INICIO

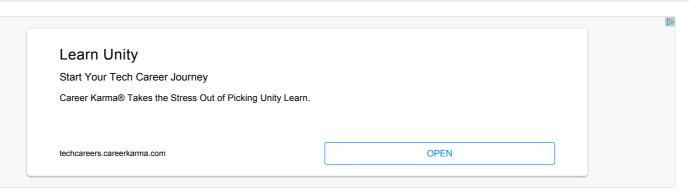
LIBROS

TUTORIALES

EVENTOS

F

BUSCAR



Pro Git, el libro oficial de Git

2.7. Consejos y trucos

Antes de que terminemos este capitulo de Git básico, unos pocos trucos y consejos que harán de tu experiencia con Git más sencilla, fácil, o más familiar. Mucha gente usa Git sin usar ninguno de estos consejos, y no nos referiremos a ellos o asumiremos que los has usado más tarde en el libro, pero probablemente debas saber como hacerlos.

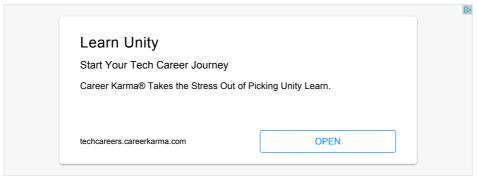
2.7.1. Autocompletado

Si usas el shell Bash, Git viene con un buen script de autocompletado que puedes activar. Descarga el código fuente de Git y busca en el directorio contrib/completion; ahí debe haber un archivo llamado git-completion.bash. Copia este fichero en tu directorio home y añade esto a tu archivo .bashrc:

source ~/.git-completion.bash

Si quieres que Git tenga automáticamente autocompletado para todos los usuarios, copia este script en el directorio /opt/local/etc/bash_completion.d en sistemas Mac, o en el directorio /etc/bash_completion.d/ en sistemas Linux. Este es un directorio de scripts que Bash cargará automáticamente para proveer de autocompletado.

Si estas usando Windows con el Bash de Git, el cual es el predeterminado cuando instalas Git en Windows con msysGit, el autocompletado debería estar preconfigurado.



Presiona el tabulador cuando estés escribiendo un comando de Git, y deberían aparecer un conjunto de sugerencias para que escojas:



En este caso, escribiendo git co y presionando el tabulador dos veces sugiere commit y config. Añadiendo m y pulsando el tabulador completa git commit automáticamente.

Esto también funciona con optiones, que probablemente es más útil. Por ejemplo, si quieres ejecutar git log y no recuerdas una de las opciones, puedes empezar a escribirla y presionar el tabulador para ver que coincide:

Indice de contenidos

1. Empezando

Capítulo 2. Fundamentos de Git

- 2.1. Obteniendo un repositorio Git
- 2.2. Guardando cambios en el repositorio
- 2.3. Viendo el histórico de confirmaciones
- 2.4. Deshaciendo cosas
- 2.5. Trabajando con repositorios remotos
- 2.6. Creando etiquetas
- 2.7. Consejos y trucos
- 2.8. Resumen
- 3. Trabajando con ramas en Git
- 4. Git en un servidor
- 5. Git en entornos distribuidos
- 6. Las herramientas de Git
- 7. Personalizando Git
- 8. Git y otros sistemas
- 9. Funcionamiento interno de Git

and/or the others like AngularIS, ReactIS, jQuery, etc.

We can write for you the application front-ends or the whole application with back-end in SQL/PHP/ASP.NET/JAVA/NodeJS.

```
$ git log --s<tab>
--shortstat --since= --src-prefix= --stat --summary
```

Es un pequeño truco que puede ahorrarte tiempo y tener que leerte mucha documentación.

2.7.2. Alias de Git

Git no infiere tu comando si lo escribes parcialmente. Si no quieres escribir el texto entero de cada uno de los comandos de Git, puedes establecer fácilmente un alias para cada comando usando git config. Aquí hay un par de ejemplos que tal vez quieras establecer:

```
$ git config --global alias.co checkout
$ git config --global alias.br branch
$ git config --global alias.ci commit
$ git config --global alias.st status
```

Esto significa que, por ejemplo, en vez de escribir git commit, simplemente necesitas escribir git ci. A medida que uses Git, probablemente uses otros comandos de forma frecuente. En este caso no dudes en crear nuevos alias.

Esta técnica también puede ser muy útil para crear comandos que creas que deben existir. Por ejemplo, para corregir el problema de usabilidad que encontramos al quitar del área de preparación un archivo, puedes añadir tu propio alias:

```
$ git config --global alias.unstage 'reset HEAD --'
```

Esto hace los siguientes dos comandos equivalentes:

```
$ git unstage fileA
$ git reset HEAD fileA
```

Esto parece un poco mas claro. También es común añadir un comando last, tal que así:

```
$ git config --global alias.last 'log -1 HEAD'
```

De esta forma puedes ver la última confirmación fácilmente:

```
$ git last
commit 66938dae3329c7aebe598c2246a8e6af90d04646
Author: Josh Goebel <dreamer3@example.com>
Date: Tue Aug 26 19:48:51 2008 +0800
```

test for current head Signed-off-by: Scott Chacon <schacon@example.com>

Como puedes ver, Git simplemente reemplaza el nuevo comando con lo que le pongas como alias. Sin embargo, tal vez quieres ejecutar un comando externo en lugar de un subcomando de Git. En este caso, empieza el comando con el caracter 1. Esto es útil si escribes tus propias herramientas que trabajan con un repositorio de Git. Podemos demostrarlo creando el alias git visual para ejecutar gitk:

\$ git config --global alias.visual '!gitk'

Anterior

2.6. Creando etiquetas

Siguiente 2.8. Resumen

© 2006-2022 uniwebsidad Contacto Aviso legal

Recursos sobre: CSS

JavaScript

Python

diseño drupal

programación

ruby Symfony

PHP

5.516 días online