

Creación de ramas y fusión

Comandos utilizados en esta práctica de laboratorio

git branch : enumera todas las ramas en la máquina local y muestra cuál es la rama de trabajo actual

git checkout -b *branchname* : crea una nueva rama con el nombre de rama especificado usando la rama actual como base

git checkout *branchname* : cambia a la rama especificada, trabajo que no está comprometido con no ser guardado

git merge *branchname* - fusiona la rama especificada en la rama de trabajo actual, las fusiones son operaciones unidireccionales

Paso uno

En la terminal o shell de git, usa el comando, `git branch`, para ver que solo hay una y es la rama maestra.

```
git branch
* master
```

Segundo paso

Cree una nueva sucursal llamada `foreign-languages` con el comando `git checkout -b foreign-languages`.

Usar `git branch` nuevamente muestra que ahora hay dos ramas y en la que está trabajando es en idiomas extranjeros.

```
git branch
* foreign-languages
  master
```

Al crear una nueva sucursal o cambiar a una sucursal diferente, todo el trabajo sin seguimiento se transfiere a la nueva sucursal; sin embargo, es una mala práctica confiar en esta funcionalidad. Por lo tanto, asegúrese de confirmar o guardar su trabajo antes de cambiar de rama.

Paso tres

Edita el archivo **languages.txt** agregando cualquier idioma extranjero que conozca, si no sabe ningún idioma extranjero, incluya un idioma que desee aprender. **Asegúrese de incluir una etiqueta para los tipos de idiomas (esto ayudará a simular conflictos más adelante en el laboratorio)** . También puede eliminar la línea de texto agregada del laboratorio 2 si lo desea.

Ejemplo: `path/to/repo/languages.txt`

```
programming languages:
  python
  javascript
  c++

foreign languages:
  french
```

Guarda el archivo.

Paso cuatro

Ahora agregue y confirme los cambios con un mensaje de confirmación descriptivo, como `add foreign languages`. Siéntase libre de mirar hacia atrás en el laboratorio 1 como referencia.

También envíe esta nueva rama a GitHub con `git push --set-upstream origin foreign-languages`. La `--set-upstream` opción le dice a git dónde enviar futuras inserciones, esto solo es necesario cuando se realiza la primera inserción de una nueva rama. Corre `git help push` para aprender más.

Si busca en el sitio web de GitHub, ahora debería poder ver la nueva rama.

```
git agregar idiomas.txt
git commit -m " agregar idiomas extranjeros "
git push --set-upstream origen idiomas extranjeros
```

Paso cinco

Para simular a otra persona trabajando en el proyecto, o si hay una situación que requiere trabajar en una característica más apremiante, creará otra rama. Antes de que las ramas editen el mismo archivo para mostrar cómo funciona la fusión.

En primer lugar, vuelva a la rama principal, ya que al crear una rama, utiliza la rama de trabajo actual para ramificarse. Para cambiar a la rama principal, utilice el comando `git checkout master`.

El uso `git branch` mostrará que la rama actual es maestra.

```
git branch
      foreign-languages
* master
```

Paso seis

Cree una nueva rama llamada `language-strength`. Para crear la rama use el comando, `git checkout -b language-strength`.

El uso `git branch` ahora mostrará tres ramas con la seleccionada siendo `language-strength`.

```
git branch
      foreign-languages
* language-strength
      master
```

Paso siete

Edite el archivo **languages.txt** agregando un indicador de qué tan bien lo conoce. Agregue débil, promedio o fuerte al lado de cada idioma. También puede eliminar la línea de texto agregada del laboratorio 2 si lo desea.

Ejemplo: `path/to/repo/languages.txt`

```
python - strong
javascript - strong
c++ - weak
```

Guarda el archivo.

Paso ocho

Ahora agregue y confirme los cambios con un mensaje de confirmación descriptivo, como `add language strength`. Siéntase libre de mirar hacia atrás en el laboratorio 1 como referencia.

También envíe esta nueva rama a GitHub con `git push --set-upstream origin language-strength`. La `--set-upstream` opción le dice a git dónde enviar futuras inserciones, esto solo es necesario cuando se realiza la primera inserción de una nueva rama. Corre `git help push` para aprender más.

Si busca en el sitio web de GitHub, ahora debería poder ver la nueva rama.

```
git agregar idiomas.txt
git commit -m " agregar intensidad de lenguaje "
git push --set-upstream origin language-strength
```

Paso nueve

Ahora hay dos ramas con funciones que deben fusionarse en la rama principal. Primero, vuelva a la rama maestra con `git checkout master`. Correr `git branch` lo verificará.

```
git branch
  foreign-languages
  language-strength
* master
```

paso diez

Para fusionar la rama **de idiomas extranjeros** en el maestro, use el comando, `git merge foreign-languages`. Git realizará la combinación automáticamente, ya que no hubo ningún cambio en la rama principal desde que se crearon los idiomas extranjeros. Si aparece un editor de texto en la terminal, utilícelo `ctrl+x` para aceptar el mensaje predeterminado para la fusión.

```
git merge foreign-languages
Merge made by the 'recursive' strategy.
 languages.txt | 10 ++++++----
 1 file changed, 6 insertions(+), 4 deletions(-)
```

Primera fusión hecha

Paso once

Ahora, combine la rama **de fortaleza del lenguaje** con `git merge language-strength`. Esta vez habrá conflictos porque el **archivo languages.txt** fue editado por ambas ramas.

```
git merge language-strength
Auto-merging languages.txt
CONFLICT (content): Merge conflict in languages.txt
Automatic merge failed; fix conflicts and then commit the result.
```

La ejecución `git status` mostrará qué archivos tienen conflictos.

```
On branch master
Your branch is ahead of 'origin/master' by 2 commits.
  (use "git push" to publish your local commits)
You have unmerged paths.
  (fix conflicts and run "git commit")

Unmerged paths:
  (use "git add <file>..." to mark resolution)

    both modified:   languages.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

Paso Doce

Resolviendo conflictos

Cuando hay un conflicto, git agrega varias líneas a su archivo para indicar dónde están los conflictos. Git usa las líneas `<<<<<<< HEAD, =====y >>>>>>> branchname` para decirte qué cambios fueron de cada rama.

- Cualquier cosa entre las líneas `<<<<<<< HEADy =====` es contenido de la rama en la que se está fusionando (la rama maestra en este ejemplo)
- Cualquier cosa entre las líneas `=====y >>>>>>> branchname` es contenido de la rama que especificó `git merge branchname` (la rama de fuerza del idioma en este ejemplo)

La resolución de conflictos se puede hacer de tres maneras, y depende de la persona que se fusiona decidir

- Solo use el código de la rama que se fusiona con: el código entre `<<<<<<< HEADy=====`
- Solo use el código de la rama que se fusiona desde: el código entre `=====y>>>>>>> branchname`
- Reescribir la sección de código

Asegúrese de eliminar las líneas `<<<<<<< HEAD, =====y después>>>>>>> branchname`

Así es como se verá el archivo languages.txt después de que git detecte el conflicto:

```
<<<<<< HEAD
programming languages:
    python
    javascript
    c++

foreign languages:
    french
=====
python - strong
javascript - strong
c++ - weak
>>>>>> language-strength
```

Para resolver el conflicto, mantenga los idiomas extranjeros y agregue la fuerza del idioma a todos los idiomas.

Ejemplo: path/to/repo/languages.txt

```
programming languages:
    python - strong
    javascript - strong
    c++ - weak

foreign languages:
    french - weak
```

Guarda el archivo.

Paso Doce

Una vez resuelto el conflicto, se puede completar la fusión. Los cambios para realizar la resolución actualmente no están rastreados, por lo que los cambios deben agregarse y luego confirmarse. El mensaje de confirmación debe incluir una notificación de la fusión.

```
git add languages.txt
git commit -m "Merge branch 'language-strength'"
```

Finalmente, las fusiones se pueden enviar a GitHub con un archivo `git push`. Verifique en el sitio web de GitHub para verificar.