**Manual Git**

**Mostrar info**

git status

Mostra la branca i fitxers modificats no staged

git log --oneline

Historial de la branca actual. Mostra el hash dels commit fets

git branch -a

Mostra totes les branques. L'actual té un asterisc

**Fer un commit**

git commit -m "missatge"

**Tornar enera d'un commit**

git reset --hard *hash*

No preserva modificacions fetes i no commit

També: git reset --hard HEAD~1 desfà l'últim commit

**Crear una branca**

git branch *branca*

git checkout -b *branca*

Crea la branca i hi va

**Canviar de branca**

git checkout branca

**Eliminar una branca**

git branch -D *branca*

**Fusionar branques**

git merge *branca*

Cal estar en la branca troncal (git checkout master)

Si la fusió no és directa, mostrarà les diferències. Corregir-les (en el tronc

**Conceptes Git**

Llocs

* Working dir
  + Conté els fitxers
* Staging / index / zona intermèdia
  + Manté els fitxers seguits
* Repository (local)
  + Conté els commit fets i els snapshots pertinents dels fitxers

Apuntadors

* HEAD
* branca (master o el nom de la branca)

Crear un repositori local nou

* Anar al directori arrel del projecte
* git init

Inicialitzar

* git config --global user.name "<el meu nom>"
* git config --global user.email " <la meva adreça>"
* git config --global --list

Copiar un repositori d'un remot

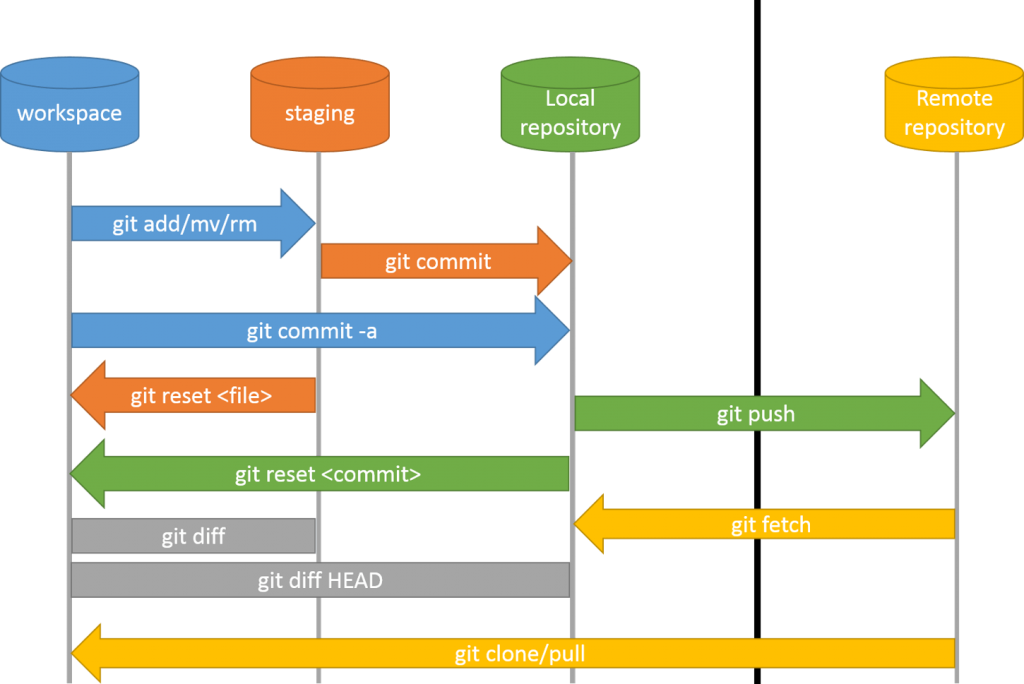
* Anar a la carpeta que vull que sigui l'arrel
* git clone <camí al repositori original>

Afegir un fitxer a l'staging (seguir-lo) / Indicar que s'ha modificat un fitxer seguit

* git add <fitxer>
* git add . tot el directori; també git add \*
* Afegeix (add) a allò que serà commit

Esborrar un fitxer de l'staging

* git rm <fitxer>



Treure de l'staging (unstage)

* git reset <fitxer>

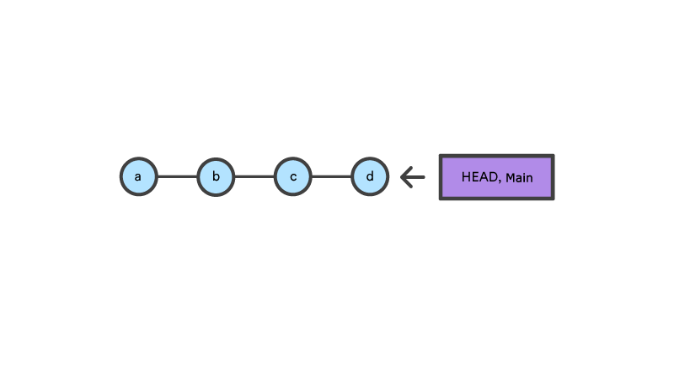
Fer un commit. Guardar en el HEAD

* git commit -m "comentari"
* El comentari ha d'indicar breument què s'ha fet en el projecte
* git commit -am "comentari" fa l'add i el commit a l'hora
* HEAD apunta al commit actual (pot no ser l'últim)

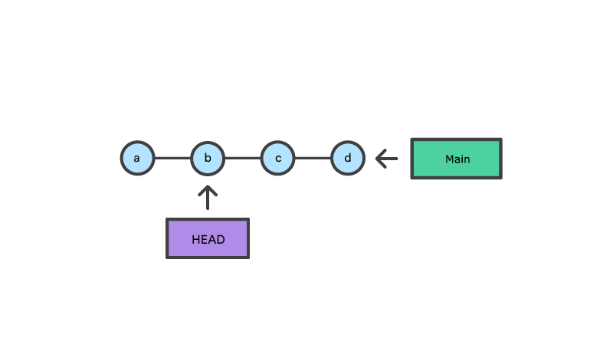
Consultes

* git status situació general
  + git status -s simplificat
* git branch llistat de branques i en quina estic
* git log mostra els commits
  + git log --oneline i també --graph --decorate --color

Situació inicial



Revisar un commit anterior



* git checkout <commit>
* Canvia el HEAD al commit indicat
* No hauríem de modificar res

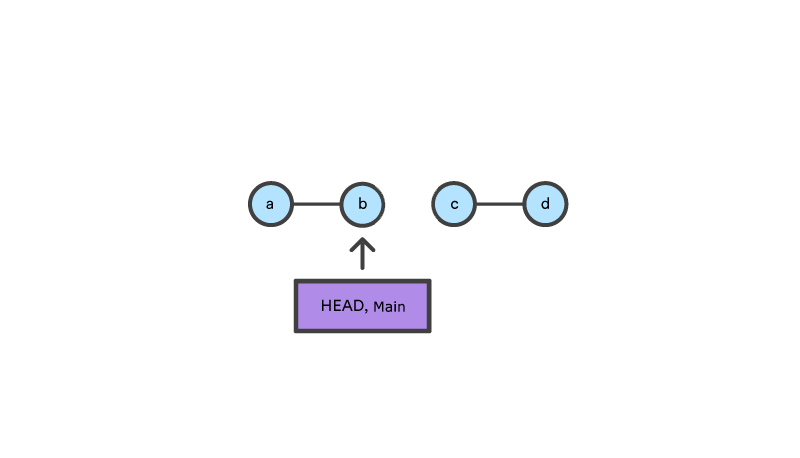
Retornar després d'una revisió

* git checkout master (o Main)
* Torna al "final" de la branca master: a l'últim commit fet de la branca indicada

"Detached HEAD" state

* Es produeix amb git checkout
* El HEAD no està apuntant a l'últim de la branca
* No s'aconsella desenvolupar a partir d'aquí: fer una branca nova

Tornar a un commit anterior



* git reset --hard <commit\_anterior>
* Canvia els apuntadors HEAD i master (el de la branca) al commit indicat
* Tot i això no s'esborren els commits posteriors i podem retornar a un commit posterior (però hem de recordar-ne el número; HEAD i master ja no serveixen!). O bé usar git reflog
* git log no mostra els commits posteriors a aquest
* --soft, --mixed i --hard indiquen quins llocs queden modificats:
  + --soft: només repository. Només canvia els dos índexs (HEAD i master)
  + --mixed (per defecte): staged i repository. Canvia índex i info de l'staged
  + --hard: tots: working, staged i repository
* El --hard perd tots els canvis pendents en l'staged i working

**Branques**

Mostar branques existents

* git branch

Canviar de branca

* git checkout <branca>

Canviar de nom la branca actual

* git branch -m <nom\_nou\_branca>

Crear una branca

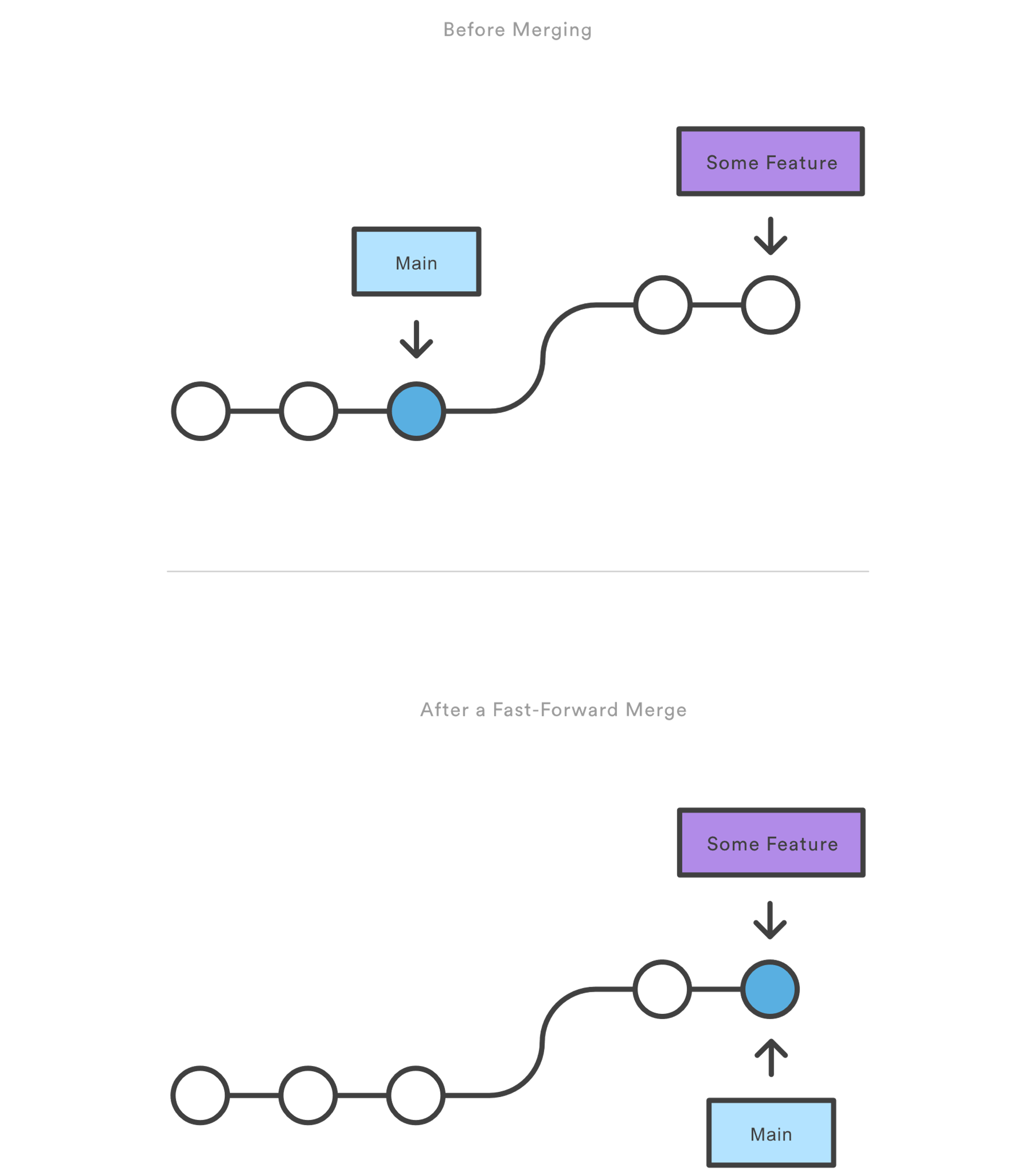
* git branch <nova\_branca>

Eliminar una branca

* S'ha de fer des d'una altra branca
* git branch -d <branca> No elimina si hi ha modificacions pendents
* git branch -D <branca>

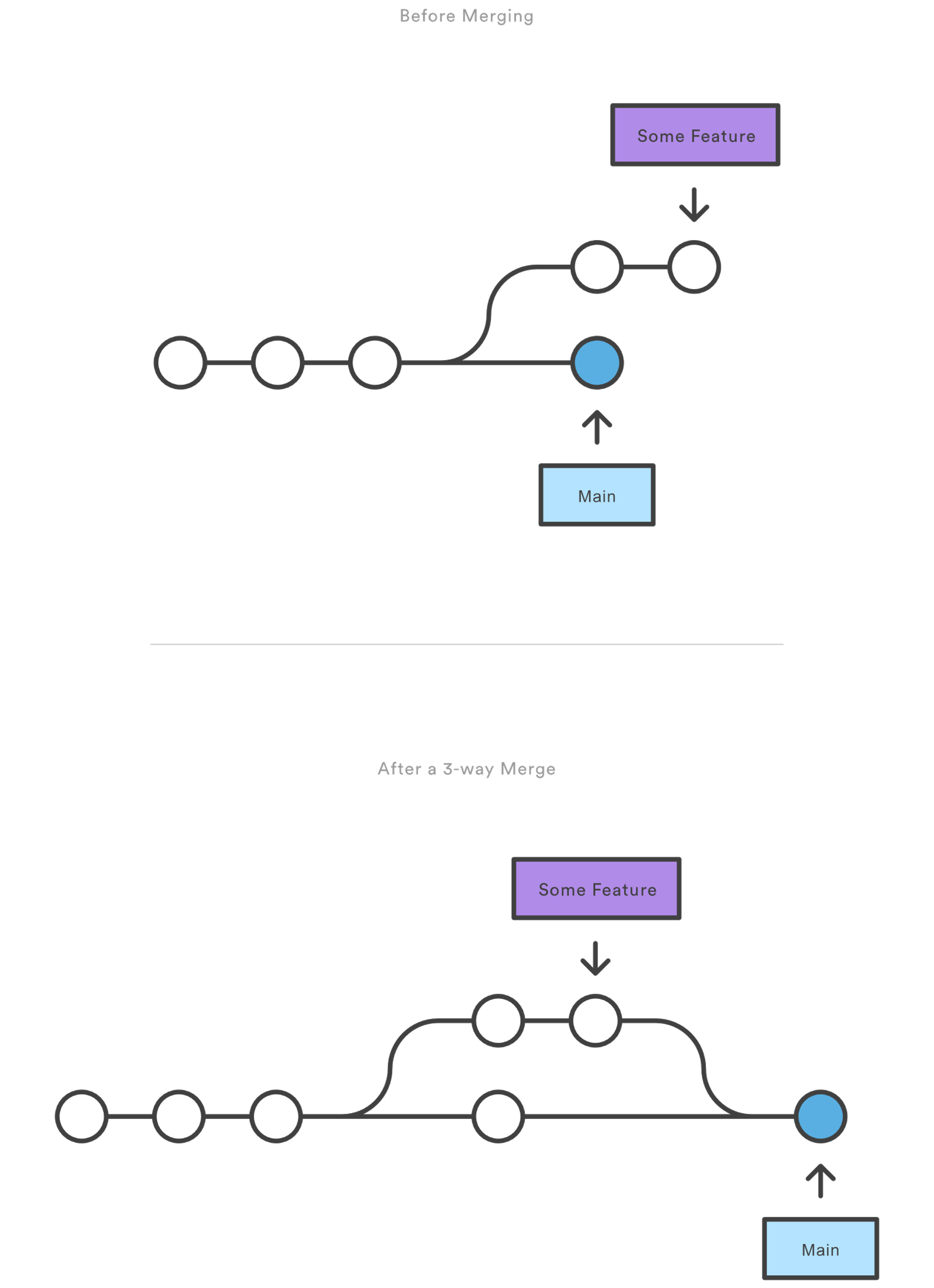
**Merge**

Fast-forward merge



* Quan no hem modificat la branca original, només la nova branca
* Converteix el HEAD de la nova branca en el HEAD de la branca principal
* git merge <branca> Des de la branca principal (o allà on volem fusionar-la)
* git merge --no-ff <branca> Fa el merge i també fa un commit del merge (altrament no queda guardat que s'ha fet un merge d'una branca)

3-way merge



* Quan hem modificat la branca original
* Solucionar, fer git add i git commit
* Usar git diff
* git merge --abort Cancel·la el procés de merge

Enviar commits fets en local al remot

* git push <origen> <master>
* L'origen pot ser una branca

Si el repositori local és l'únic, podem crear-ne un de remot:

* git remote add <origen> <server>

GitHub

jordi.surinyac@uvic.cat

@JordiSu

KulD3Fur4

<http://localhost:59890>

jordi.surinyac@gmail.com

@jordisuri

l'habitual

https://github.com/jordisuri/Sincro.git

He creat, via web, el projecte Sincro

Connexió amb el remont:

git remote add origin https://github.com/jordisuri/Sincro.git

git branch -M main

git push -u origin main

Copiar el projecte original al remot

git push -u origin main

# demana autentificació en el browser

Després de treballar localment (amb commits), pujar al repositori

git push origin

Actualitzar la versió local amb el repositori

git pull origin