

In case of fire



 1. **git commit**

 2. **git push**

 3. **leave building**

Branches

Branching



HEAD The snapshot you are viewing

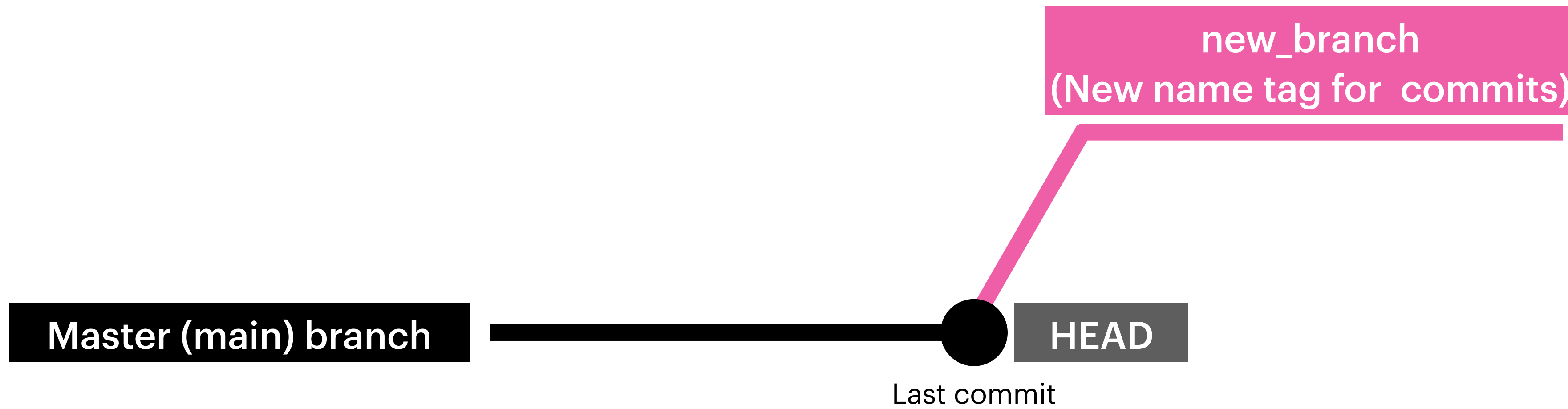
Branching



\$ git branch List of branches (Master)

HEAD The snapshot you are viewing

Branching



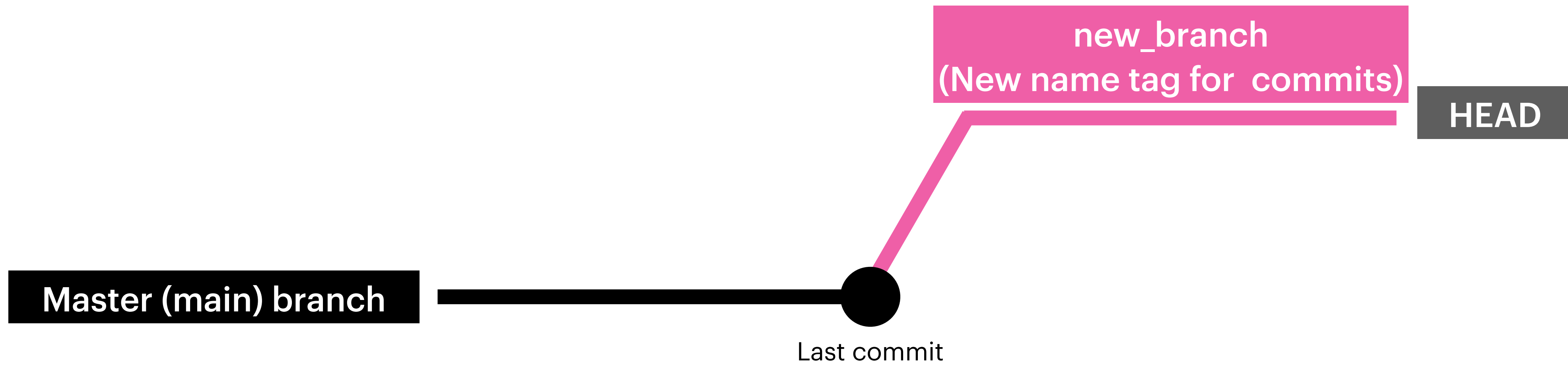
\$ git branch List of branches (Master)

\$ git branch new_branch

HEAD

The snapshot you are viewing

Branching



\$ git branch List of branches (Master)

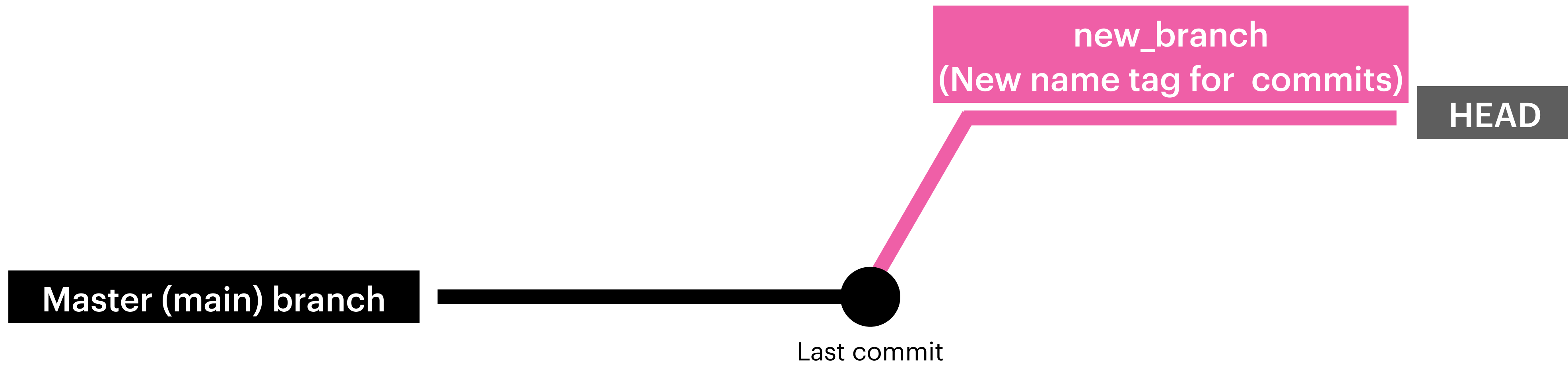
\$ git branch **new_branch**

\$ git checkout **new_branch**

HEAD

The snapshot you are viewing

Branching



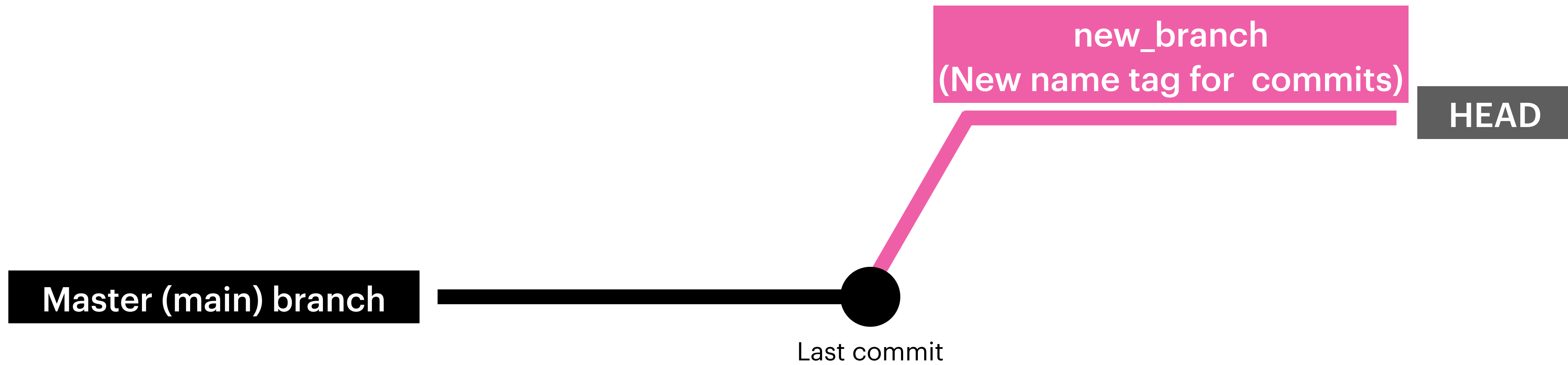
\$ git branch List of branches (Master)

\$ git branch new_branch
\$ git checkout new_branch } \$ git checkout -b new_branch

HEAD

The snapshot you are viewing

Branching



\$ git branch List of branches (Master)

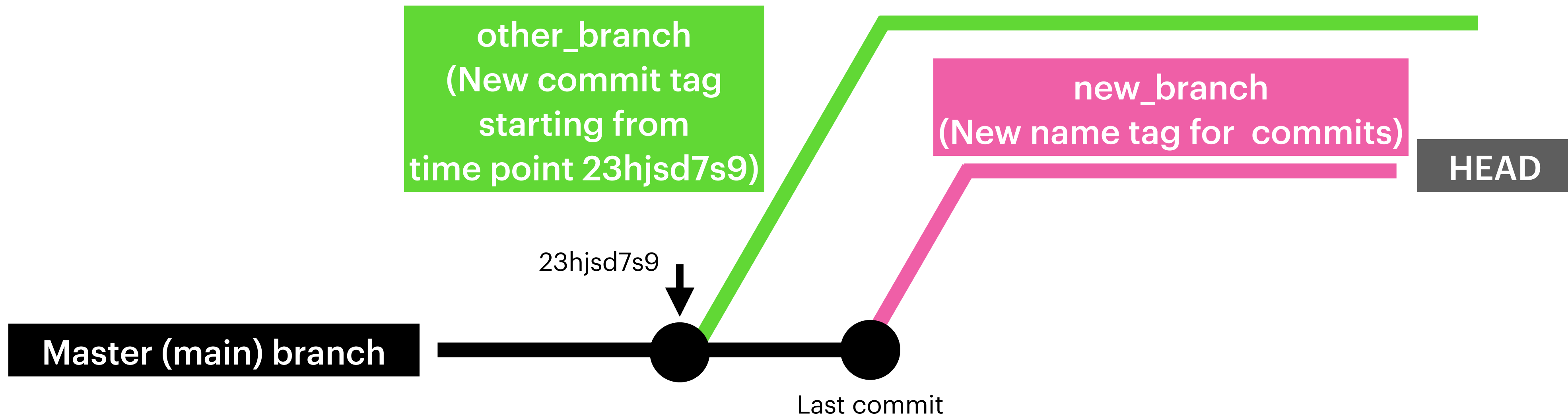
\$ git branch new_branch
\$ git checkout new_branch } \$ git checkout -b new_branch

\$ git switch new_branch

HEAD

The snapshot you are viewing

Branching



\$ git branch List of branches (Master)

\$ git branch new_branch
\$ git checkout new_branch } \$ git checkout -b new_branch

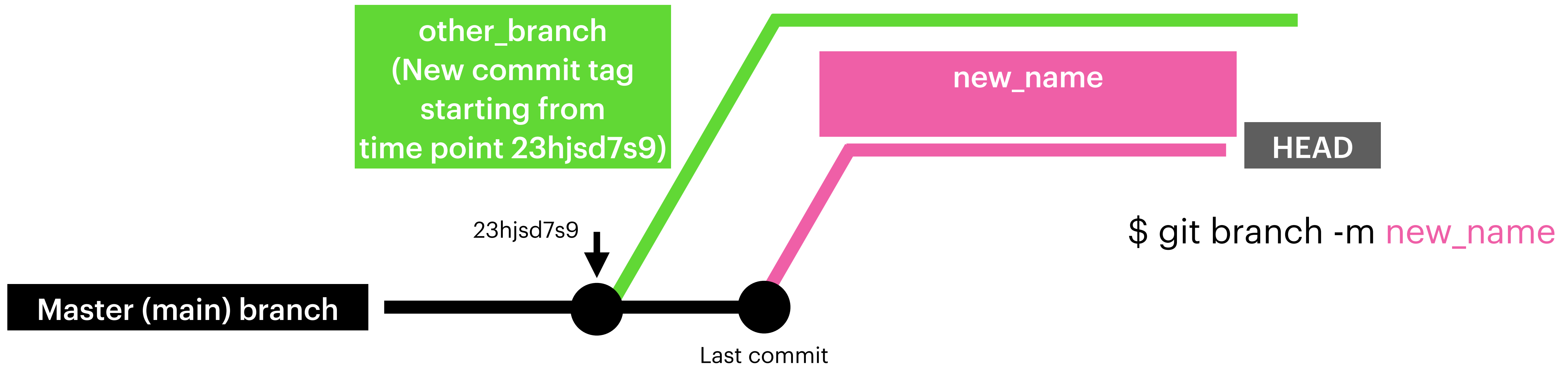
\$ git switch new_branch

\$ git branch other_branch 23hjsd7s9

HEAD

The snapshot you are viewing

Branching



\$ git branch List of branches (Master)

\$ git branch new_branch
\$ git checkout new_branch } \$ git checkout -b new_branch

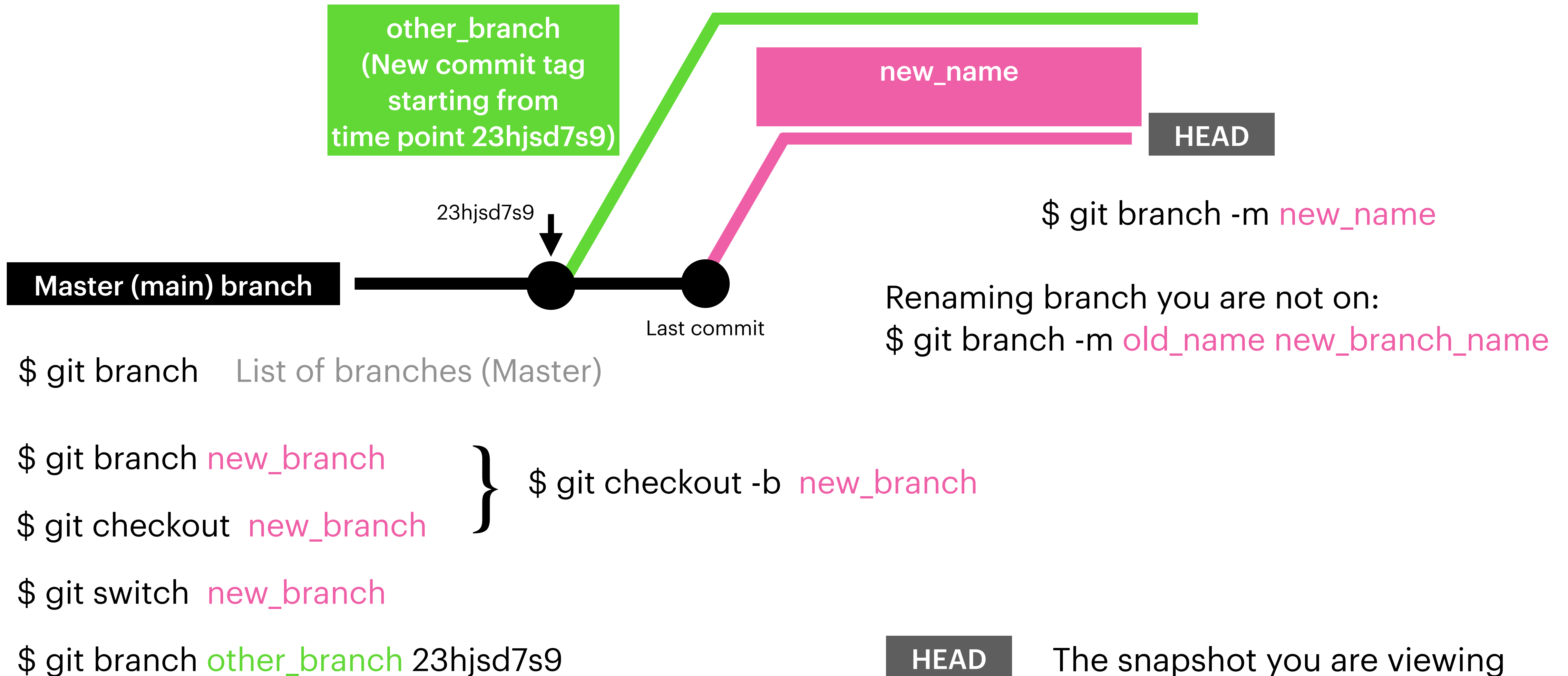
\$ git switch new_branch

\$ git branch other_branch 23hjsd7s9

HEAD

The snapshot you are viewing

Branching



Demo

```
git log --all --decorate --oneline --graph
```

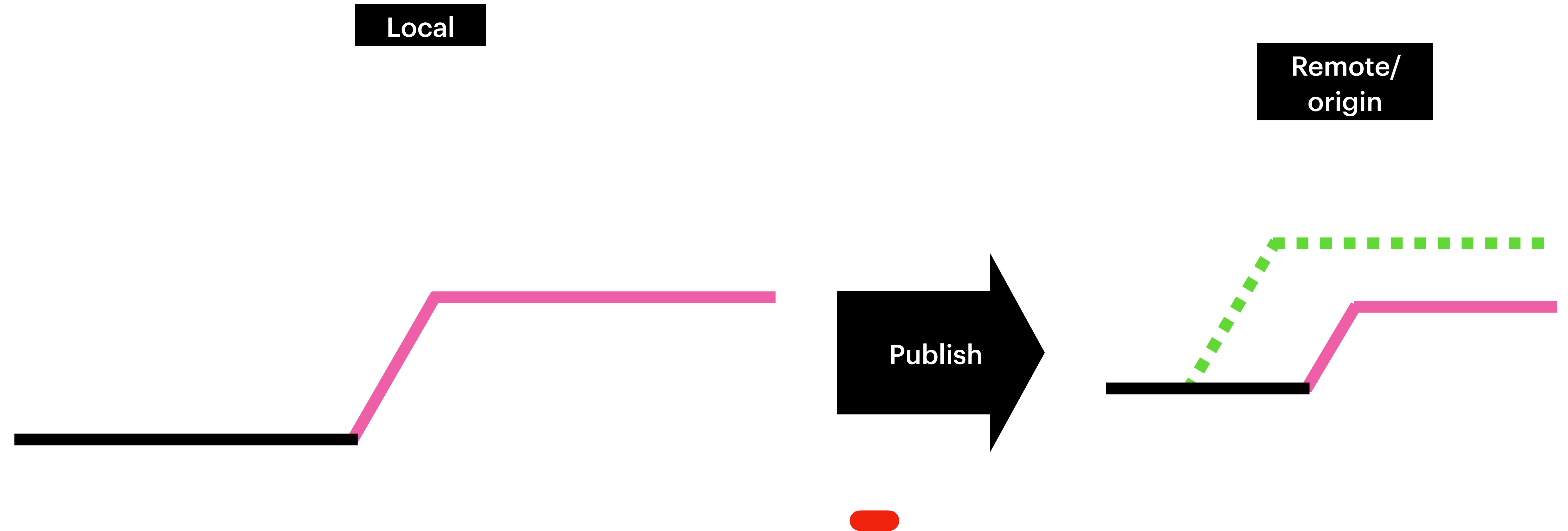
Branching

Connect local branch to remote

Local

Remote/
origin

Publish



Branching

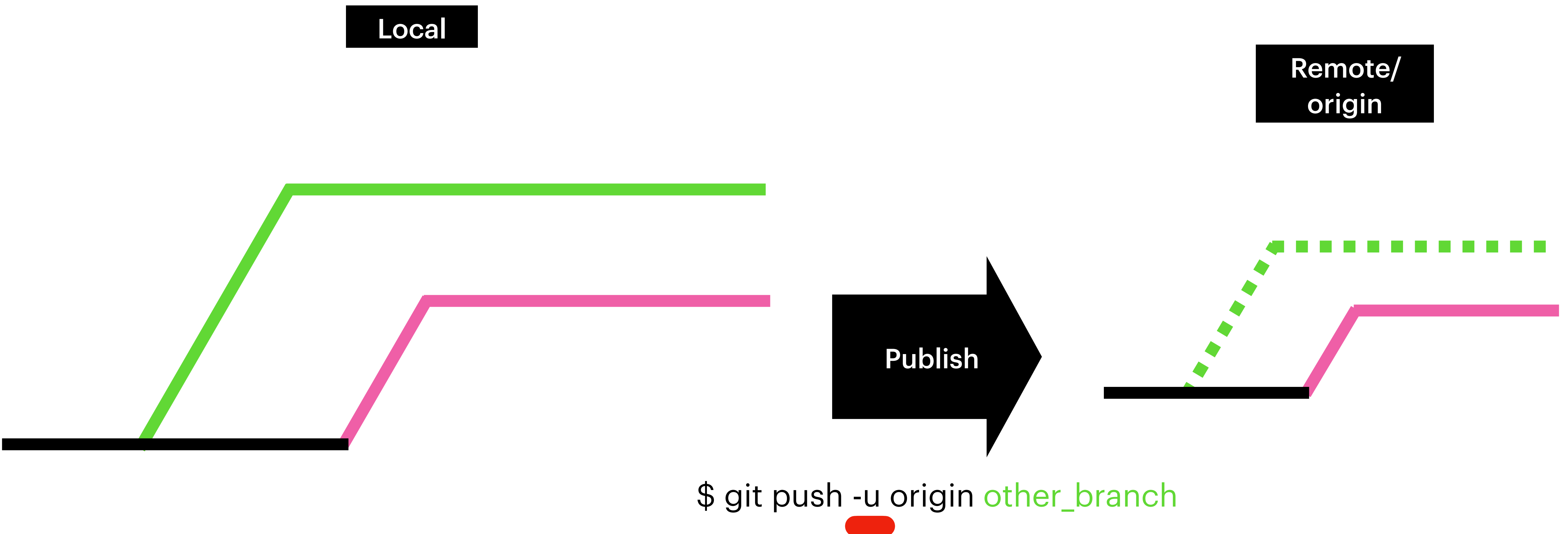
Connect local branch to remote

Local

Remote/
origin

Publish

\$ git push -u origin other_branch

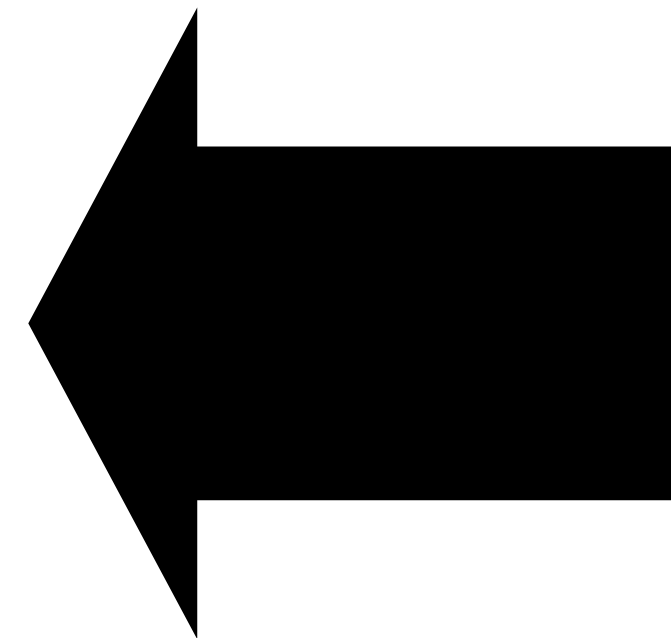


Branching

Connect remote branch to local

Local

Remote/
origin



```
$ git branch --track new-branch origin/new-branch
```

```
$ git checkout --track origin/new-branch
```

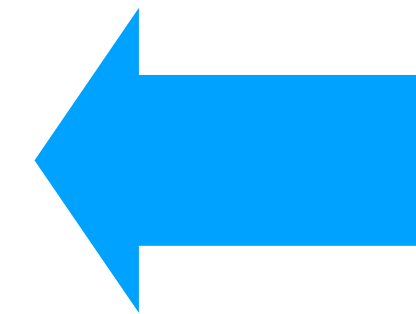
Branching

Sync local/remote branches

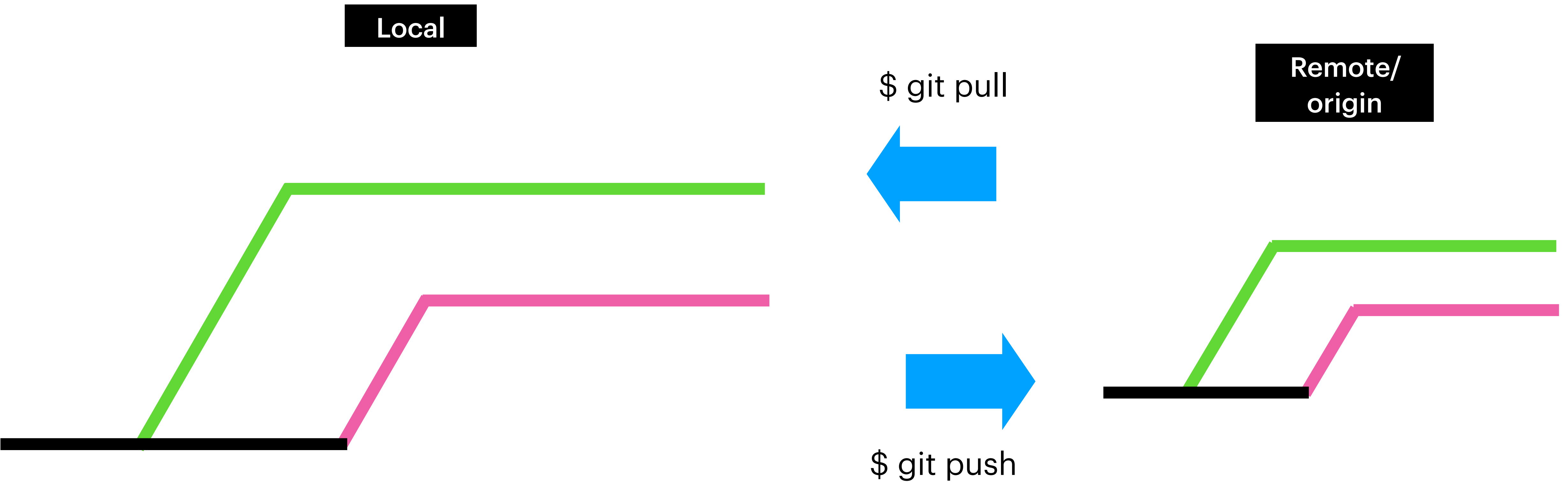
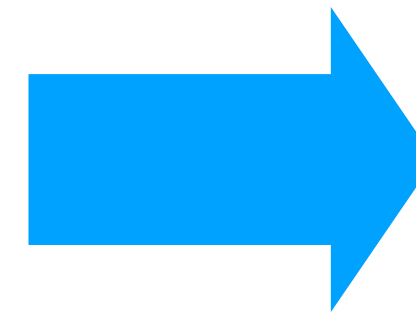
Local

Remote/
origin

\$ git pull

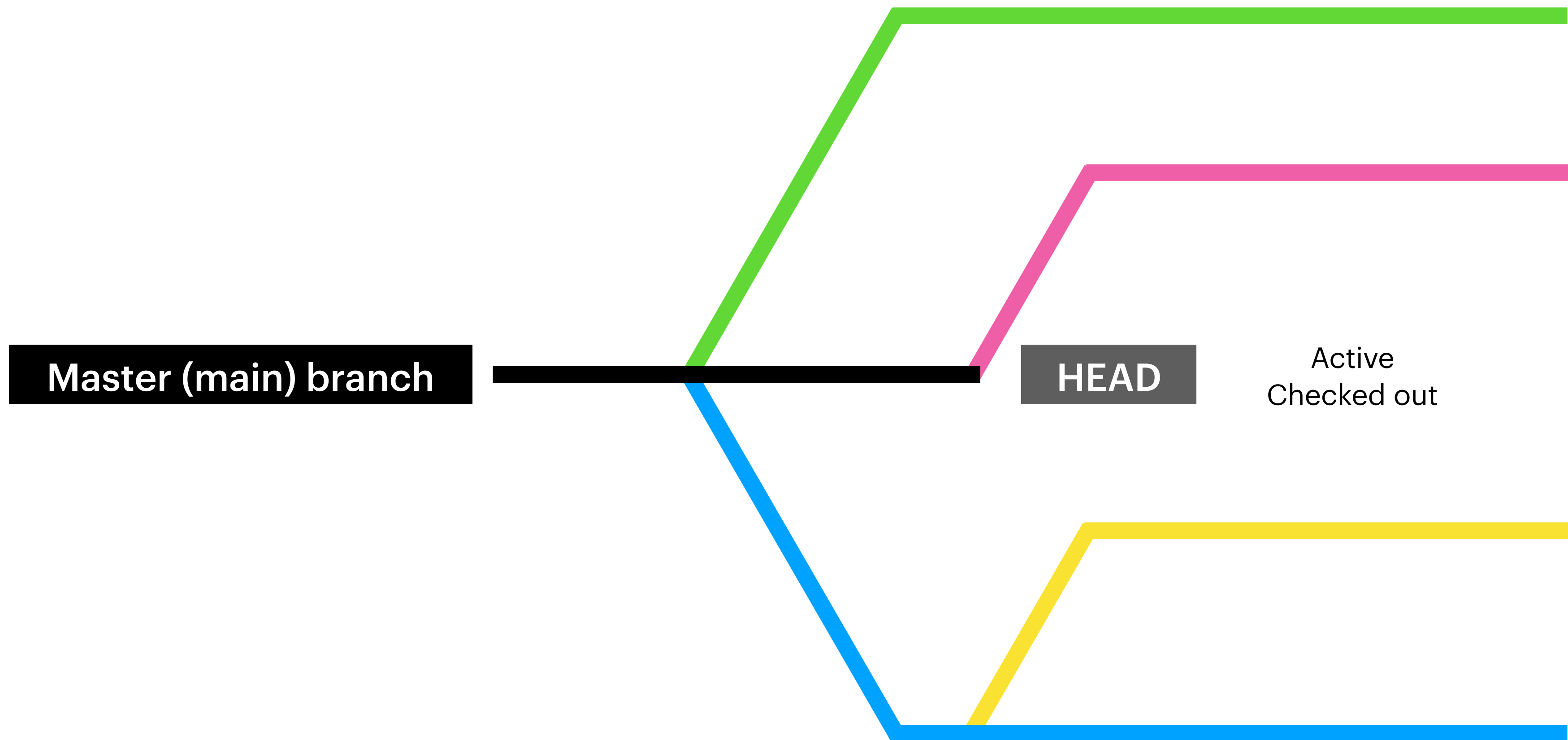


\$ git push



Branching

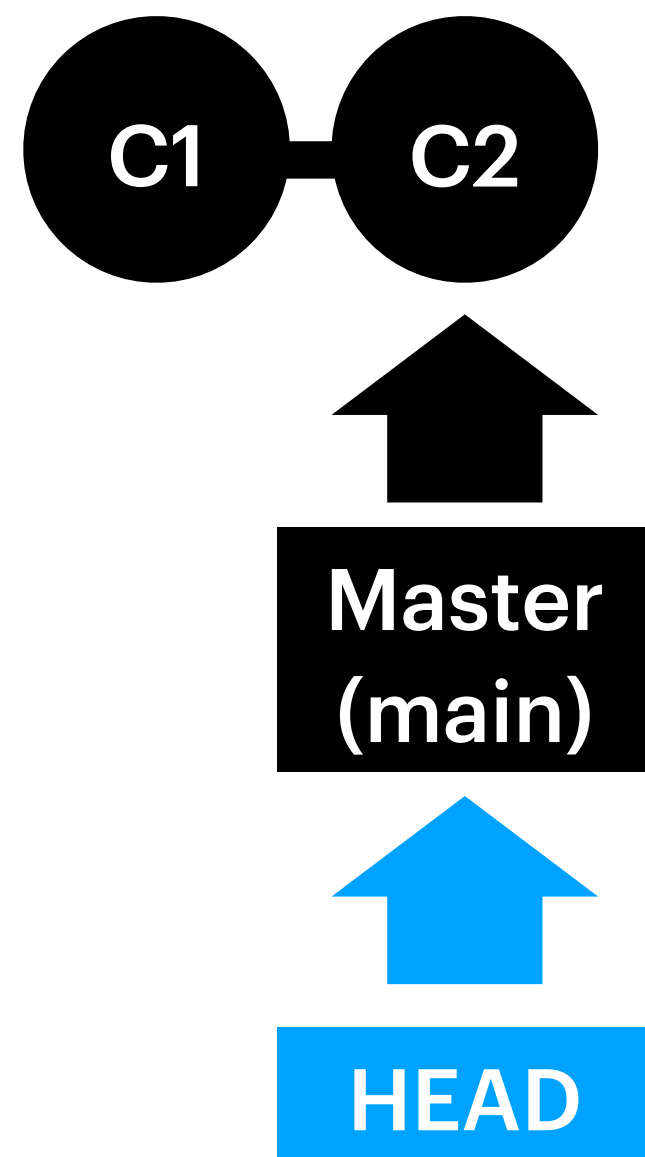
Branches can advance independently



Merge and rebase

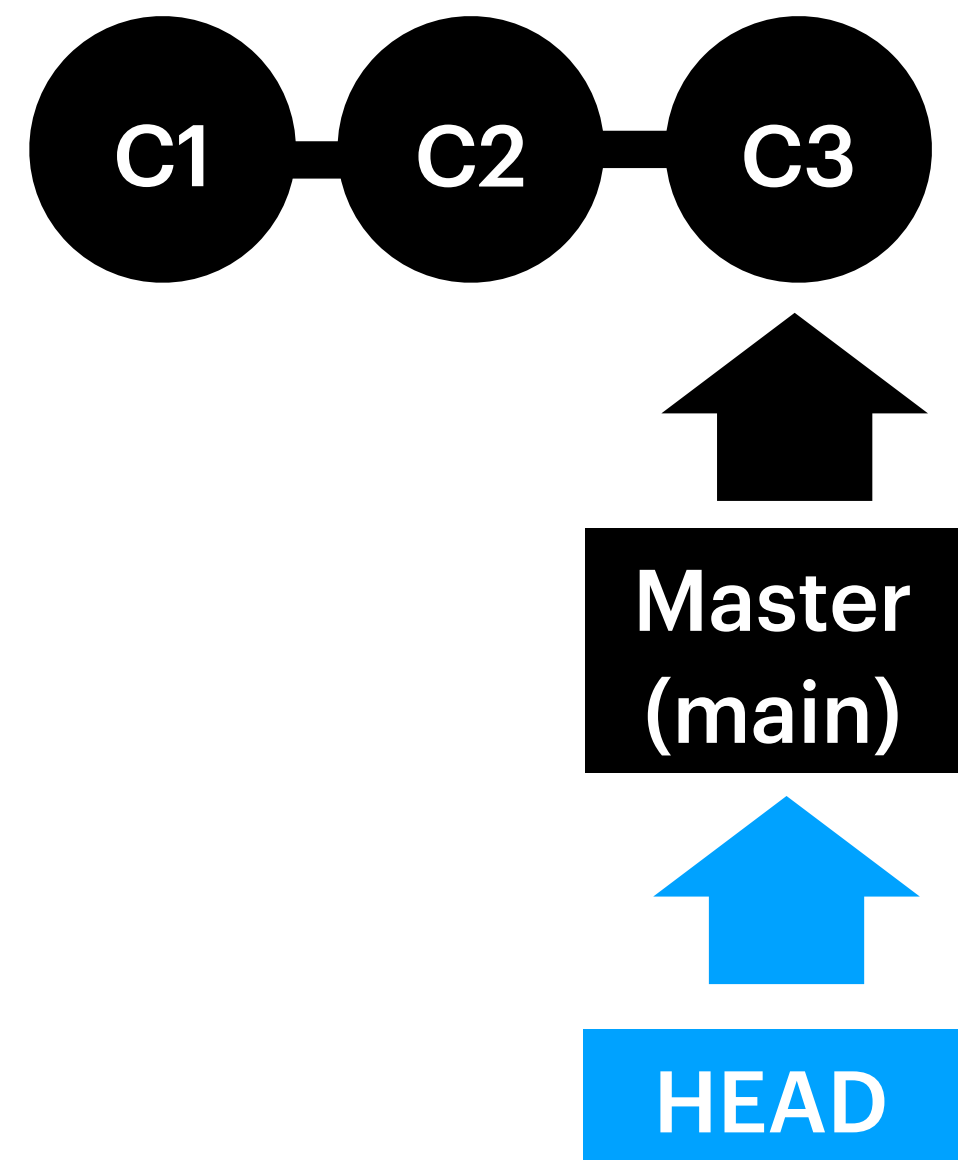
Branching

Merge



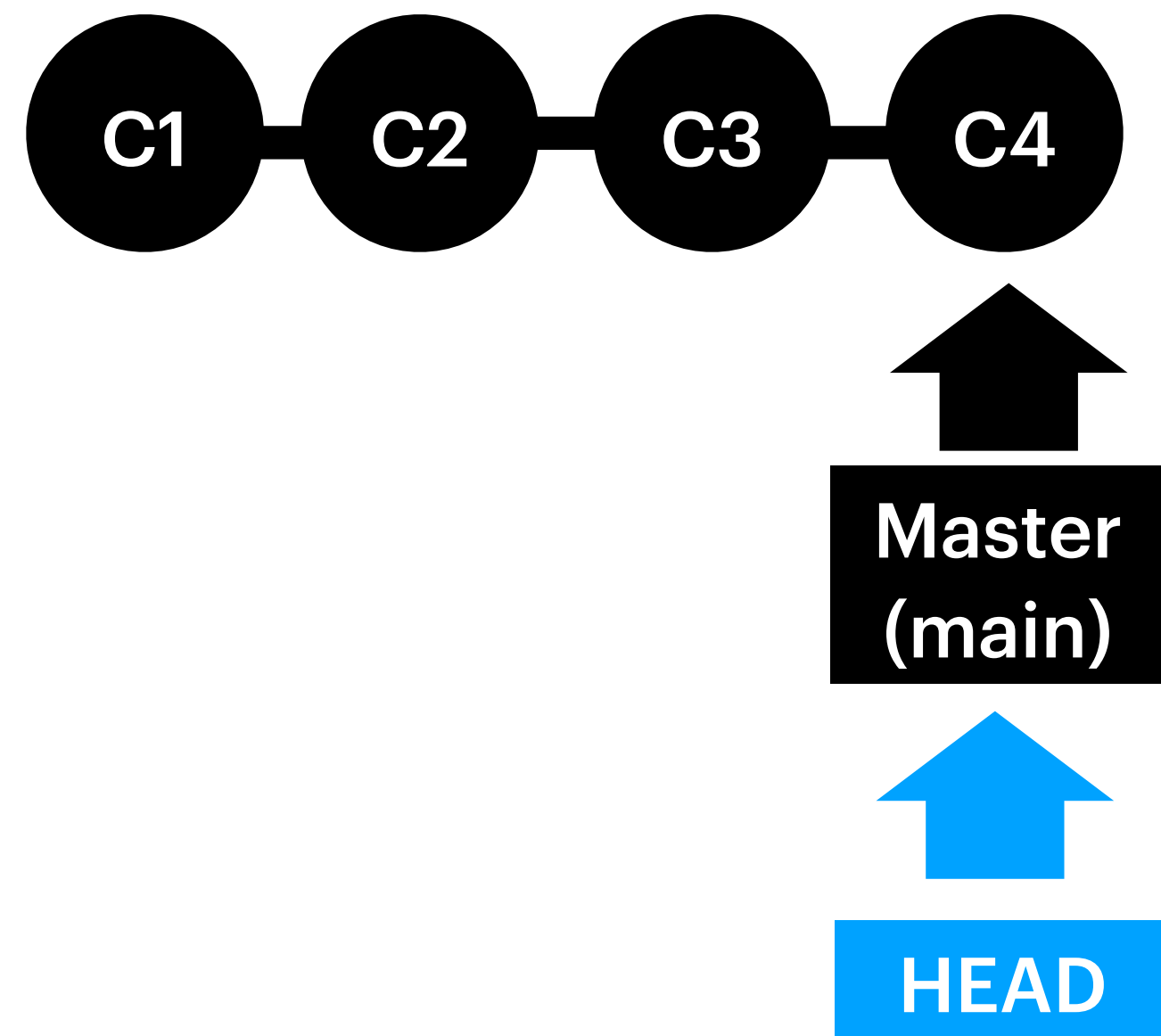
Branching

Merge



Branching

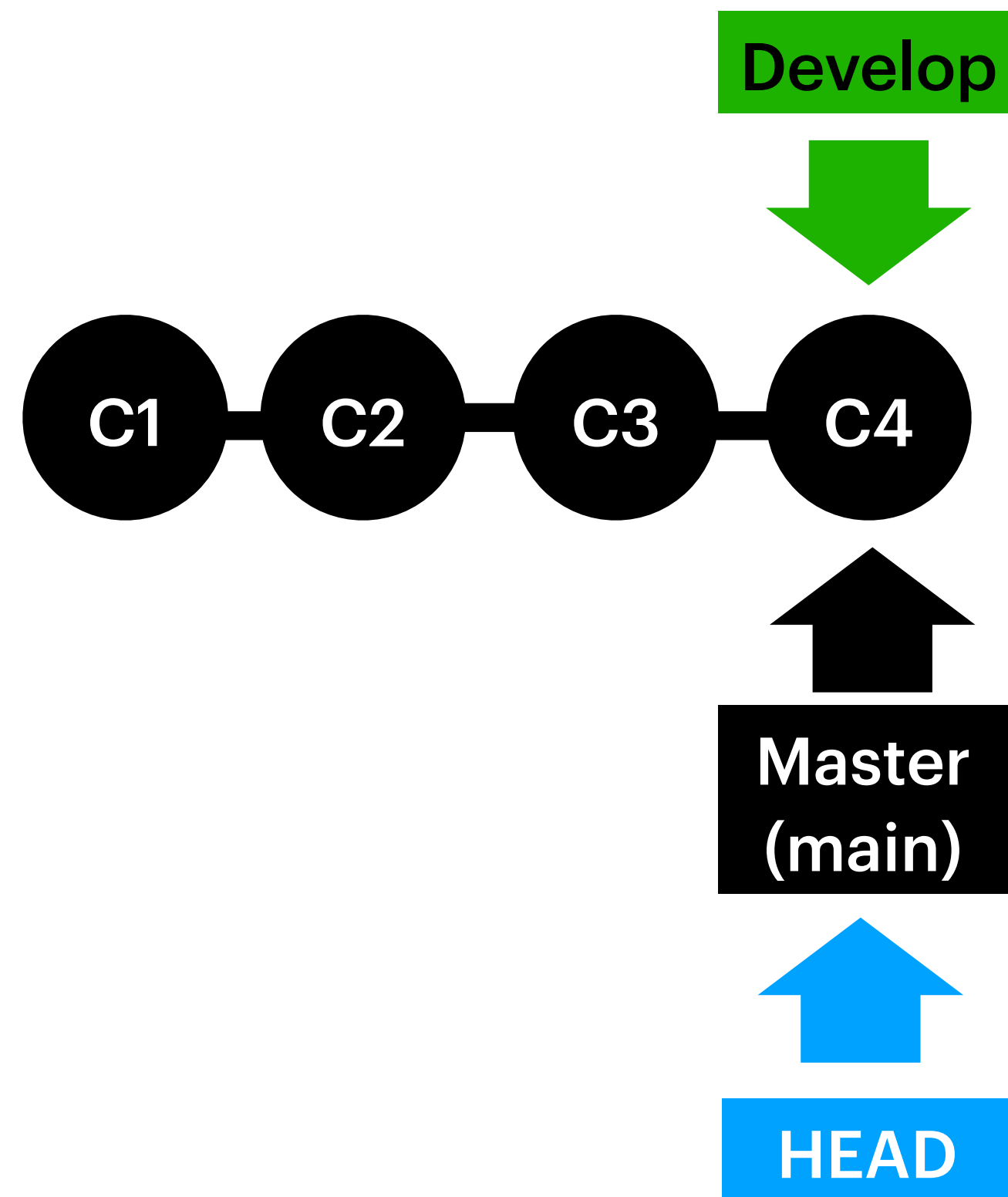
Merge



Branching

Merge

\$ git branch Develop

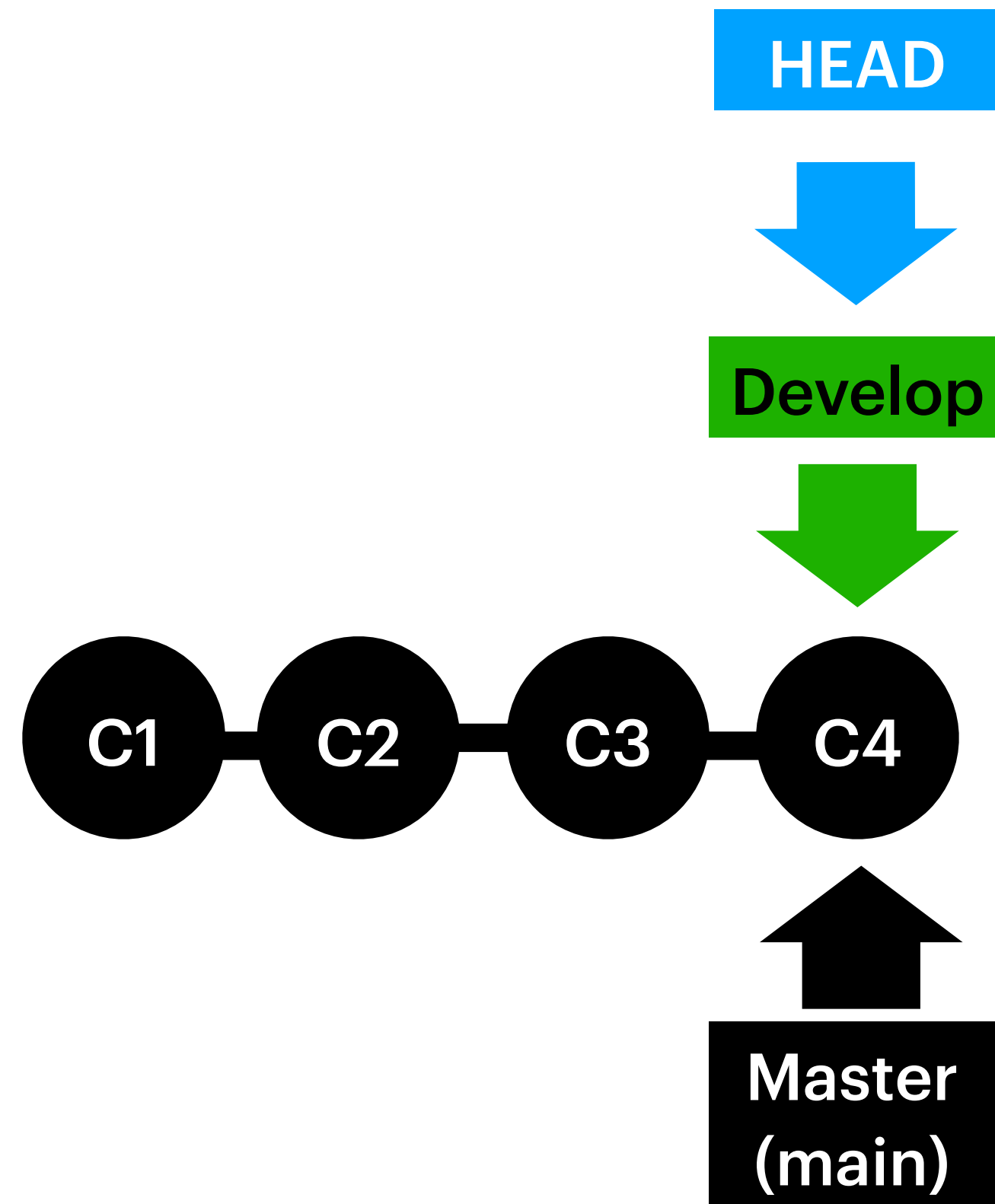


Branching

Merge

\$ git branch Develop

\$ git switch Develop

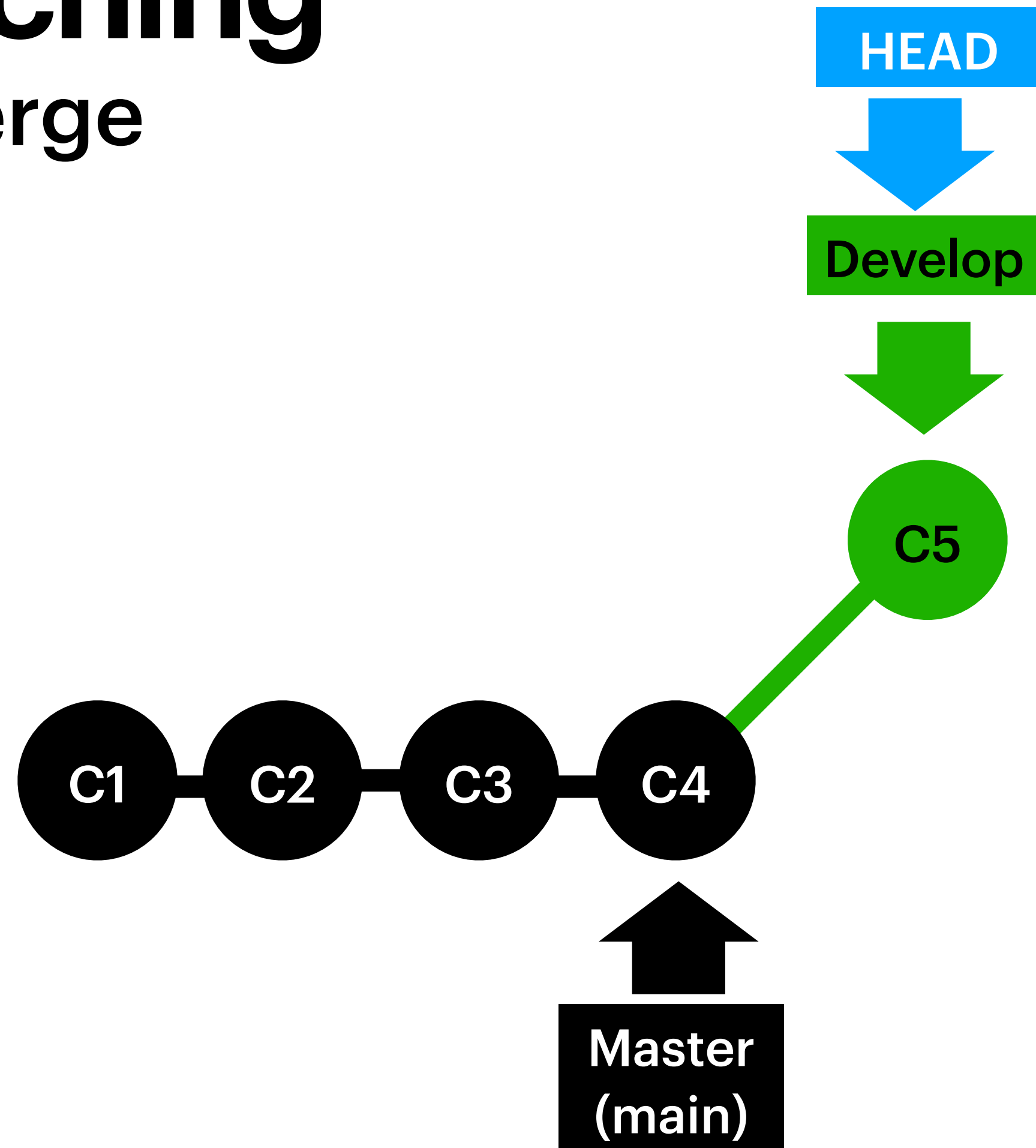


Branching

Merge

\$ git branch Develop

\$ git switch Develop

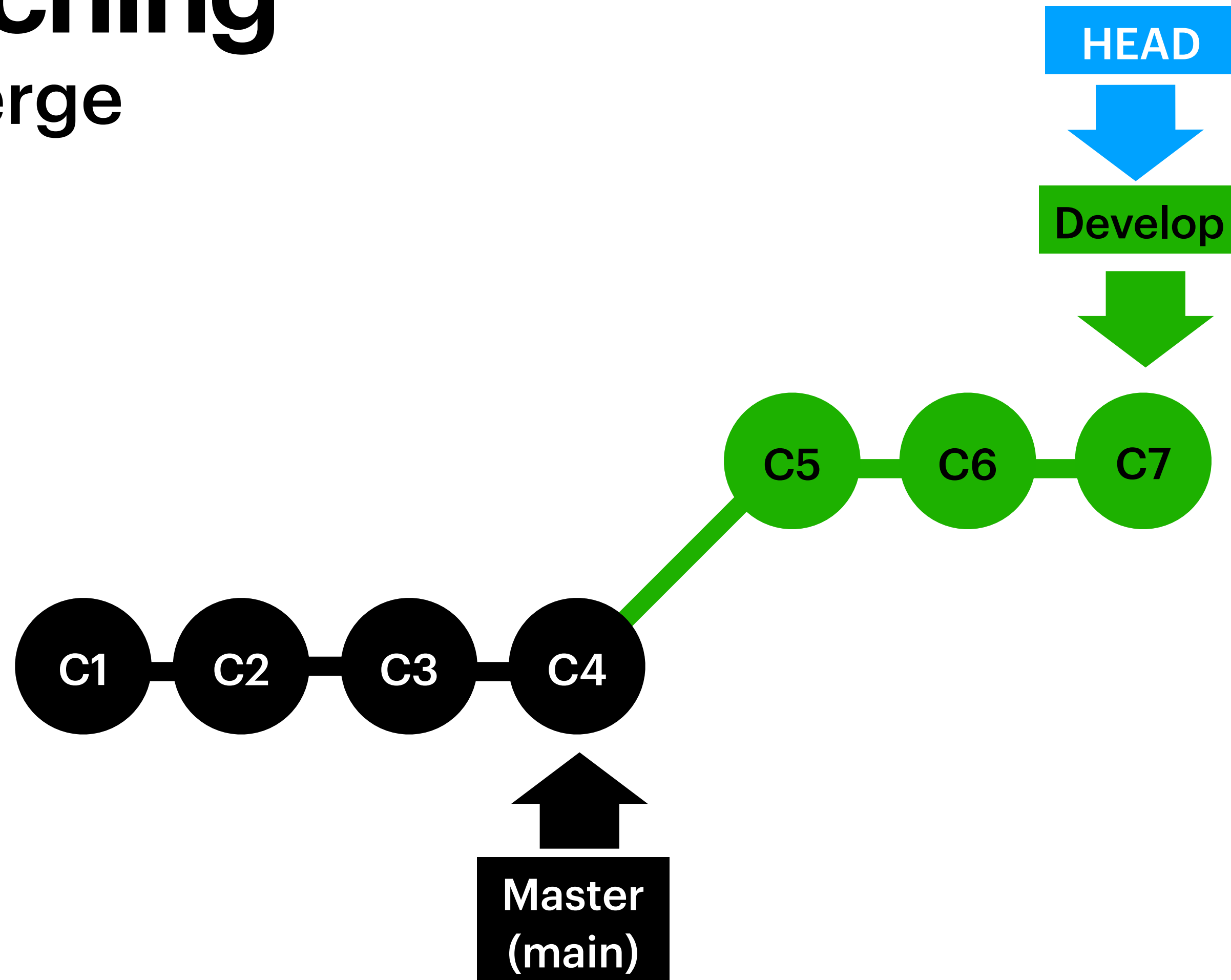


Branching

Merge

```
$ git branch Develop
```

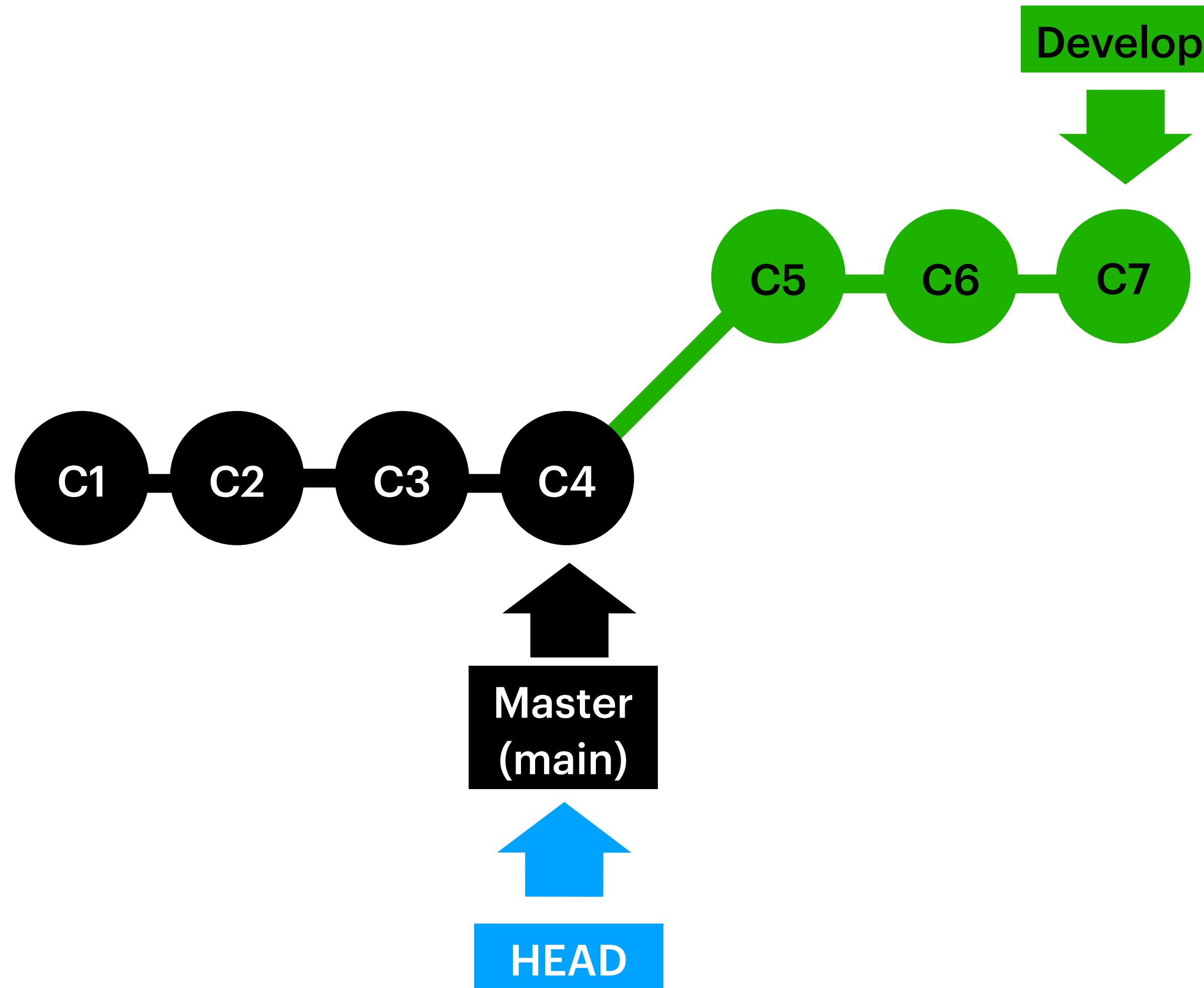
```
$ git switch Develop
```



Branching Merge

```
$ git branch Develop
```

```
$ git switch Develop
```

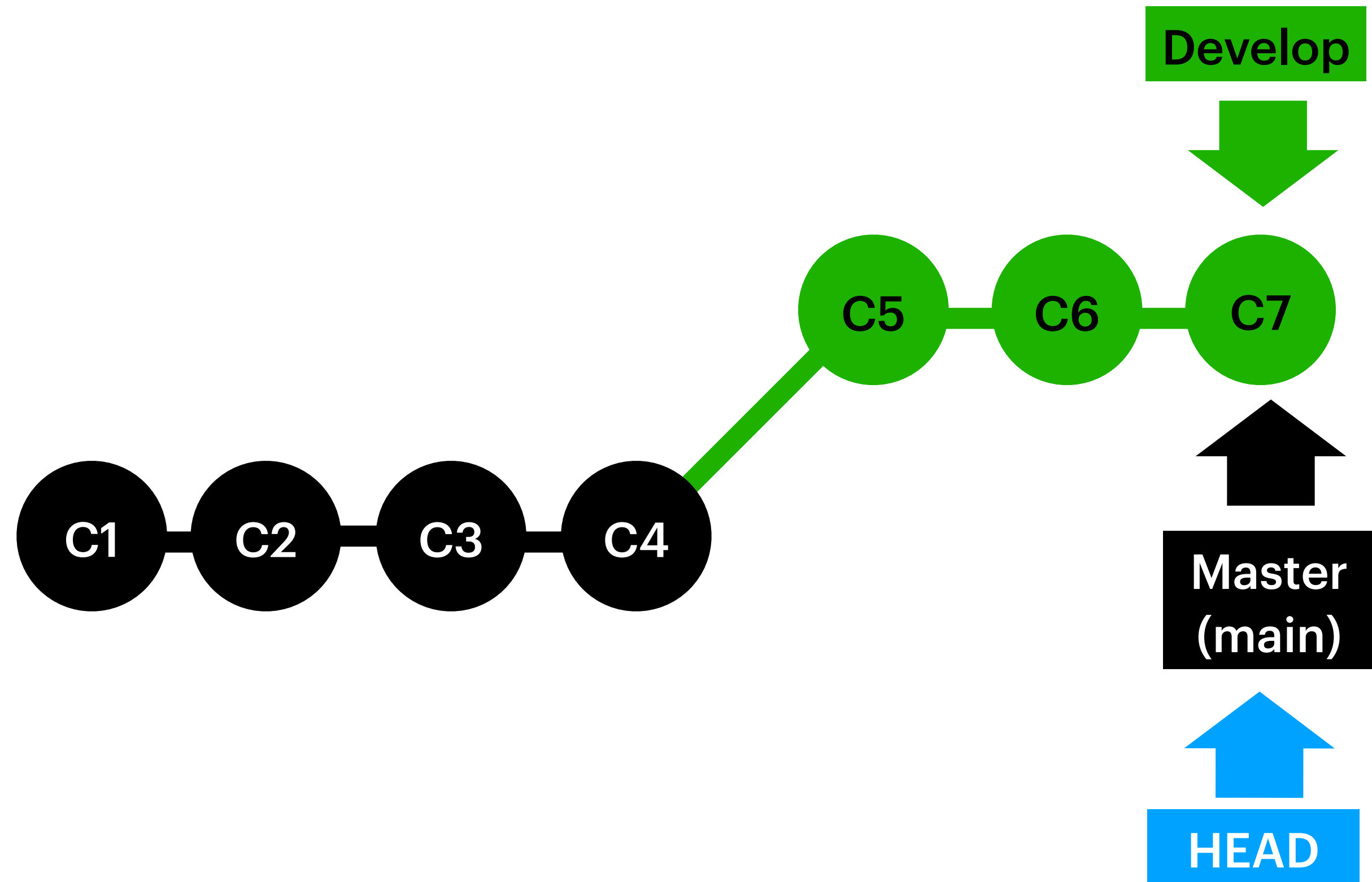


```
$ git switch Master
```

Branching Merge

```
$ git branch Develop
```

```
$ git switch Develop
```

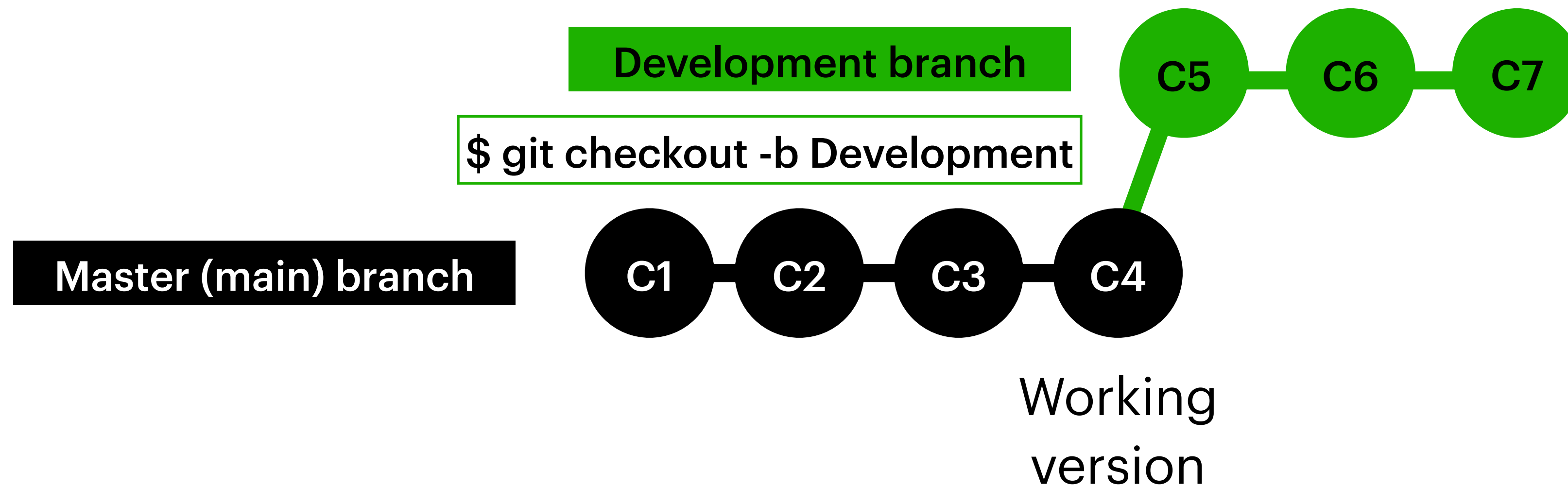


```
$ git switch Master
```

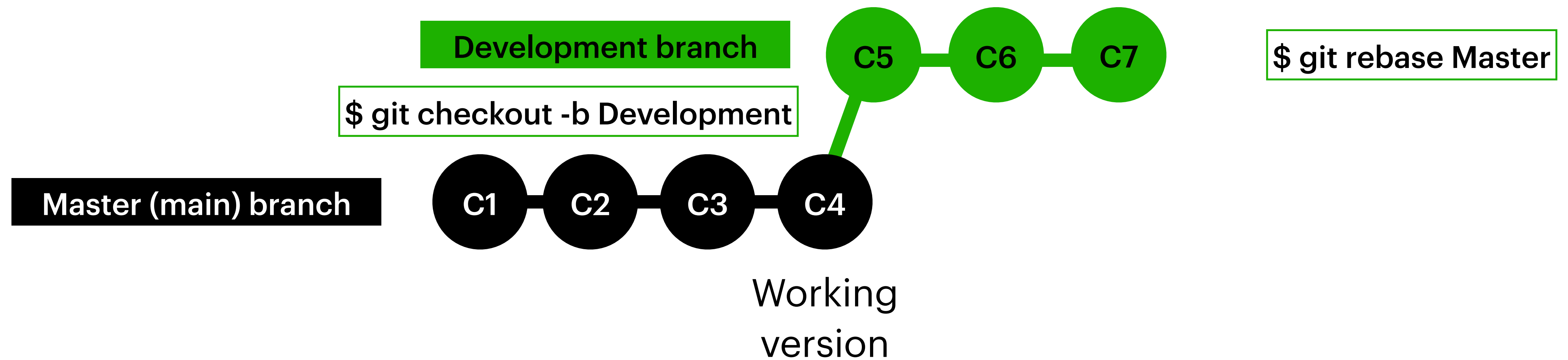
```
$ git merge Development
```

Hands-on

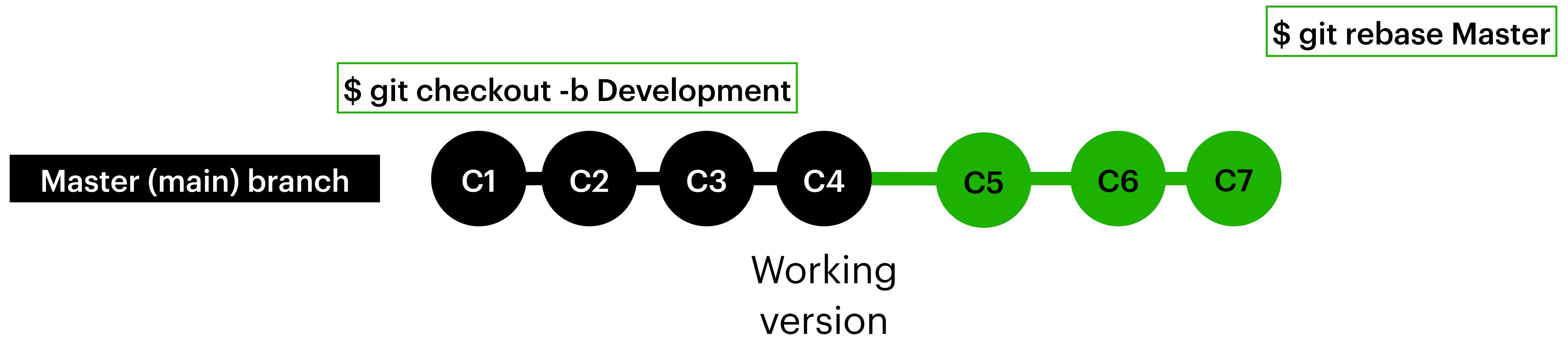
Branching Rebase



Branching Rebase

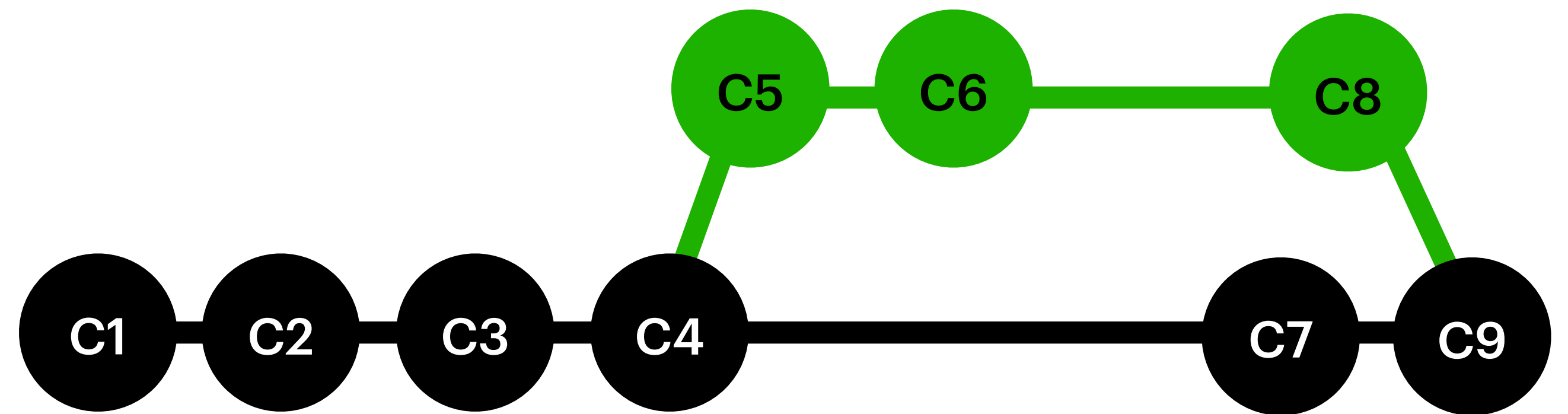


Branching Rebase



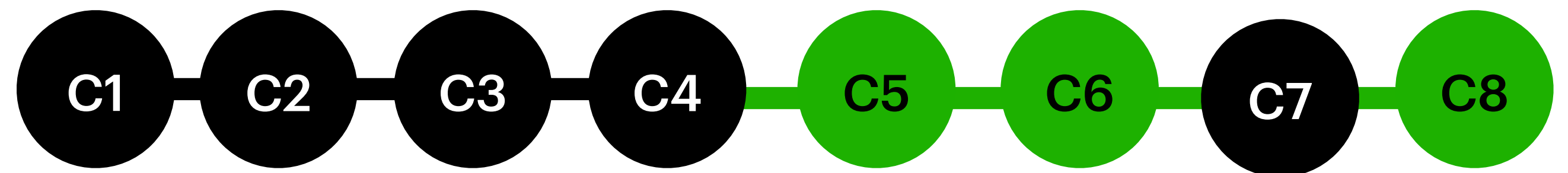
Rebase vs. Merge

- Simple and familiar
- Preserves complete history and chronological order
- Maintains the context of the branch

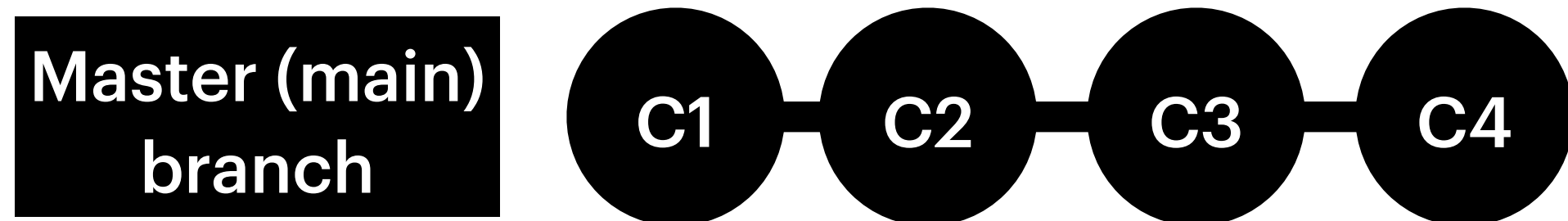


Comes down to team policy

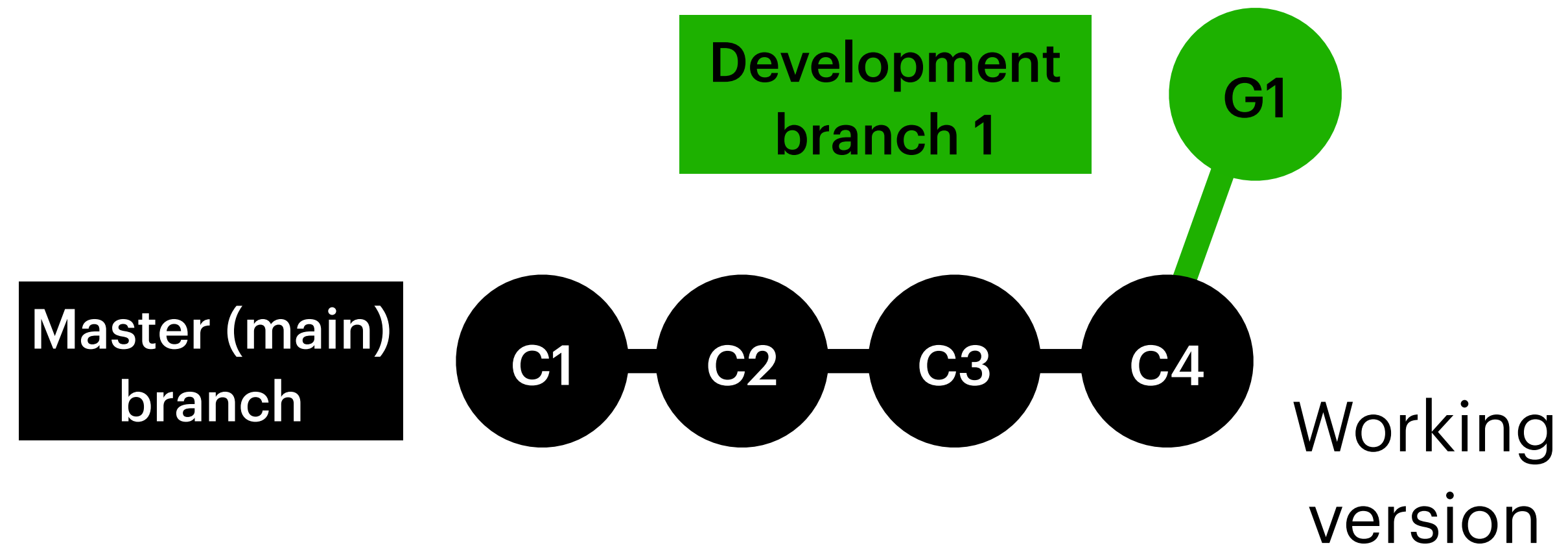
- Streamlines a potentially complex history
- Avoids merge commit “noise” in busy repos



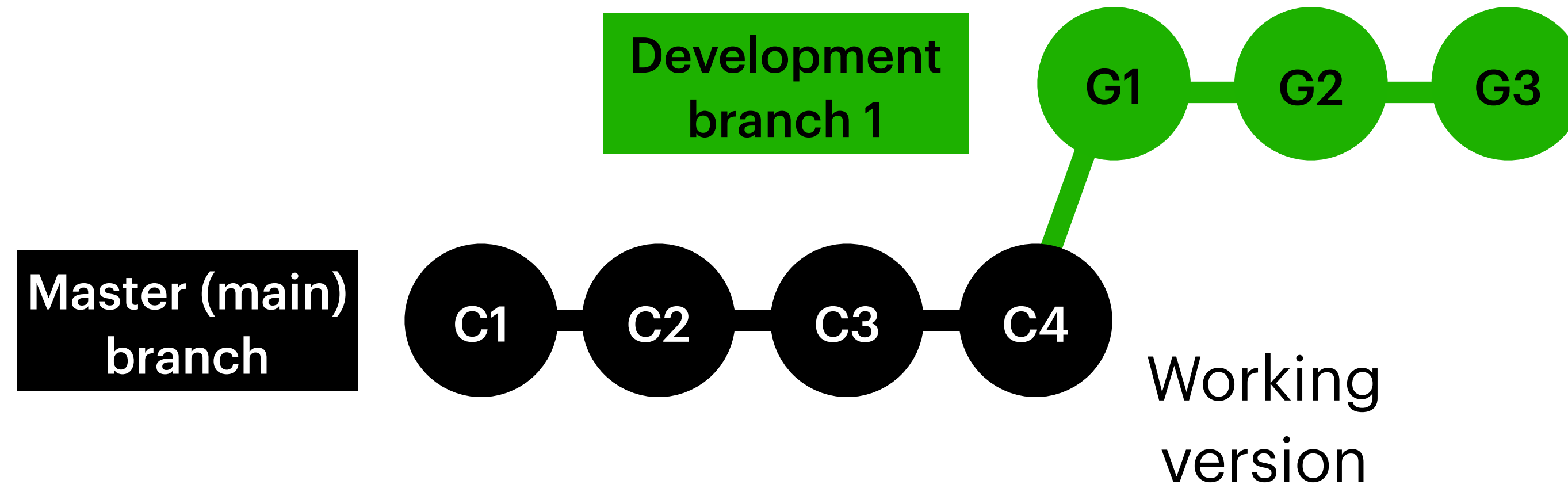
Branching



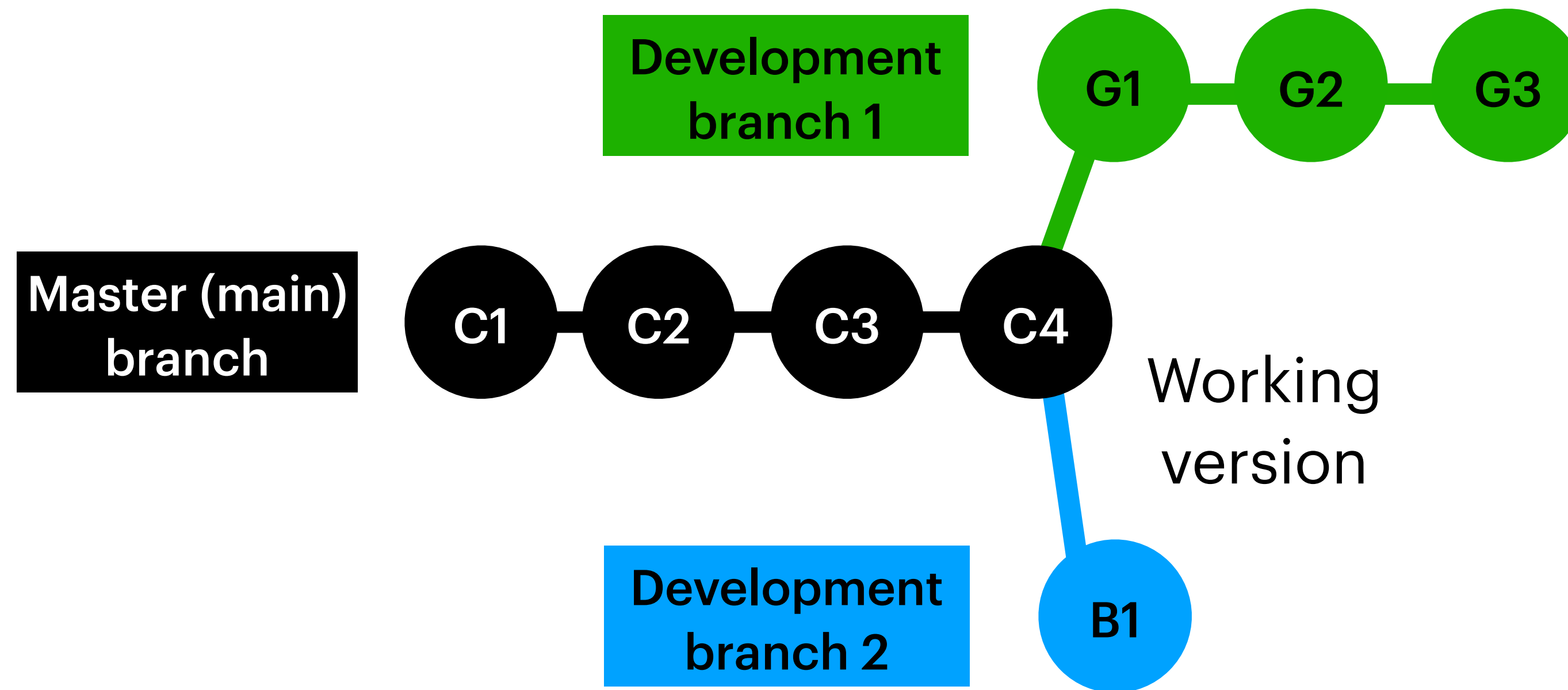
Branching



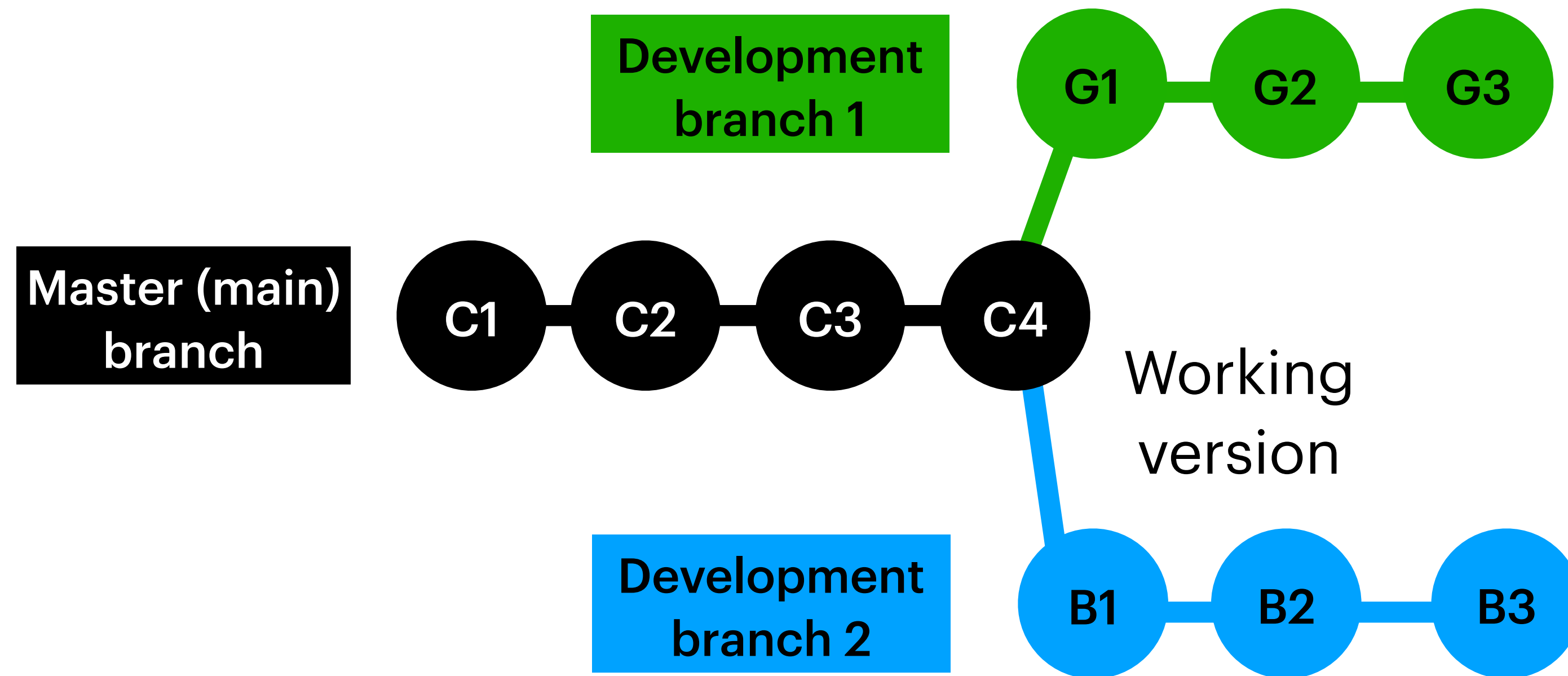
Branching



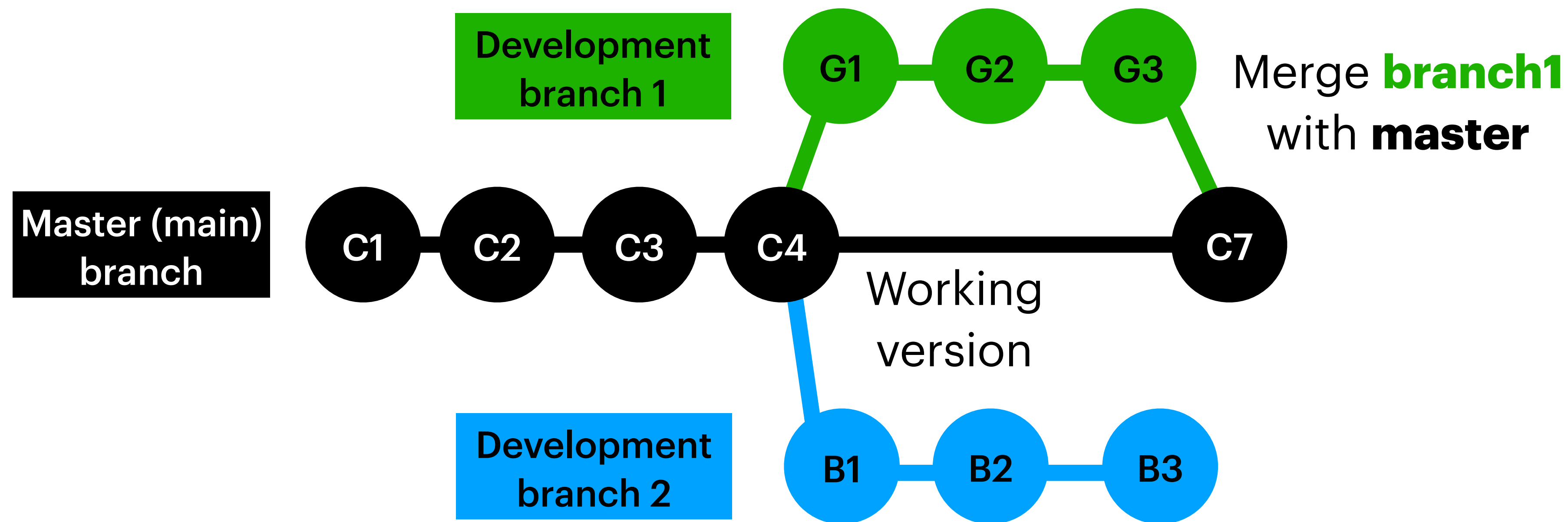
Branching



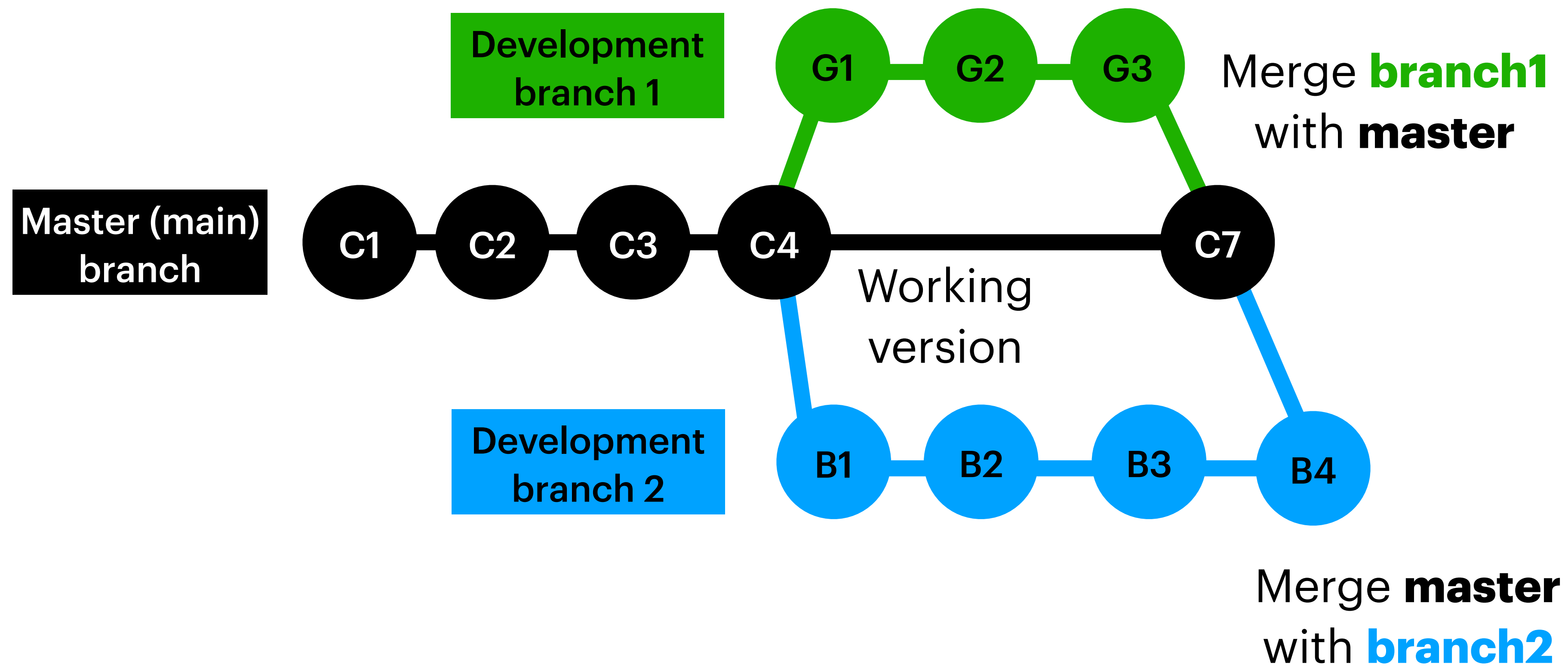
Branching



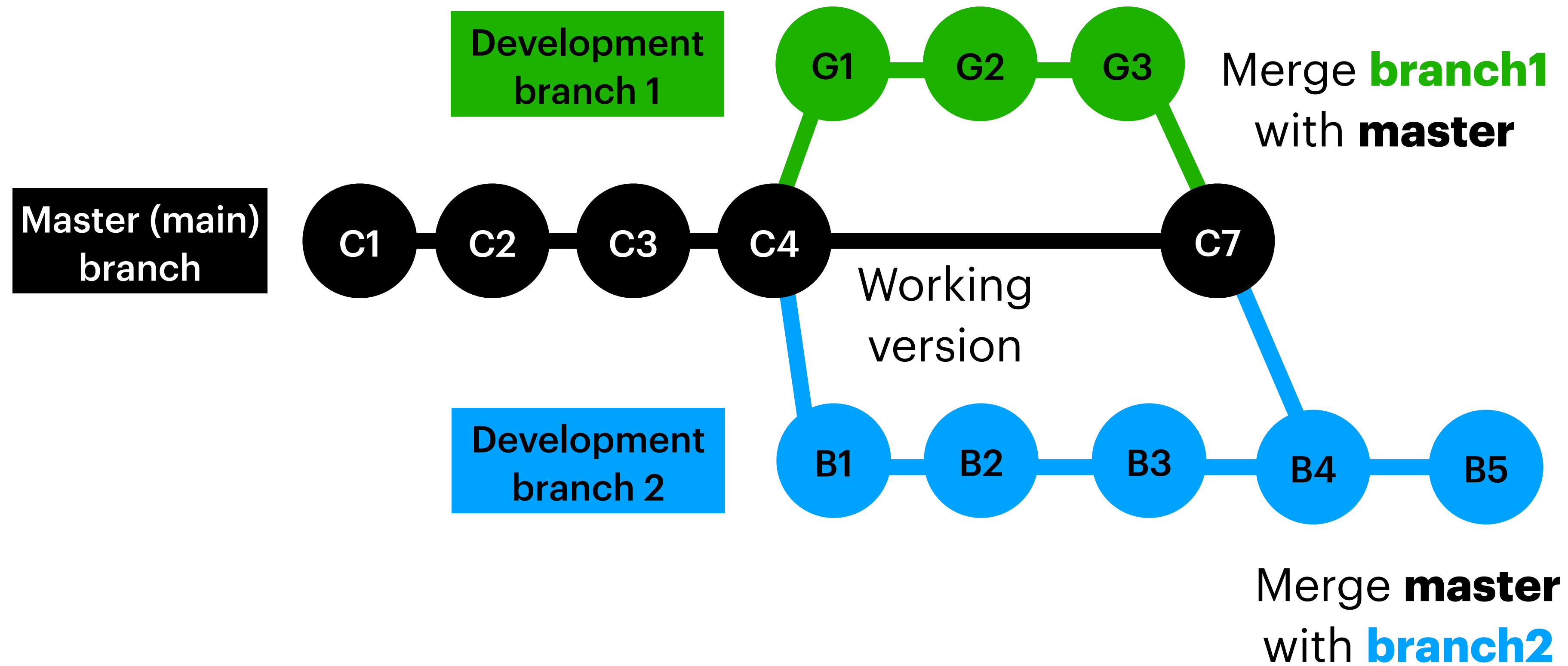
Branching



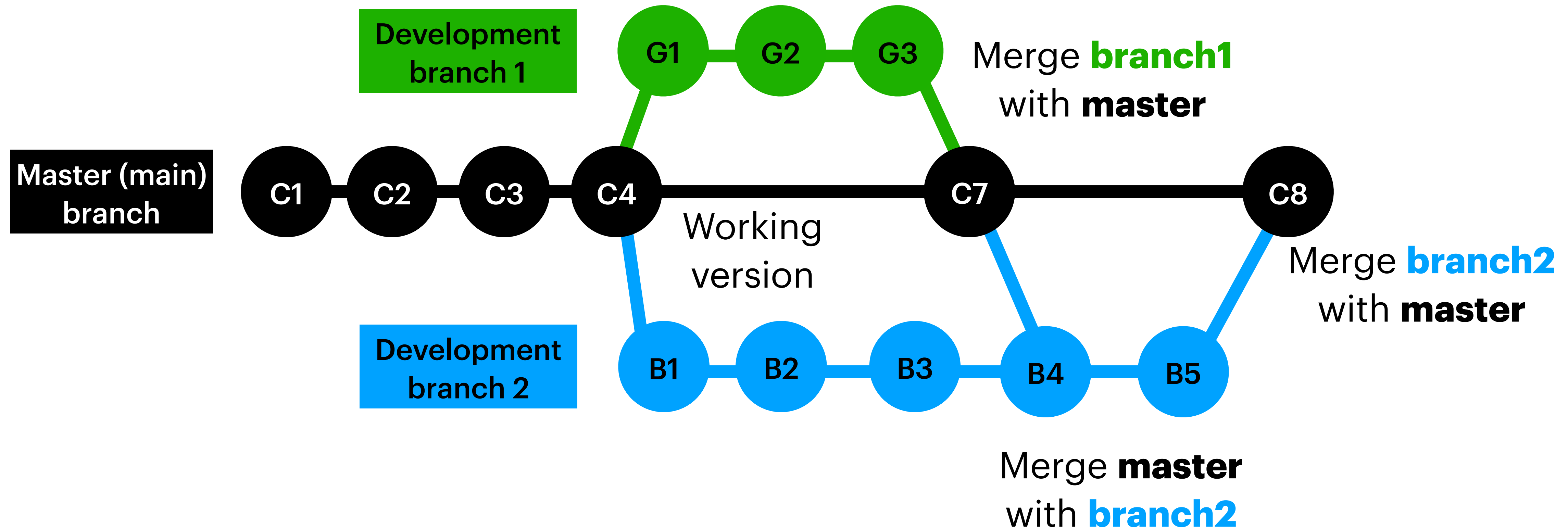
Branching



Branching



Branching



git is not perfect

git is not perfect

- Complex information model
 - It is complicated – and you need to know all of it
- Crazy command line syntax
- Documentation isn't good
- Burden of VCS maintenance pushed to contributors
- Unsafe version control

Rewrite history

git rebase

Rewrite history **only if you have to**