

Git

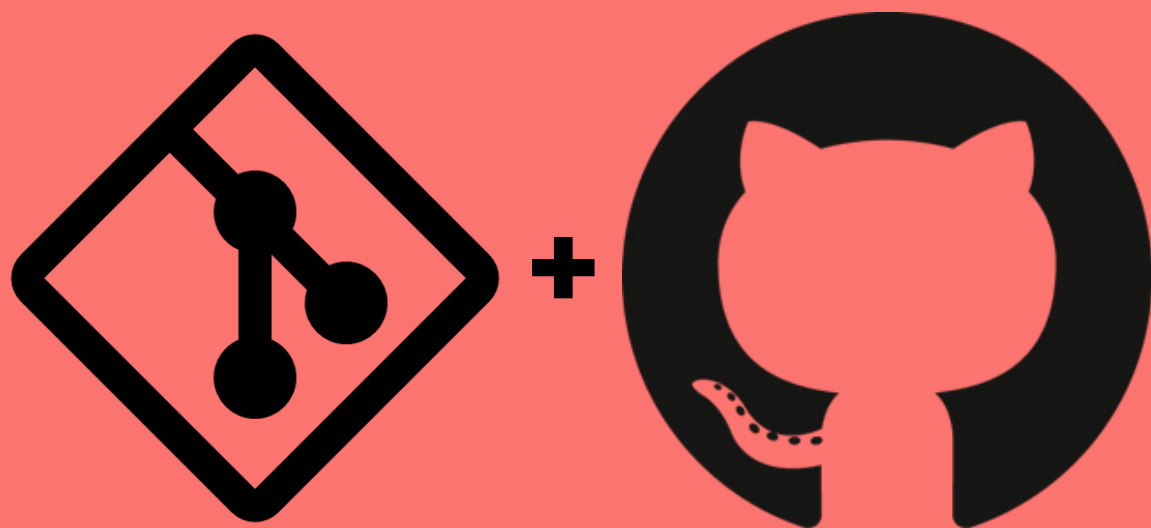
In Practice



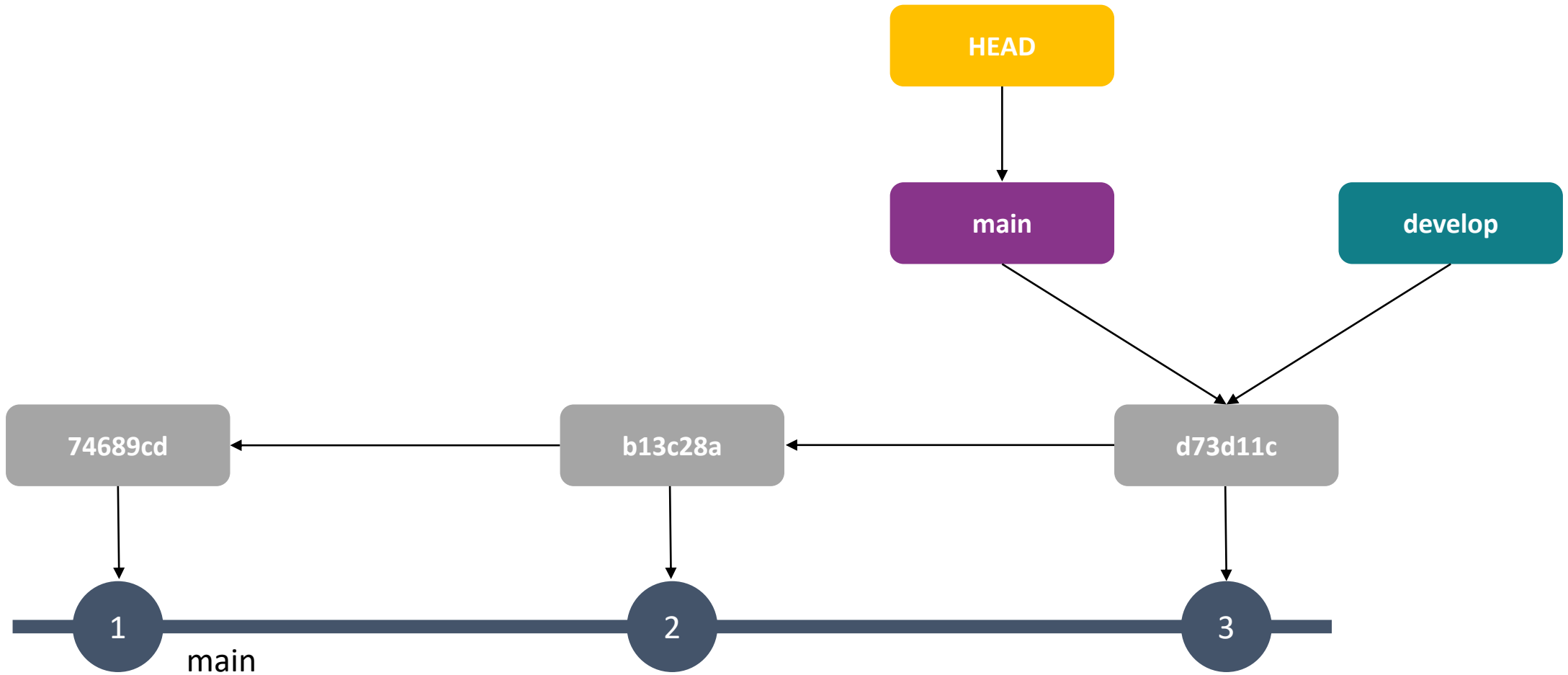
Git

Detached HEAD

In Practice

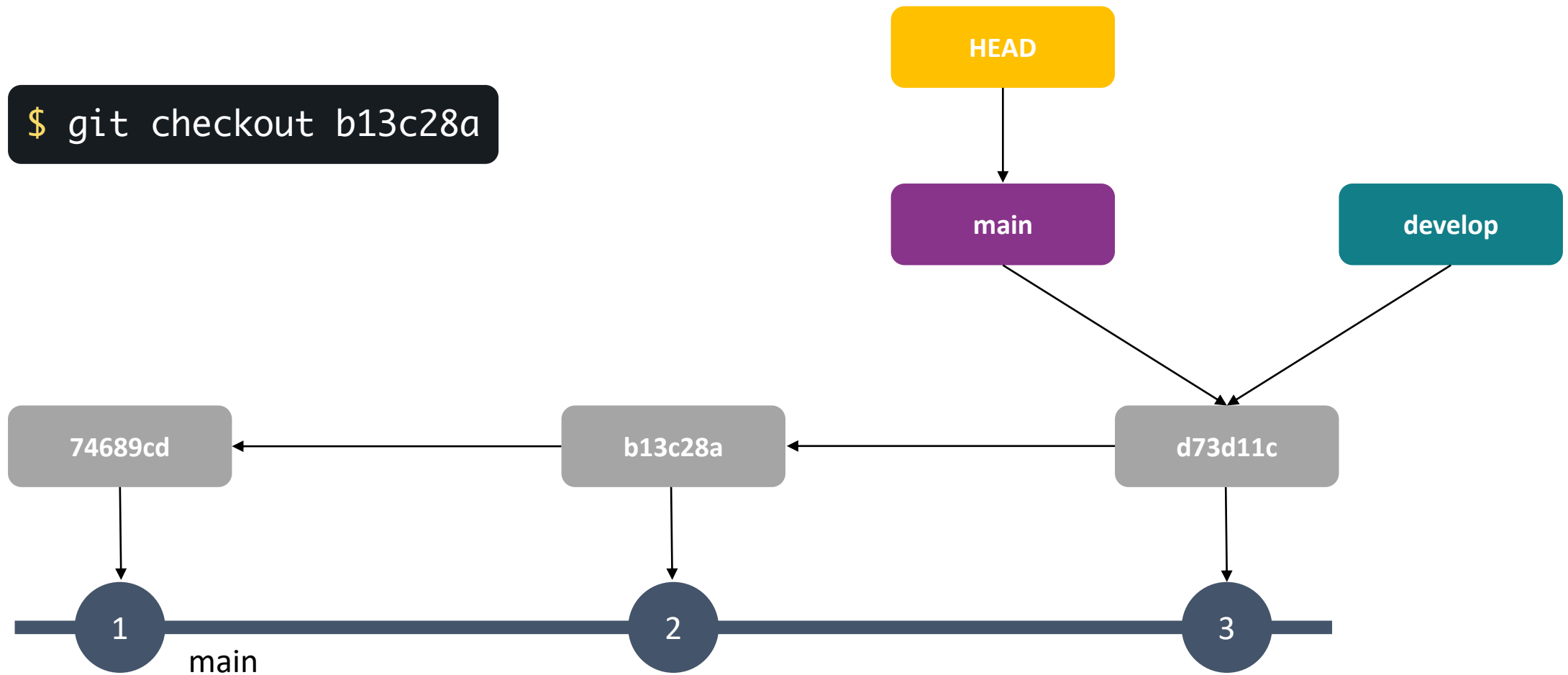


Detached HEAD



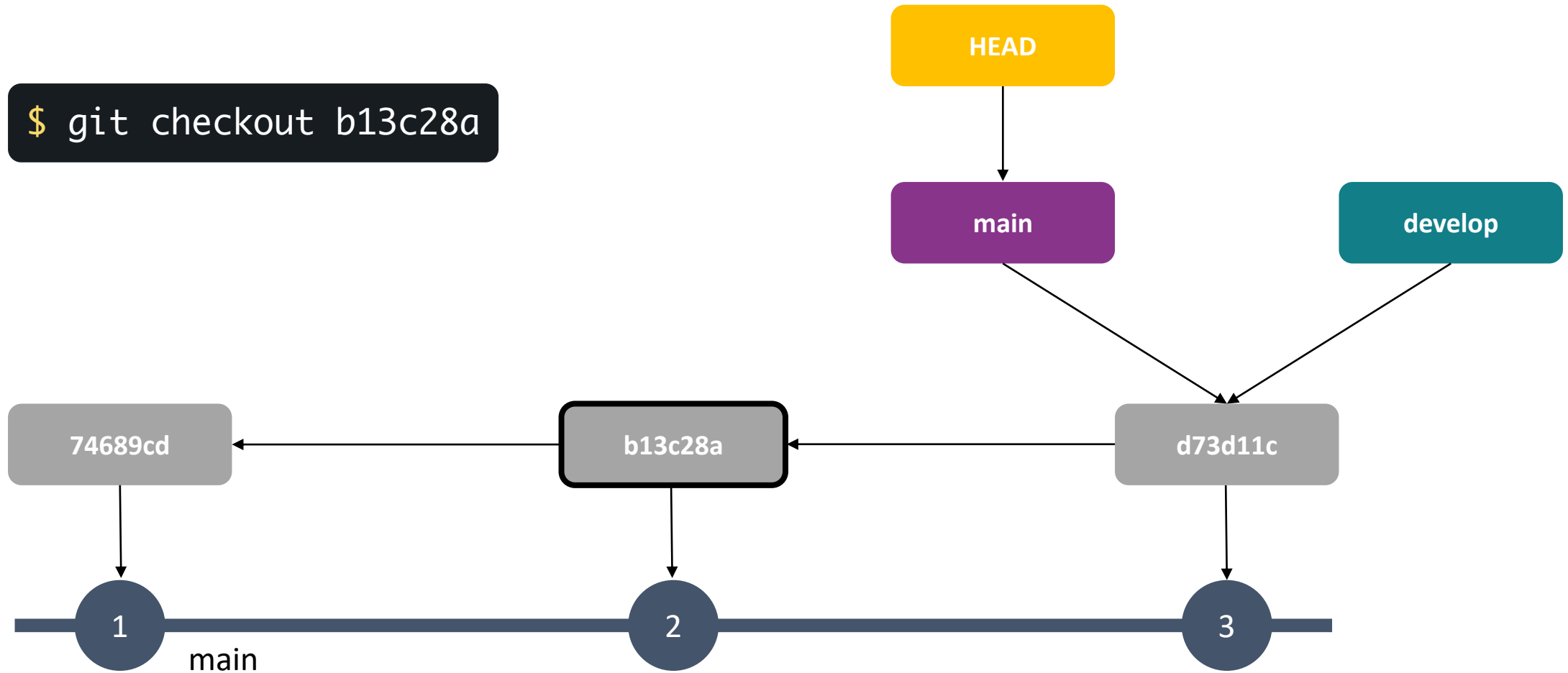
Detached HEAD

```
$ git checkout b13c28a
```

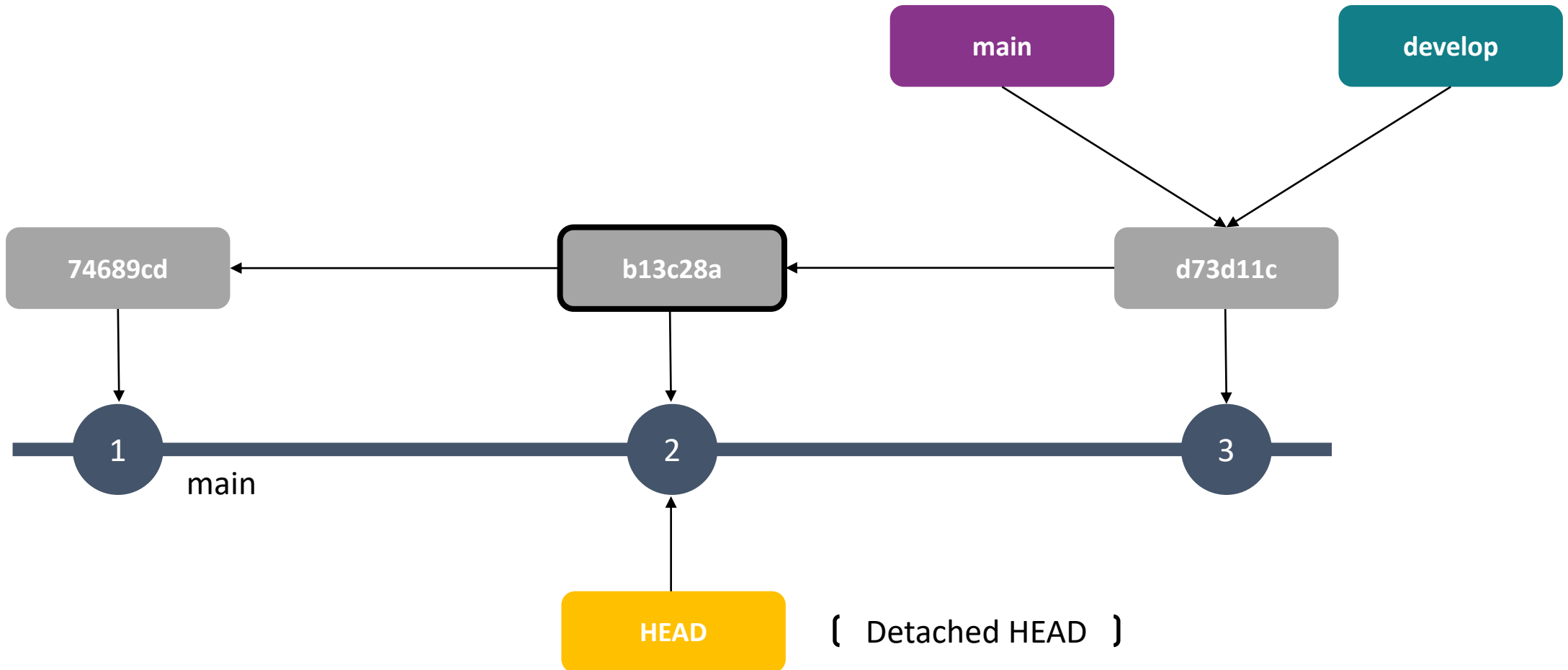


Detached HEAD

```
$ git checkout b13c28a
```



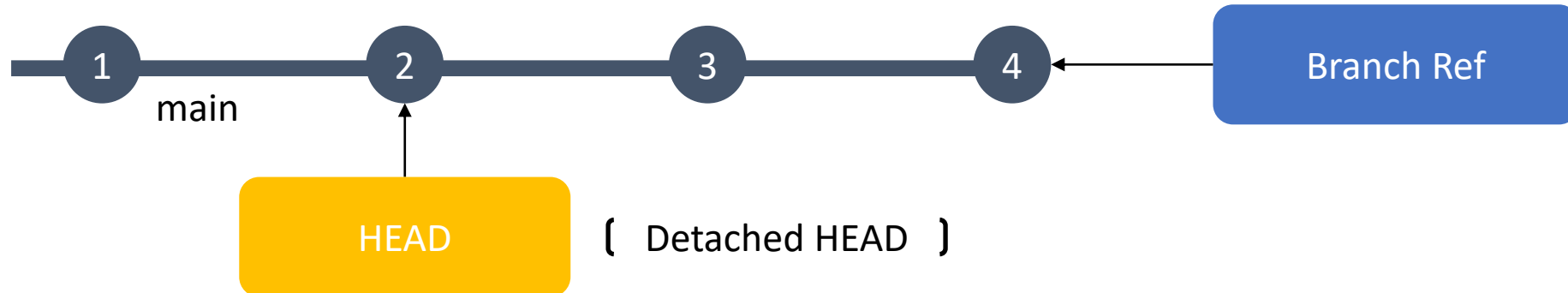
Detached HEAD



git checkout <commit ID>



```
git checkout 2 # commit ID 2
```



Detached HEAD – When?

1. `git checkout HEAD~1`
2. `git checkout 53ec1` (commit ID)

We will look into the below commands in the later sessions

1. `git checkout tags/v1.1`
2. `git rebase` (when conflicts happen)
3. Detached head is NOT a branch

HEAD And Branch Ref

```
.../.git (main)$ tree
.
├── COMMIT_EDITMSG
├── HEAD
├── config
├── description
├── hooks
│   ├── applypatch-msg.sample
│   ├── commit-msg.sample
│   ├── fsmonitor-watchman.sample
│   ├── post-update.sample
│   ├── pre-applypatch.sample
│   ├── pre-commit.sample
│   ├── pre-merge-commit.sample
│   ├── pre-push.sample
│   ├── pre-rebase.sample
│   ├── pre-receive.sample
│   ├── prepare-commit-msg.sample
│   ├── push-to-checkout.sample
│   └── update.sample
├── index
├── info
│   └── exclude
├── logs
│   ├── HEAD
│   └── refs
│       └── heads
│           ├── develop
│           └── main
├── objects
│   ├── 56
│   │   └── d202715b5e65a31440a70c
│   ├── 66
│   │   └── dac8d5243886b9ef0ce5c9
│   ├── b8
│   │   └── f40b56f689054b9f1dae1
│   ├── e6
│   │   └── 9de
│   ├── e9
│   │   └── 0b41de6e96ad76154fbf2
│   ├── info
│   └── pack
├── refs
│   ├── heads
│   │   ├── develop
│   │   └── main
│   └── tags
└── tags
```

e69de29be2d116434b8b29ae

(Detached HEAD)

\$ git checkout e69de29

X

Branch Ref

main

HEAD And Branch Ref Differ

```
.../.git (main)$ tree
.
├── COMMIT_EDITMSG
├── HEAD
├── config
├── description
├── hooks
│   ├── applypatch-msg.sample
│   ├── commit-msg.sample
│   ├── fsmonitor-watchman.sample
│   ├── post-update.sample
│   ├── pre-applypatch.sample
│   ├── pre-commit.sample
│   ├── pre-merge-commit.sample
│   ├── pre-push.sample
│   ├── pre-rebase.sample
│   ├── pre-receive.sample
│   ├── prepare-commit-msg.sample
│   ├── push-to-checkout.sample
│   └── update.sample
├── index
├── info
│   └── exclude
├── logs
│   ├── HEAD
│   └── refs
│       ├── heads
│       │   ├── develop
│       │   └── main
│       └── tags
├── objects
│   ├── 56
│   │   └── d202715b5e65a31440a70c
│   ├── 66
│   │   └── dac8d5243886b9ef0ce5c9
│   ├── b8
│   │   └── f40b56f689054b9f1deae1
│   ├── e6
│   │   └── 9de29bb2d1d6434b8b29ae
│   ├── e9
│   │   └── 0b41de6e96ad4b6154fbf2
│   └── info
├── pack
├── refs
│   ├── heads
│   │   ├── develop
│   │   └── main
│   └── tags
└── tags
```

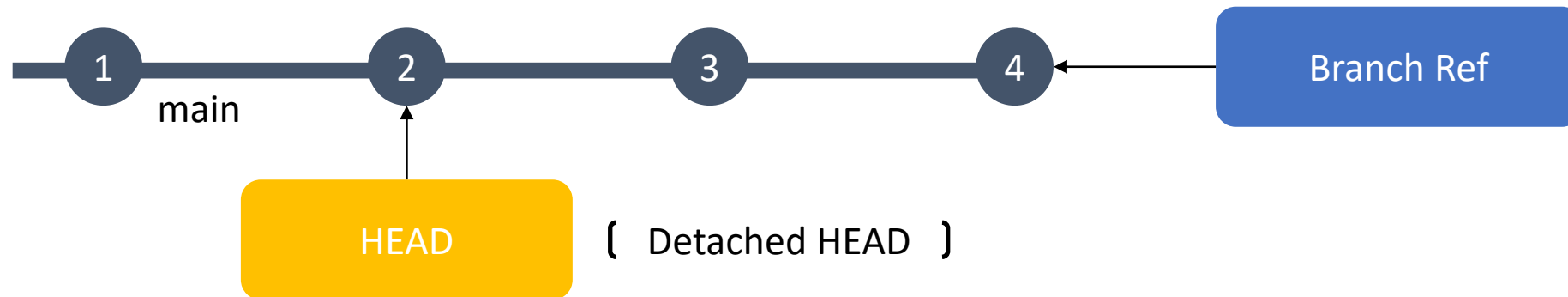
e69de29bb2d1d6434b8b29ae

[Detached HEAD]

```
$ git checkout e69de29
```

```
→ project-1$ cat ../.git/refs/heads/main
4e27ff85b94ff8dd773186bfa0e7e923e3346ce4 # last commit (previous HEAD)
```

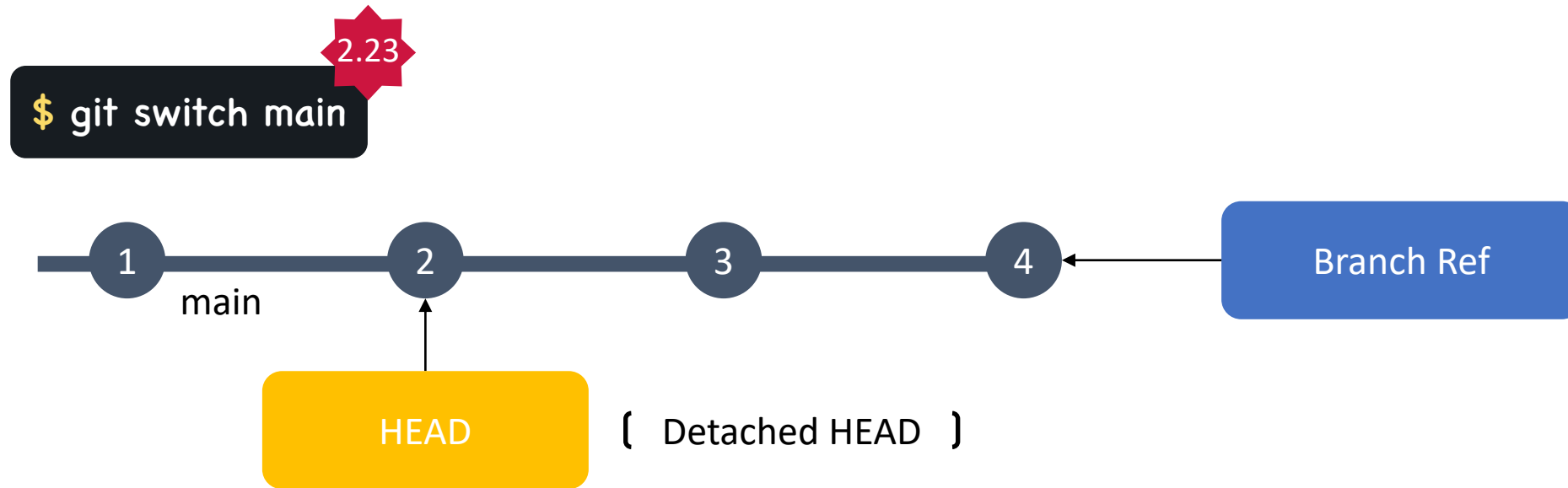
1. Detached HEAD – What to do?



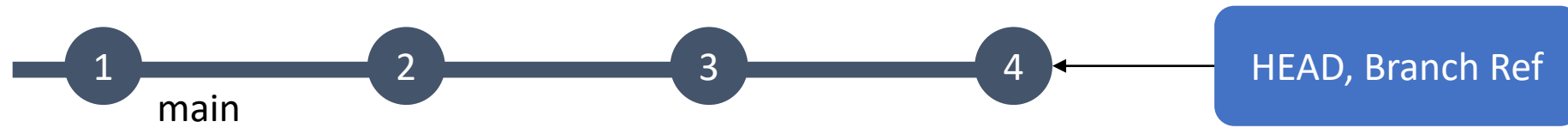
1. Discard the changes
 - a. In detached head by accident
 - b. Experiment didn't work

2. Keep the changes
 - a. Experiment worked

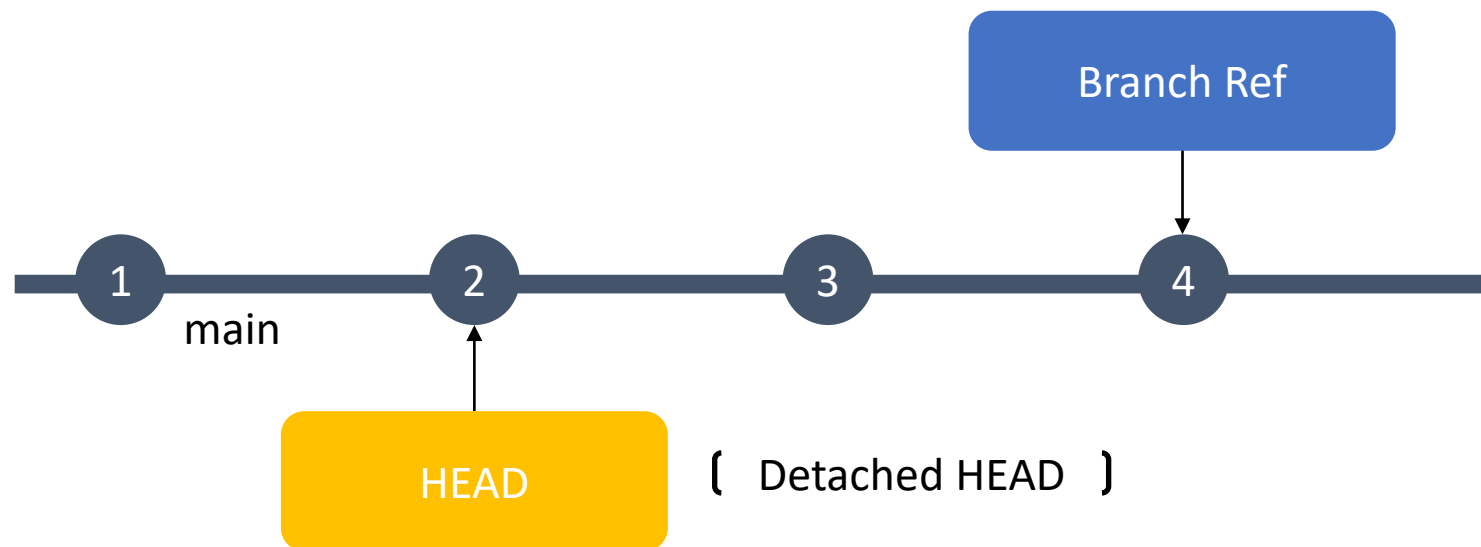
1. Discard Changes



1. Discard Changes

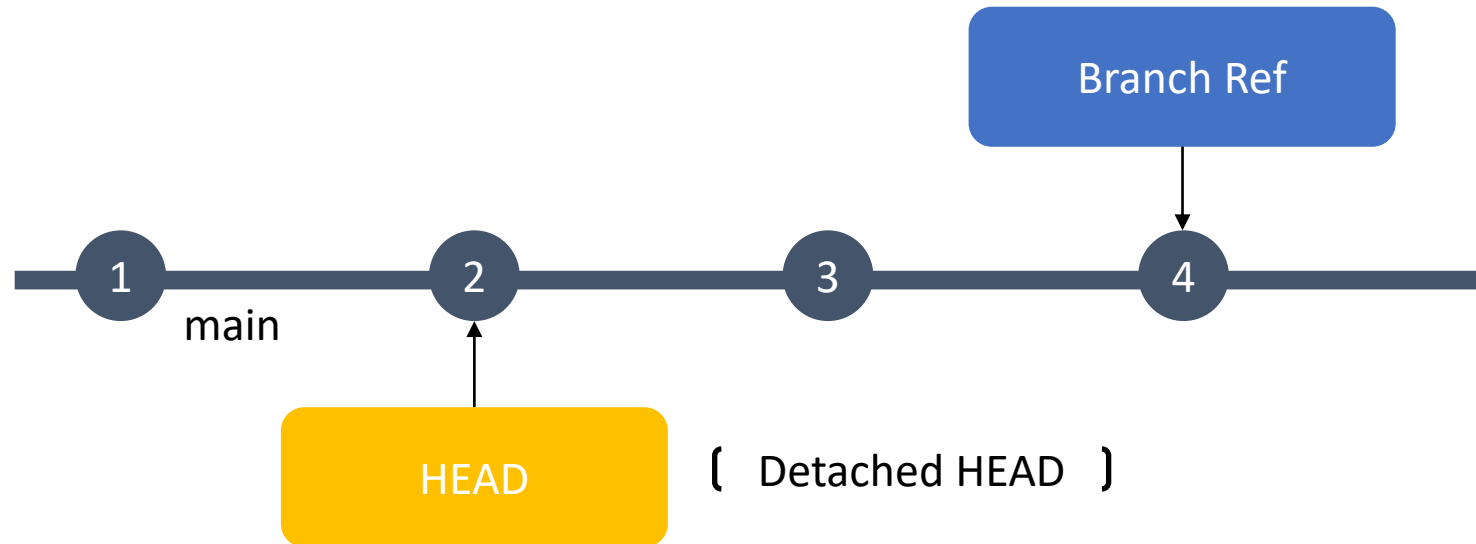


2. Keep Changes

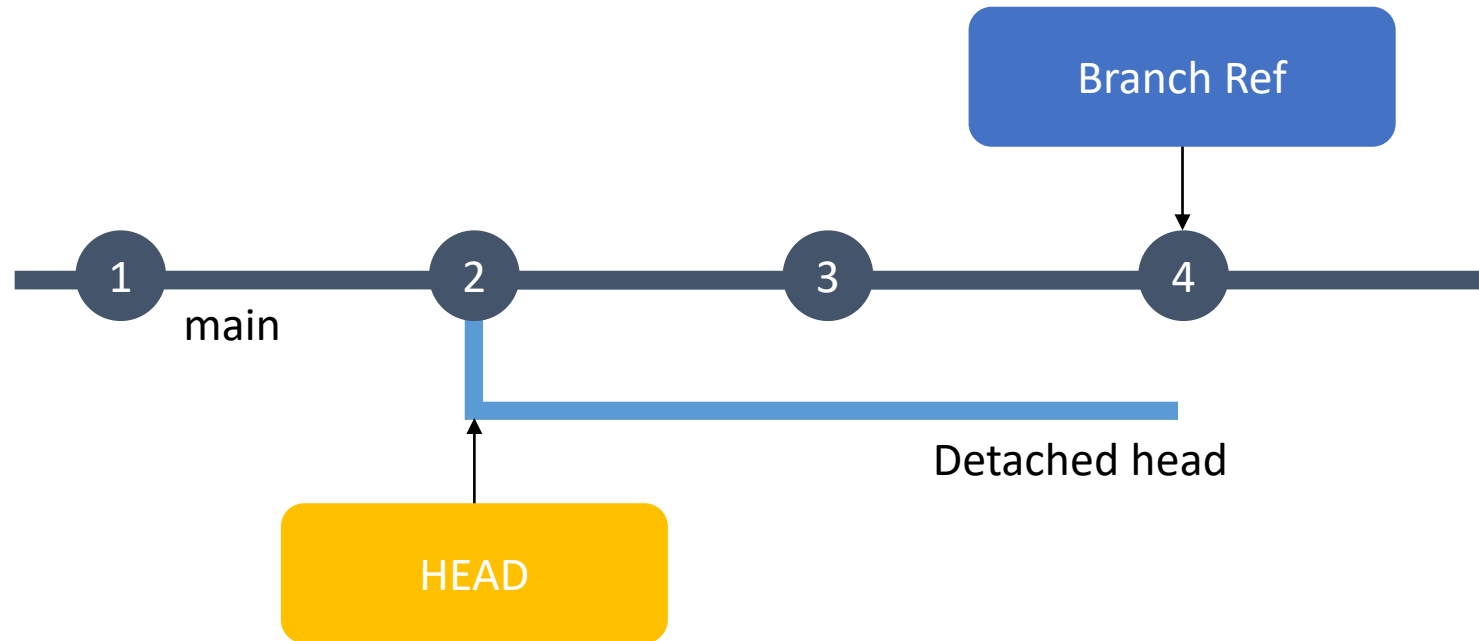


2. Keep Changes

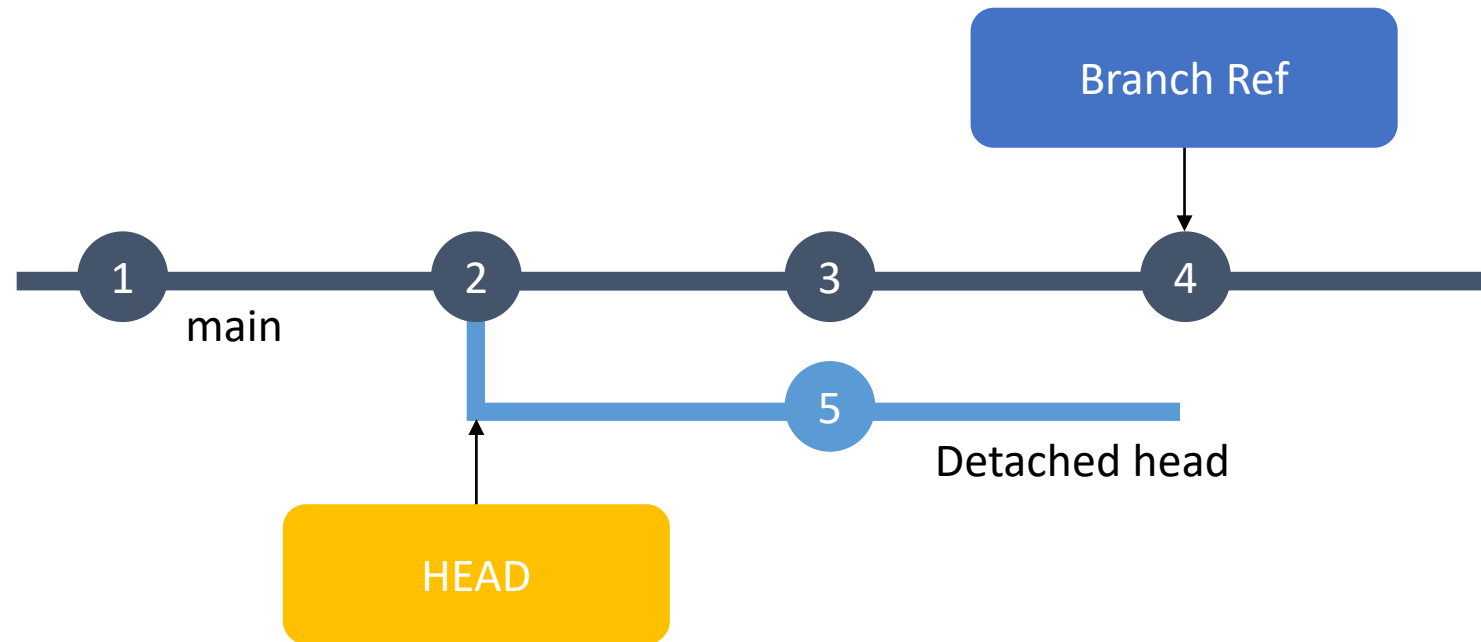
```
$ git commit -m "c5"
```



2. Keep Changes



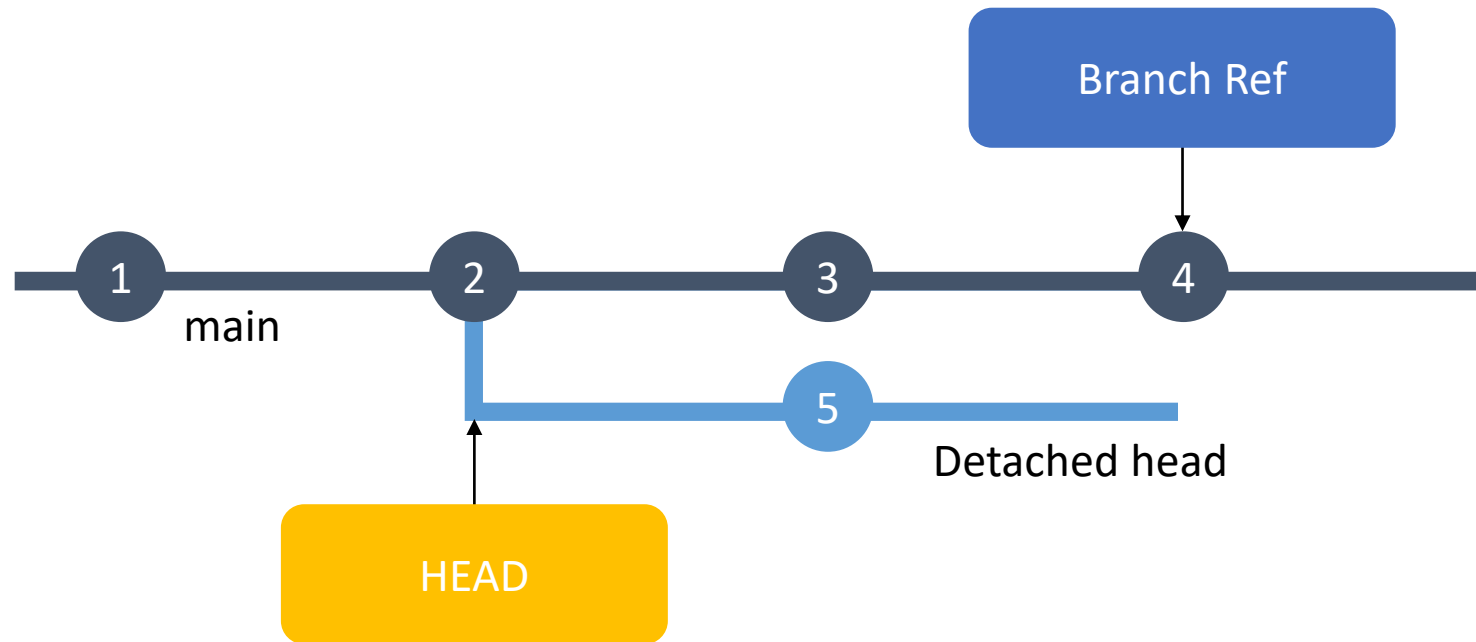
2. Keep Changes



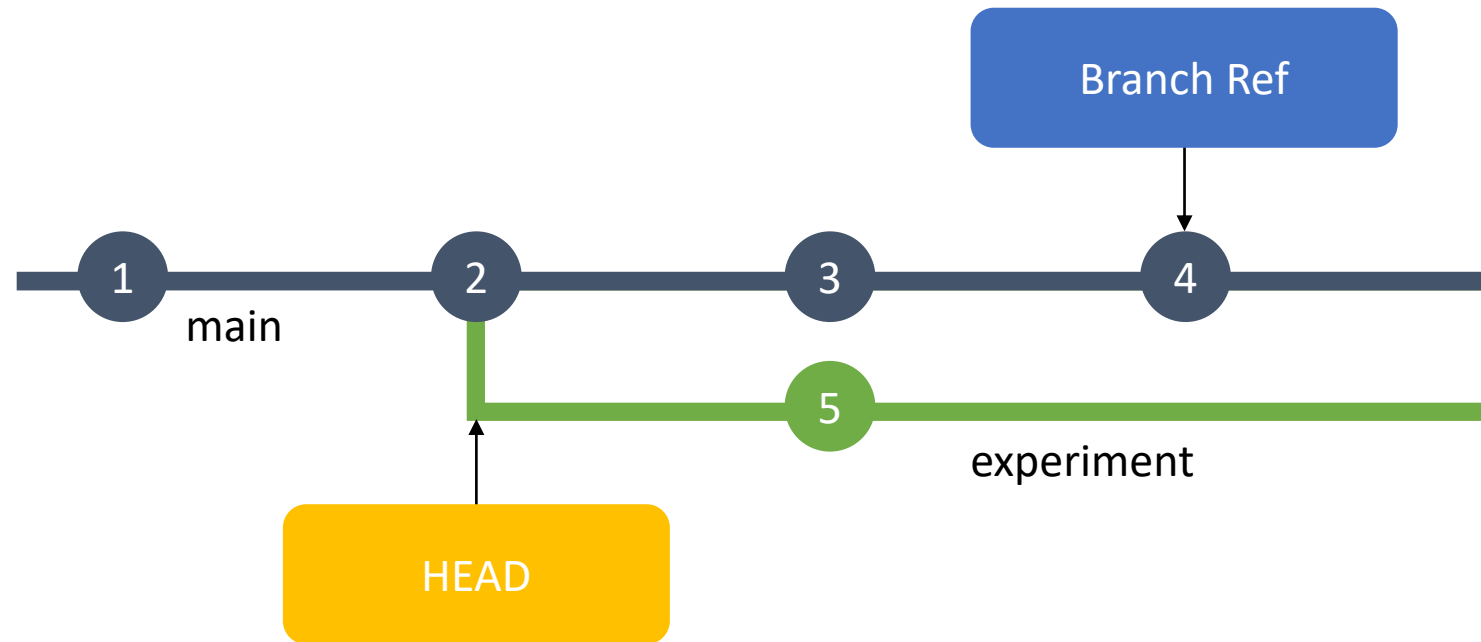
2. Keep Changes

2.23

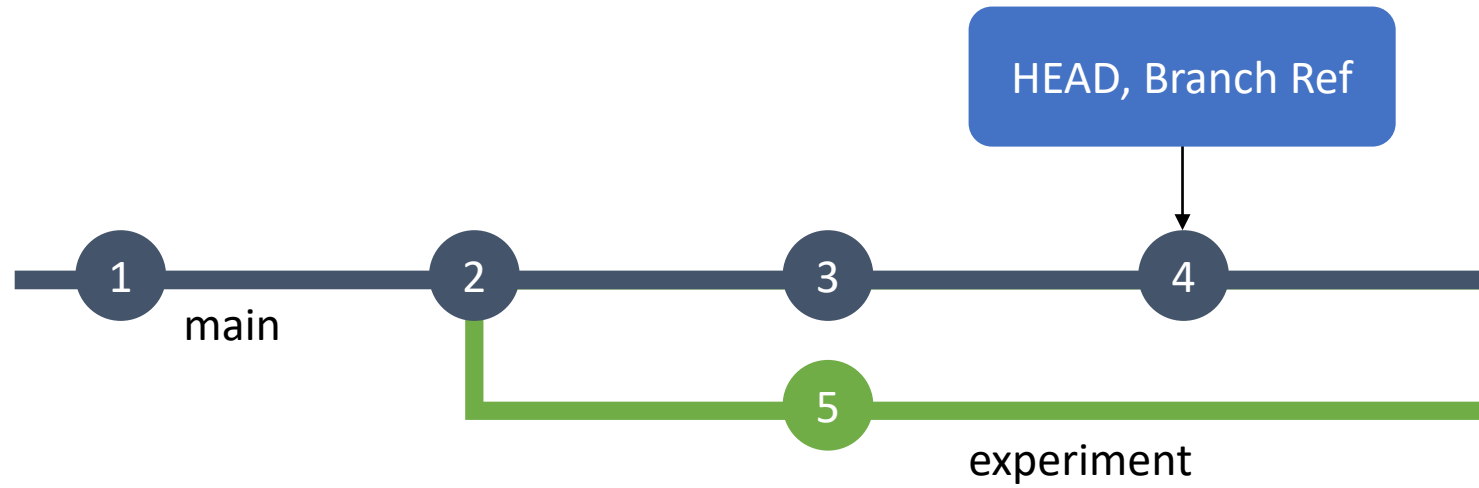
```
$ git switch -c experiment
```



2. Keep Changes

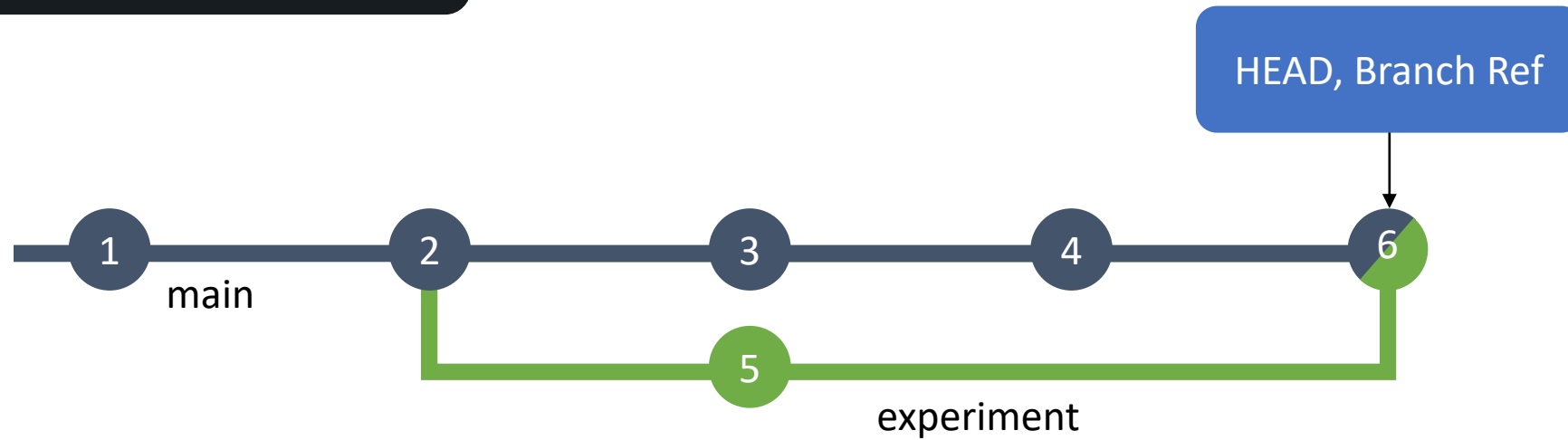


2. Keep Changes



2. Keep Changes

```
$ git switch main  
$ git merge experiment
```



Let us practice