

我们需要工具



# Sublime Text

一个代码编辑器，也是你的新陪伴



# Terminal终端 (Bash)

不用怕命令行 (command line)



# Git 和 GitHub

版本控制，协作



# Your turn!

上Setup



## 不同的命名

- 命令提示符 (Command prompt)
- 控制台 (Console)
- 终端 (Terminal)



# 基本命令



## 我在哪里？

- 1、在提示符之前查找目录名，`→` 之后
- 2、或print当前目录的路径

```
pwd
```



```
>_
```



```
seb — seb@Macbook — ~ — zsh — 80x20
[→ ~ pwd [2.3.1]
/Users/seb
[→ ~ [2.3.1]
```



这就是你的 `$HOME` 目录>-

## 显示当前目录里的内容

ls (或者ll, `ls -lh` 的别名)

```
seb — seb@Macbook — ~ — zsh — 80x20
[2.3.1]
→ ~ ls
Applications      La Loco           Sites
Applications (Par Le Wagon
Creative Cloud Fil Library
Desktop           Movies
Documents         Music
Downloads         Pictures
FullStack         Public
[2.3.1]
→ ~
```



# 进入目录

`cd <FOLDER_NAME>`

```
tmp — seb@Macbook — ~/tmp — zsh — 80x20  
[→ ~ ls [2.3.1] ]  
Applications          La Loco               Sites  
Applications (Parallels) Le Wagon              code  
Creative Cloud Files  Library              floobits  
Desktop               Movies                perso  
Documents             Music                 ruby-101  
Downloads             Pictures              tmp  
FullStack             Public  
[→ ~ cd tmp [2.3.1] ]  
[→ tmp pwd [2.3.1] ]  
/Users/seb/tmp  
[→ tmp [2.3.1]
```



# 返回上级目录

cd ..

```
seb — seb@Macbook — ~ — -zsh — 80x20
[→ ~ ls [2.3.1] ]
Applications          La Loco               Sites
Applications (Parallels) Le Wagon              code
Creative Cloud Files  Library              floobits
Desktop               Movies                perso
Documents             Music                 ruby-101
Downloads             Pictures              tmp
FullStack             Public

[→ ~ cd tmp [2.3.1] ]
[→ tmp pwd [2.3.1] ]
/Users/seb/tmp
[→ tmp cd .. [2.3.1] ]
[→ ~ pwd [2.3.1] ]
/Users/seb
[→ ~ [2.3.1] ]
```



# 创建目录

`mkdir <NEW_FOLDER>`

```
terminal-101 — seb@Macbook — ~/terminal-101 — zsh — 80x20
[→ ~ mkdir terminal-101 [2.3.1] ]
[→ ~ ls [2.3.1] ]
Applications          La Loco               Sites
Applications (Parallels) Le Wagon              code
Creative Cloud Files  Library              floobits
Desktop               Movies               perso
Documents             Music               ruby-101
Downloads             Pictures            terminal-101
FullStack             Public              tmp
[→ ~ cd terminal-101 [2.3.1] ]
[→ terminal-101 pwd [2.3.1] ]
/Users/seb/terminal-101
[→ terminal-101 [2.3.1] ]
```



# 创建文件

`touch <FILE_NAME>`

```
terminal-101 — seb@Macbook — ~/terminal-101 — zsh — 80x20
[→ terminal-101 touch hello_world.rb [2.3.1] ]
[→ terminal-101 ll [2.3.1] ]
total 0
-rw-r--r--  1 seb  staff  0 Sep  4 14:06 hello_world.rb
[→ terminal-101 [2.3.1]
```



## 把目录或者文件移动到新的位置

`mv <FILE_NAME> <FOLDER_NAME>`

```
destination — seb@Macbook — ../destination — zsh — 80x20
[→ terminal-101 ll [2.3.1] ]
total 0
-rw-r--r--  1 seb  staff   0 Sep  4 14:06 hello_world.rb
[→ terminal-101 mkdir destination [2.3.1] ]
[→ terminal-101 mv hello_world.rb destination [2.3.1] ]
[→ terminal-101 ll [2.3.1] ]
total 0
drwxr-xr-x  3 seb  staff  102 Sep  4 14:07 destination
[→ terminal-101 cd destination [2.3.1] ]
[→ destination ll [2.3.1] ]
total 0
-rw-r--r--  1 seb  staff   0 Sep  4 14:06 hello_world.rb
[→ destination [2.3.1]
```



## 重命名文件或者目录

`mv <FILE_NAME> <NEW_FILENAME>`

```
destination — seb@Macbook — ../destination — zsh — 80x20
[→ destination ll [2.3.1] ]
total 0
-rw-r--r--  1 seb  staff  0 Sep  4 14:06 hello_world.rb
[→ destination mv hello_world.rb goodbye_world.rb [2.3.1] ]
[→ destination ll [2.3.1] ]
total 0
-rw-r--r--  1 seb  staff  0 Sep  4 14:06 goodbye_world.rb
→ destination [2.3.1]
```





# 使用Sublime Text打开当前目录

使用stt快捷键开启当前的目录

```
terminal-101 — seb@Macbook — ~/terminal-101 — -zsh — 80x20
[→ destination pwd [2.3.1] ]
/Users/seb/terminal-101/destination
[→ destination cd .. [2.3.1] ]
[→ terminal-101 stt [2.3.1] ]
[→ terminal-101 [2.3.1] ]
```



# 一次显示整个文件

cat <FILE\_NAME>

```
terminal-101 — seb@Macbook — ~/terminal-101 — zsh — 80x20
[→ destination pwd [2.3.1] ]
/Users/seb/terminal-101/destination
[→ destination cd .. [2.3.1] ]
[→ terminal-101 stt [2.3.1] ]
[→ terminal-101 cat destination/goodbye_world.rb [2.3.1] ]
puts "Hello World!"

puts "Goodbye World..."
[→ terminal-101 [2.3.1] ]
```



# 还有更多的命令行!

Cheatsheet



>\_

# 我们属于知识型员工

我们创建及编辑 文件（文本、图片等）

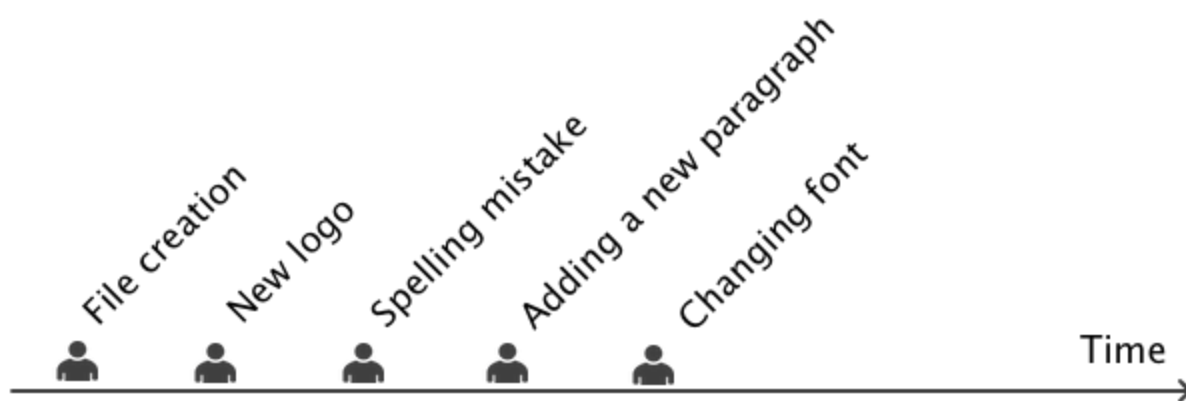


## 日常工作流程

- 1、创建文件
- 2、保存文件
- 3、编辑文件
- 4、再次保存文件
- 5、等



# 文件生命



# 人工版本控制

如何追踪文件的不同版本



Report (Christmas added).doc



Report (final version).doc



Report (John version).doc



Report (REAL FINAL VERSION).doc



Report.doc



## 如何自动化？

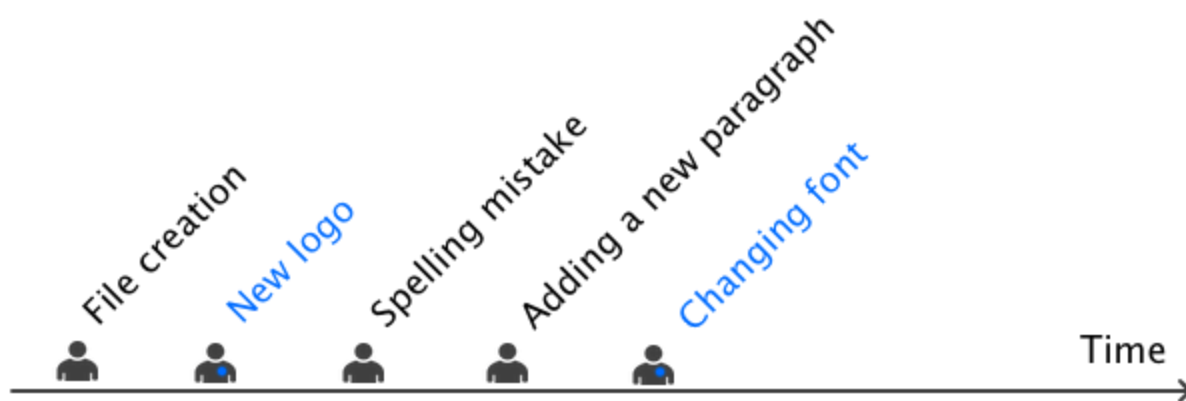
对于每个文档版本，需要知道：

- 1、文件**何时**被修改
- 2、修改的**内容**
- 3、**为何**修改





还有：团队



## 这是另一个问题

对于每个文档版本，需要知道：

- 1、文件何时被修改
- 2、修改的内容
- 3、为何修改
- 4、谁在修改



## 简而言之

我们需要一个工具，能够：

- 追踪文档版本
- 保存文档更改的历史
- 促进团队合作



该工具是



git



# Git常用命令



>\_

## 从哪里开始?

```
# 从现有的repository (例如: GitHub上) 开始  
git clone <github_ssh_clone_url>
```

```
# 或从头开始  
mkdir new_project  
cd new_project  
git init
```



## Status

git可以显示工作目录和暂存区的状态，文件是否有更改

```
git status
```



# Commit

`commit` (类似于文件夹的快照) 有3个步骤

*# 首先, 查看哪些文件已被更改过*  
`git status`

*# 其次, 将文件内容添加到索引 (将修改添加到暂存区)*  
`git add <file_1_which_has_been_modified>`  
`git add <file_2_which_has_been_modified>`

*# 可以再次查看暂存区*  
`git status`

*# 最后, 更改记录 (提交) 到存储库*  
`git commit --message "A meaningful message about this change"`





## Diff

显示更改记录（提交）到存储库，此命令比较的是工作目录中当前文件和暂存区域快照之间的差异

```
git diff  
git diff <a_specific_file_or_folder>
```



## Log

显示提交日志信息

```
git log
```

```
# 在~/.gitconfig中有更多fancy的命令  
git lg
```



>\_

## Live-code: git init

接下来，我们将创建一个项目并追踪该项目

```
mkdir -p ~/code/$GITHUB_USERNAME/git-101  
cd ~/code/$GITHUB_USERNAME/git-101  
git init  
ls -a # 创建了一个.git隐藏文件夹
```



## Live-code: 第一个commit

接下来，我们将创建一个 `index.html` 文件并编写一些HTML代码

```
touch index.html  
stt # 编写一些HTML代码
```

然后再commit!

```
git status # 查看仓库状态  
git add index.html  
git status # 将文件添加到暂存区，可以commit了  
git commit -m "Basic HTML content for home page"  
git status
```



## Live-code: 第二个commit

接下来，我们在项目中添加一个图片

```
curl https://raw.githubusercontent.com/lewagon/karr-image:  
stt # 添加到HTML代码中
```

然后再commit!

```
git status  
git diff index.html # 哪些内容已被更改过?  
git add index.html  
git add logo.png  
git status  
git commit -m "Adding logo to home page"  
git status  
git log # 查看commit历史
```



我们将使用push来推送

```
hub create  
hub browse # can you see your new repo on Github?  
git status # check again - anything to be pushed?  
git push origin master
```



# Remote



# Fork和clone



Le Wagon

[git-101-boilerplate](#)

Simple boilerplate to show how to use git with Github on @lewagon setup day.

Updated 10 minutes ago



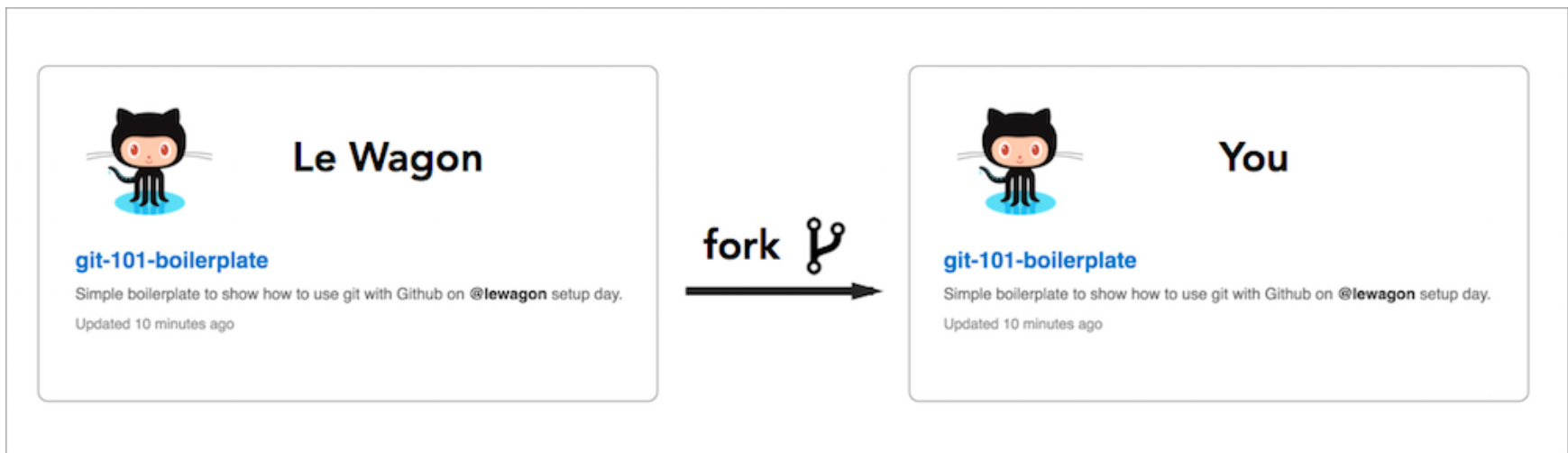
You



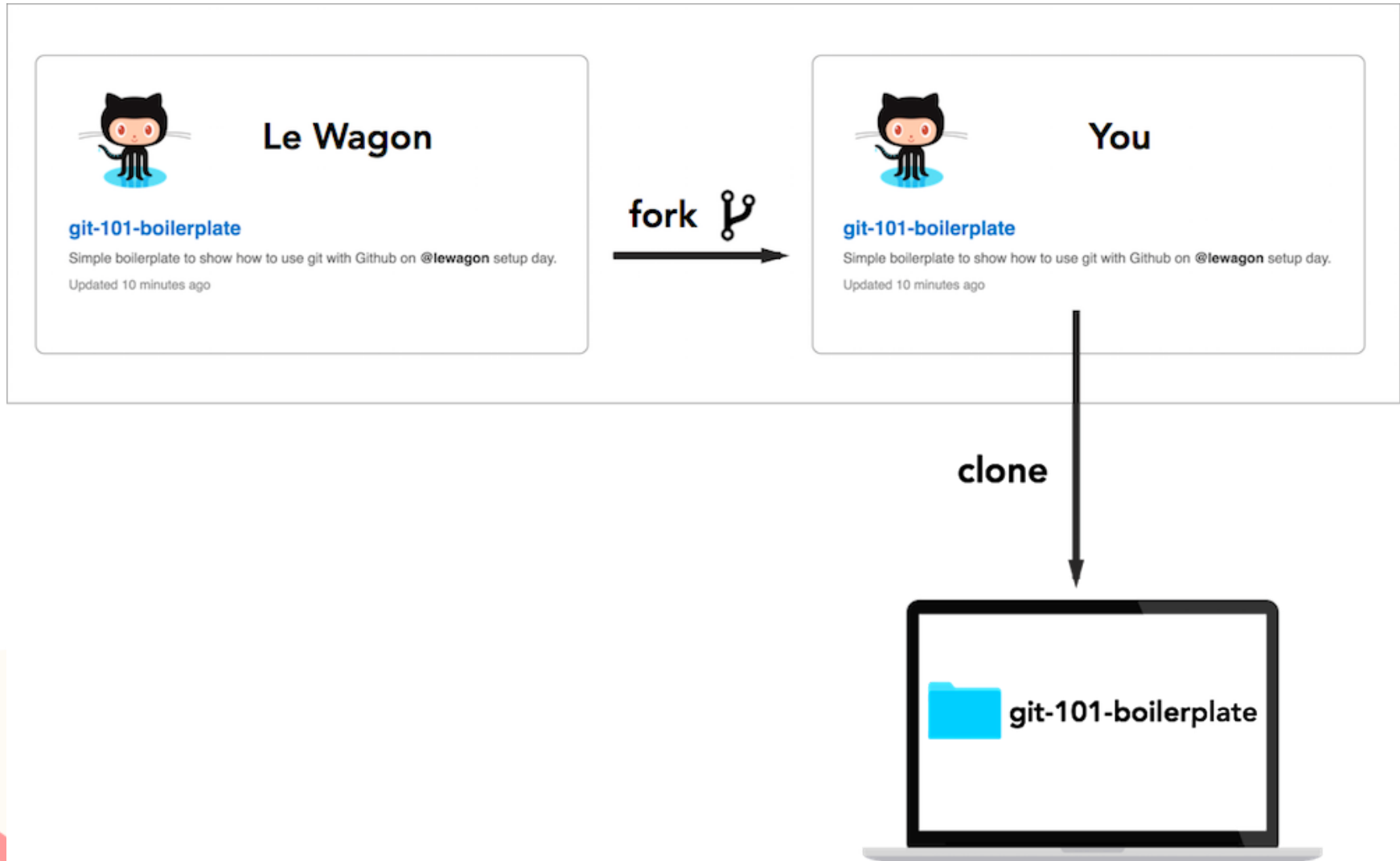
>\_



# Fork和clone



# Fork和clone



## 将本地分支的更新，推送到远程主机

一旦完成commit，把更改的内容推送（push）到Github上

# 常规的命令

```
git push <remote> <branch>
```

# 我们将使用

```
git push origin master
```



## Live-code: fork并clone一个remote

1. 首先, fork [Le Wagon's git 101 boilerplate](#)
2. 其次, clone到本地, 使用以下代码:

```
cd ~/code/$GITHUB_USERNAME  
git clone git@github.com:$GITHUB_USERNAME/git-101-boilerp  
git status # git已进行追踪
```



## Live-code: commit并push

接下来，我们将更改一些内容，commit，和push

```
stt # 更改HTML代码  
git add index.html  
git commit -m "adding some custom text"  
git status  
git push origin master # Push到 Github上
```

检查该项目已在Github上已被更新过



# 高级Git

在接下来的课程中，我们将学到git如何：

- 解决冲突（conflicts）
- 帮助协作（通过使用branches）
- 生产环境部署（production deployment）（使用多个 remote）



## Learn.lewagon.com Demo

- 课程系列
- 课程
- 同班同学
- 当天的buddy
- Exercises