Git and GitHub

ihower@gmail.com

2011/3

部分插圖取自 Pro Git 一書

我是誰?

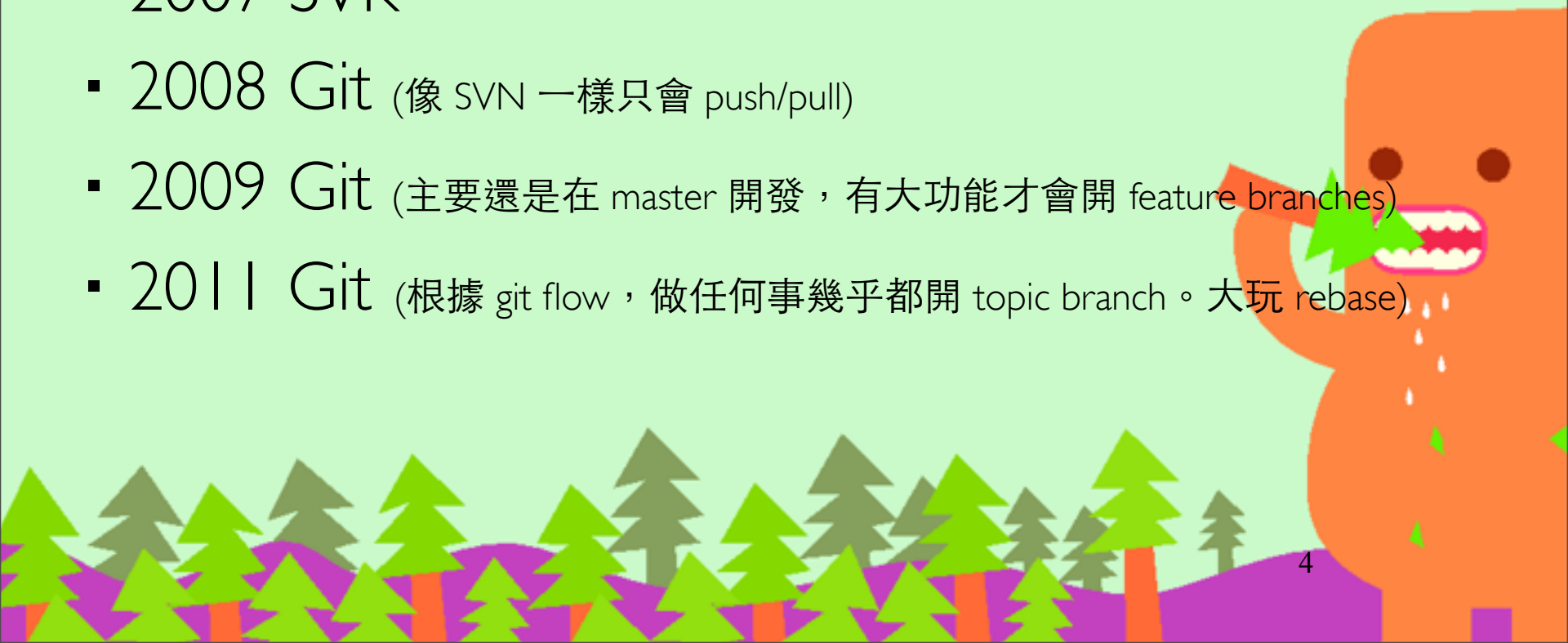
- 張文鈿 a.k.a. ihower
  - http://ihower.tw
  - http://twitter.com/ihower
- Ruby on Rails Developer since 2006
- The organizer of Ruby Taiwan Community
  - http://ruby.tw

Agenda

- 1. 什麼是 DVCS
- 2. 什麼是 Git
- 3. Git 指令基礎
- 4. Github 簡介
- 5. Git 開發模式及流程

我的版本控制之路

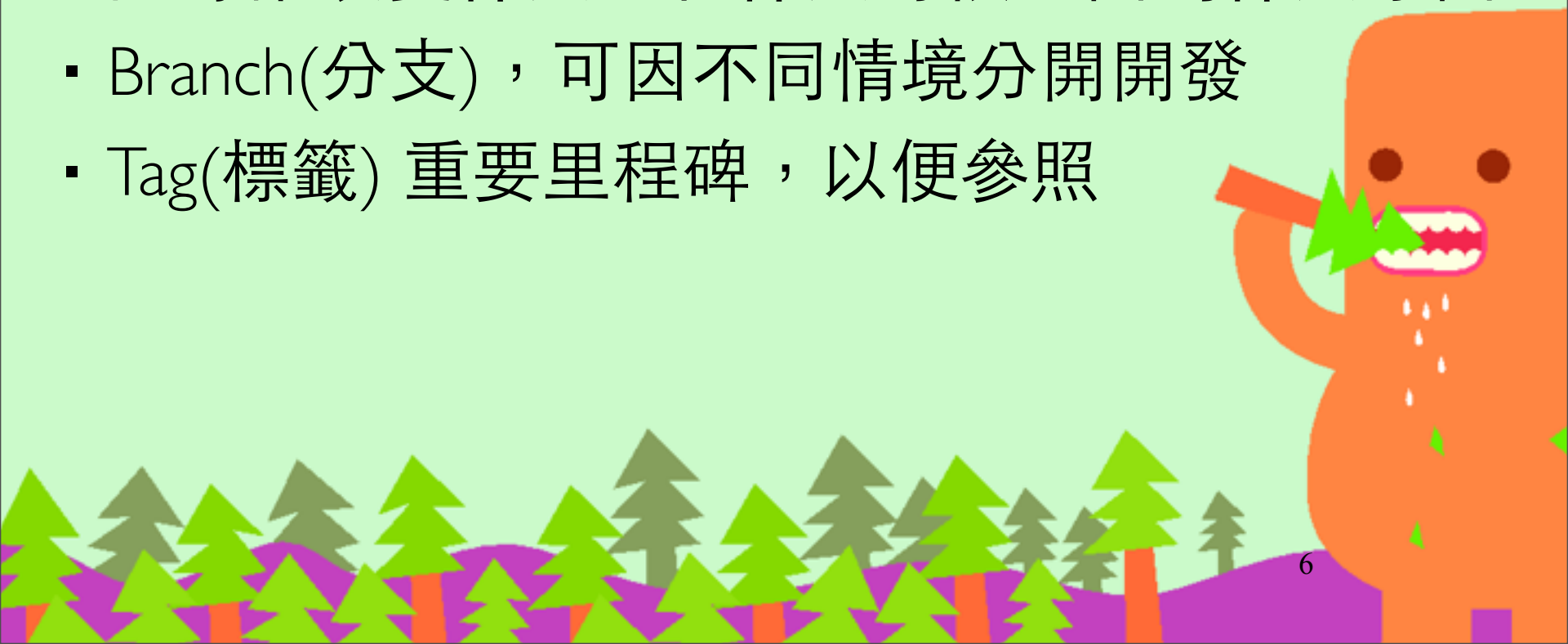
- 2002 目錄管理法 (定時 copy 備份一次)
- 2005 SubVersion
- 2007 SVK
- 2008 Git (像 SVN 一樣只會 push/pull)
- 2009 Git (主要還是在 master 開發，有大功能才會開 feature branches)
- 2011 Git (根據 git flow，做任何事幾乎都開 topic branch。大玩 rebase)



1. 什麼是 DVCS ?

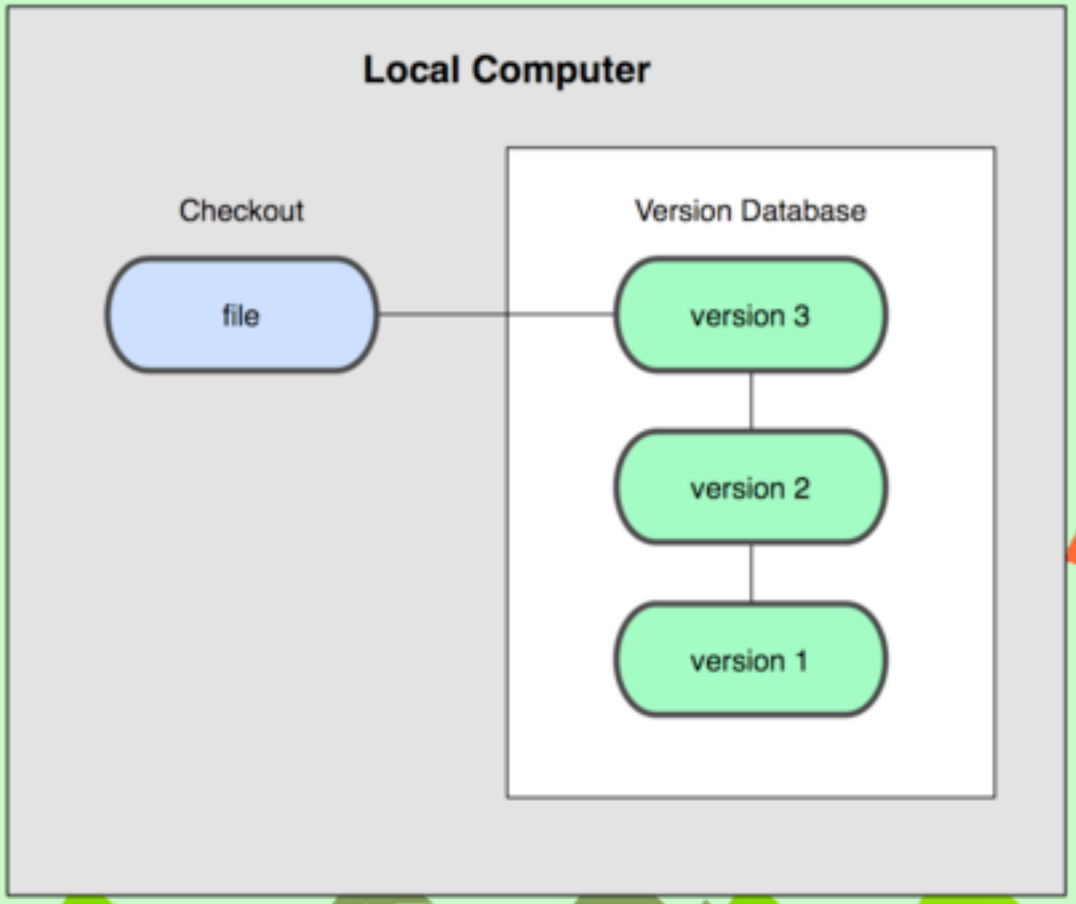
Version Control System

- 建立 repository，保存備份程式碼
- 方便散佈程式給團隊，有效率協同開發
- 記錄誰改變什麼、在什麼時候、因為什麼原因
- Branch(分支)，可因不同情境分開開發
- Tag(標籤) 重要里程碑，以便參照



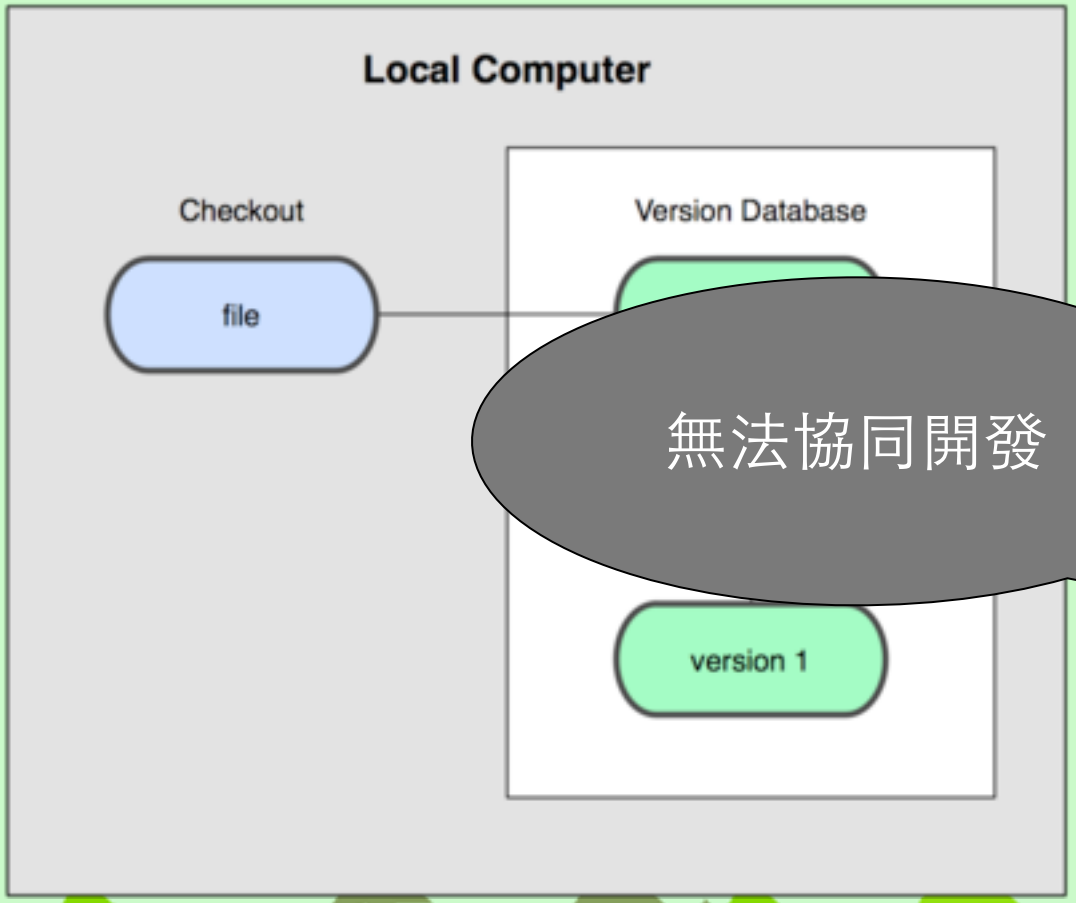
Local VCS

copy paste 資料夾管理, rcs



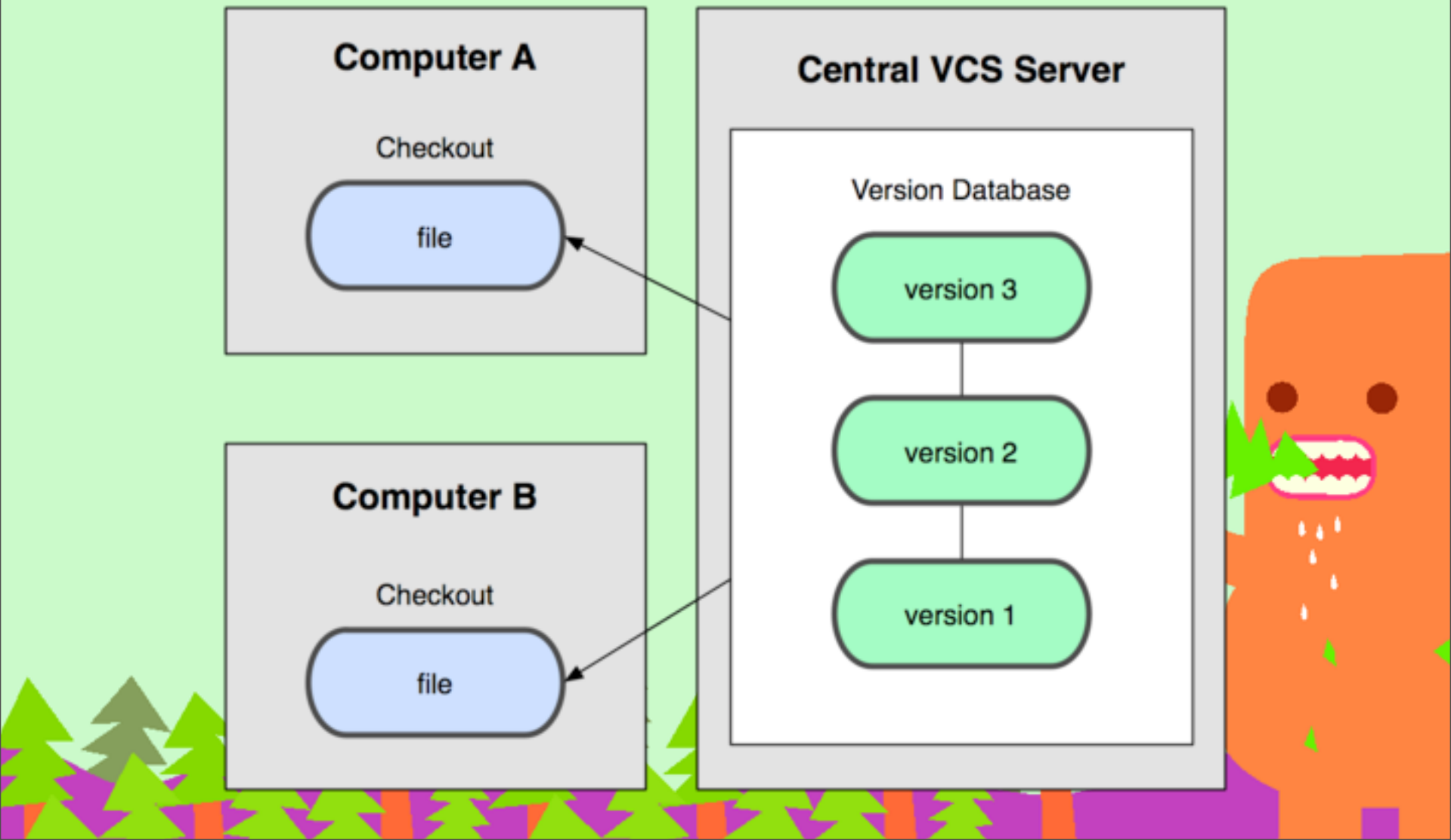
Local VCS

copy paste 資料夾管理, rcs



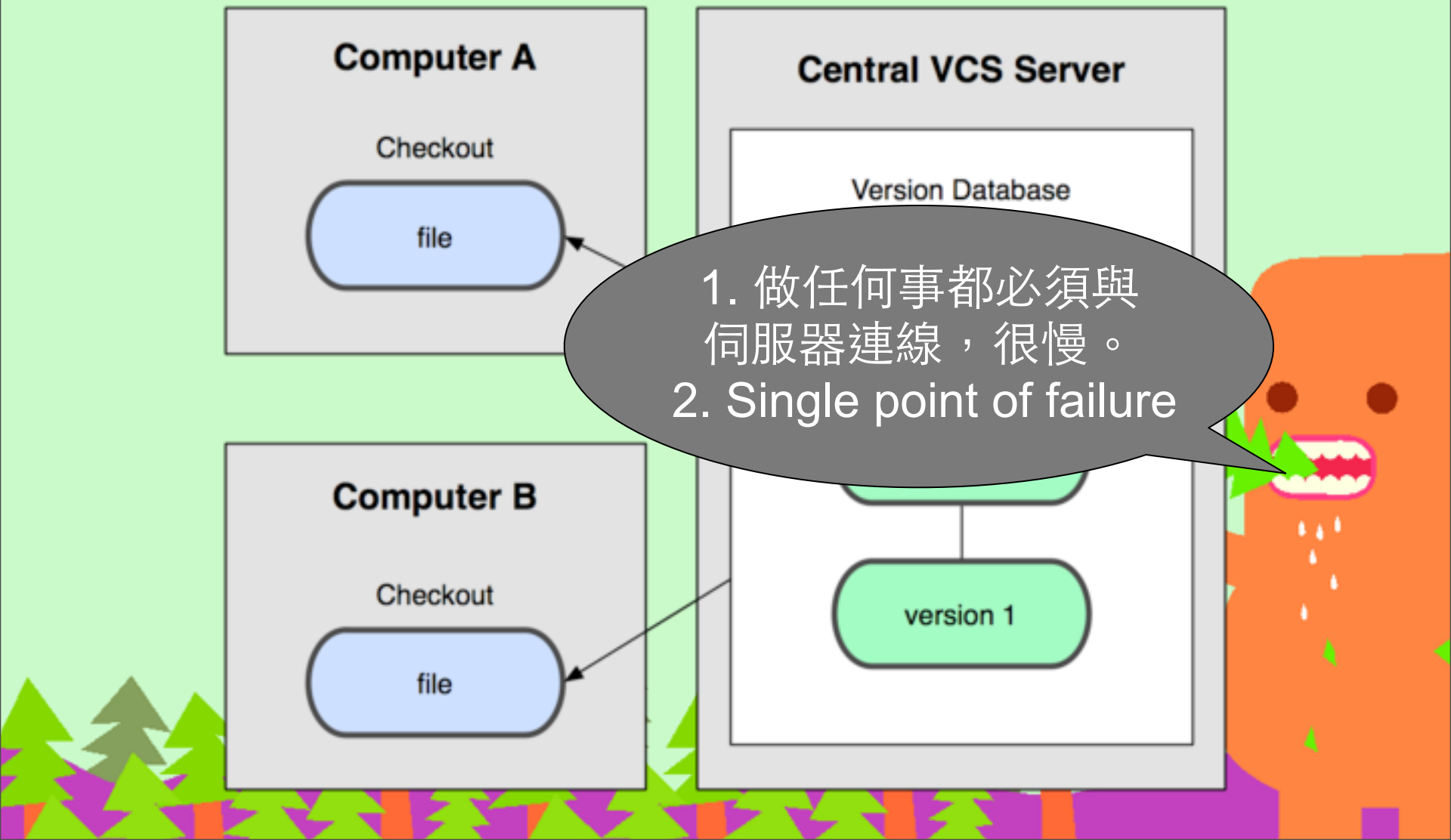
Centralized VCS

CVS, Subversion, Perforce



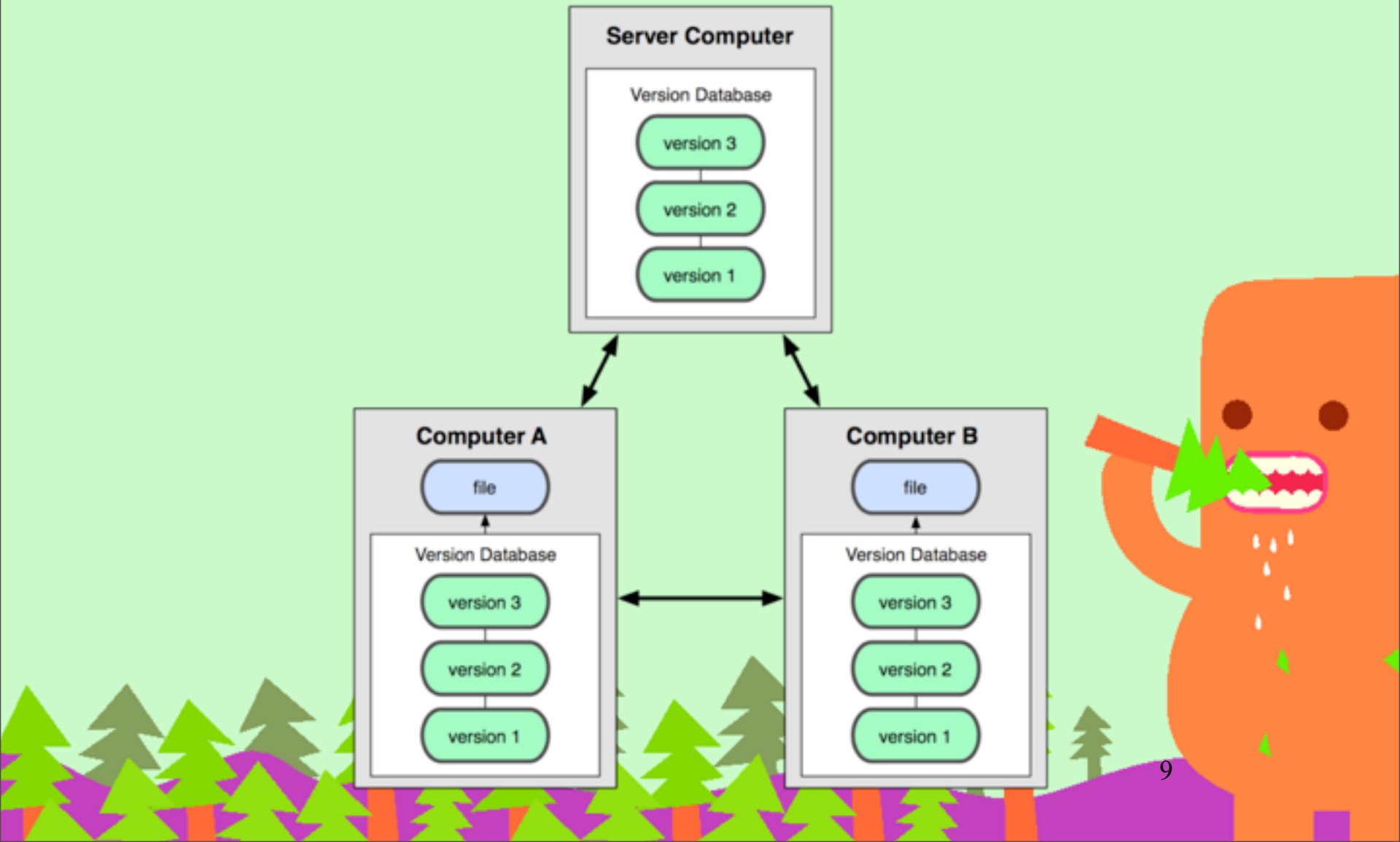
Centralized VCS

CVS, Subversion, Perforce



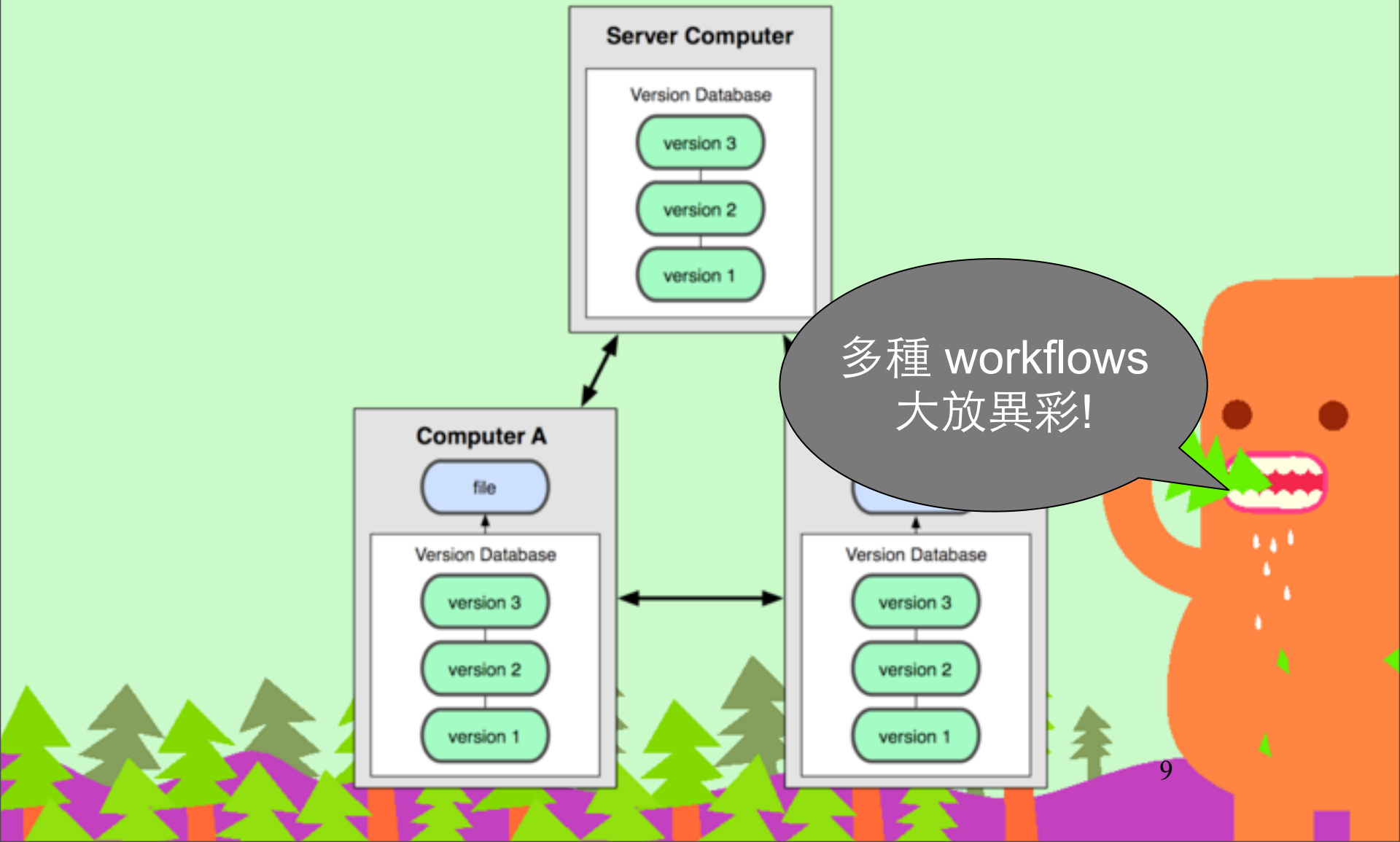
Distributed VCS

Git, Mercurial(Hg), Bazaar



Distributed VCS

Git, Mercurial(Hg), Bazaar



Local development

- 集中式的 CSV 系統，沒網路就不能開發，無法 commit，無法看 history log。
  - 坐高鐵，坐飛機的時候
  - 網路故障的時候
  - 咖啡店沒有無線網路的時候
- 分散式 CSV 系統即使沒網路，照常可以 commit 和看 history log。

2. 什麼是 Git ?

Git

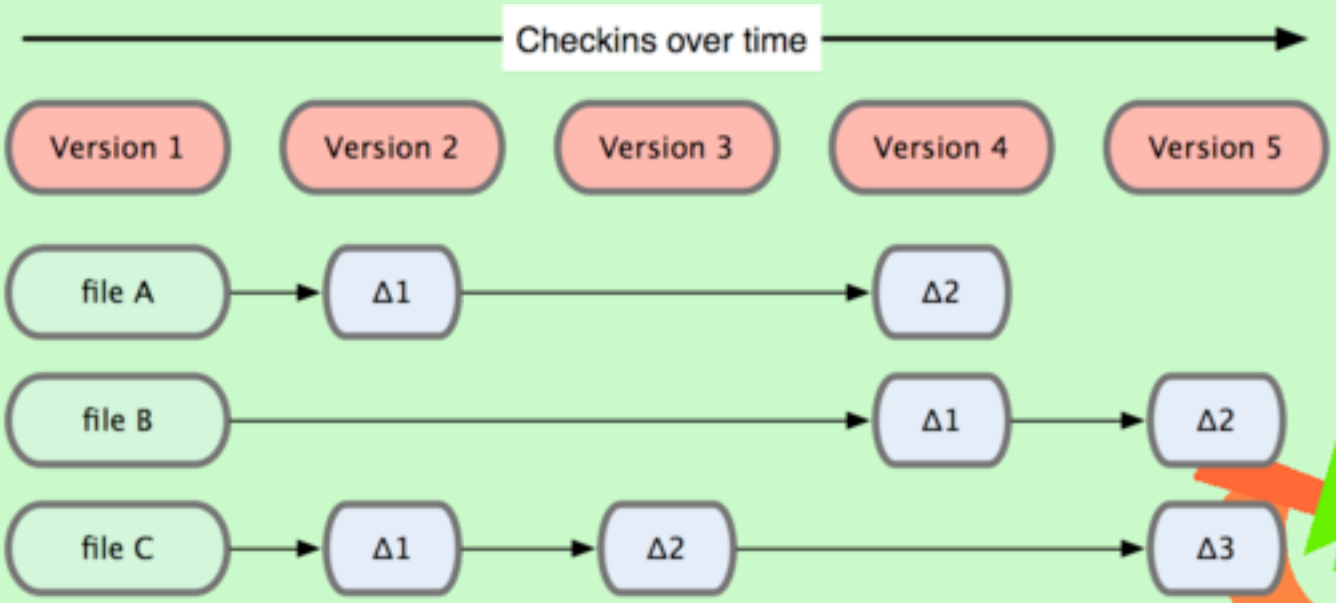
- 發明人 Linux Torvalds (Linux creator)
- 目的是管理 Linux Kernel 原始碼
- 2005/4 開始開發，2005/6 開始管理 Linix Kernel，2005/12 釋出 1.0 版。

誰在用 Git?

- Git
- Linux Kernel
- Perl
- Eclipse
- Gnome
- KDE
- Qt
- Ruby on Rails
- Android
- PostgreSQL
- Debian
- X.org

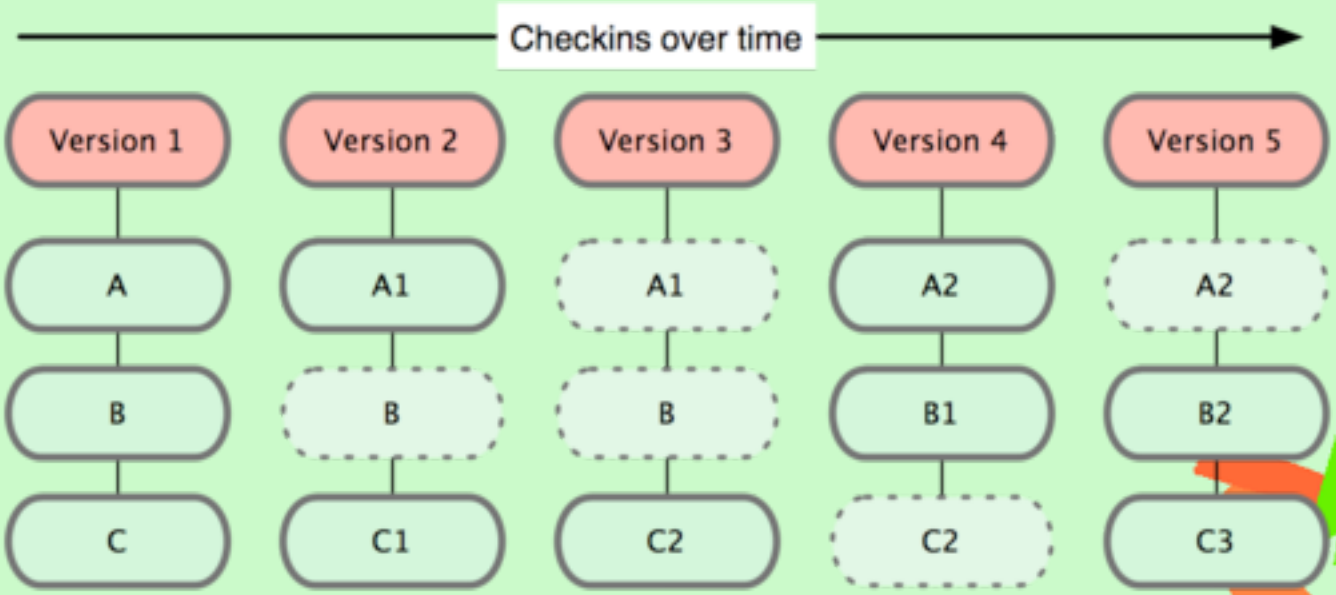
other VCS way

記錄每次檔案變更，開分支需要建立副本。



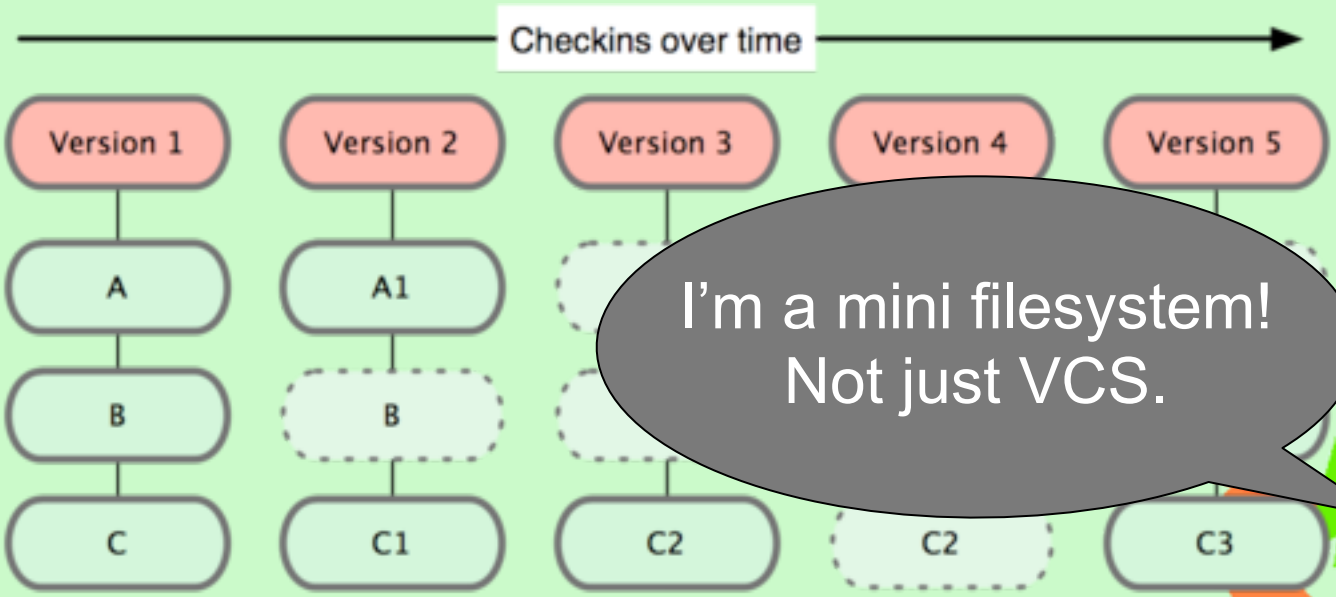
Git Way

記錄檔案內容，利用 metadata 建構出 snapshots 。
相同內容只會有一份記錄。



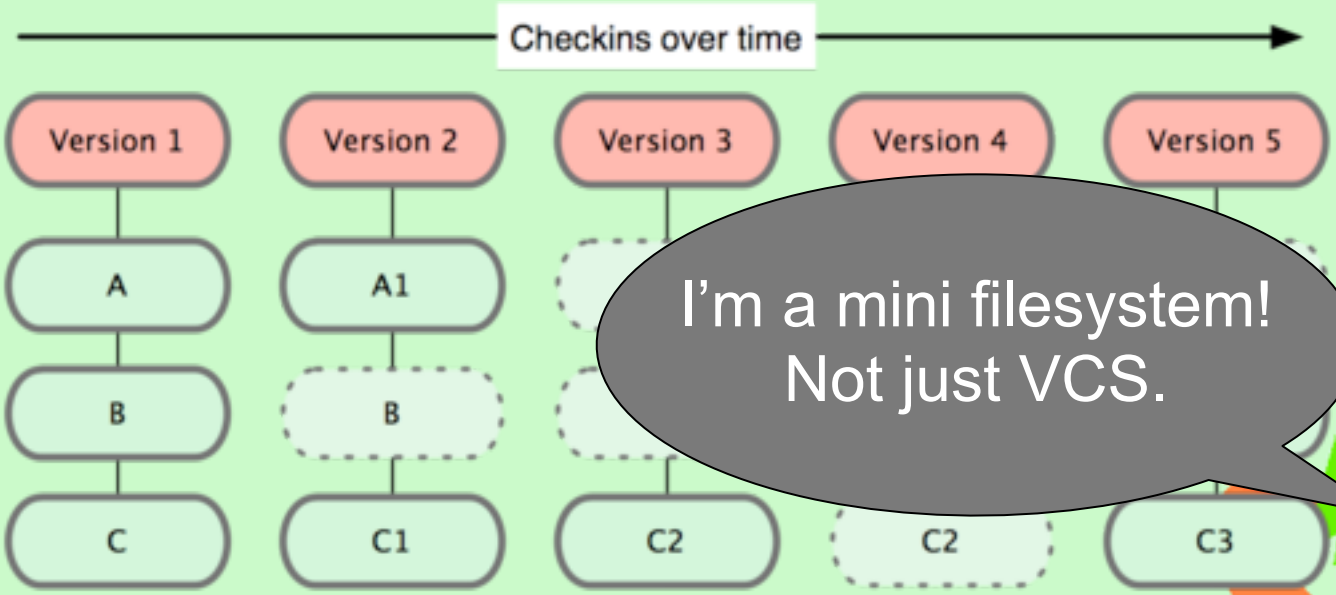
Git Way

記錄檔案內容，利用 metadata 建構出 snapshots 。
相同內容只會有一份記錄。



Git Way

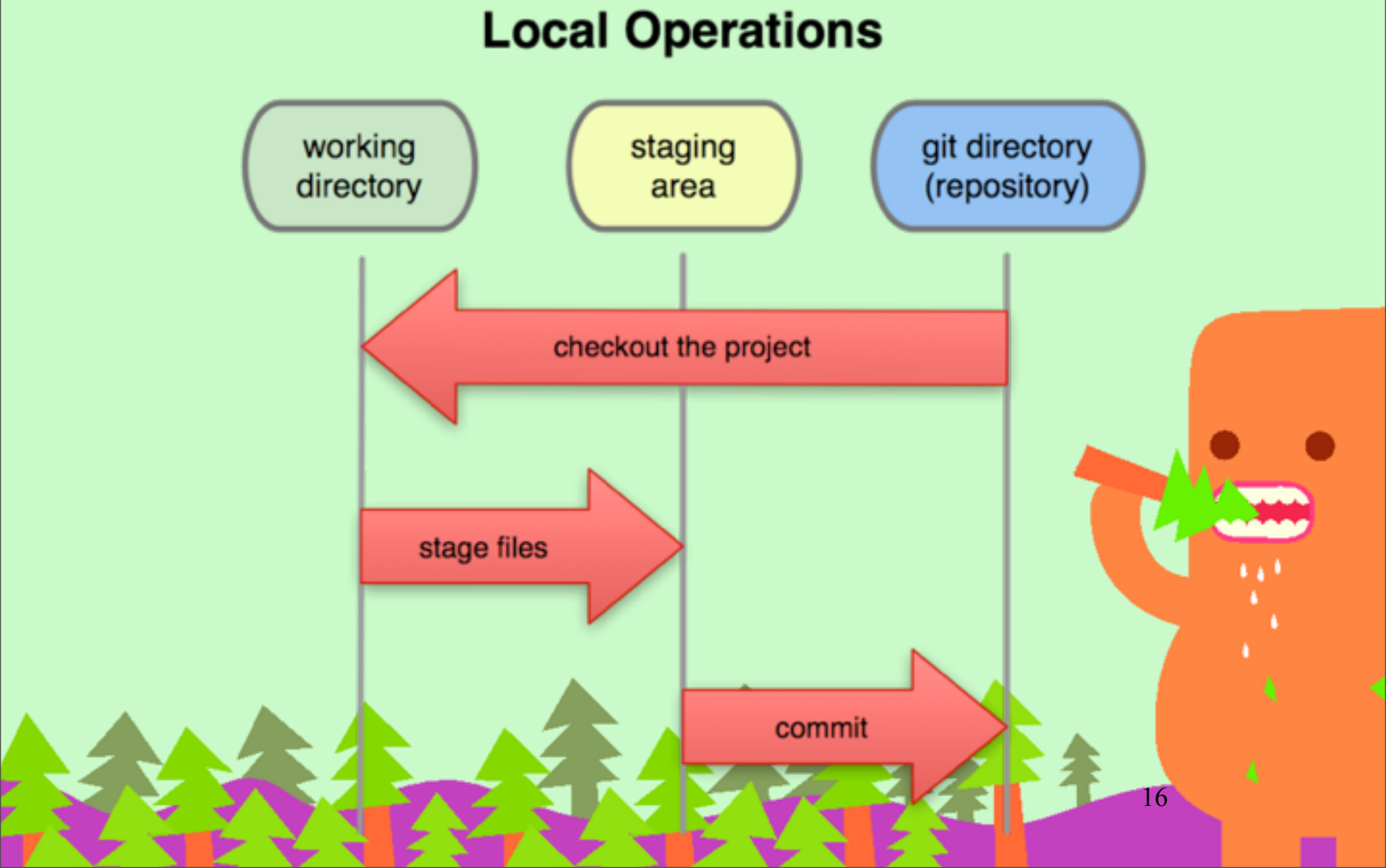
記錄檔案內容，利用 metadata 建構出 snapshots 。
相同內容只會有一份記錄。



I’m a mini filesystem!
Not just VCS.

延伸應用：可當做備份儲存格式，就跟 Apple 的 Time Machine 概念類似!

三種區域: Working tree/Staging area/Repository



Why Git is Better than X

http://zh-tw.whygitisbetterthanx.com

- 便宜快速的本地分支
- 所有內容都在本地端
- Git 很快
- Git 很省空間
- Staging 功能
- 它是分散式的
- 適用任何工作流程
- 我們有 GitHub!

自從用了 Git，就再也不會想用 SVN 了

- svn 開分支和 merge 超痛苦，能避免開 branch 就避免開 branch
- svn 看 log 超痛苦，需要等待網路連線
- svn commit 或 checkout 超慢，主機放國外常 commit 一半中斷

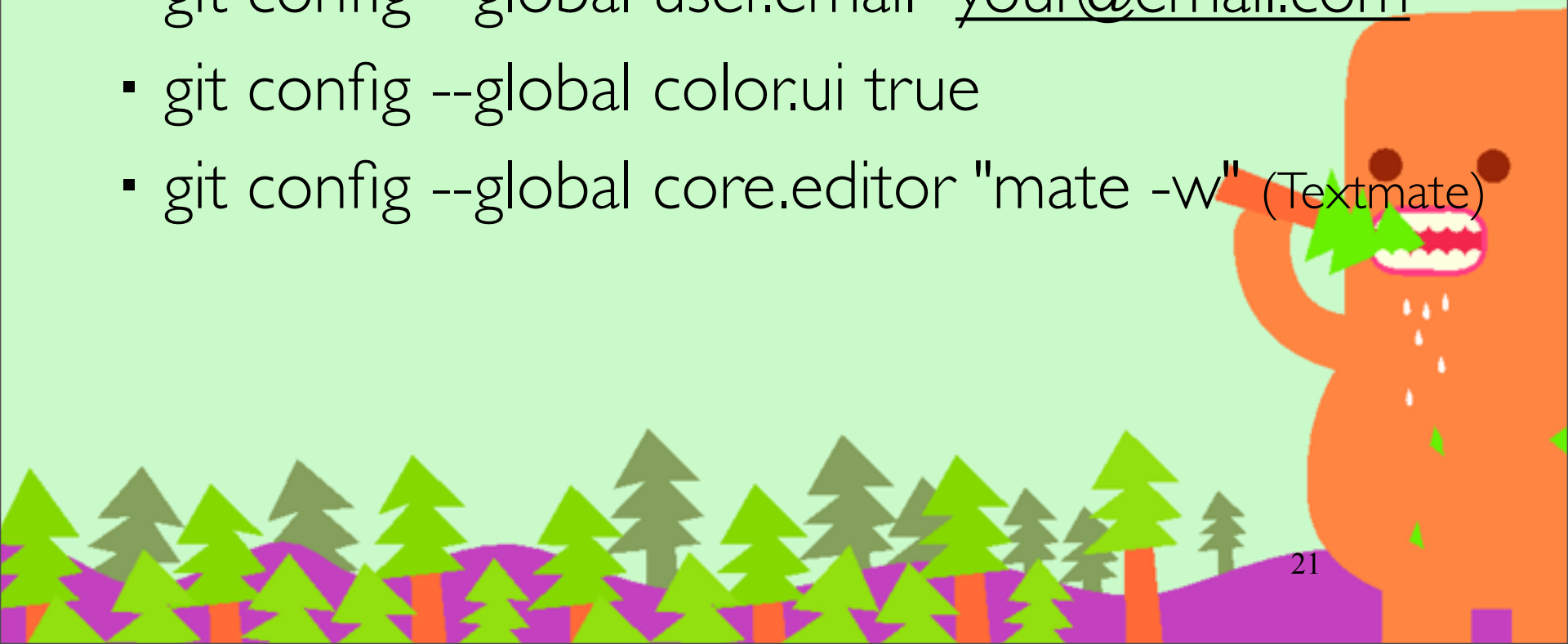
3-1. Git 指令基礎

Install

- 安裝 Git
  - http://help.github.com/set-up-git-redirect
- 安裝 Git GUI(optional)
  - GitX (Mac)
  - TortoiseGit/Git Extensions (Windows)
  - qgit (Linux)
- 產生 SSH Key
  - ssh-keygen -t rsa -C "your_email@youremail.com"

Setup

- 編輯 ~/.gitconfig 或輸入以下指令
- git config --global user.name "Your Name"
- git config --global user.email "your@email.com"
- git config --global color.ui true
- git config --global core.editor "mate -w" (Textmate)



Setup (Windows only)

- 處理換行字元誤認為變更
- git config --global core.autocrlf true
- git config --global core.safecrlf true

從頭建立 Repository

- mkdir sandbox
- cd sandbox
- git init

從一個已經存在的 Repository

- SSH 安全性最佳
  - git clone git@github.com:ihower/sandbox.git
- HTTP/HTTPS 速度最差，但能突破防火牆限制
  - git clone https://ihower@github.com/ihower/sandbox.git
- Git protocol 速度最快，但缺認證機制(因此只能做成唯讀)
  - git clone git://github.com/ihower/sandbox.git
- File 本機目錄 (有人用 Dropbox 分享 git init --bare --shared 目錄!! Crazy!!)
  - git clone file://path/to/repo.git

第一次 commit

- touch README
- git add README
- git status
- git commit -m “First Commit”

修改看看

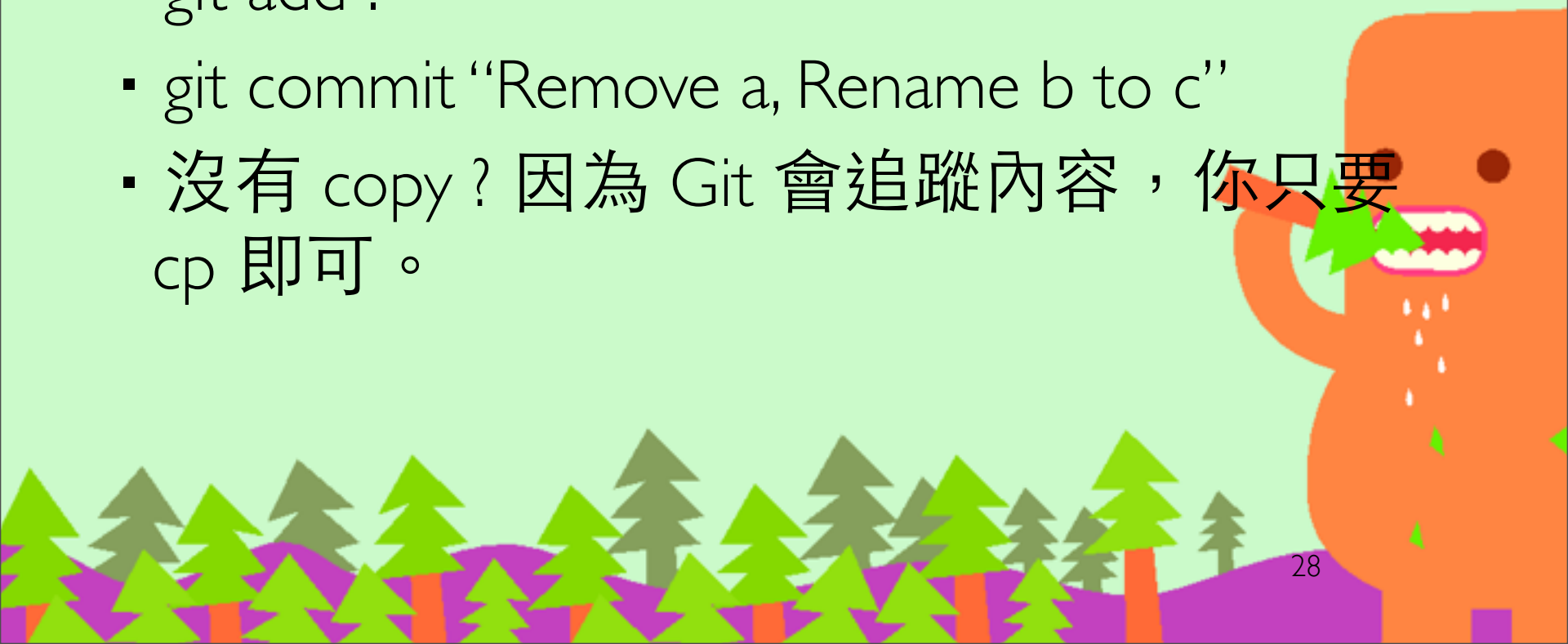
- 編輯 README 做些變更
- git status
- git diff
- git add .
  (一次加入所有變更跟新增檔案，但不包括刪除的檔案!)
- git status
- git diff --cached
- git commit -m “Update README”

Staging 準備要 commit 的檔案

- touch a.rb
- touch b.rb
- git add a.rb
- git commit “Update a”
- git add b.rb
- git commit “Update b”

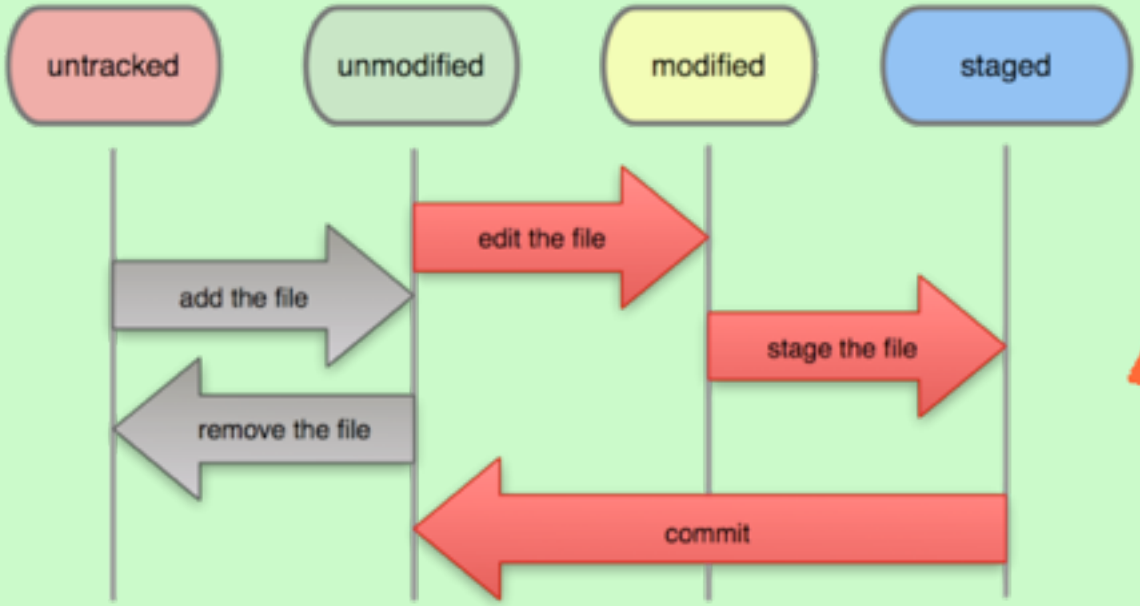
刪除和搬移檔案

- git rm a.rb
- git mv b.rb c.rb
- git add .
- git commit “Remove a, Rename b to c”
- 沒有 copy？因為 Git 會追蹤內容，你只要 cp 即可。



尚未 commit 前的三種檔案狀態

Untracked files	Changes not staged for commit	Changes to be committed



歷史紀錄

- git log
- git log --oneline
- git log --oneline --decorate --graph
- 還是用 GUI 吧

忽略不需要追蹤的檔案

- 編輯 .gitignore (屬於專案的設定，會 commit 出去)
- 編輯 ~/.gitignore (你個人的 Global 設定)
- 空目錄不會 commit 出去，必要時需要放 .gitkeep 檔案
- .gitignore 大集合
  https://github.com/github/gitignore

通常什麼檔案不需要 commit 出去?

- tmp/*
- log/*
- 你個人環境的設定檔(例如你偏愛的編輯器設定檔)
- 可以自動產生的檔案
- build/* 等 compile 之後的檔案
- .DS_Store
- Thumbs.rb

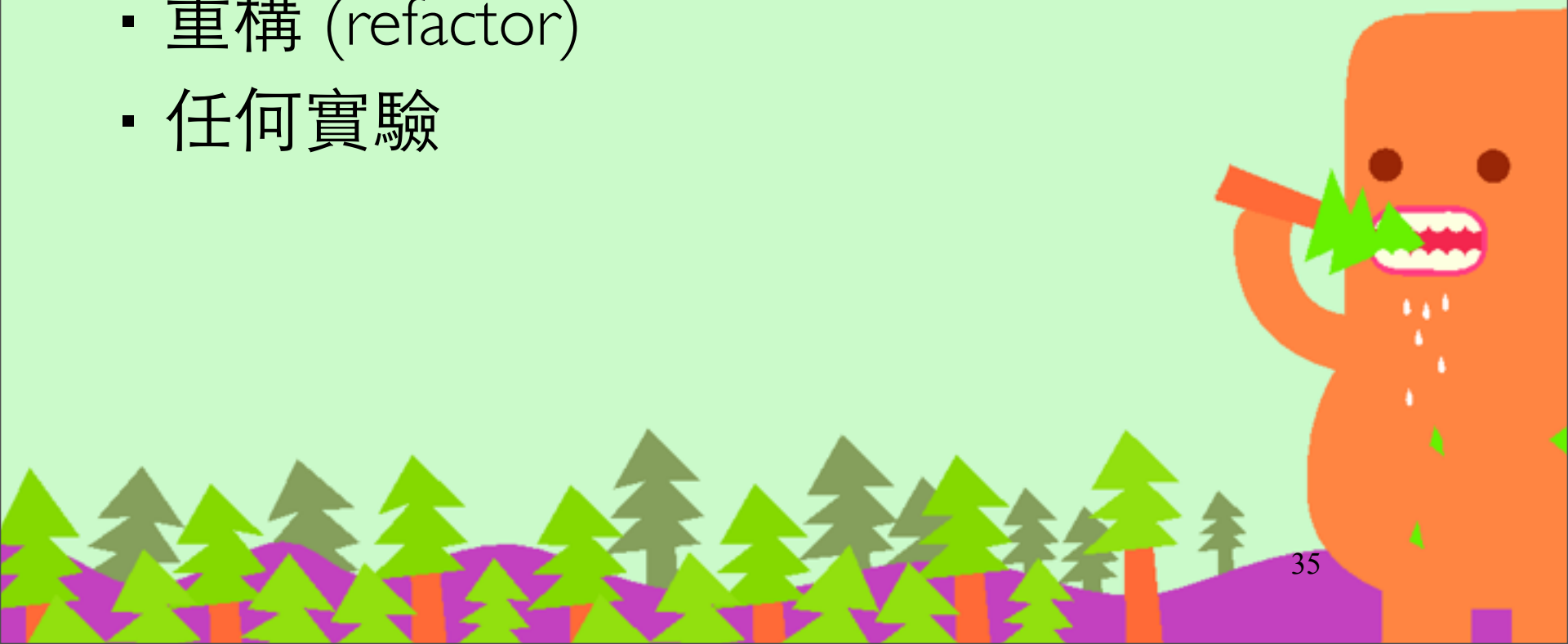
Commit 基本原則

- 適當的粒度/相關性/獨立性
  - 以一個小功能、小改進或一個 bug fixed 為單位。
  - 對應的 unit test 程式在同一個 commit
  - 無相關的修改不在同一個 commit
  - 語法錯誤的半成品程式不能 commit
- git diff --check 可以檢查多餘的空白
- commit 訊息很重要
  - 第一行寫摘要
  - 有需要寫實作細節的話，放第二行之後

3-2. Git 指令基礎：開分支

何時開 Branch ?

- topic feature 開發新功能
- Bug fixes
- 重構 (refactor)
- 任何實驗



開 branch

- git branch new_feature
- git branch
- git checkout new_feature
- touch new_feature.rb
- git add new_feature.rb
- git commit -m “New feature”

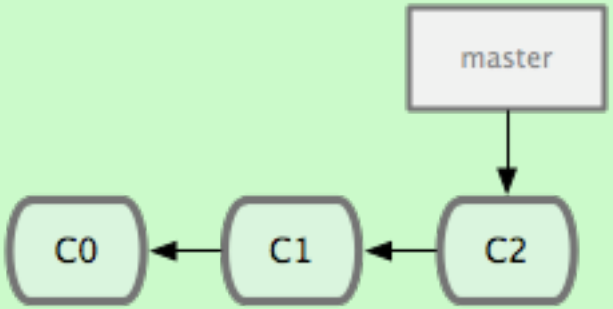
開 branch

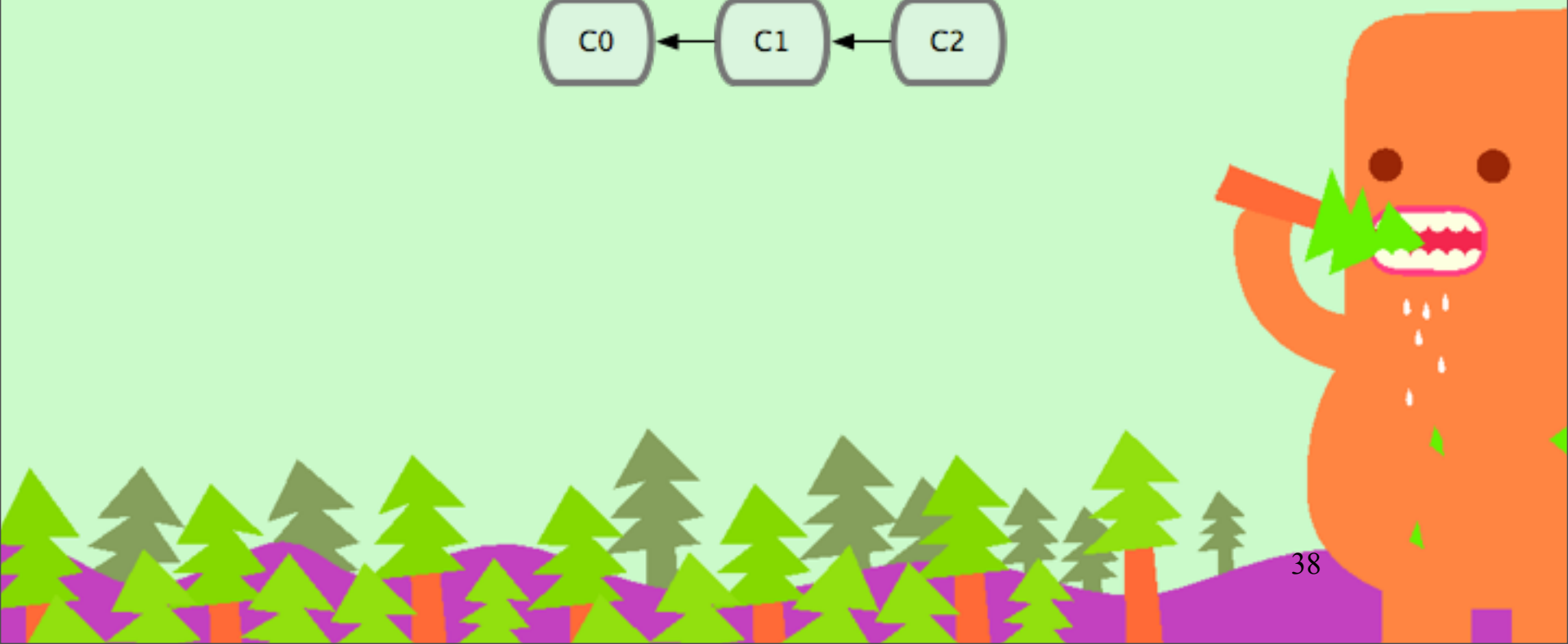
- git branch new_feature
- git branch
- git checkout new_feature
- touch new_feature.rb
- git add new_feature.rb
- git commit -m “New feature”

合併 branch 回來

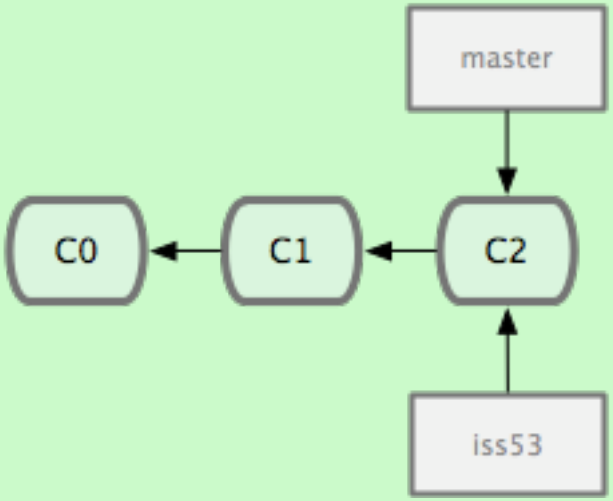
- git checkout master
- git merge new_feature

一個流程範例

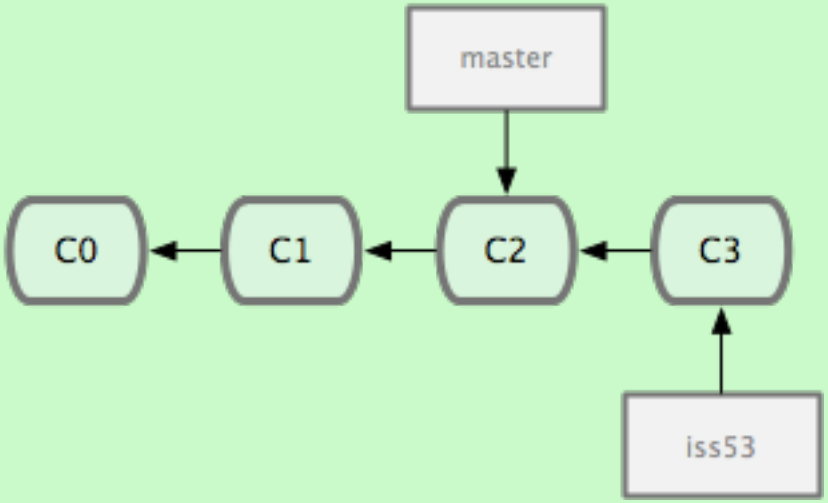




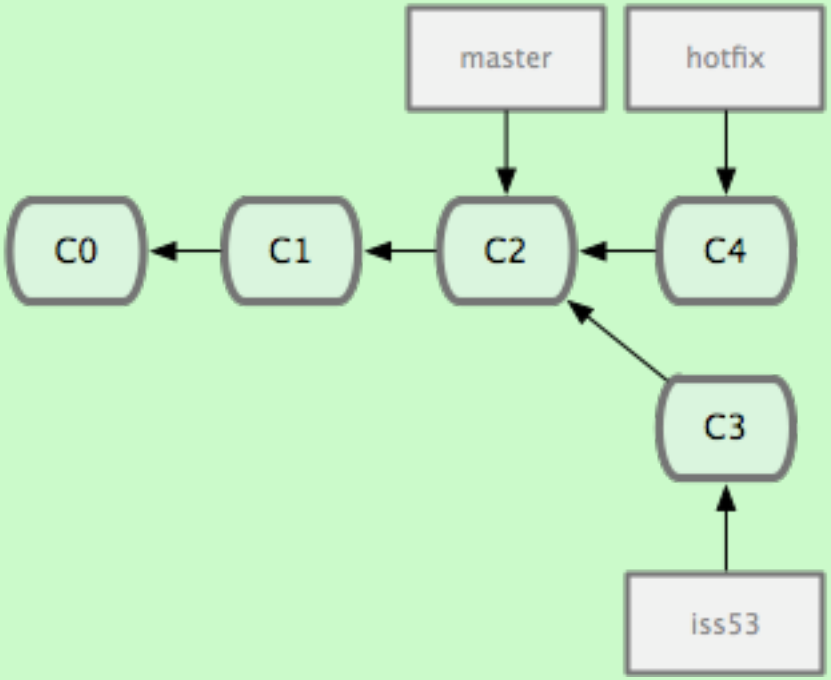
(master) git branch iss53



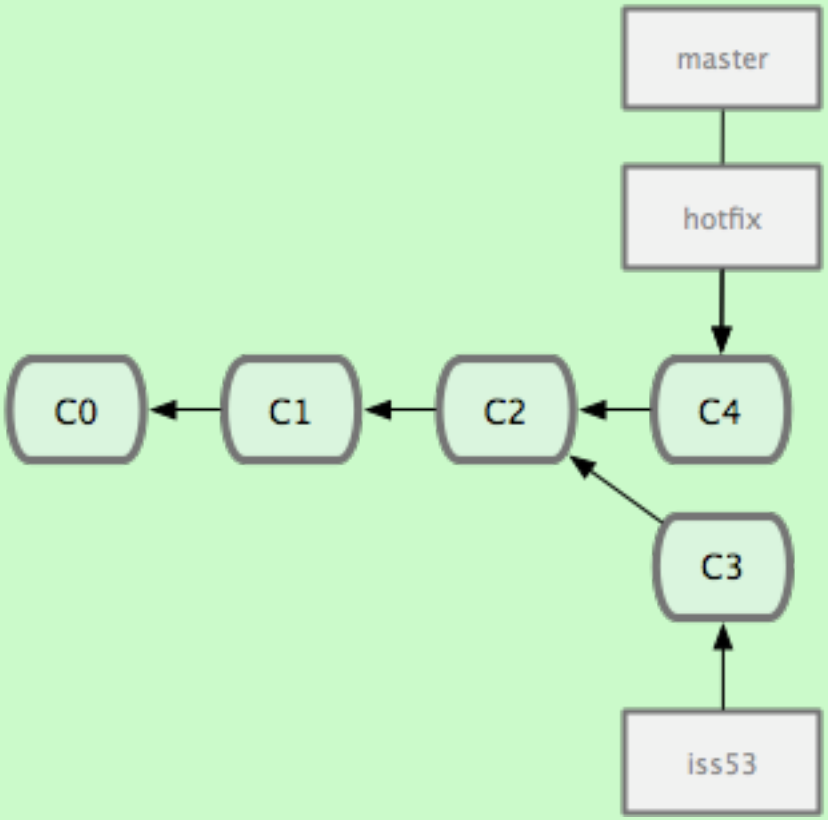
(iss53) git commit



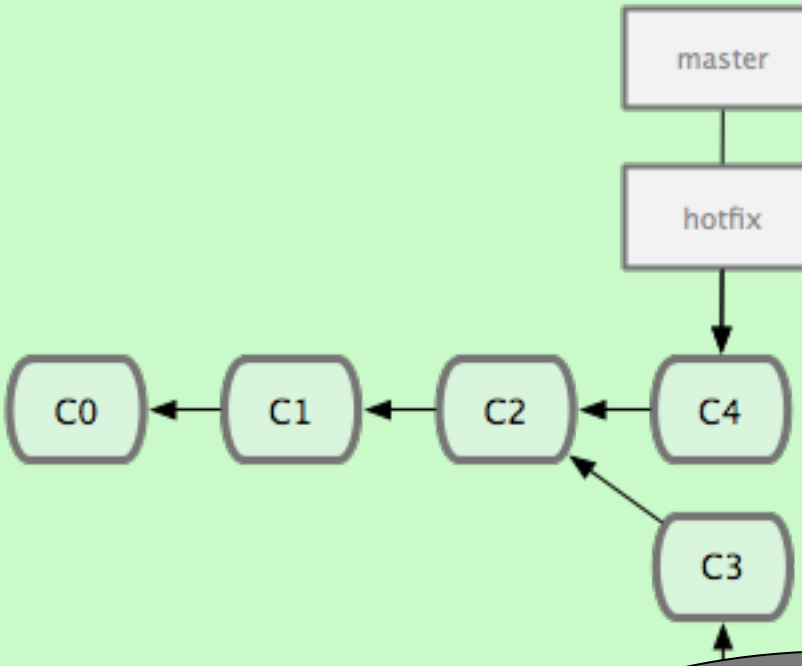
(master) git branch hotfix
(hotfix) git commit



(master) git merge hotfix

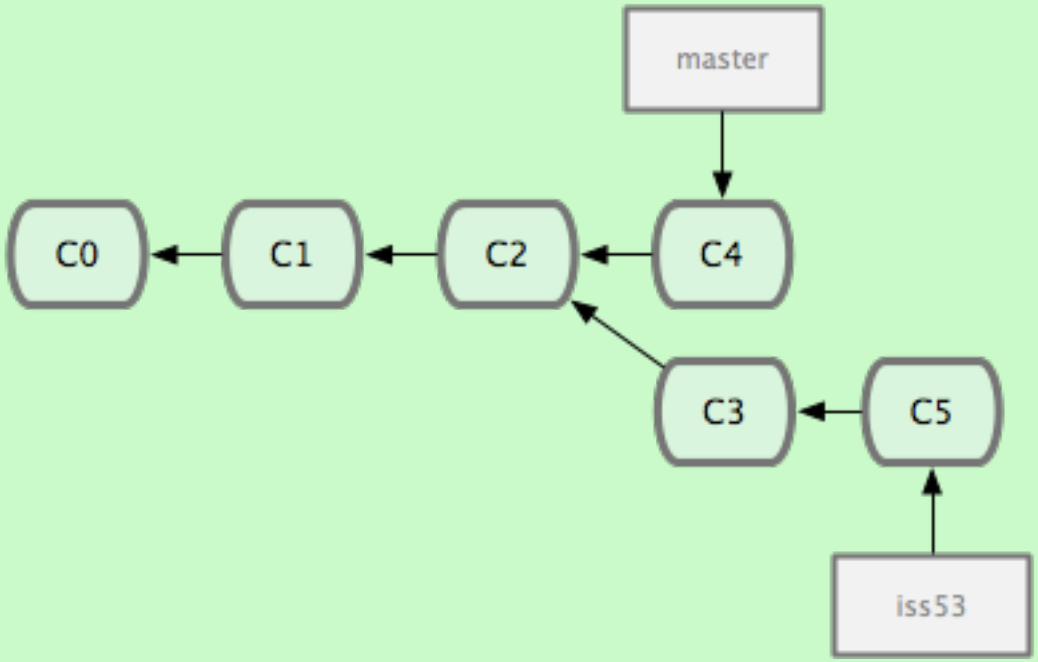


(master) git merge hotfix

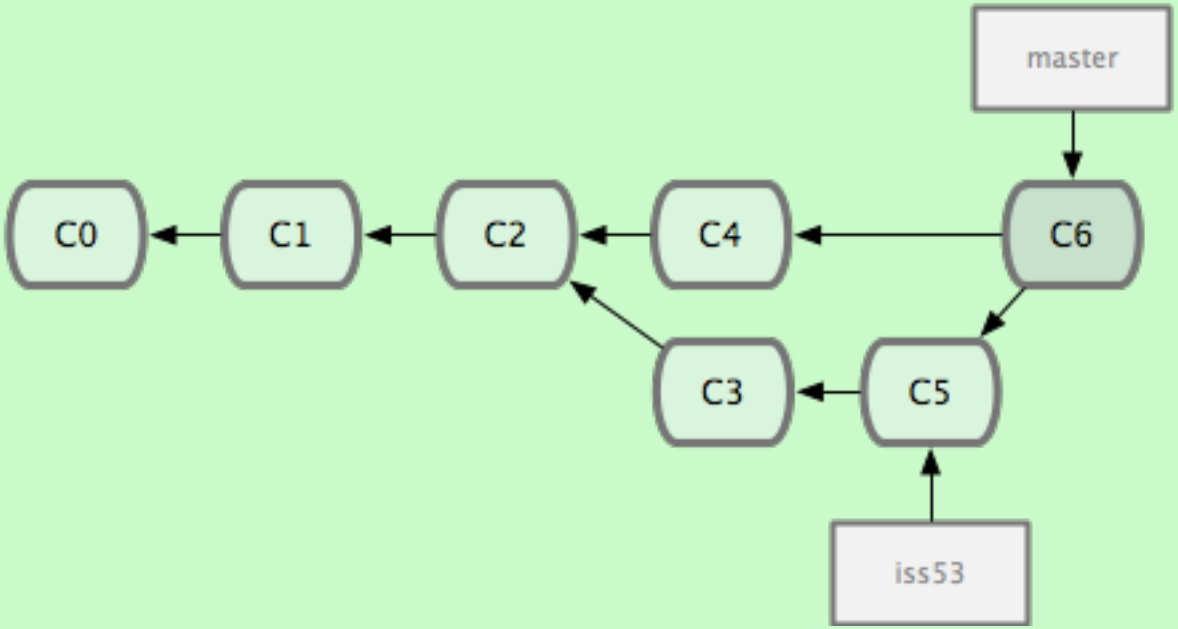


這種只需要改變
參考的合併叫作
fast-forward

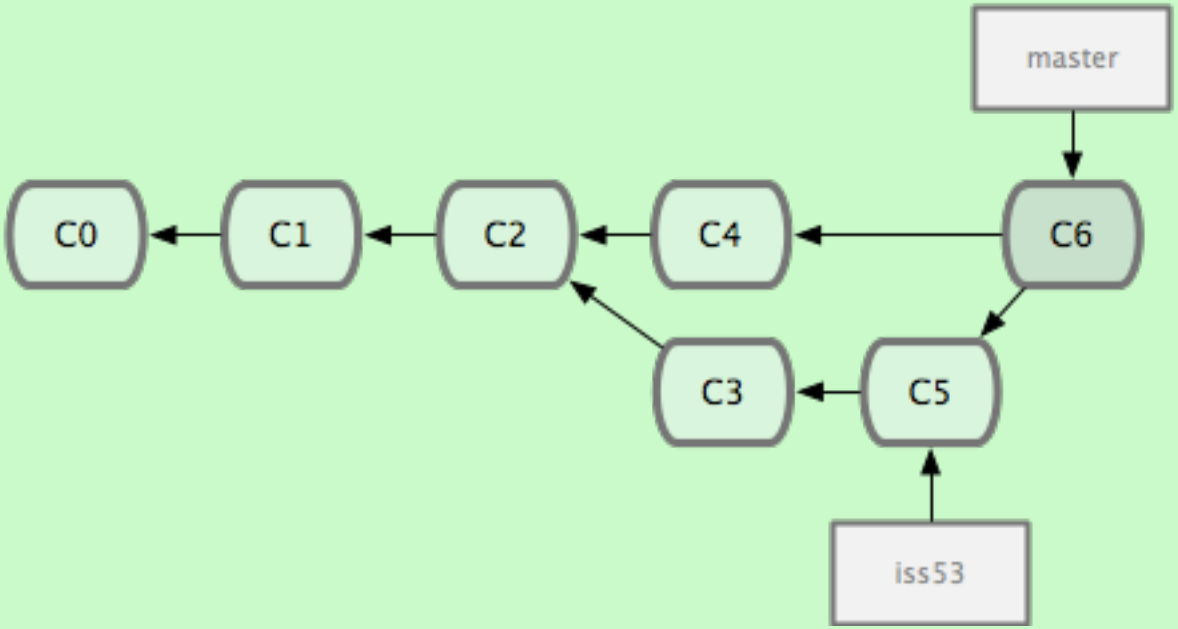
(iss53) git commit



(master) git merge iss53



(master) git merge iss53



C6 是個 merge commit

四種合併模式

- Straight merge 預設的合併模式
  - 保留被合併的 branch commits 記錄，並加上一個 merge commit，看線圖會有兩條 Parents 線。
  - 如果是 fast-forward，就不會有 merge commit。除非加上 --no-ff 參數。

四種合併模式 (cont.)

- Squashed commit
  - git merge new_feature --squash
  - 壓縮成只有一個 merge-commit，不會有被合併的 log。SVN 的 merge 即是如此。
- cherry-pick
  - git cherry-pick 714cba
  - 只合併指定的 commit

使用 branch 注意事項

- 切換 working tree 的 branch 時，如果有檔案在 staging area 或是 modified，會無法切換。
- 可以先 commit，反正只要不 push 出去，再 reset 回來即可
- 或是用 stash 指令暫存起來
  - git stash
  - git stash apply
  - git stash clear

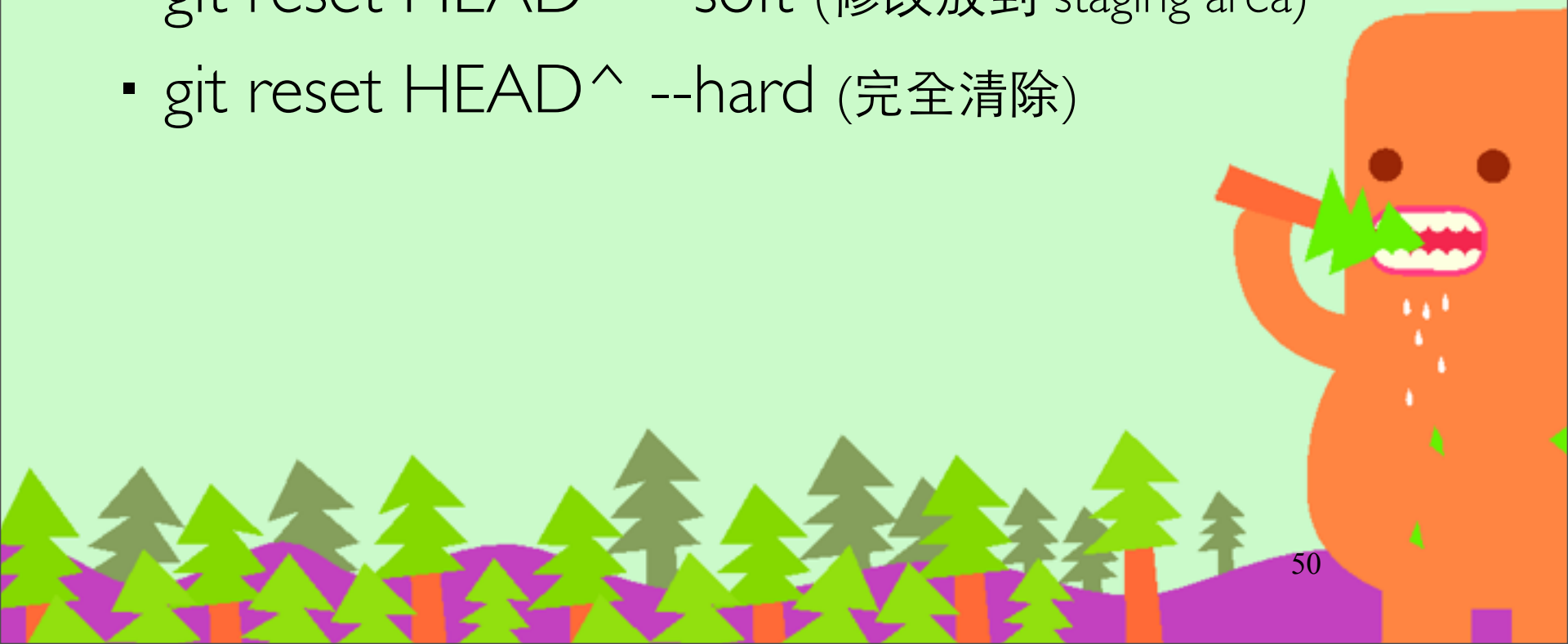
3-3. Git 指令基礎：還沒 push 前可以做的事

還沒 push 分享出去的 branch，你可以...

- 不同於 SVN，Git 的 Commit 其實還在本地端，所以可以修改 commit logs!!
  - 合併 commits
  - 修改 commit 內容或訊息
  - 打散 commit 成多個 commits
  - 調換 commits 順序

reset (砍掉 commit 記錄)

- git reset e37c75787
- git reset HEAD^ (留著修改在 working tree)
- git reset HEAD^ --soft (修改放到 staging area)
- git reset HEAD^ --hard (完全清除)



revert (還原 commit 記錄)

- 和 reset 不同，revert 是新增一筆 commit 來做還原。
- git revert e37c75787
- git revert HEAD^

rebase (重新 commit 一遍)

- git rebase -i e37c7578

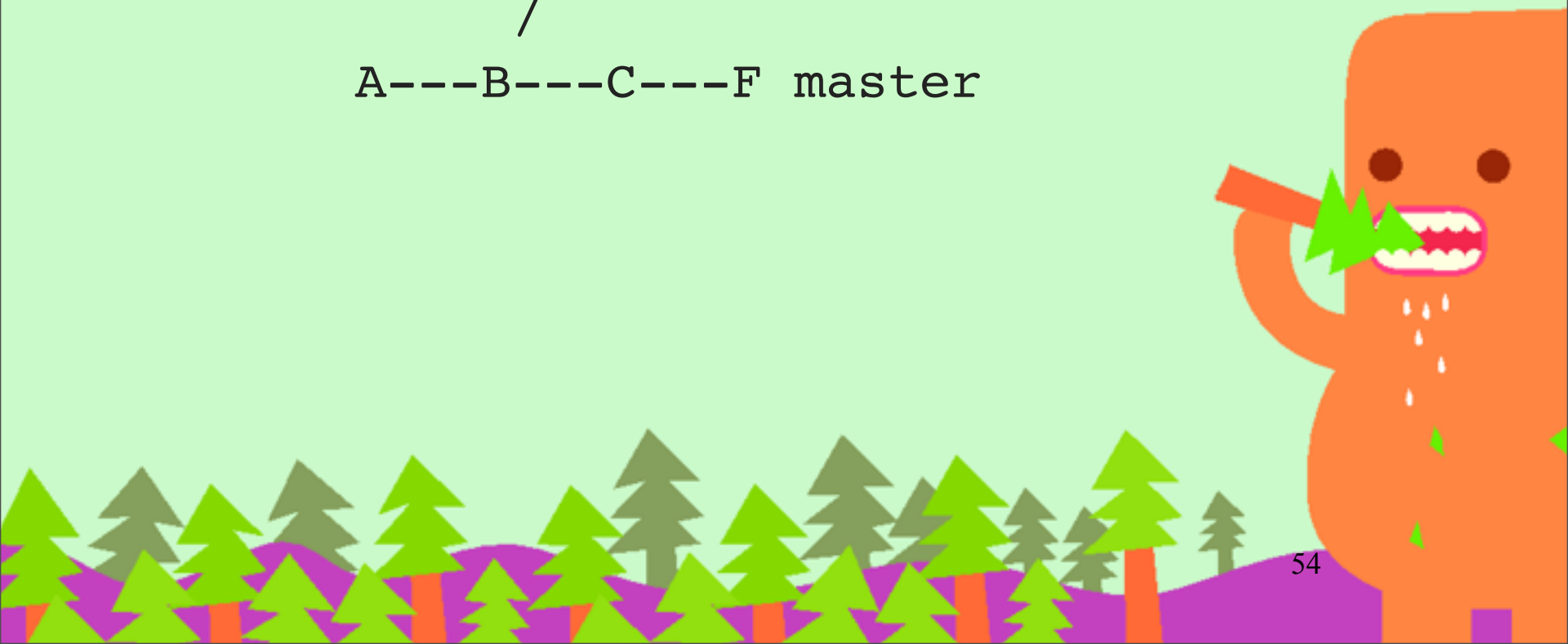
```
pick 048b59e first commit
pick 995dbb3 change something
pick aa3e16e changed

# Rebase 0072886..1b6475f onto 0072886
#
# Commands:
#  p, pick = use commit
#  r, reword = use commit, but edit the commit message
#  e, edit = use commit, but stop for amending
#  s, squash = use commit, but meld into previous commit
#  f, fixup = like "squash", but discard this commit's log message
#  x, exec = run command (the rest of the line) using shell
#
# If you remove a line here THAT COMMIT WILL BE LOST.
# However, if you remove everything, the rebase will be aborted.
```

第四種合併模式: rebase

- rebase 調整分支點 (僅適用於 local branch)
  - 1. 從要被合併的 branch 頭重新分支
  - 2. 將目前 branch 的 commits 記錄一筆一筆重新 apply/patch 上去
  - 等於砍掉重練 commit log，僅適合還沒分享給別人的 local branch。

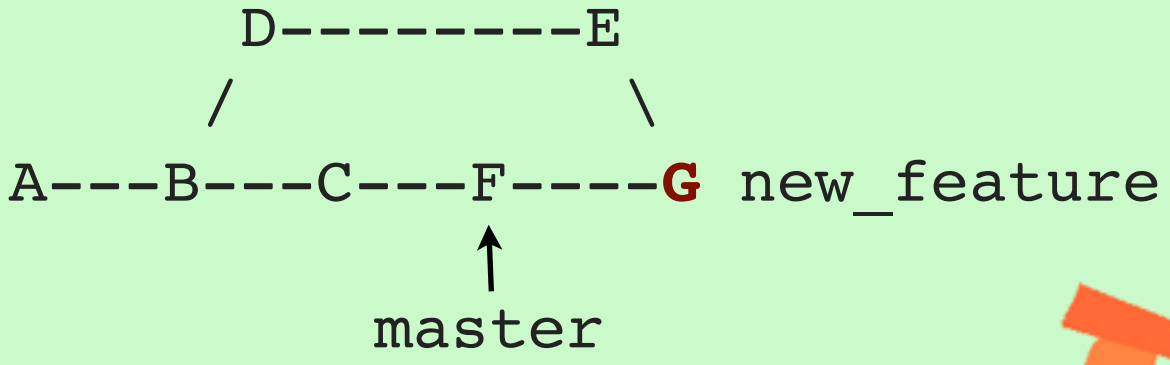
一個範例: 開發中的 new_feature 想要合併 master 主幹的變更

```
          D---E new_feature
         /
A---B---C---F master
```



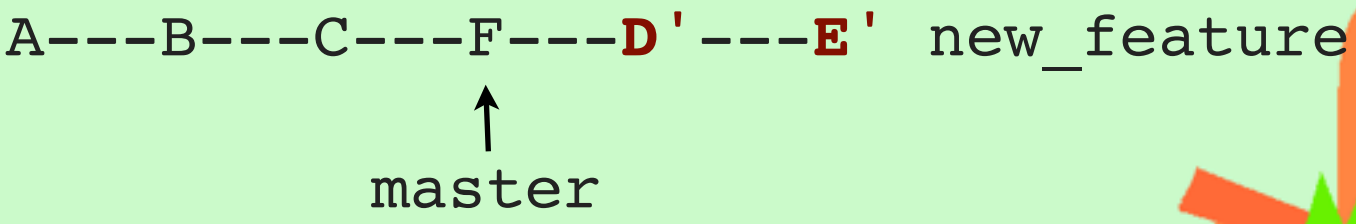
如果是 merge

(new_feature) git merge master



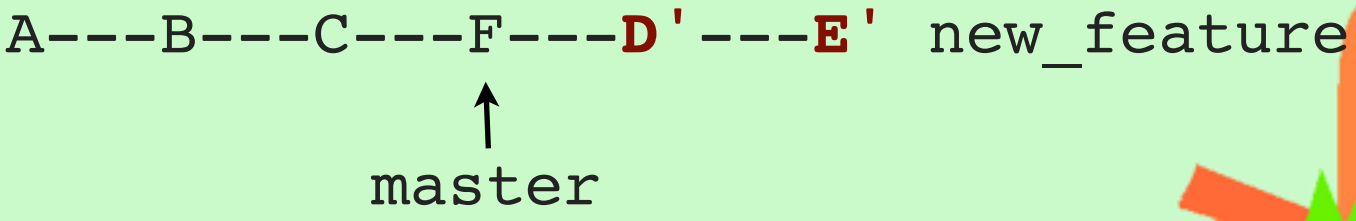
如果用 rebase，就沒有 G 點!

(new_feature) git rebase master



如果用 rebase，就沒有 G 點!

(new_feature) git rebase master

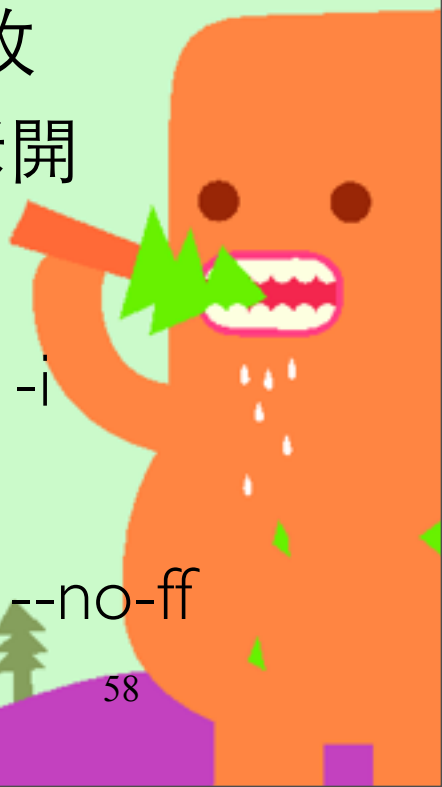


已經 push 分享出去的 commits 記錄，請千萬不要再修改記錄 push 出去!!

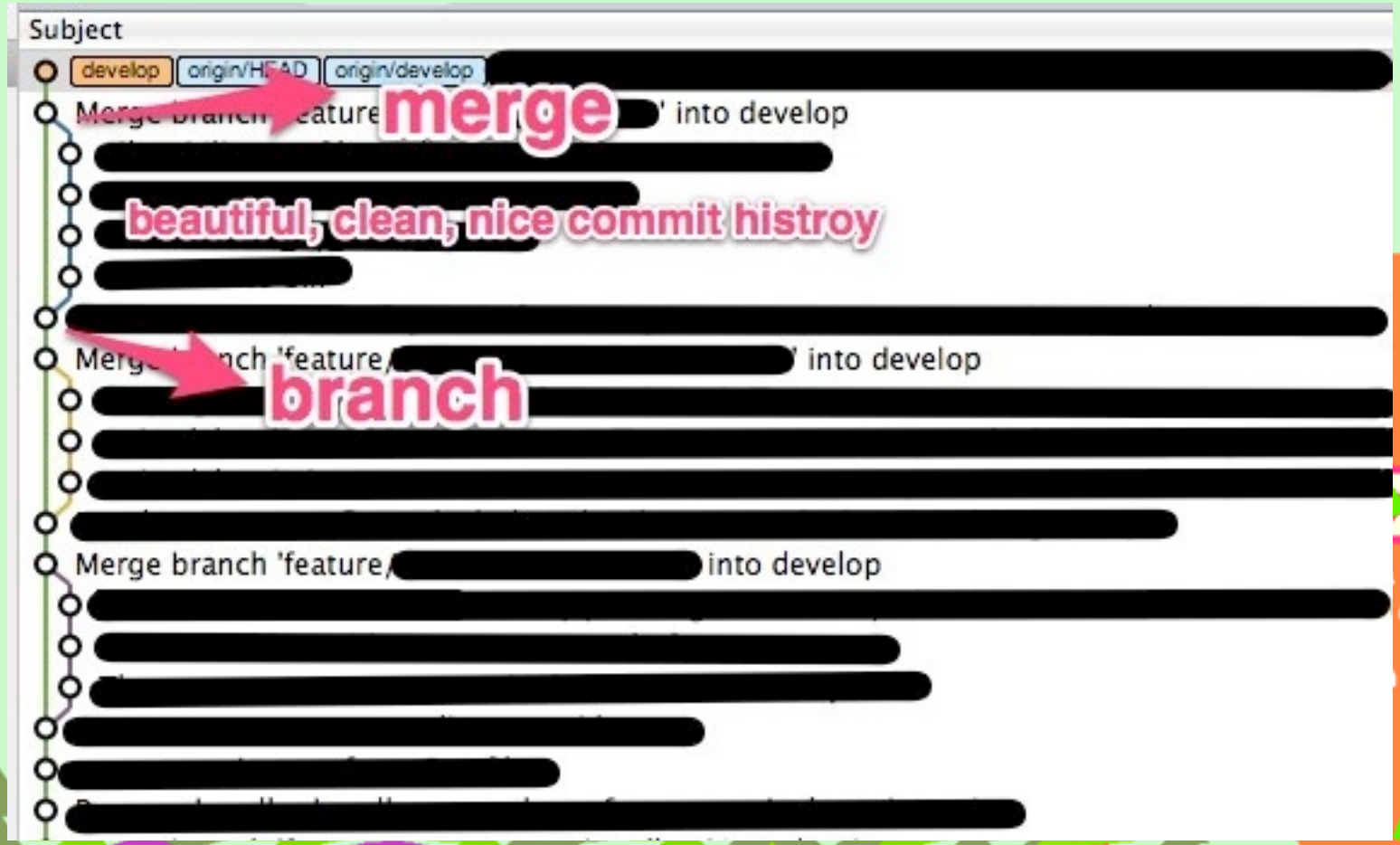
集大成：rebase + merge 的完美合併法

(假設我們要將 feature branch 合併回主幹 develop)

- 原因
  - feature branch 很亂，不時 merge 與主幹同步
  - feature branch 有 typo，commit 訊息想改
  - feature branch 有些 commits 想合併或拆開
- 作法
  - 先在 feature branch 做 git rebase develop -i
  - (反覆整理直到滿意) git rebase 分岔點 -i
  - 在從 develop branch 做 git merge feature --no-ff



超級乾淨，每一次的 merge commit 就代表一個功能完成



注意事項 (I)

- 必須要加 --no-ff 才會有 merge commit。不然會是 fast-forward。
- rebase 之後的 feature branch 就不要再 push 出去了
- 如果有遠端的 feature branch，合併完也砍掉

注意事項 (2)

- 不求一次 rebase 到完美，不然中間的 conflict 會搞混。
- 可以一次改點東西就 rebase 一次，然後開個臨時的 branch 存檔起來，再繼續 rebase 自己直到滿意為止。

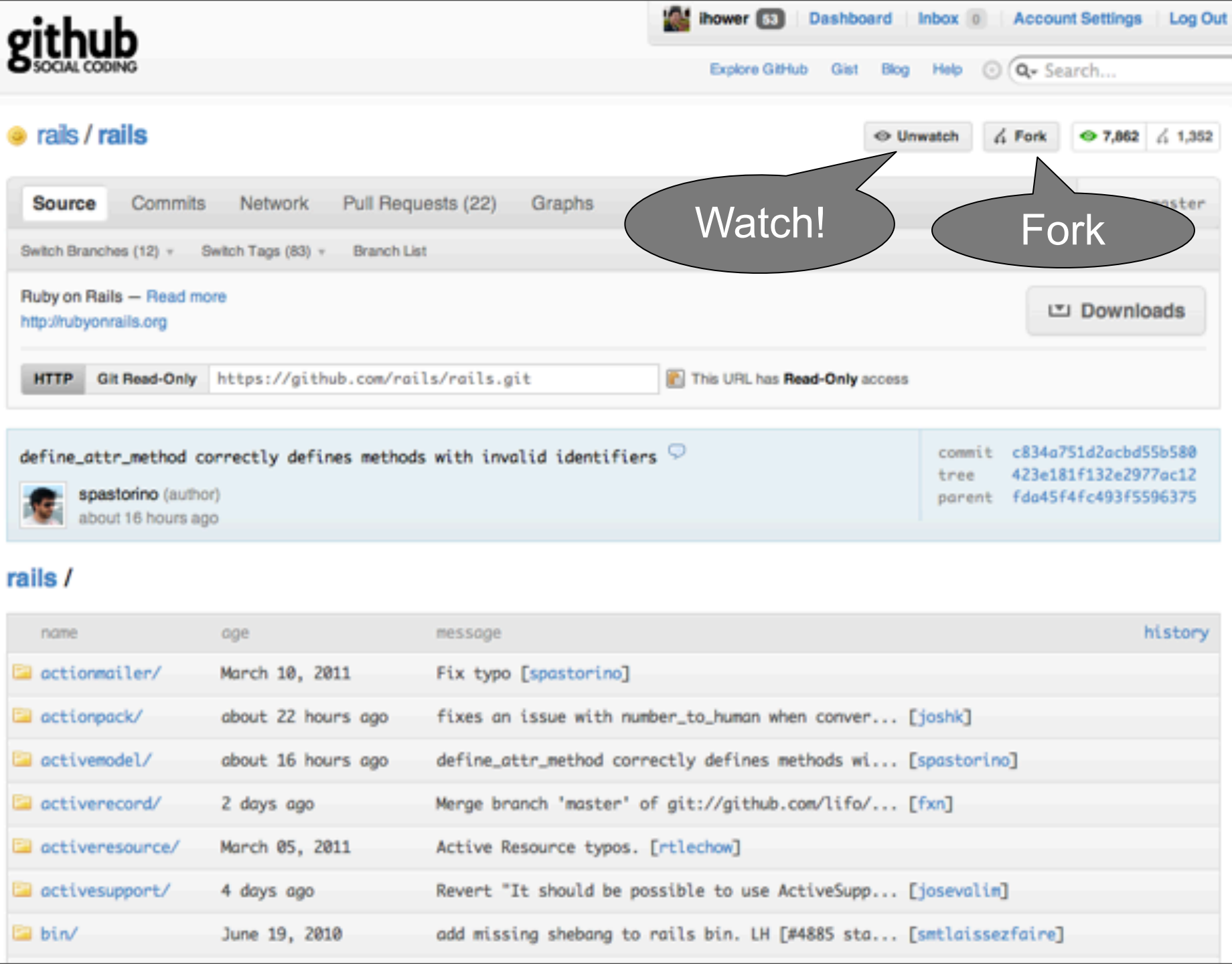
4. Github 是什麼？

= github

SOCIAL CODE HOSTING

誰在用 GitHub?

- twitter
- facebook
- rackspace
- digg
- Yahoo!
- shopify
- EMI
- six apart
- jQuery
- YUI 3
- mootools
- Ruby on Rails
- node.js
- symfony
- mongodb
- Erlang


github
SOCIAL CODING
ihower 53
Dashboard
Inbox 0
Account Settings
Log Out
Explore GitHub
Gist
Blog
Help
Search...
rails / rails
Unwatch
Fork
7,862
1,352
Watch!
Fork
Source
Commits
Network
Pull Requests (22)
Graphs
Switch Branches (12)
Switch Tags (83)
Branch List
Ruby on Rails — Read more
http://rubyonrails.org
Downloads
HTTP
Git Read-Only
https://github.com/rails/rails.git
This URL has Read-Only access
define_attr_method correctly defines methods with invalid identifiers
spastorino (author)
about 16 hours ago
commit c834a751d2acbd55b580
tree 423e181f132e2977ac12
parent fda45f4fc493f5596375
rails /
name
age
message
history
actionmailer/ March 10, 2011 Fix typo [spastorino]
actionpack/ about 22 hours ago fixes an issue with number_to_human when conver... [joshk]
activemodel/ about 16 hours ago define_attr_method correctly defines methods wi... [spastorino]
activerecord/ 2 days ago Merge branch 'master' of git://github.com/lifo/... [fxn]
activeresource/ March 05, 2011 Active Resource typos. [rtlechow]
activesupport/ 4 days ago Revert "It should be possible to use ActiveSupp... [josevalim]
bin/ June 19, 2010 add missing shebang to rails bin. LH [#4885 sta... [smtlaissezfaire]

Ruby is the #2 most popular language on GitHub

Explore | Repositories | **Languages** | Timeline | Search | Tips

Ruby | Recently Created | Recently Updated

Most Watched Today

- wbailey / **kata**
- injekt / **slop**
- apneadiving / **Google-Maps-for-Rails**
- codegram / **resort**
- shuber / **attr_encrypted**

Most Forked Today

- mxcl / **homebrew**
- thoughtbot / **capybara-webkit**
- opscode / **cookbooks**
- edavis10 / **redmine**
- drbrain / **meme**

Most Watched This Week

- injekt / **slop**
- mxcl / **homebrew**
- rails / **rails**
- NoamB / **sorcery**
- diaspora / **diaspora**

Most Forked This Week

- mxcl / **homebrew**
- rails / **rails**
- diaspora / **diaspora**
- chicagoruby / **booksiebot7000**
- drbrain / **meme**

Most Watched This Month

- postrank-labs / **goliath**
- mxcl / **homebrew**
- rails / **rails**
- meskyanichi / **backup**
- diaspora / **diaspora**

Most Forked This Month

- mxcl / **homebrew**
- rails / **rails**
- diaspora / **diaspora**
- mojombo / **jekyll**
- plataformatec / **devise**

Most Watched Overall

- rails / **rails**
- mxcl / **homebrew**
- diaspora / **diaspora**
- joshuaclayton / **blueprint-css**
- plataformatec / **devise**

Most Forked Overall

- mxcl / **homebrew**
- rails / **rails**
- diaspora / **diaspora**
- thoughtbot / **paperclip**
- Shopify / **active_merchant**

All Languages

ActionScript
Ada
Arc
ASP
Assembly
Boo
C
C#
C++
Clojure
CoffeeScript
ColdFusion
Common Lisp
D
Delphi
Duby
Eiffel
Emacs Lisp
Erlang
F#
Factor
FORTRAN
Go
Groovy
Haskell
HaXe
Io
Java
JavaScript

Fork network

The rails network graph

All branches in the network using **rails/rails** as the reference point. Read our blog post about how it works.

Show Help

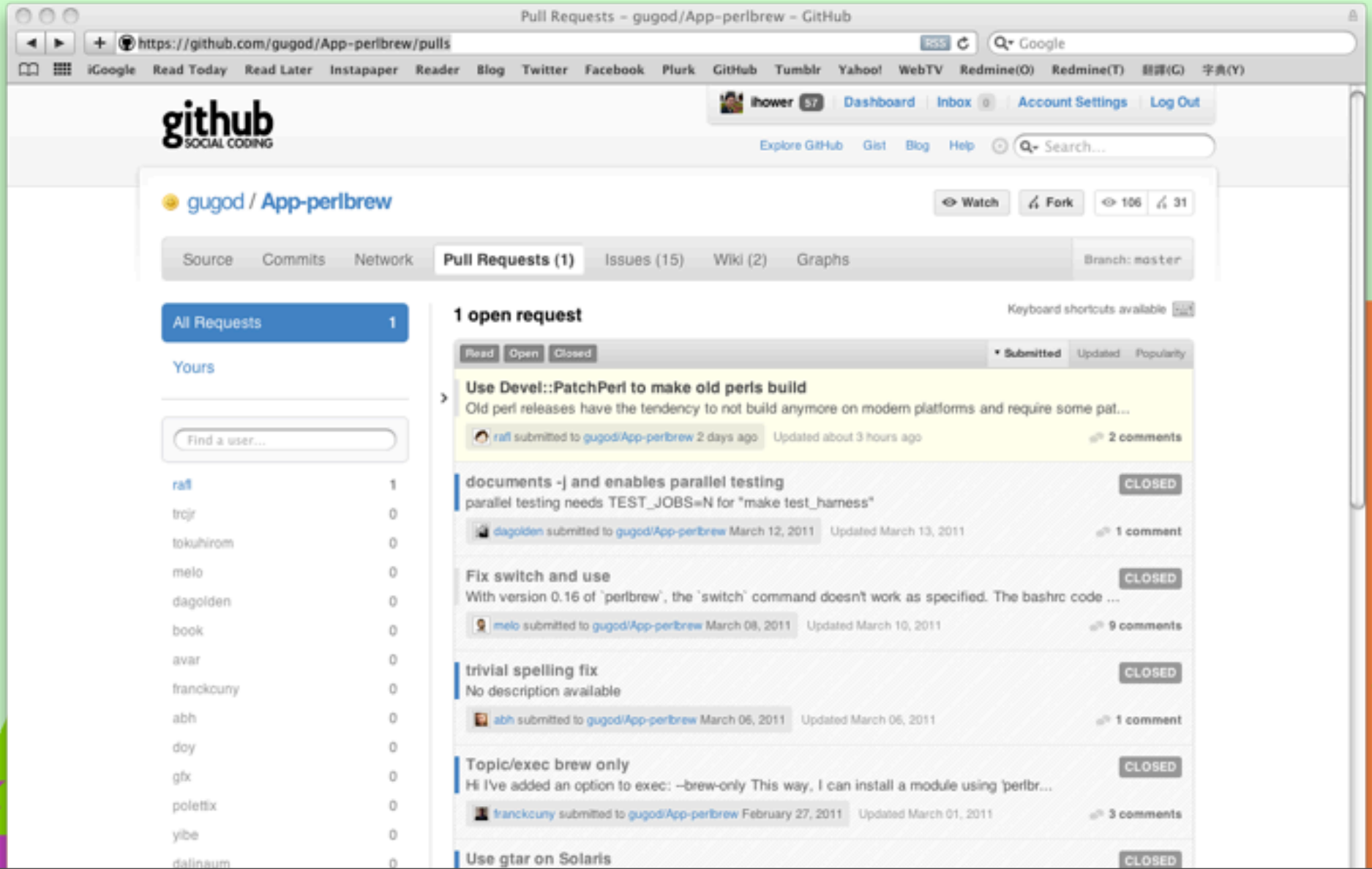
如何送 Patch 給開源專案?

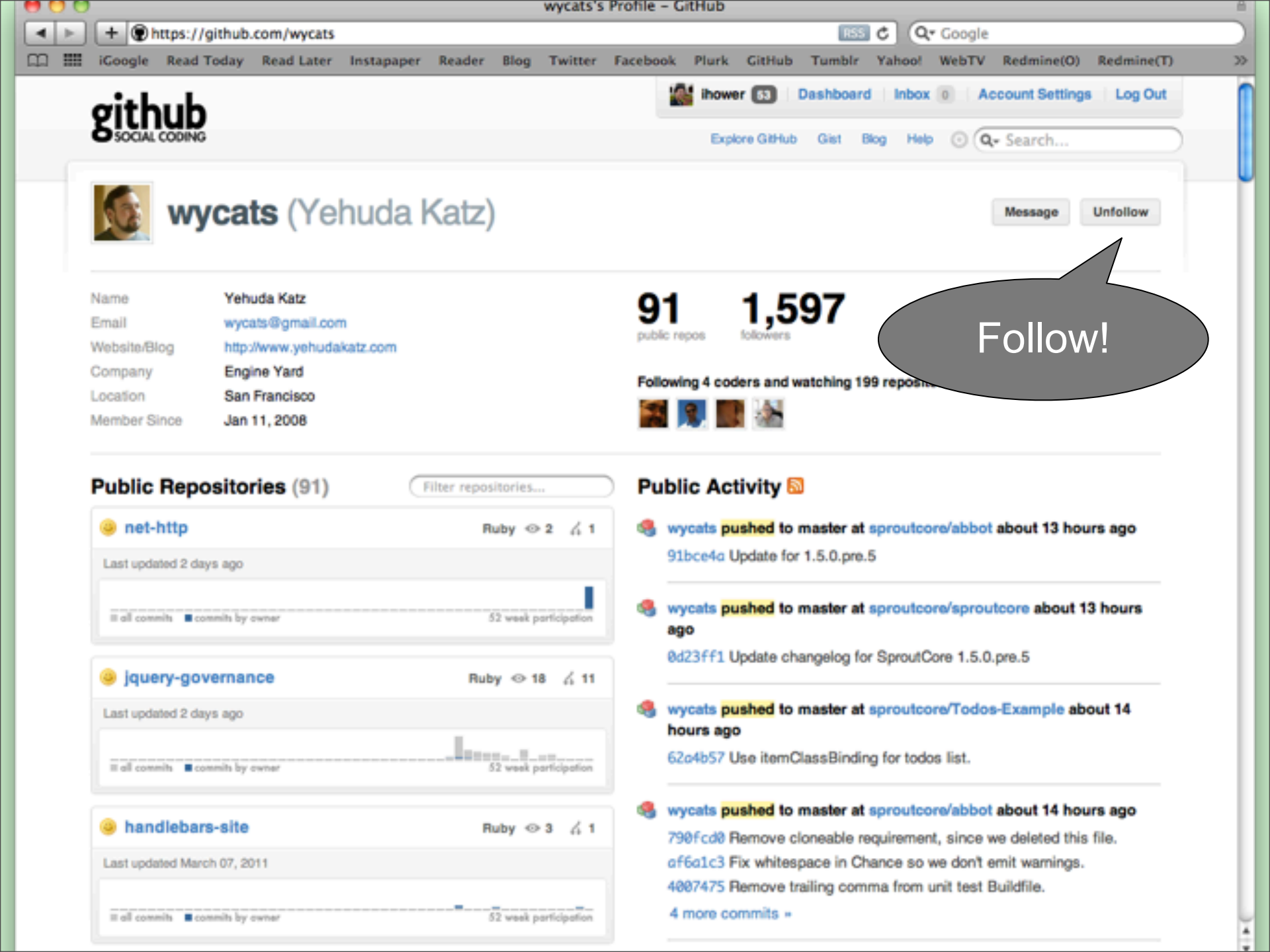
- Fork 成我的遠端 Repository
- Clone
- Fix bugs 或 Enhance features
- Push
- 送 Pull request 給原作者，請他合併

Fork & Pull Requests

有 75% 的被 fork 開源專案，有 outside merge 回來的記錄!!

被 watch 超過一人以上的開源專案，有 25% 有 outside contributions 紀錄!!




wycats's Profile – GitHub
https://github.com/wycats
Google
iGoogle Read Today Read Later Instapaper Reader Blog Twitter Facebook Plurk GitHub Tumblr Yahoo! WebTV Redmine(O) Redmine(T)
github
SOCIAL CODING
ihower 63 Dashboard Inbox 0 Account Settings Log Out
Explore GitHub Gist Blog Help Search...
wycats (Yehuda Katz)
Message Unfollow
Follow!
Name Yehuda Katz
Email wycats@gmail.com
Website/Blog http://www.yehudakatz.com
Company Engine Yard
Location San Francisco
Member Since Jan 11, 2008
91 public repos
1,597 followers
Following 4 coders and watching 199 repos
Public Repositories (91)
Filter repositories...
net-http Ruby 2 1
Last updated 2 days ago
all commits commits by owner 52 week participation
jquery-governance Ruby 18 11
Last updated 2 days ago
all commits commits by owner 52 week participation
handlebars-site Ruby 3 1
Last updated March 07, 2011
all commits commits by owner 52 week participation
Public Activity
wycats pushed to master at sproutcore/abbot about 13 hours ago
91bce4a Update for 1.5.0.pre.5
wycats pushed to master at sproutcore/sproutcore about 13 hours ago
0d23ff1 Update changelog for SproutCore 1.5.0.pre.5
wycats pushed to master at sproutcore/Todos-Example about 14 hours ago
62a4b57 Use itemClassBinding for todos list.
wycats pushed to master at sproutcore/abbot about 14 hours ago
790fcd0 Remove cloneable requirement, since we deleted this file.
af6a1c3 Fix whitespace in Chance so we don't emit warnings.
4007475 Remove trailing comma from unit test Buildfile.
4 more commits »

Gem.refresh to reload the gemfile - GitHub

https://github.com/carlhuda/bundler/wiki/Gem.refresh-to-reload-the-gemfile

iGoogle Read Today Read Later Instapaper Reader Blog Twitter Facebook Plurk GitHub Tumblr Yahoo! WebTV Redmine(O) Redmine(T)

github SOCIAL CODING

ihower 63 Dashboard Inbox 0 Account Settings Log Out

Explore GitHub Gist Blog Help Search...

carlhuda / bundler

Watch Fork 1,064 196

Source Commits Network Pull Requests (19) Issues (147) Wiki (14) Graphs Branch: 1-0-stable

Home Pages Wiki History Git Access

Gem.refresh to reload the gemfile

New Page Edit Page Page History

From issue 739:

Hi guys,

For working unicorn you introduced Gem.refresh (back in f0a65fea). And it looks like this:

```
gem_class.send(:define_method, :refresh) { }
```

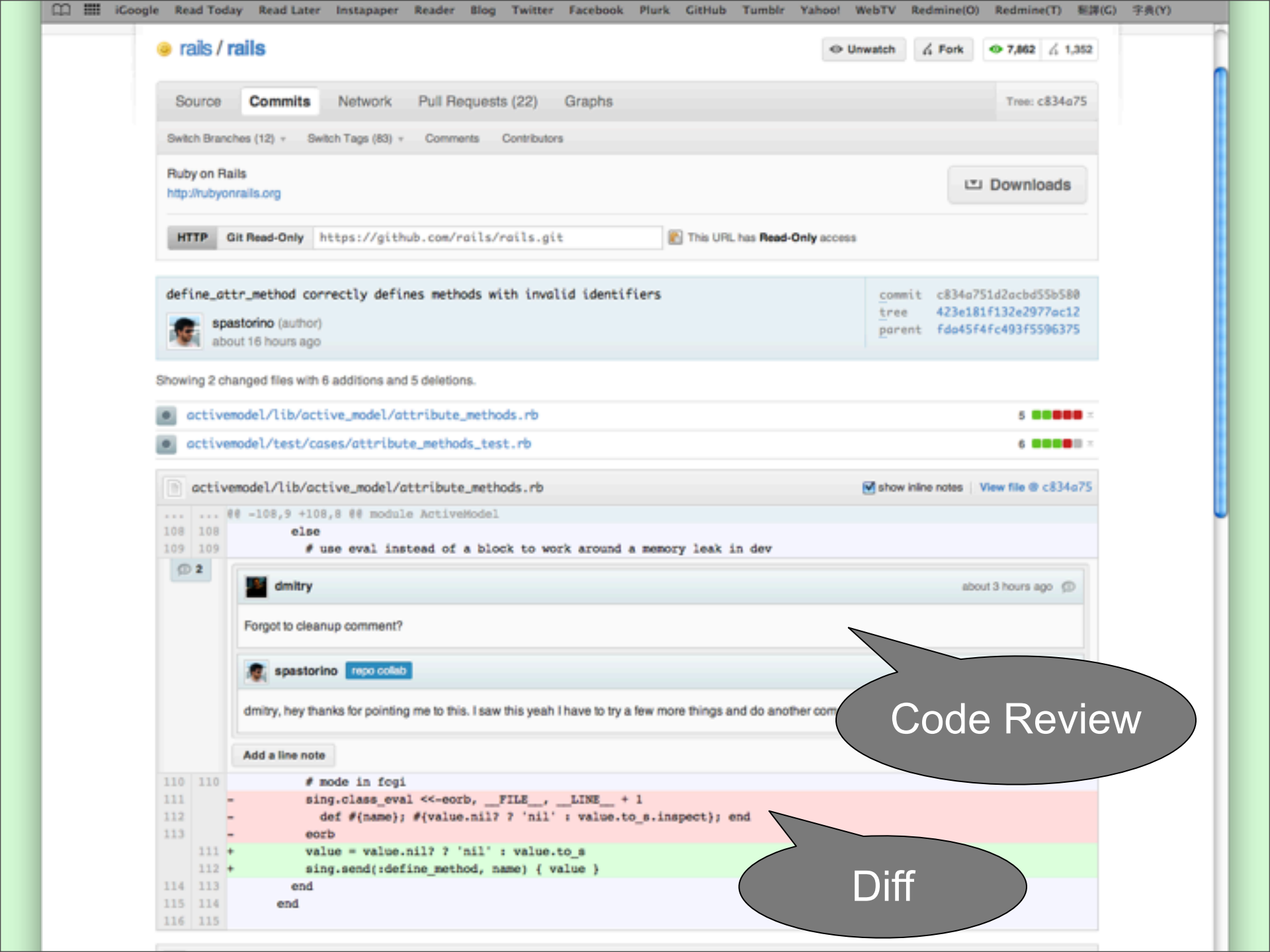
The problem is it's in the unicorn for a reason. Unicorn offers zero-downtime restarts. When you send it USR2 it forks and reloads rails/rack and then starts responding to users requests.

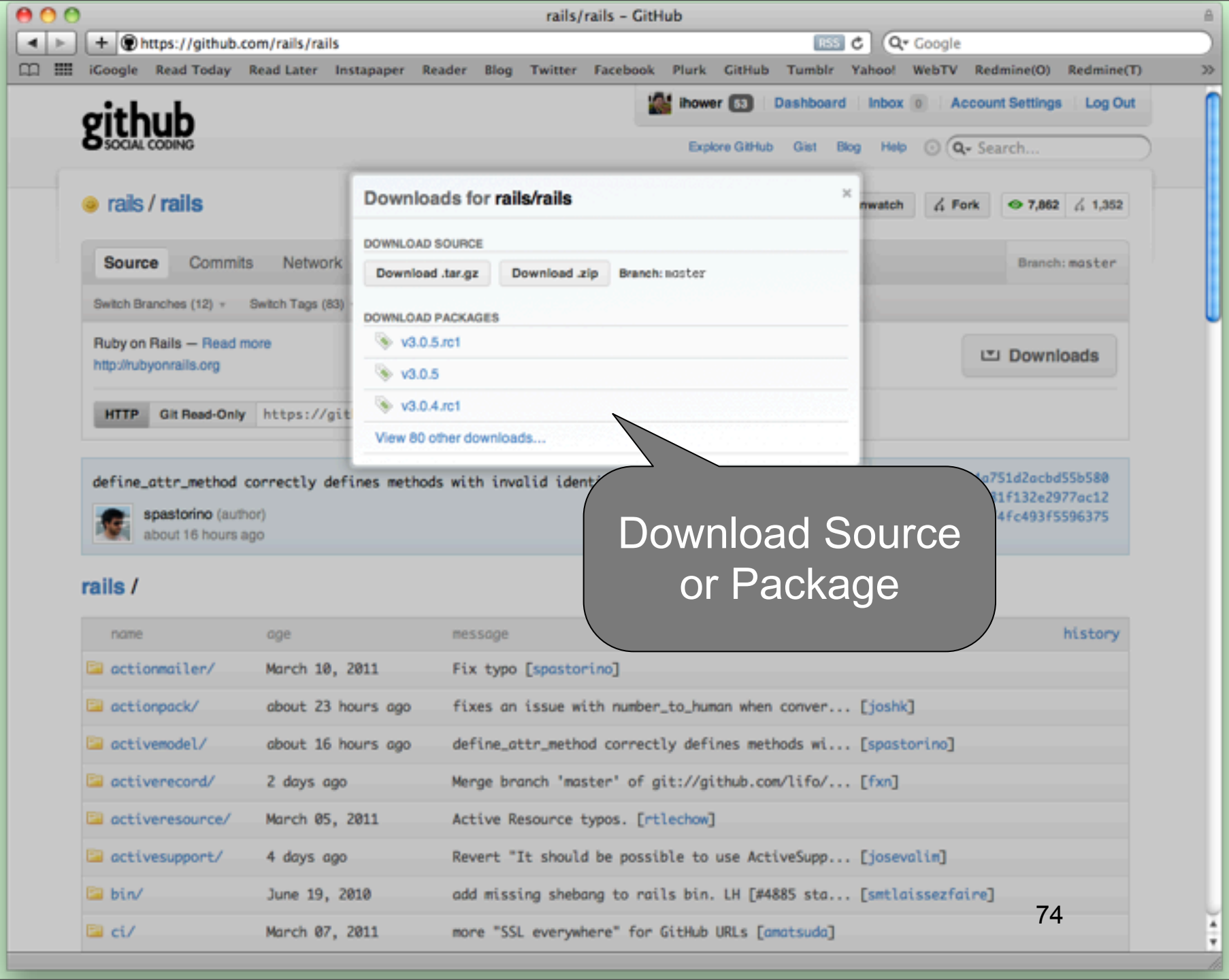
When Gemfile is changed it doesn't pick up changes. For example, I have working unicorn instance, then I add a gem, I deploy, send USR2, and unicorn couldn't restart. Because it loaded old Gemfile and have no idea about new gem. Practically it dumps this backtrace (https://gist.github.com/5c27d25d134e1cfa0eae) to the unicorn.stderr.log and continues to run old instance with the new current directory.

I propose Gem.refresh should reload Gemfile. What do you guys think?

elucid says:

I don't think this is actually a problem with Bundler. I've put up a simple repo at https://github.com/elucid/unicorn-reload to demonstrate how USR2 reloads with Unicorn work. If the old Gemfile is being reloaded after an upgrade, I think this probably has to do with the way that Unicorn was started in the first place. In particular, new Unicorn master processes will be started in the exact same directory that the original was. If you update your app code in place and nothing else changes, that should be fine. However if each deploy has its own directory you could be pointing at an old one. You can manually set this to something stable (e.g. a symlink that gets updated on each release) by calling `working_directory somedir` in your Unicorn config file.


rails / rails
Unwatch
Fork
7,862
1,352
Source
Commits
Network
Pull Requests (22)
Graphs
Tree: c834a75
Switch Branches (12)
Switch Tags (83)
Comments
Contributors
Ruby on Rails
http://rubyonrails.org
Downloads
HTTP
Git Read-Only
https://github.com/rails/rails.git
This URL has Read-Only access
define_attr_method correctly defines methods with invalid identifiers
spastorino (author)
about 16 hours ago
commit c834a751d2acbd55b580
tree 423e181f132e2977ac12
parent fda45f4fc493f5596375
Showing 2 changed files with 6 additions and 5 deletions.
activemodel/lib/active_model/attribute_methods.rb
activemodel/test/cases/attribute_methods_test.rb
activemodel/lib/active_model/attribute_methods.rb
show inline notes
View file @ c834a75
@@ -108,9 +108,8 @@ module ActiveModel
else
# use eval instead of a block to work around a memory leak in dev
2
dmitry
about 3 hours ago
Forgot to cleanup comment?
spastorino
repo collab
dmitry, hey thanks for pointing me to this. I saw this yeah I have to try a few more things and do another com
Add a line note
mode in fcgi
- sing.class_eval <<-eorb, __FILE__, __LINE__ + 1
- def #{name}; #{value.nil? ? 'nil' : value.to_s.inspect}; end
- eorb
+ value = value.nil? ? 'nil' : value.to_s
+ sing.send(:define_method, name) { value }
end
end
Code Review
Diff


rails/rails - GitHub
https://github.com/rails/rails
Google
iGoogle Read Today Read Later Instapaper Reader Blog Twitter Facebook Plurk GitHub Tumblr Yahoo! WebTV Redmine(O) Redmine(T)
github
SOCIAL CODING
ihower 53 Dashboard Inbox 0 Account Settings Log Out
Explore GitHub Gist Blog Help Search...
rails / rails
Fork 7,862 1,352
Source Commits Network
Branch: master
Switch Branches (12) Switch Tags (83)
Ruby on Rails — Read more
http://rubyonrails.org
Downloads
HTTP Git Read-Only
Downloads for rails/rails
DOWNLOAD SOURCE
Download .tar.gz Download .zip Branch: master
DOWNLOAD PACKAGES
v3.0.5.rc1
v3.0.5
v3.0.4.rc1
View 80 other downloads...
define_attr_method correctly defines methods with invalid ident
spastorino (author)
about 16 hours ago
Download Source or Package
rails /
name age message history
actionmailer/ March 10, 2011 Fix typo [spastorino]
actionpack/ about 23 hours ago fixes an issue with number_to_human when conver... [joshk]
activemodel/ about 16 hours ago define_attr_method correctly defines methods wi... [spastorino]
activerecord/ 2 days ago Merge branch 'master' of git://github.com/lifo/... [fxn]
activeresource/ March 05, 2011 Active Resource typos. [rtlechow]
activesupport/ 4 days ago Revert "It should be possible to use ActiveSupp... [josevalim]
bin/ June 19, 2010 add missing shebang to rails bin. LH [#4885 sta... [smtlaissezfaire]
ci/ March 07, 2011 more "SSL everywhere" for GitHub URLs [amatsuda]

Github pages

http://pages.github.com/

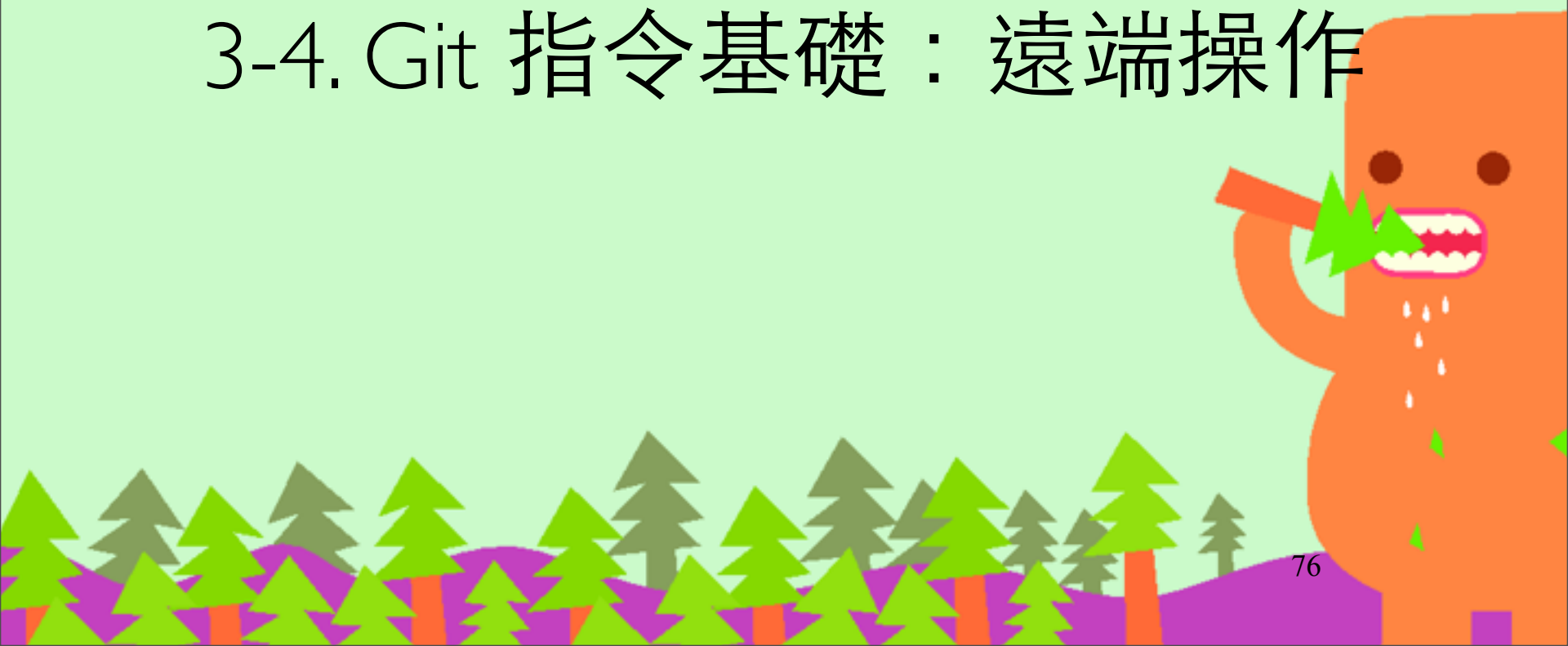
- 叫做 your.github.com 的 repository，Github 會自動建立 http://your.github.com 的靜態網頁。
- 擁有 gh-pages 的分支的 your_project，Github 會自動建立 http://your.github.com/your_project 靜態網頁。

Github pages

http://pages.github.com/

- 叫做 your.github.com 的 repository，Github 會自動建立 http://your.github.com 的靜態網頁。
- 擁有 gh-pages 的分支的 your_project，Github 會自動建立 http://your.github.com/your_project 靜態網頁。

3-4. Git 指令基礎：遠端操作

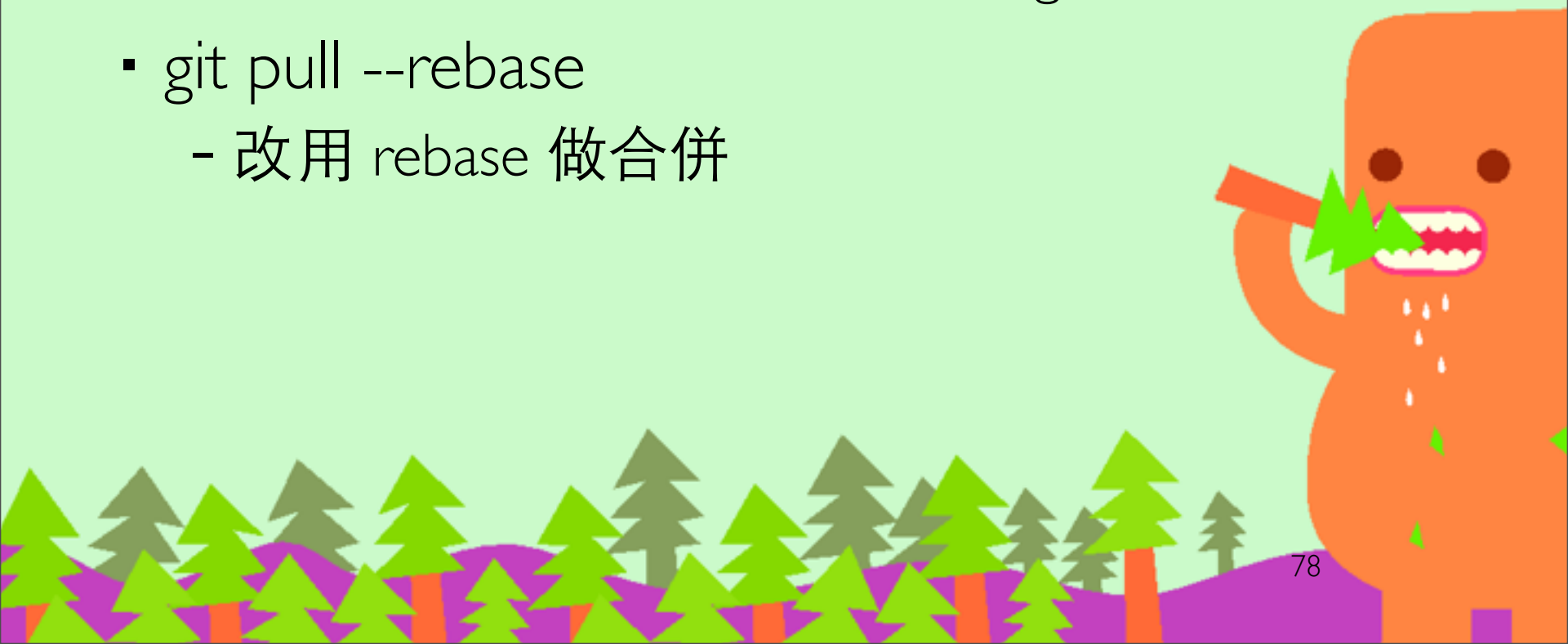


Push 到遠端 GitHub Reporitory

- 至 Github 開一個專案
- git remote add origin git@github.com:your_account/sandbox.git
- git push -u origin master
- 之後只需要 git push
  - 出現 ![rejected] 表示需要先做 git pull

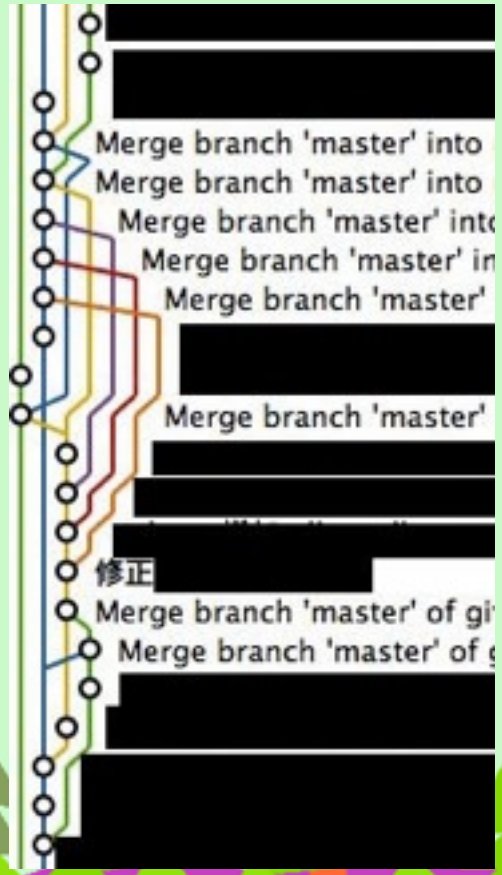
Pull 從遠端更新

- git pull origin master 或 git pull
  - git fetch 遠端的 branch
  - 然後與本地端的 branch 做 merge
- git pull --rebase
  - 改用 rebase 做合併



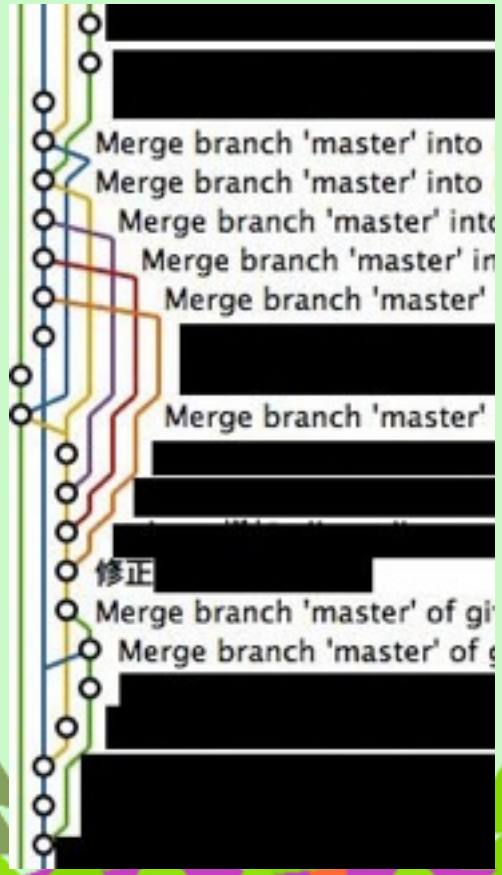
使用 git pull --rebase 可以避免無謂的 merge，讓線圖變乾淨

http://ihower.tw/blog/archives/3843



使用 git pull --rebase 可以避免無謂的 merge，讓線圖變乾淨

http://ihower.tw/blog/archives/3843



特別是很多人在同一個 branch 同時開發，例如主幹 master，容易發生這種情況!

git pull 何時用 merge? 何時用 rebase?

- 修改比較多，預期會有 conflict，用 merge
  - 是不是一開始就應該開 feature branch 來做呢?
- 修改範圍較小，不太預期有 conflict，建議可以用 --rebase

git pull 何時用 merge? 何時用 rebase?

- 修改比較多，預期會有 conflict，用 merge
  - 是不是一開始就應該開 feature branch 來做呢?
- 修改範圍較小，不太預期有 conflict，建議可以用 --rebase

Git Submodule

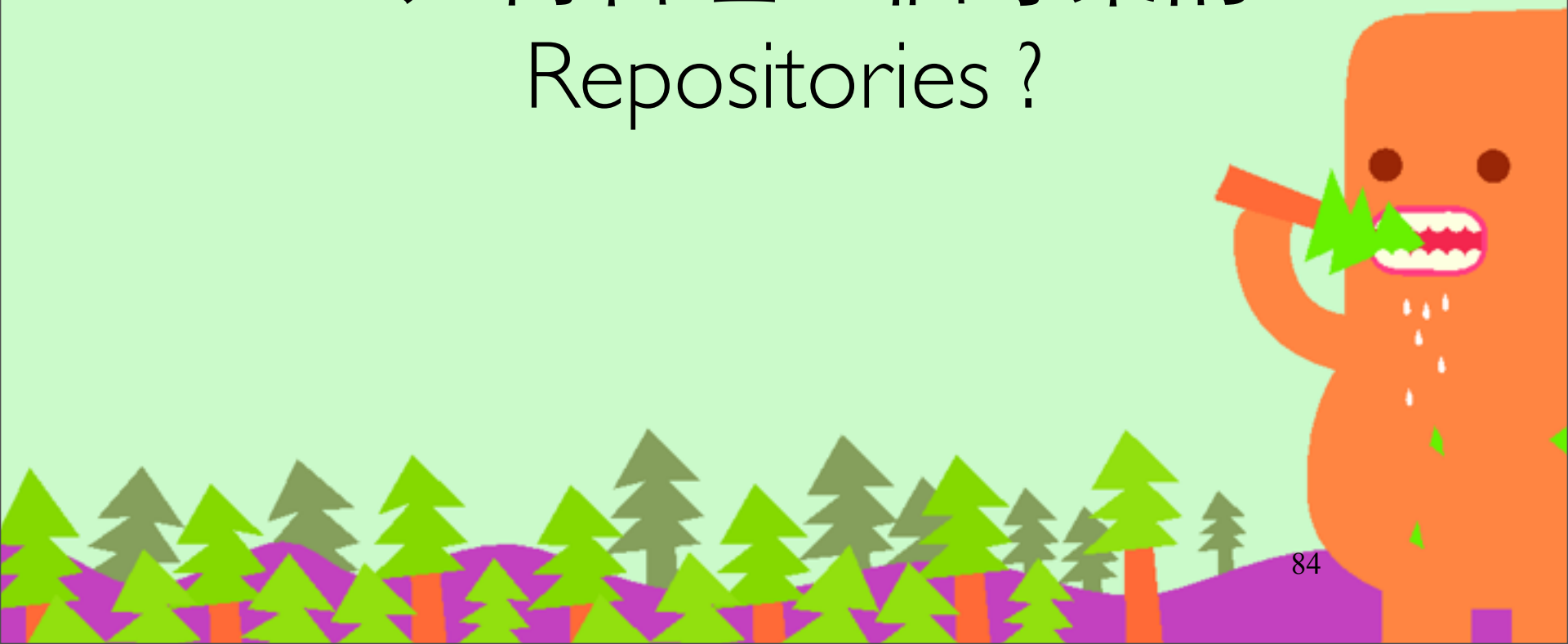
- Git 不像 SVN 可以只取出其中的子目錄
- 如何將第三方程式碼加進來? 像 SVN Externals?
- 參考 http://josephjiang.com/entry.php?id=342
- 有險路? http://josephjiang.com/entry.php?id=357
- 我沒用到
  - 因為 Ruby/Rails Develops 用不太到 XD
  - 我們用 Bundler 工具來管理 dependencies
  - 可以確保大家都跑相同的套件版本，又可以解決套件相依性問題

其他更多指令

- git tag
- git archive
- git bisect
- git blame
- git grep
- git show
- git reflog (預設會保留孤兒 commits 90天)
- git gc

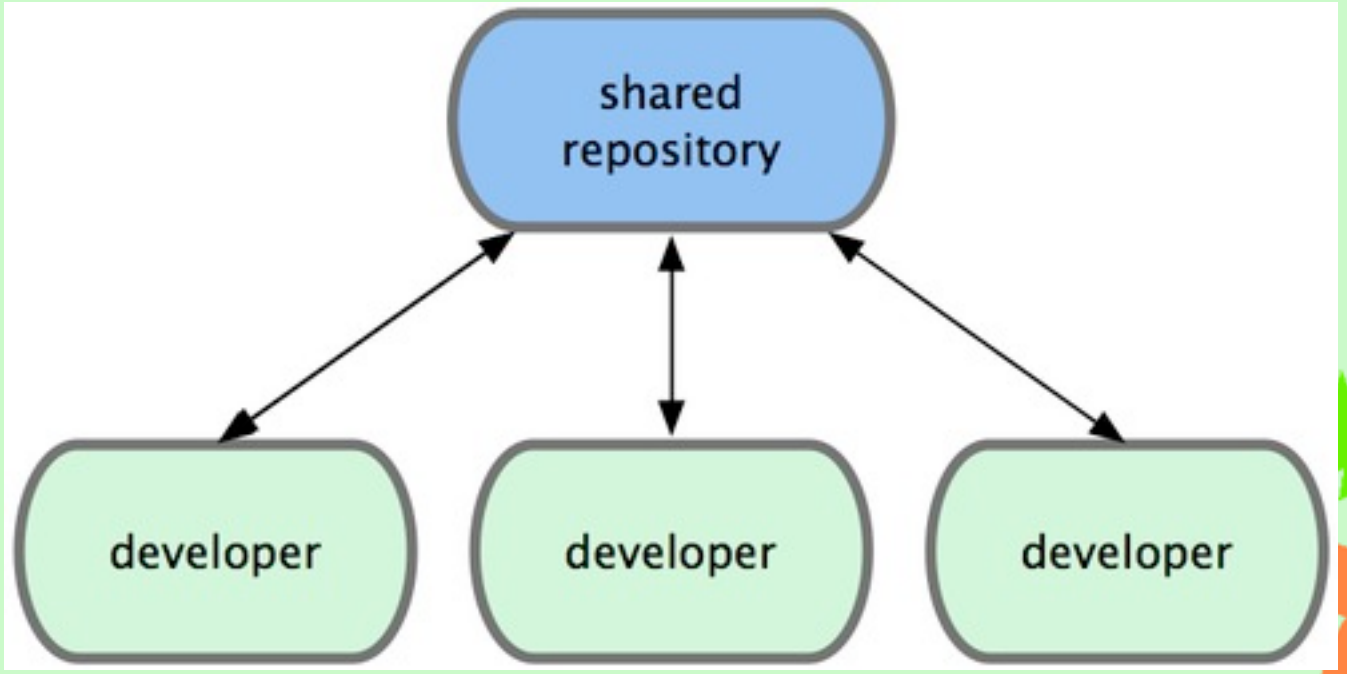
5. Git 開發模式及流程

1. 如何管理一個專案的 Repositories ?



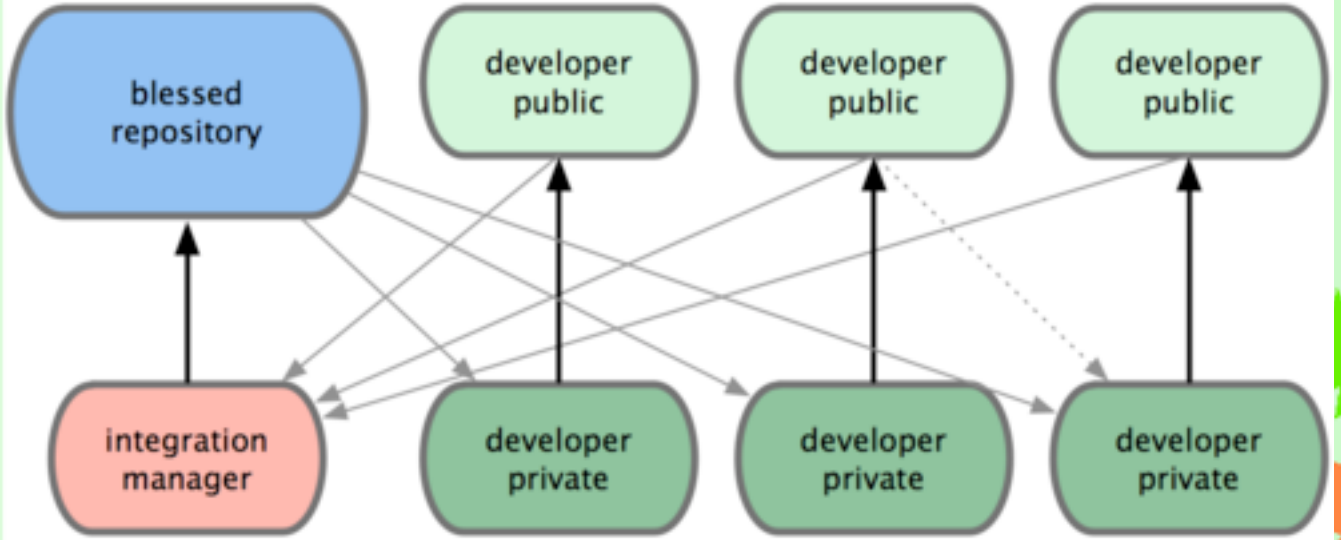
Centralized Workflow

團隊內部私有專案，大家都有權限 Push 到共用的 Repository。
使用 Branches 來管理流程(下述)



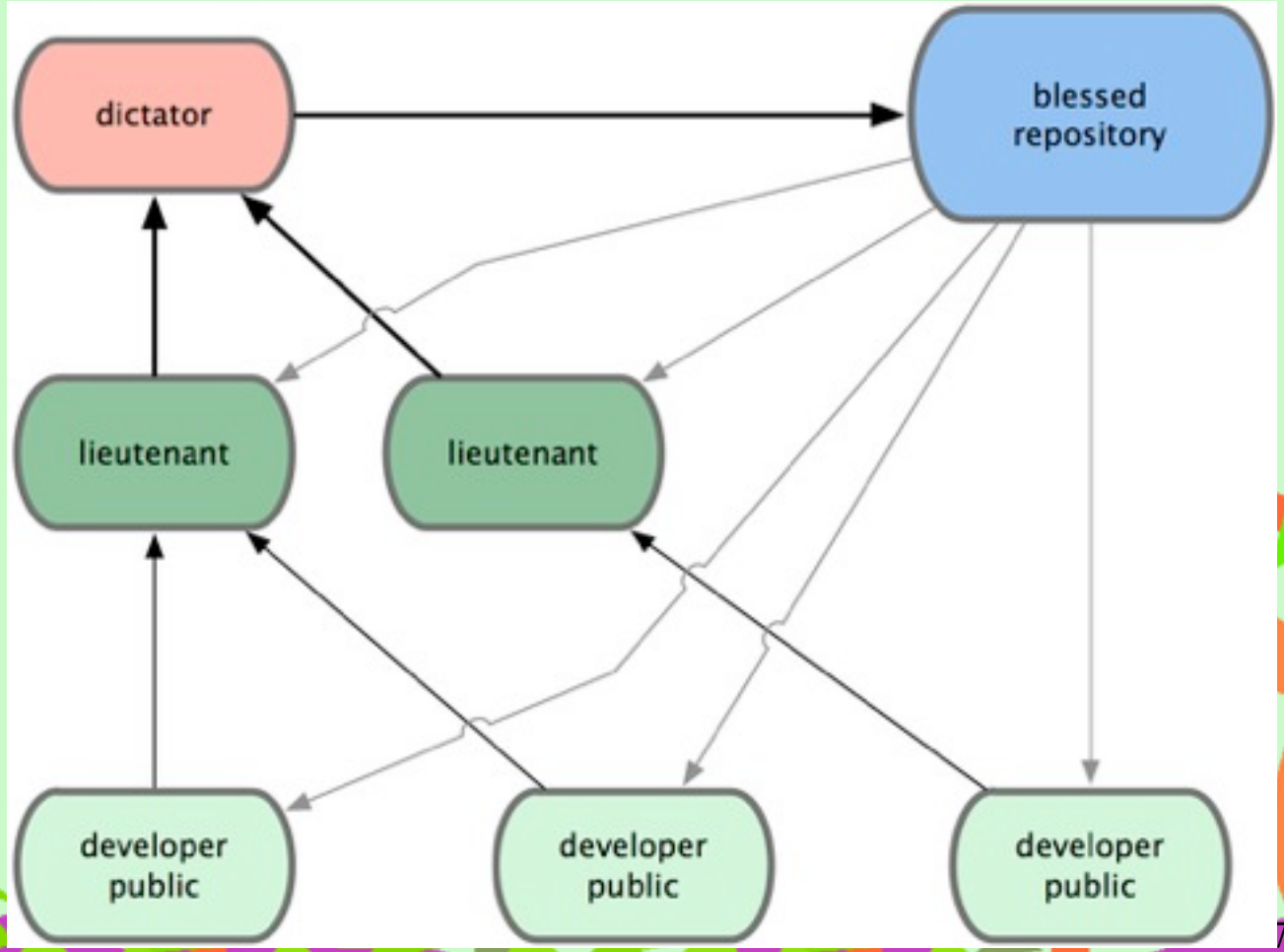
Integration-Manager Workflow

適合一般 Open Source 專案，只有少部分人有權限可以 Push 到 Repository，其他開發者用 request pull 請求合併。
例如 GitHub 提供的 Fork 和 Pull Request 功能



Dictator and Lieutenants Workflow

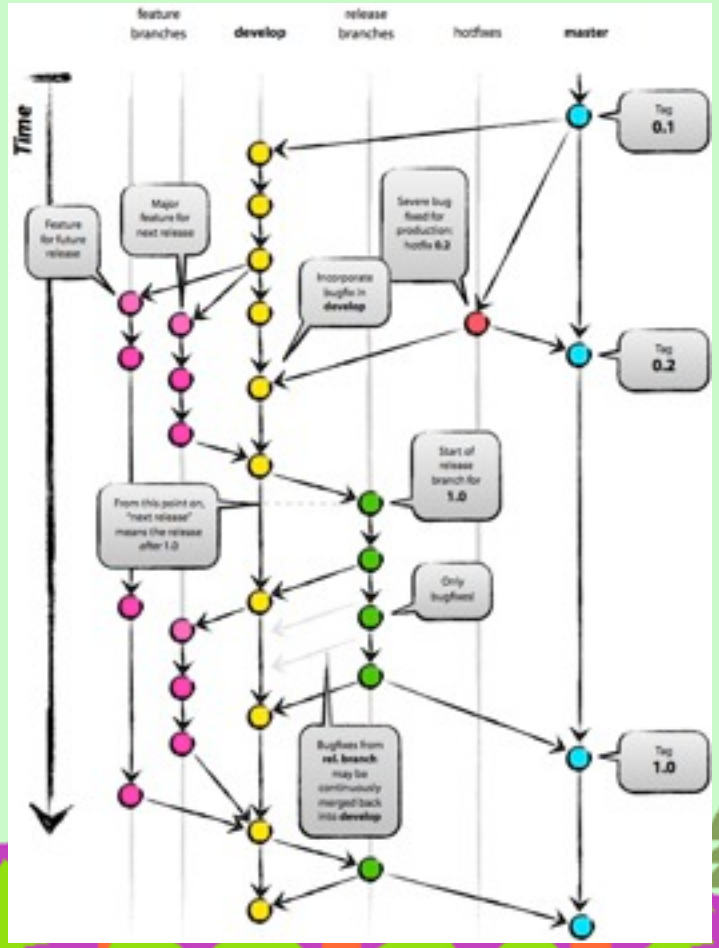
多層權限控管，適合大型 Open Source 專案，例如 Linux Kernel
又分成 patch 型或 pull request 型



2. 如何管理一個 Repository 的 Branches ?

Git flow: 一套如何管理 branches 的最佳實踐

http://nvie.com/posts/a-successful-git-branching-model/
http://ihower.tw/blog/archives/5140



兩個主要分支

- master: 永遠處在 production-ready 狀態
- develop: 最新的下次發佈開發狀態

三種支援性分支(1)

- Feature branches
  - 開發新功能或修 bugs
  - 從 develop 分支出來
  - 完成後 merge 回 develop
  - 如果開發時間較長，則需定期同步 develop 主幹的程式(初學可用 merge，建議改用 rebase)，不然最後會合併不回去。

三種支援性分支(1)

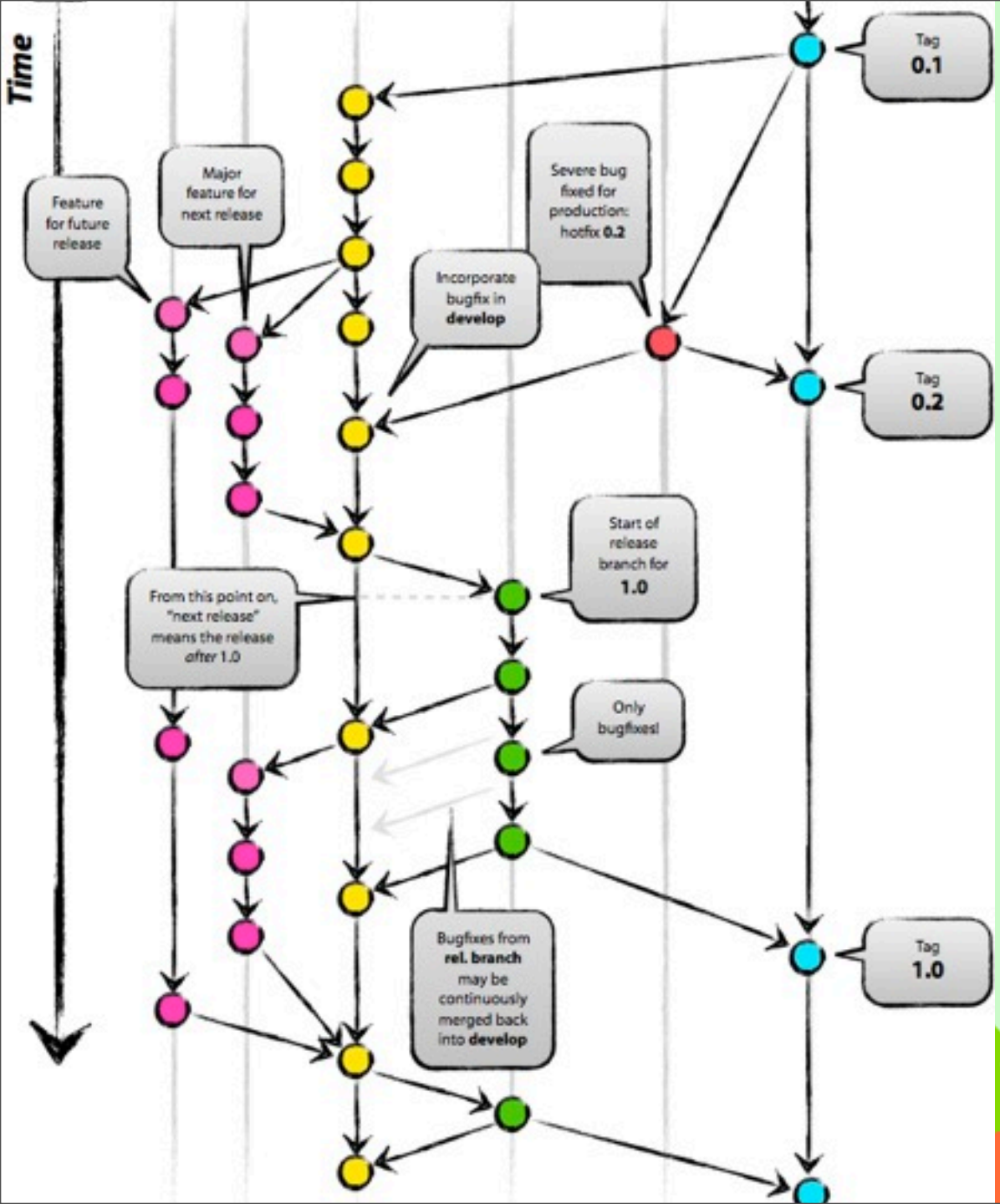
- Feature branches
  - 開發新功能或修 bugs
  - 從 develop 分支出來
  - 完成後 merge 回 develop
  - 如果開發時間較長，則需定期同步 develop 主幹的程式(初學可用 merge，建議改用 rebase)，不然最後會合併不回去。

三種支援性分支(2)

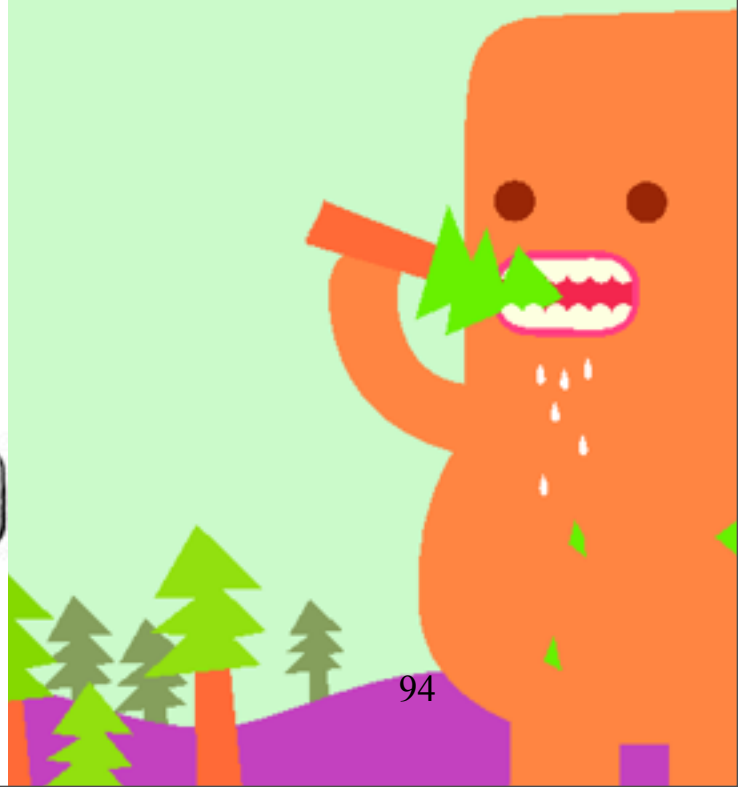
- Release branches
  - 準備要 release 的版本，只修 bugs
  - 從 develop 分支出來
  - 完成後 merge 回 master 和 develop

三種支援性分支(3)

- Hotfix branches
  - 等不及 release 版本就必須馬上修 master 趕著上線
  - 會從 master 分支出來
  - 完成後 merge 回 master 和 develop

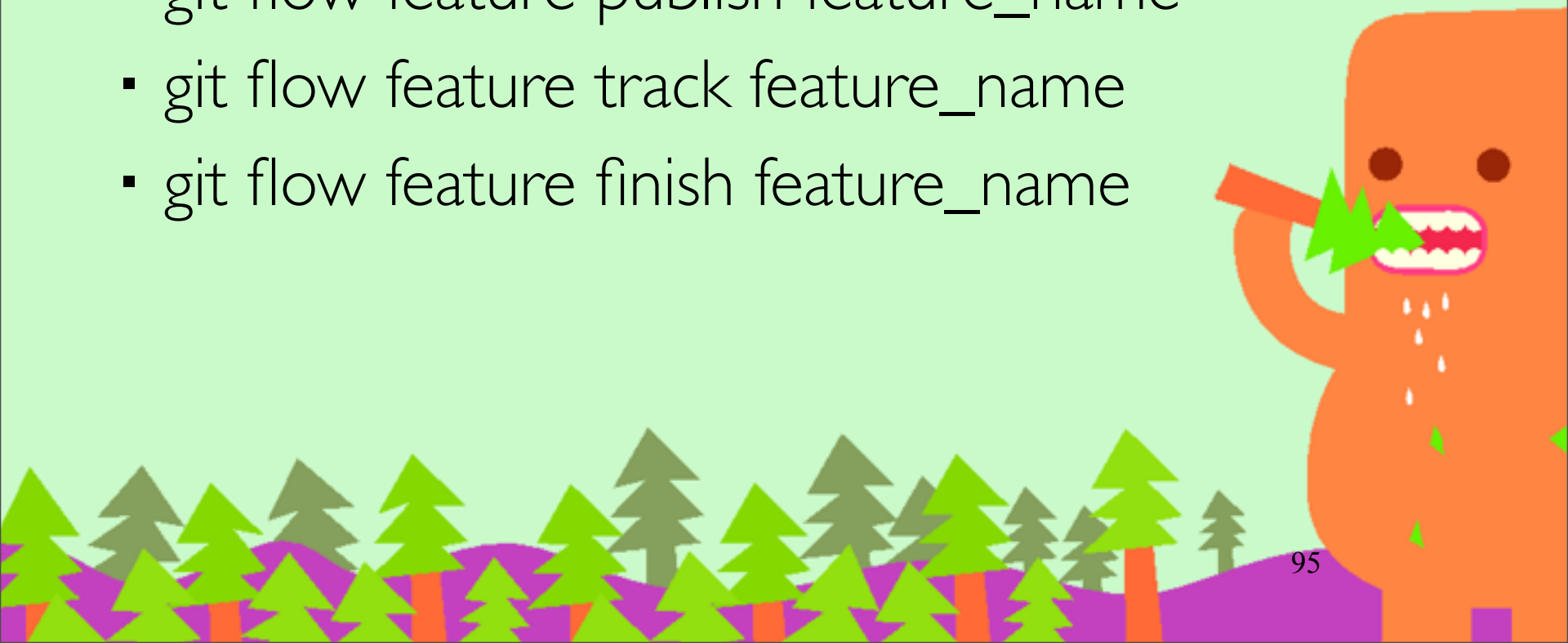


粉紅色: Feature branches
黃色: develop
綠色 release branches
紅色: hotfix branches
藍色: master



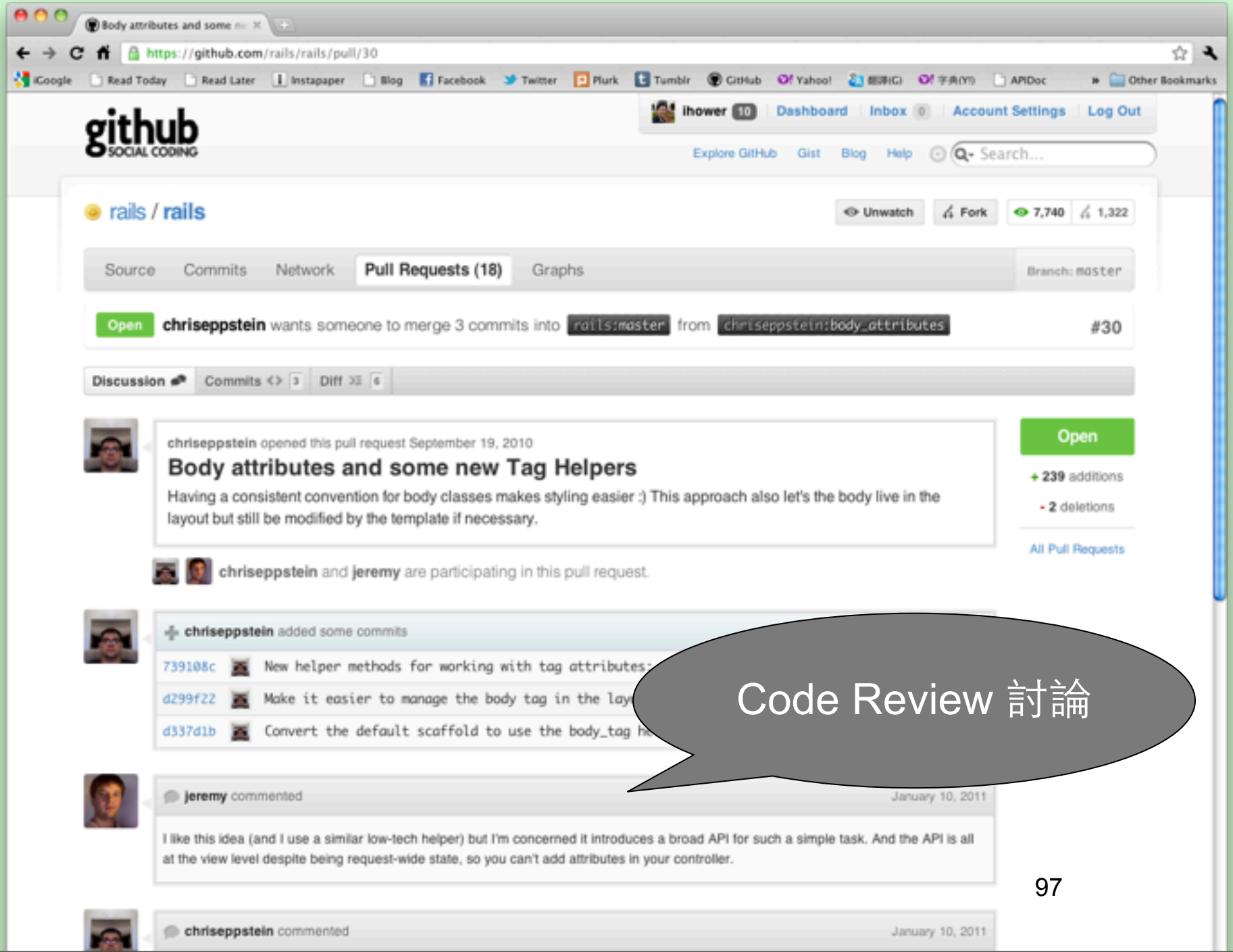
git-flow 工具

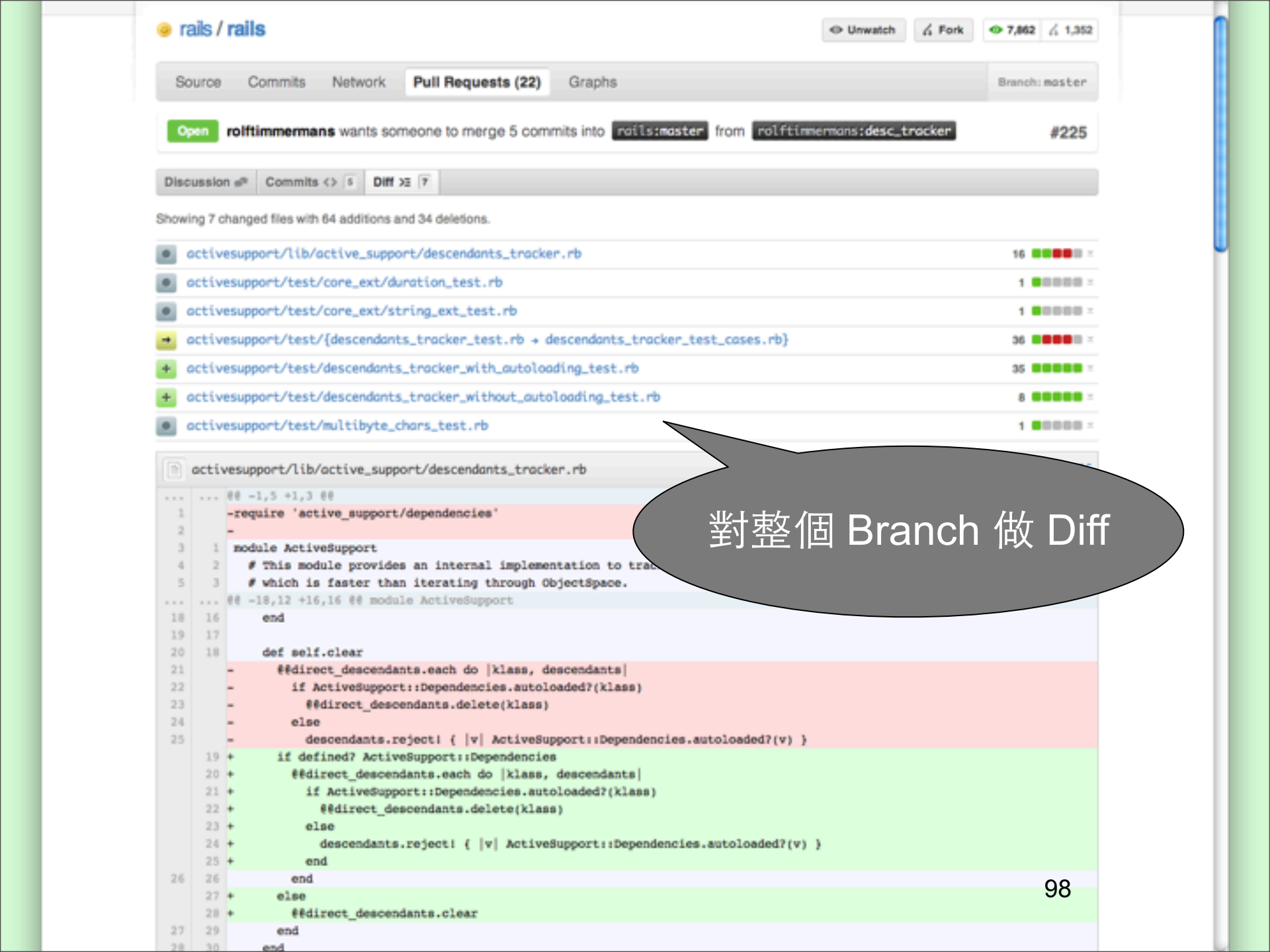
- git flow init
- git flow feature finish feature_name
- git flow feature publish feature_name
- git flow feature track feature_name
- git flow feature finish feature_name



Branch-Owner Policy

- 規定只有 project leaders 可以 commit/merge 進 develop branch。
- 規定只有 release team 可以管理 master branch。
- 其他開發者都開 topic branches，完成後發 pull request，大家做 code review 後，沒問題才 merge 進 develop。
- GitHub 的 pull request 功能強力推薦


Body attributes and some
https://github.com/rails/rails/pull/30
iGoogle
Read Today
Read Later
Instapaper
Blog
Facebook
Twitter
Plurk
Tumblr
GitHub
Yahoo!
APIDoc
Other Bookmarks
github
SOCIAL CODING
ihower 10
Dashboard
Inbox 0
Account Settings
Log Out
Explore GitHub
Gist
Blog
Help
Search...
rails / rails
Unwatch
Fork
7,740
1,322
Source
Commits
Network
Pull Requests (18)
Graphs
Branch: master
Open
chriseppstein wants someone to merge 3 commits into rails:master from chriseppstein:body_attributes
#30
Discussion
Commits 3
Diff 6
chriseppstein opened this pull request September 19, 2010
Body attributes and some new Tag Helpers
Having a consistent convention for body classes makes styling easier :) This approach also let's the body live in the layout but still be modified by the template if necessary.
chriseppstein and jeremy are participating in this pull request.
Open
+ 239 additions
- 2 deletions
All Pull Requests
chriseppstein added some commits
739108c
New helper methods for working with tag attributes
d299f22
Make it easier to manage the body tag in the lay
d337d1b
Convert the default scaffold to use the body_tag
Code Review 討論
jeremy commented
January 10, 2011
I like this idea (and I use a similar low-tech helper) but I'm concerned it introduces a broad API for such a simple task. And the API is all at the view level despite being request-wide state, so you can't add attributes in your controller.
chriseppstein commented
January 10, 2011


rails / rails
Unwatch
Fork
7,862
1,352
Source
Commits
Network
Pull Requests (22)
Graphs
Branch: master
Open
rolftimmermans wants someone to merge 5 commits into rails:master from rolftimmermans:desc_tracker
#225
Discussion
Commits 5
Diff 7
Showing 7 changed files with 64 additions and 34 deletions.
activesupport/lib/active_support/descendants_tracker.rb 16
activesupport/test/core_ext/duration_test.rb 1
activesupport/test/core_ext/string_ext_test.rb 1
activesupport/test/{descendants_tracker_test.rb → descendants_tracker_test_cases.rb} 36
activesupport/test/descendants_tracker_with_autoloading_test.rb 35
activesupport/test/descendants_tracker_without_autoloading_test.rb 8
activesupport/test/multibyte_chars_test.rb 1
activesupport/lib/active_support/descendants_tracker.rb
@@ -1,5 +1,3 @@
-require 'active_support/dependencies'
-
module ActiveSupport
This module provides an internal implementation to tra
which is faster than iterating through ObjectSpace.
@@ -18,12 +16,16 @@ module ActiveSupport
end
def self.clear
-@@direct_descendants.each do |klass, descendants|
-if ActiveSupport::Dependencies.autoloaded?(klass)
-@@direct_descendants.delete(klass)
-else
-descendants.reject! { |v| ActiveSupport::Dependencies.autoloaded?(v) }
+if defined? ActiveSupport::Dependencies
+@@direct_descendants.each do |klass, descendants|
+if ActiveSupport::Dependencies.autoloaded?(klass)
+@@direct_descendants.delete(klass)
+else
+descendants.reject! { |v| ActiveSupport::Dependencies.autoloaded?(v) }
+end
end
+else
+@@direct_descendants.clear
end
end
end
對整個 Branch 做 Diff

the maintainer workflow for git itself

http://www.kernel.org/pub/software/scm/git/docs/gitworkflows.html

- **master** tracks the commits that should go into the next release
- **maint** tracks the commits that should go into the next "maintenance release"
- **next** is intended as a testing branch for topics being tested for stability for master.
- **pu** (proposed updates branches) is an integration branch for things that are not quite ready for inclusion yet
- **topic branches** for any nontrivial feature

Ruby on Rails 的 workflow

- 例如：Rails 目前的版本是 3.0.5
- master 是開發版本，指向下一次主要 release 3.1
- stable branches
  - 2-2-stable 已停止維護
  - 2-3-stable bug fixes
  - 3-0-stable bug fixes

One more thing...

git-svn

- 公司用 SubVersion 的朋友們，沒關係。今天就可以開始改用 Git 體驗 Git 的快感!
- git svn clone http://url/svn/trunk/ project_name
- (normal git operations)
- git svn dcommit

參考資料

- http://progit.org/book/
- http://git-scm.com
- http://help.github.com/
- http://gitimmersion.com
- http://www.gitready.com/
- http://gitref.org
- http://ihower.tw/blog/category/git
- Pragmatic Version Control Using Git, Pragmatic

Thank you.

請 google “ihower” 就可以找到我及這份投影片。