"  
git config --global user.email you@foo.com
```

# Install & configure git

<http://git-scm.com>

## Set CLI colors

```
git config --global user.name "Your Name"  
git config --global user.email you@foo.com  
git config --global color.ui auto
```

# Git frontend basics

# Git frontend basics

workdir

# Git frontend basics

workdir

`'git init'`

# Git frontend basics

local repo — workdir

# Git frontend basics

local repo – workdir

```
`git add $PATH`
```



# Git frontend basics

repo — index ← workdir

# Git frontend basics

repo — index ← workdir

`'git commit'`

# Git frontend basics

repo ← index ← workdir

# Git frontend basics

repo ← index ← workdir

Lather, rinse, repeat.

# Git frontend basics

repo ← index ← workdir

Lather, rinse, repeat.  
Change, add, commit.

# Git frontend basics

repo ← index ← workdir

Lather, rinse, repeat.

Change, add, commit.

Commit OFTEN.

# Git frontend basics

DEMO

# Git backend basics



# Git backend basics

blob

tree

commit

tag

# Git backend basics

    /b1  
t1-<  
    \b2

# Git backend basics

        /b1  
c1-t1-<  
        \b2

# Git backend basics

```
          /b1  
c1-t1-<  
          \b2  
          /  
t2-<  
          \b1'
```

# Git backend basics

```
          /b1  
c1-t1-<  
          \b2  
          /  
c2-t2-<  
          \b1'
```

# Git backend basics

```
                /b1
c1-t1-<
↑           \b2
|           /
c2-t2-<
                \b1'
```

# Git backend basics

# Git backend basics

c1



c2



# Git backend basics

c1

↑

c2

(master\*)

# Git backend basics

c1

↑

c2

(master\*)

```
`git branch feature1`
```

# Git backend basics

c1

↑

c2

(master\*, feature1)

# Git backend basics

c1

↑

c2

(master\*, feature1)

```
`git checkout feature1`
```

# Git backend basics

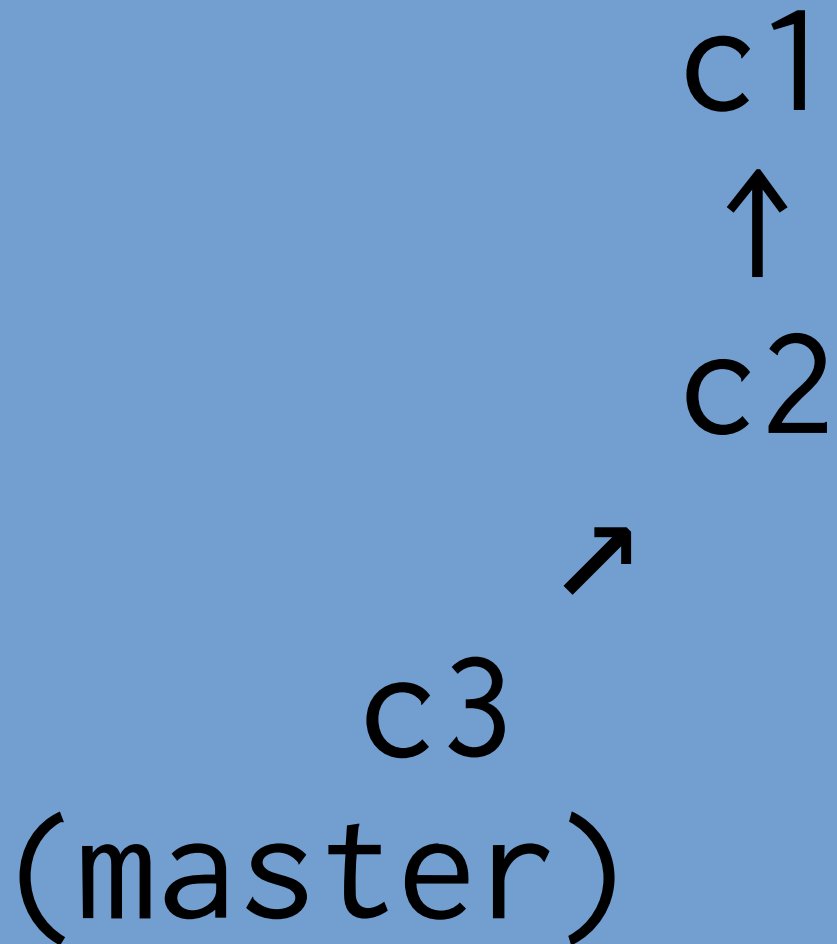
c1

↑

c2

(master, feature1\*)

# Git backend basics



# Git backend basics

c1



c2



c3

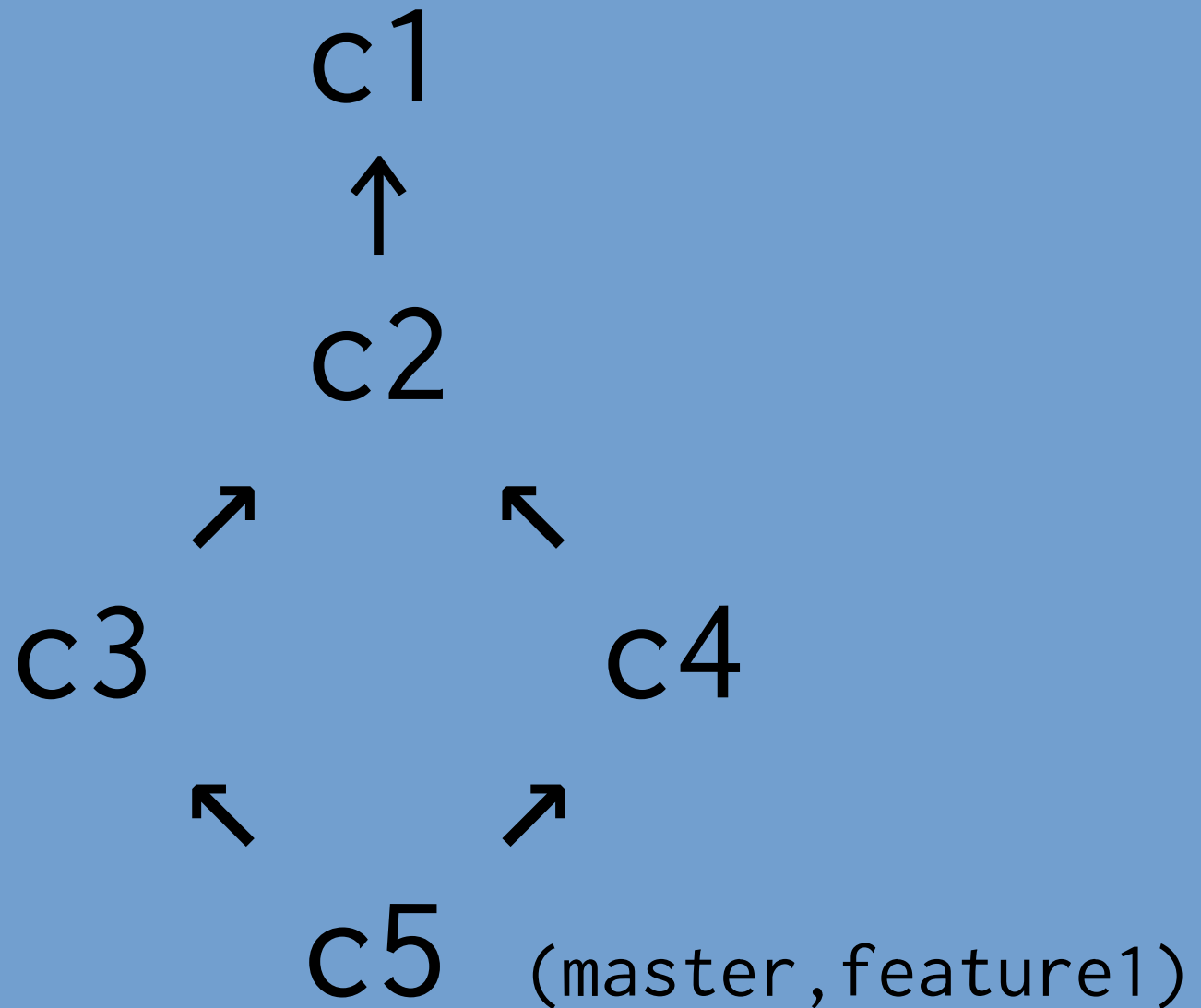
c4

(master\*)

(feature1)

`'git merge feature1'`

# Git backend basics





# Resources

Pro Git (free)

# Resources

Pro Git (free)

Version Control with Git  
(O'Reilly)

# Resources

Pro Git (free)

Version Control with Git  
(O'Reilly)

Git Intro: Randal Schwartz  
(Vimeo)

# Q&A

- Branching
- Collaboration
- SVN migration