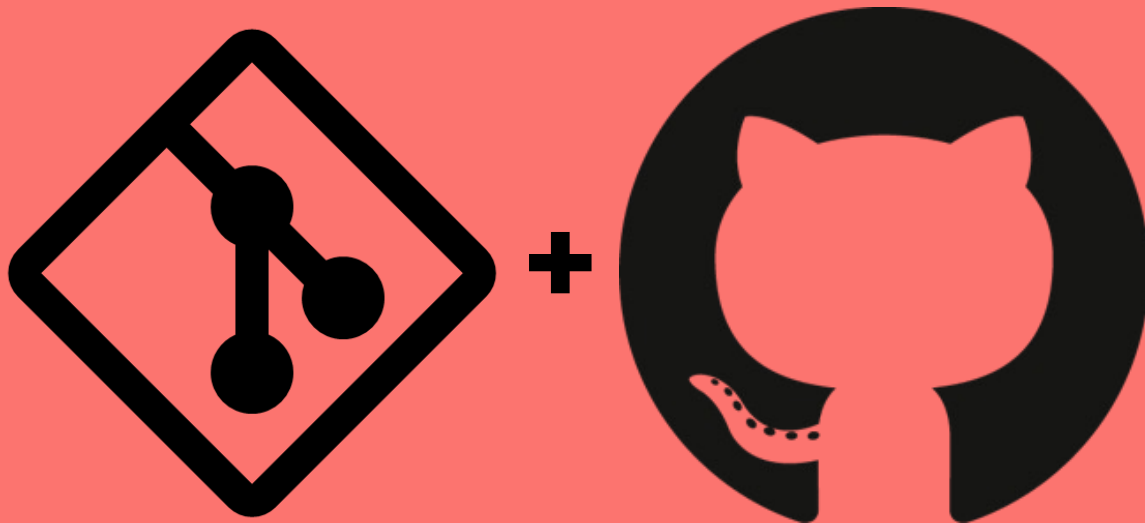


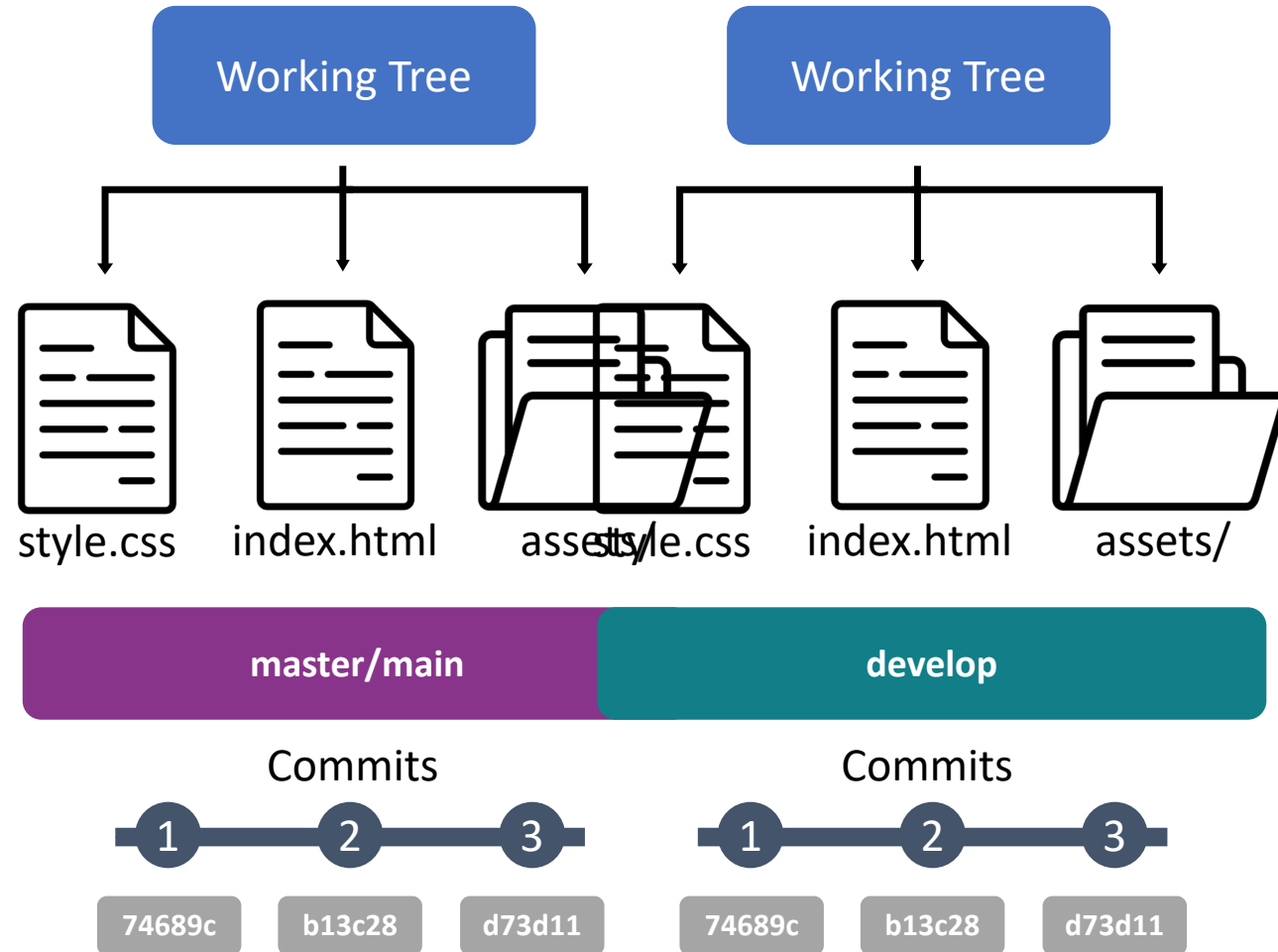
Git

Git Branches

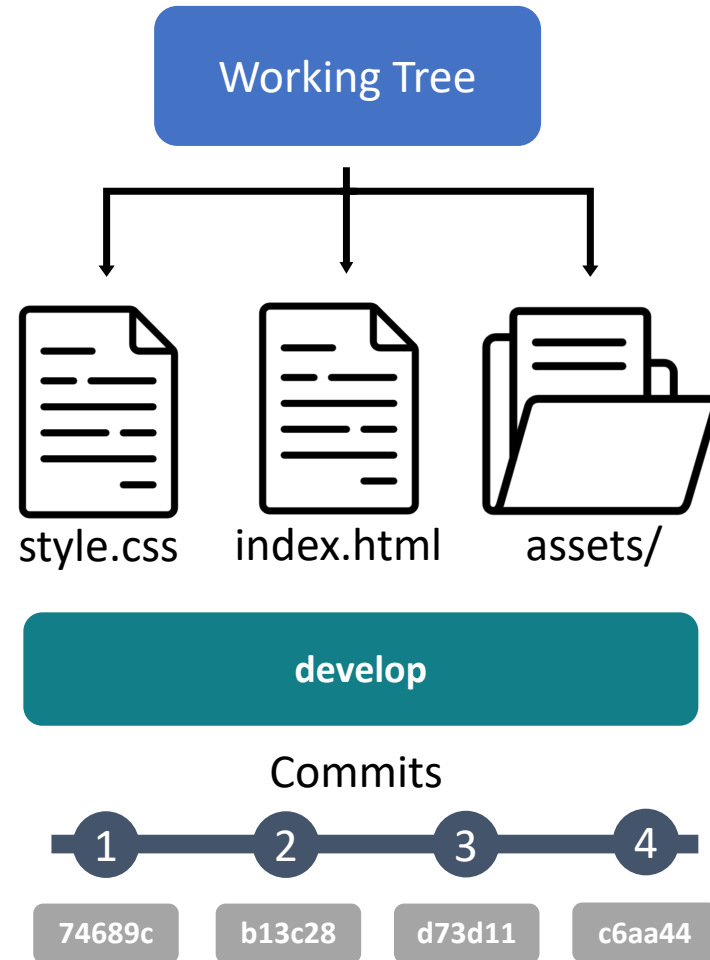
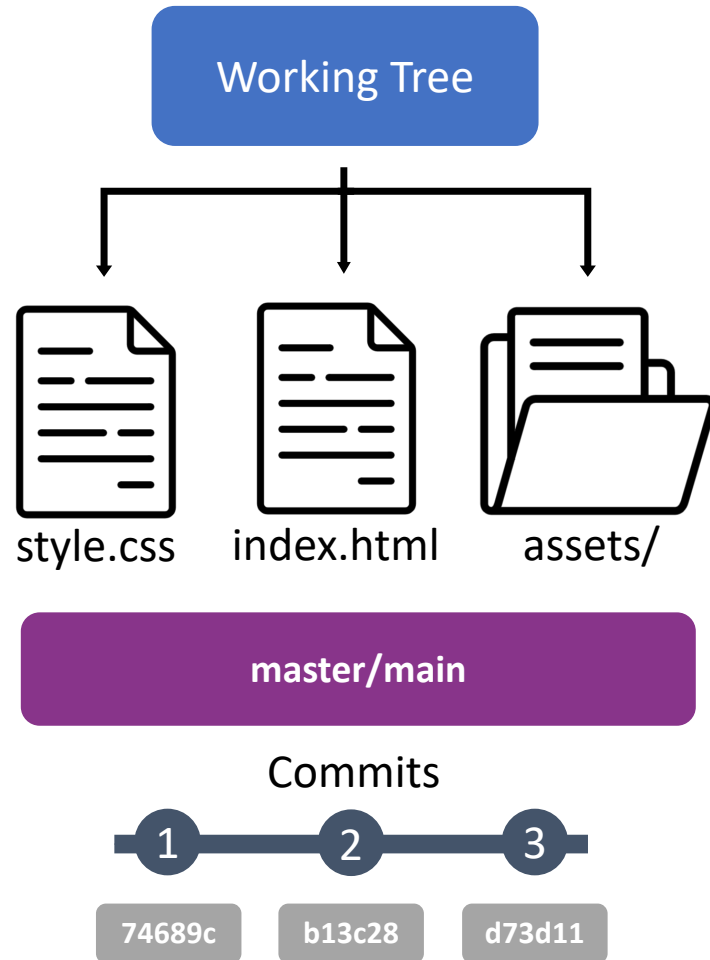
In Practice



What Is A Branch?



What Is A Branch?



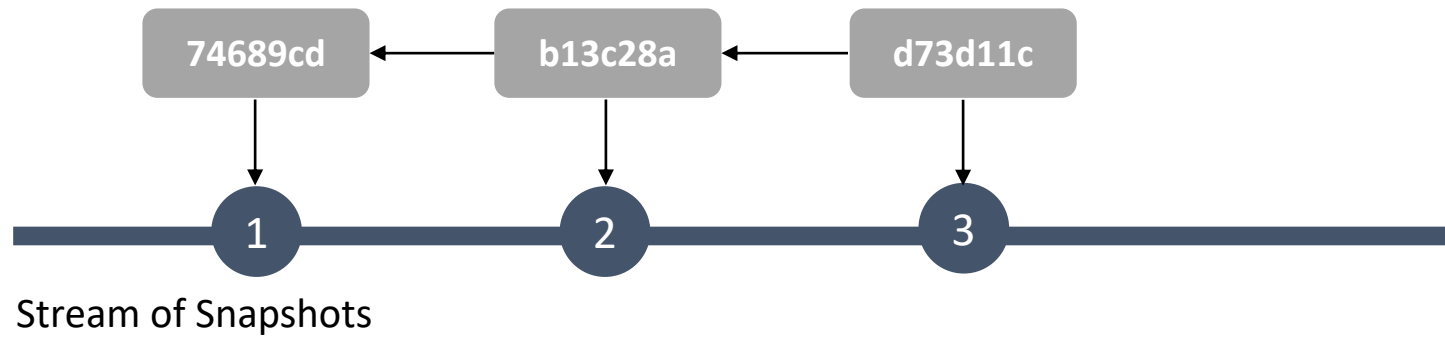
What Is A Branch?

1. A branch represents an independent line of development
2. It is a set of code changes grouped together with a unique name
3. The implementation behind Git branches is more lightweight than other version control systems

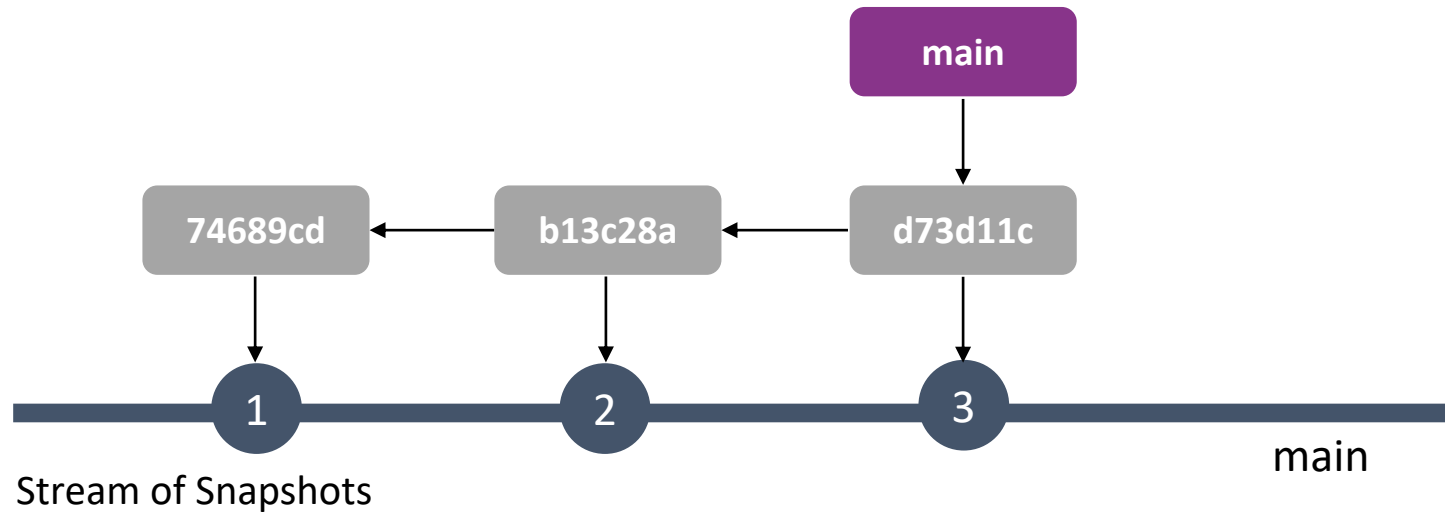
What Is A Branch?

- 4. Git branches are effectively a pointer to a snapshot of your changes
- 5. Instead of copying files one by one, Git stores a branch as a reference to a commit

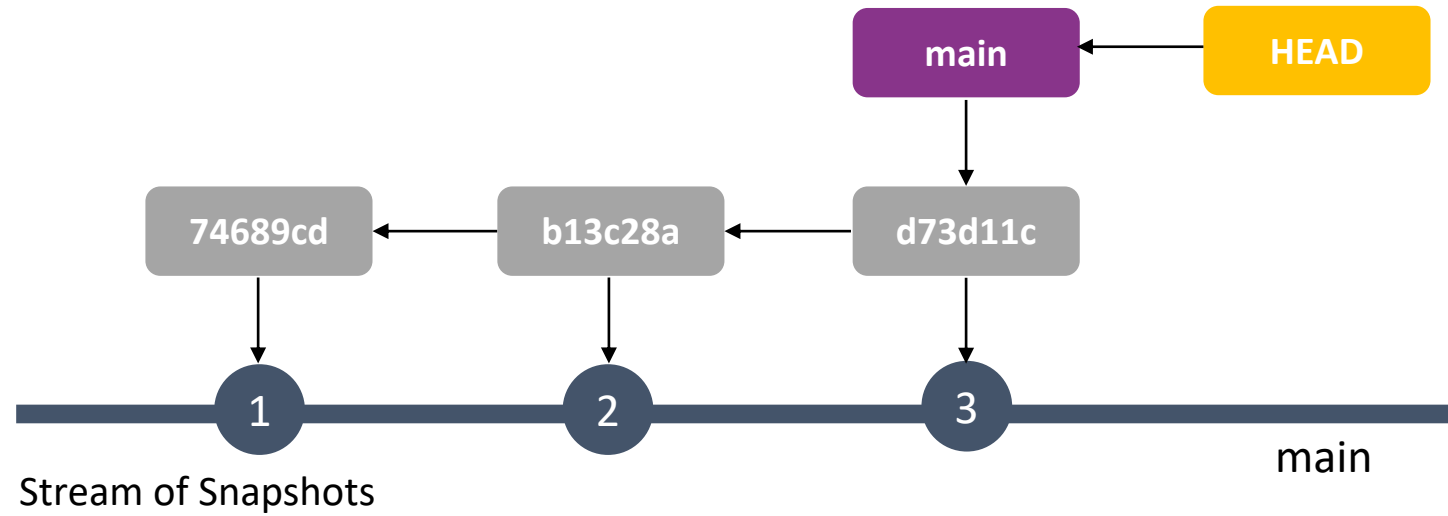
What Is A Branch?



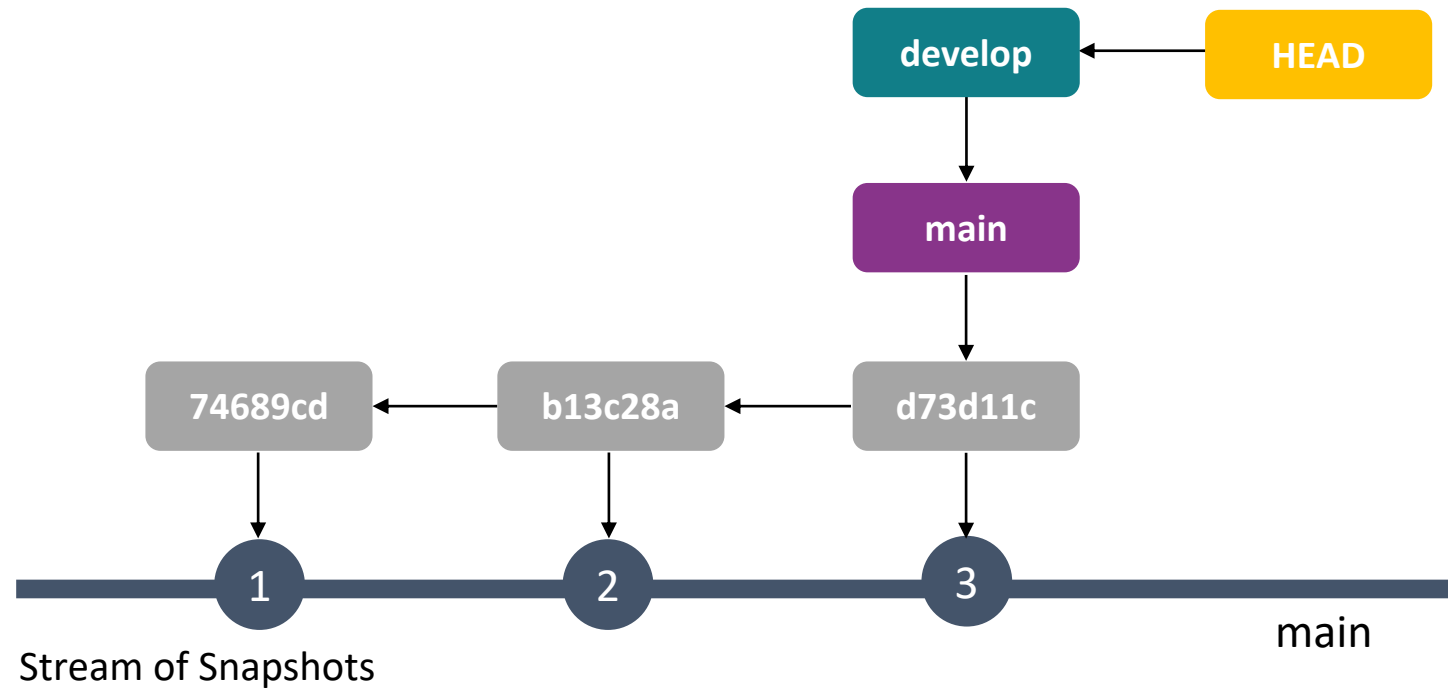
What Is A Branch?



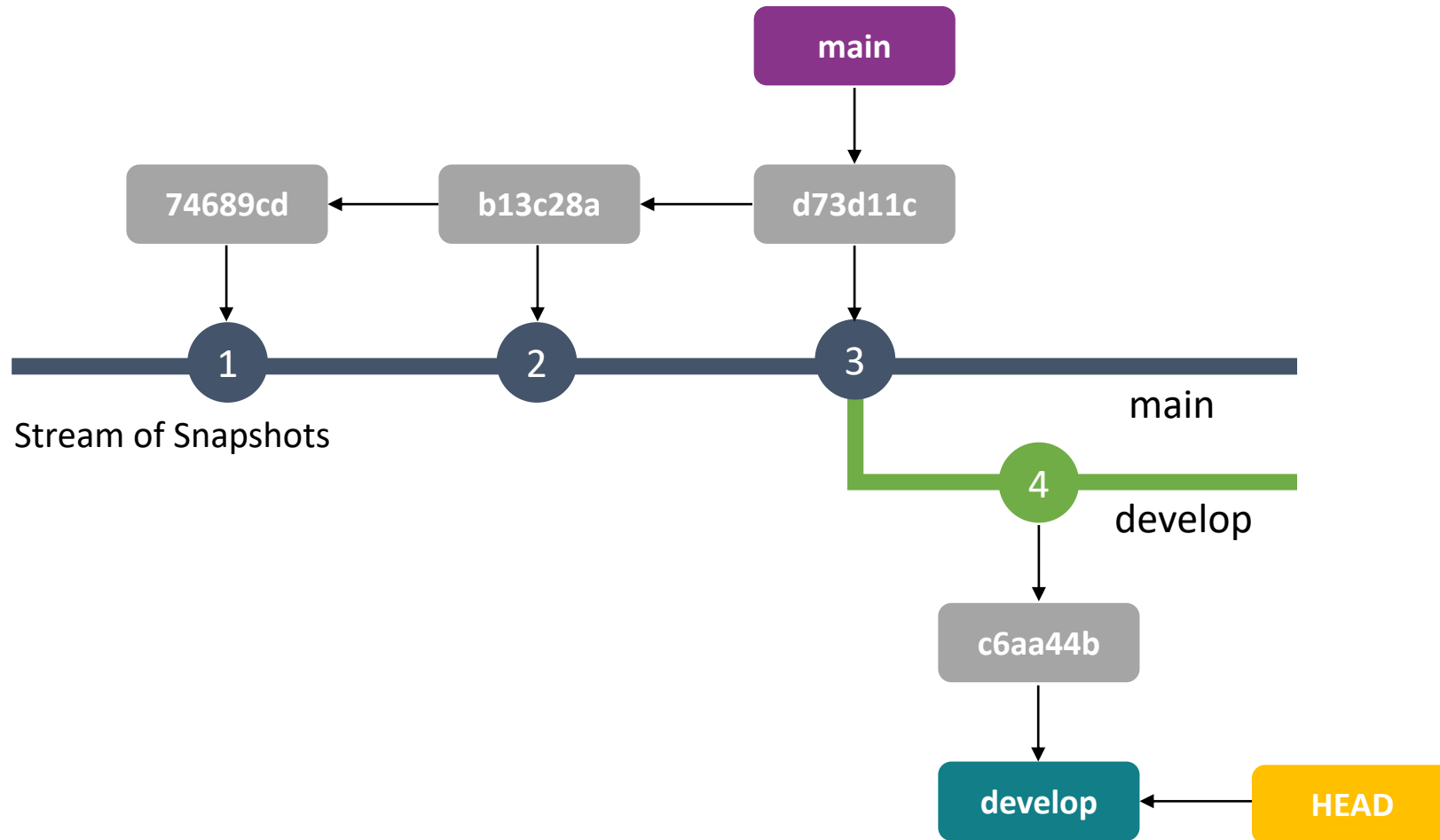
What Is A Branch?



What Is A Branch?



What Is A Branch?



HEAD

1. A HEAD is a reference which points to where you are now
2. HEAD points to either a specified branch or a commit
3. HEAD typically moves along with each new commit made in a branch

HEAD

4. Internally, HEAD is a file stored under the top level `.git/`
5. It contains the path to
 - a. a branch in `refs/` or
 - b. a commit under `objects/`
6. References are stored under `.git/refs/*/`

HEAD

```
.../.git (develop)$ tree
```

```
HEAD
```

ref: refs/heads/develop

```
→ project-1$ cat ../.git/refs/heads/develop
7e051346a7fc41a96d1fcdcb5300da38542b5c9f # commit ID
```

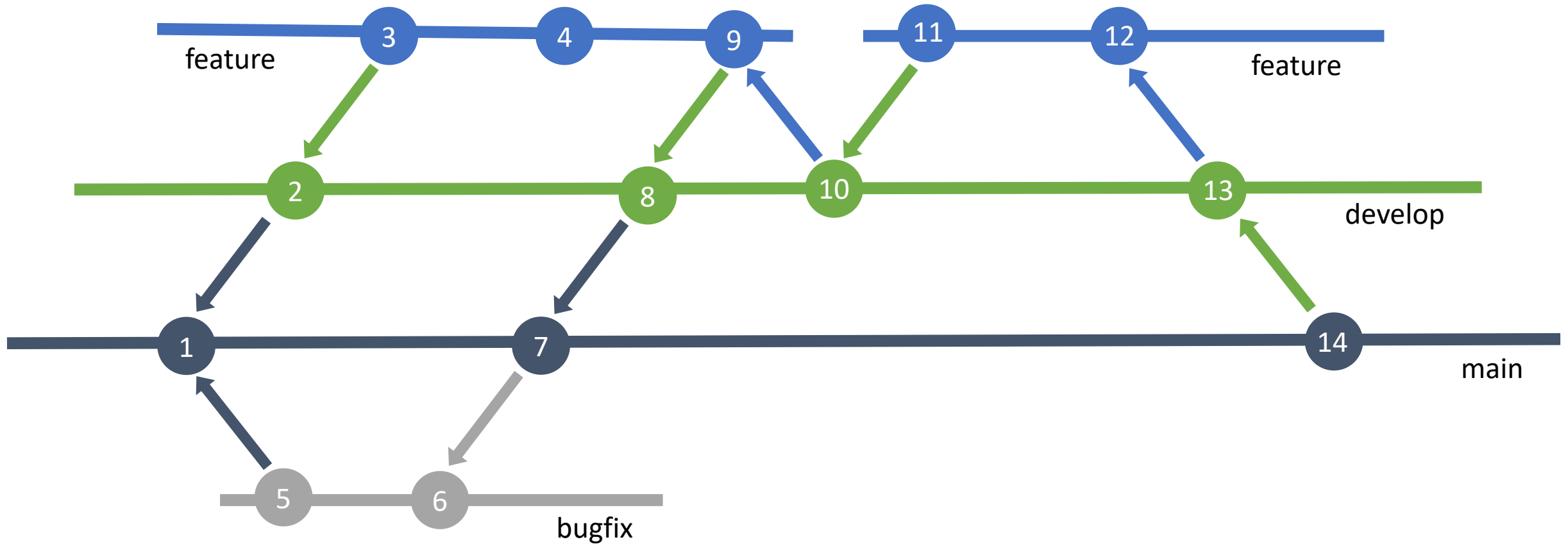
```
logs
├── HEAD
├── refs
│   └── heads
│       ├── develop
│       └── main
objects
├── 56
│   └── d202715b5e65a31440a70c1
├── 66
│   └── dac8d1243886b9ef0ce5c9l
├── b8
│   └── f40b51f689054b9f1deae1
├── e6
│   └── 9de291b2d1d6434b8b29ae
├── e9
│   └── 0b41d16e96ad4b6154fbf2
├── info
└── pack
```

```
refs
├── heads
│   ├── develop
│   └── main
└── tags
```

HEAD

HEAD refs under .git/	When
HEAD	Created by default. Contains the currently checked-out branch or commit
ORIG_HEAD	Created before drastic changes to history happen— e.g., reset, rebase. Contains a backup reference to previous state of HEAD
FETCH_HEAD	Created when fetching from a remote repo. Contains the most recently fetched branch
MERGE_HEAD	Created during a merge. Contains the commits (<i>other</i> commit) to be merged into the current branch
CHERRYPICK_HEAD	Created during a cherrypick. Contains the commit ID to be cherry picked

A Typical “Branched” Repo



Creating A Branch

2.23

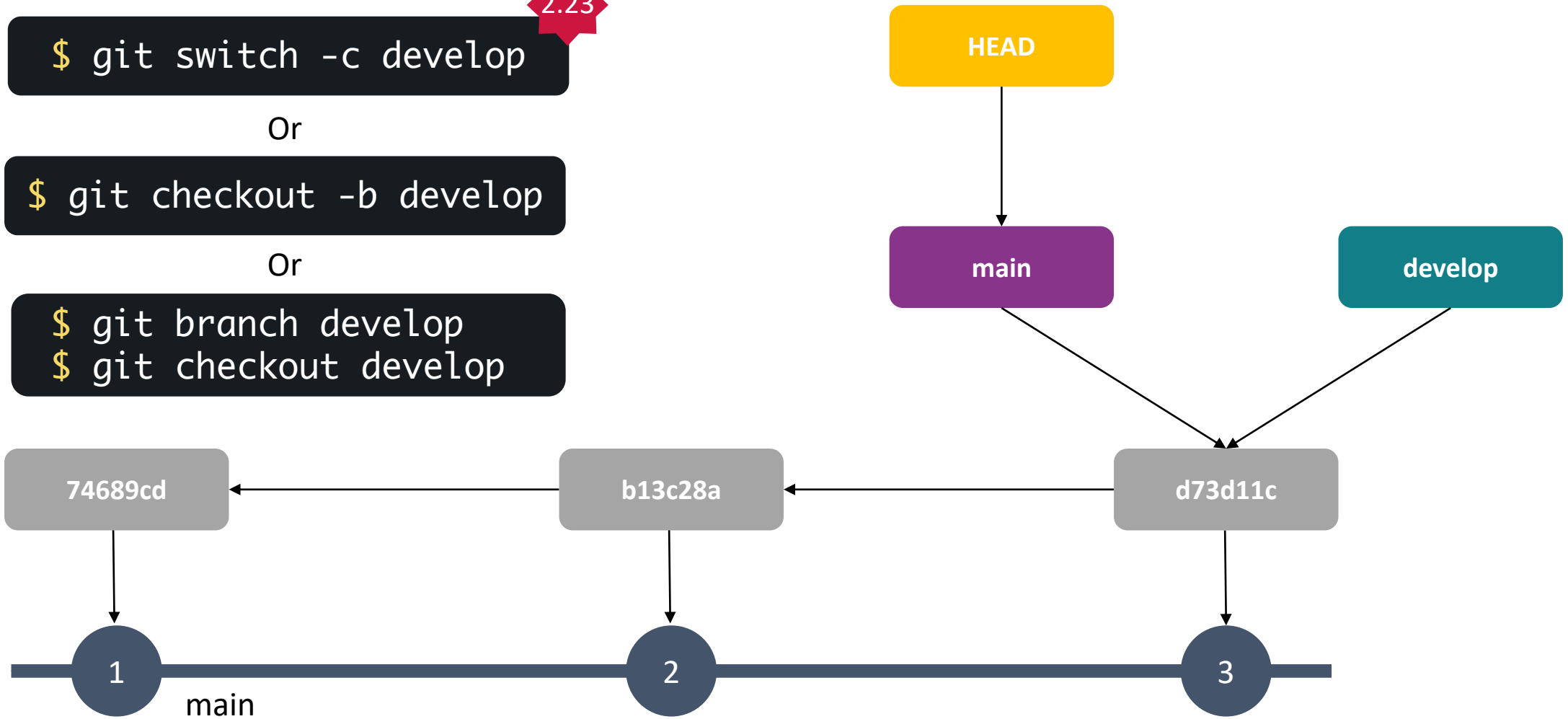
```
$ git switch -c develop
```

Or

```
$ git checkout -b develop
```

Or

```
$ git branch develop  
$ git checkout develop
```



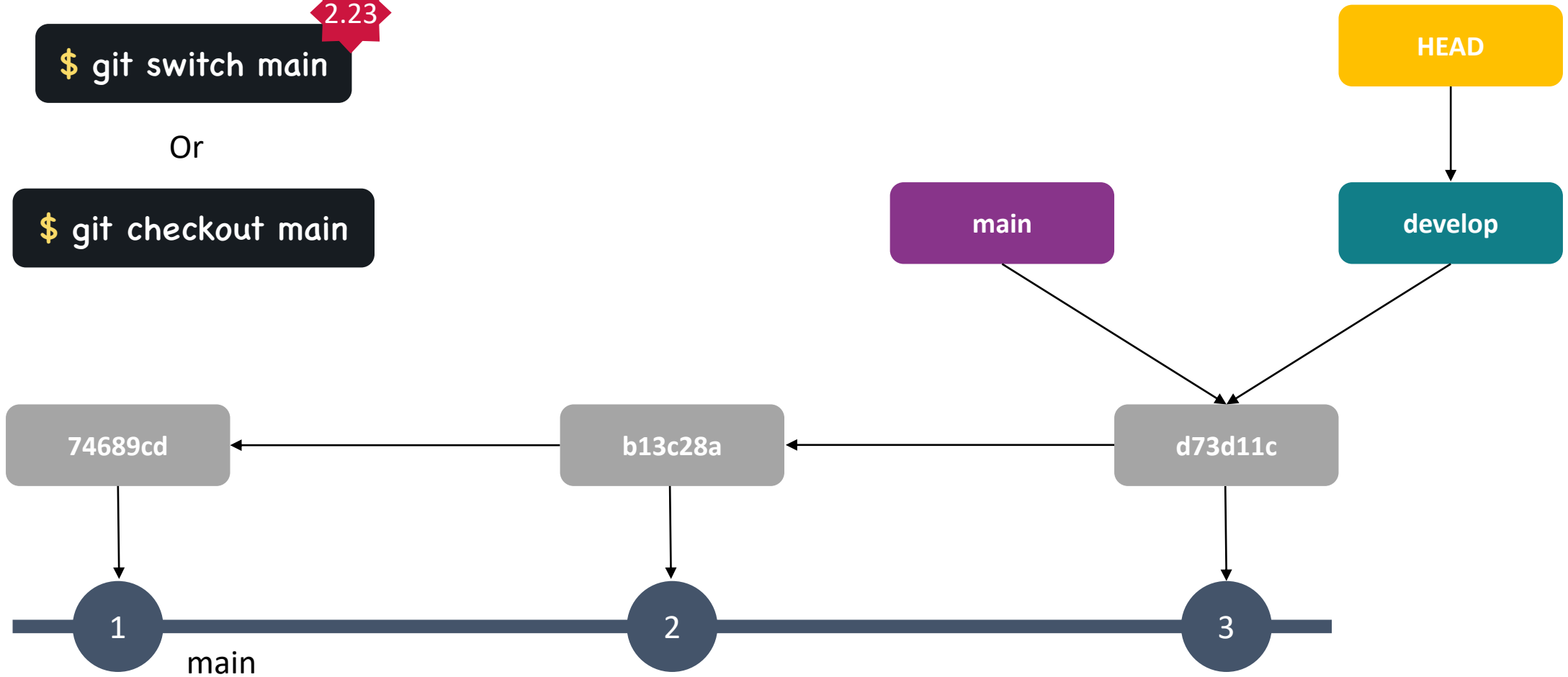
Switch Branch

`$ git switch main`

2.23

Or

`$ git checkout main`

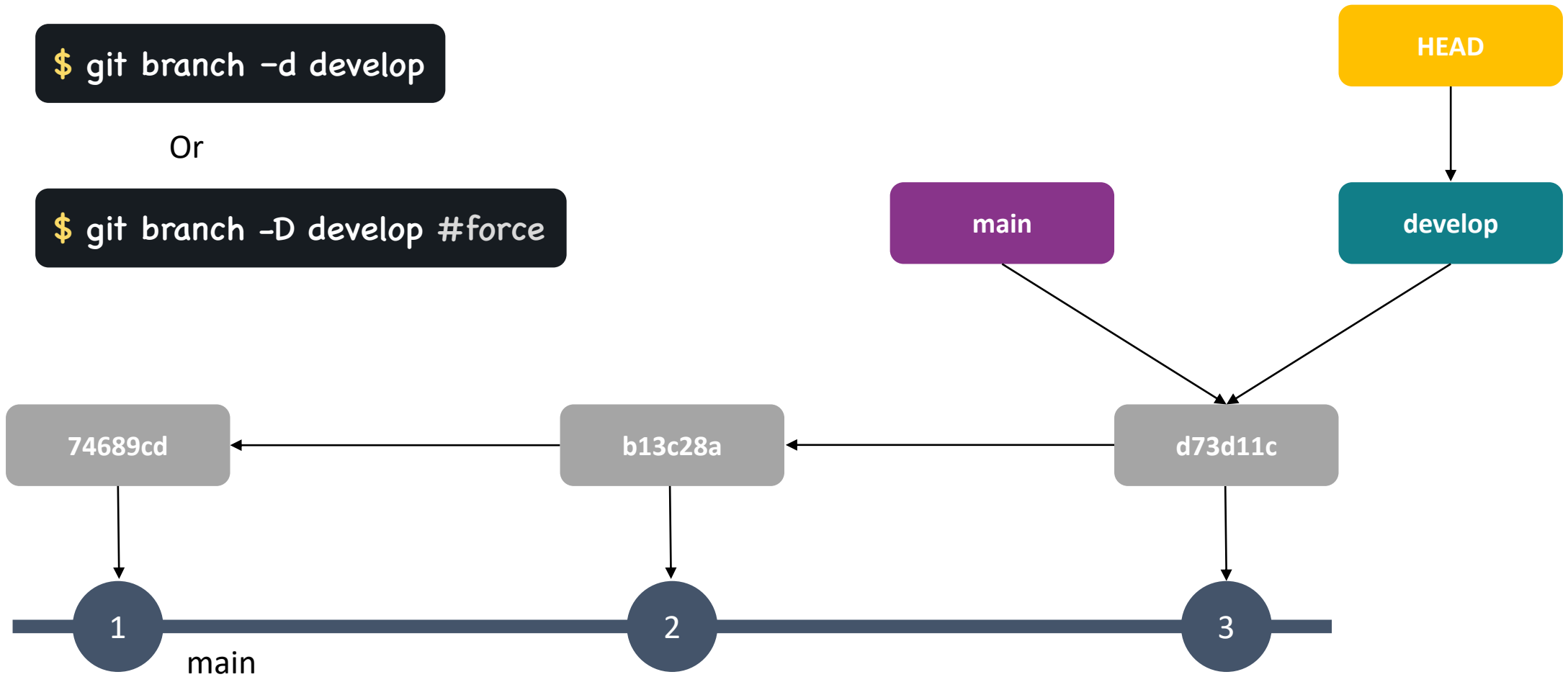


Delete A Branch

```
$ git branch -d develop
```

Or

```
$ git branch -D develop #force
```



Let us practice

HEAD



```
→ project-1$ ls -1 ../.git
```

```
COMMIT_EDITMSG
```

```
HEAD
```

```
config
```

```
description
```

```
hooks
```

```
index
```

```
info
```

```
logs
```

```
objects
```

```
refs
```

```
→ project-1$ ls -1 ../.git/refs/heads
```

```
develop
```

```
main
```

HEAD



```
→ project-1$ git branch  
* develop  
main
```

```
→ project-1$ cat ../.git/HEAD  
ref: refs/heads/develop
```

```
→ project-1$ cat ../.git/refs/heads/develop  
7e051346a7fc41a96d1fcdcb5300da38542b5c9f # commit ID
```