

# Git

## In Practice



# Git

## In Practice



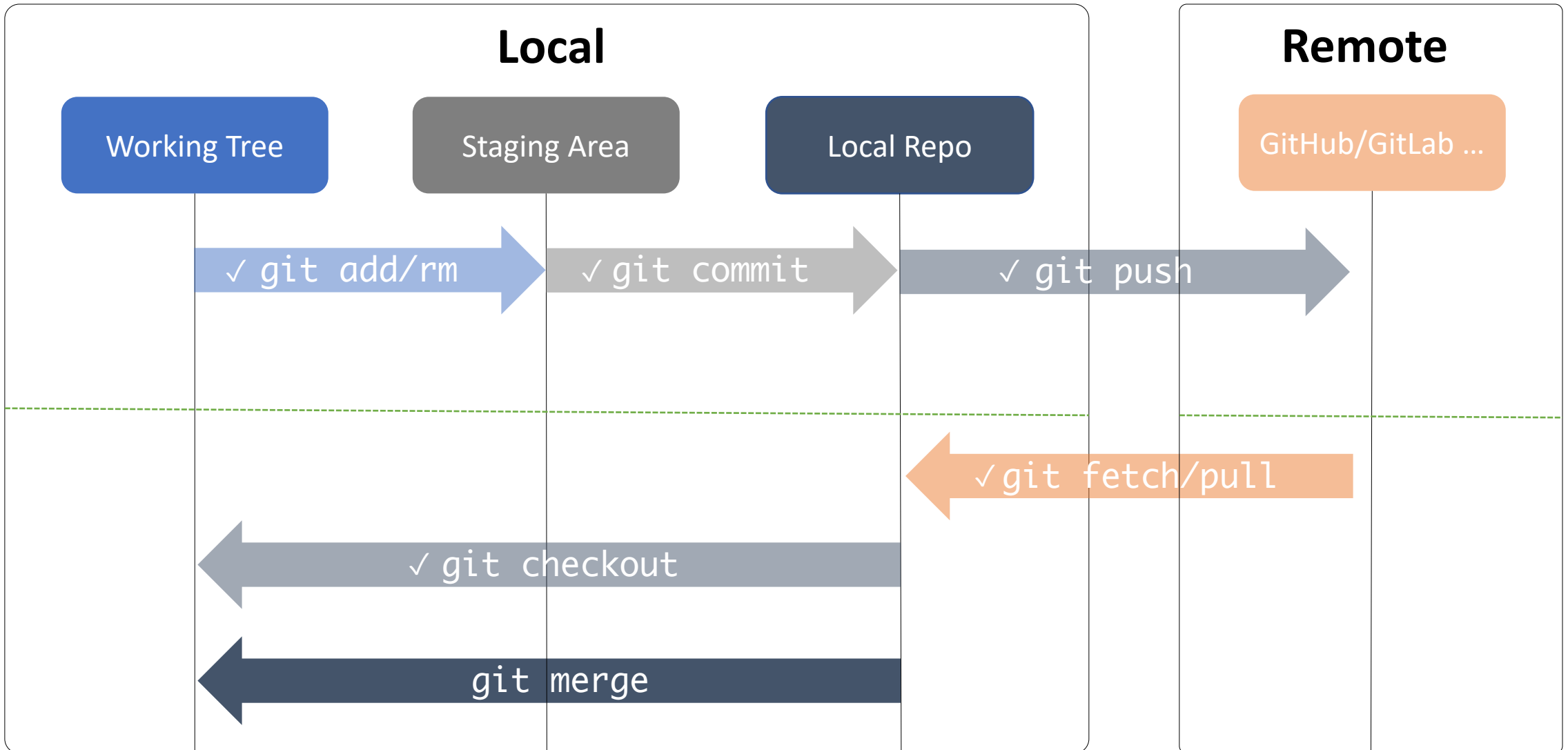
1 Merge and Merge Strategies

2

3

# Basic Merges

# Project Components – Merge



# What is a merge?



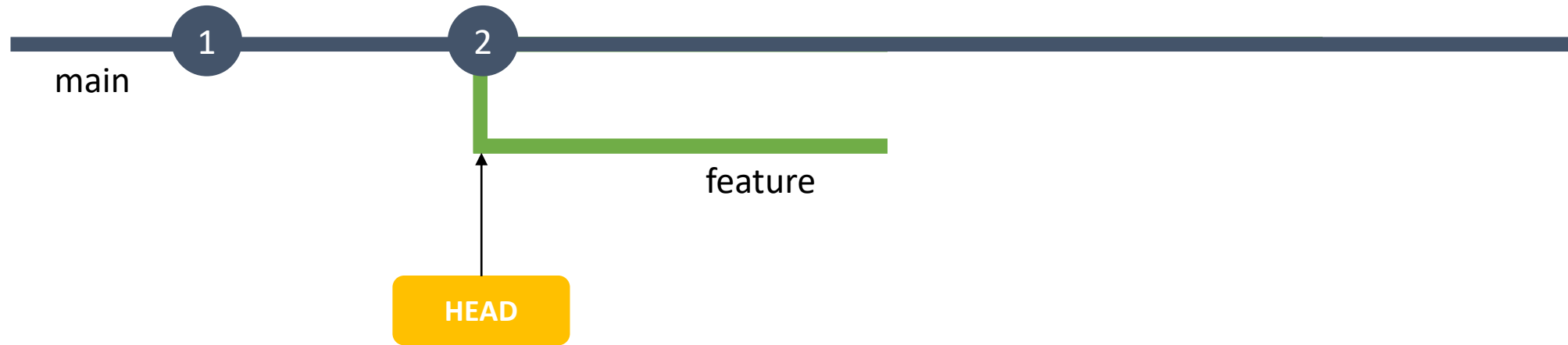
# What is a merge?



```
$ git switch -C feature
```

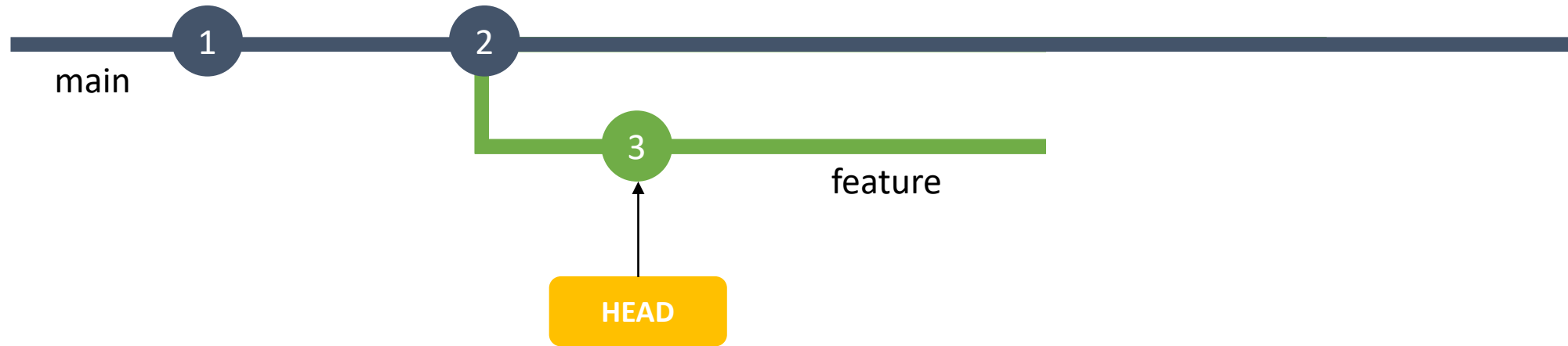
2.23

# What is a merge?



# What is a merge?

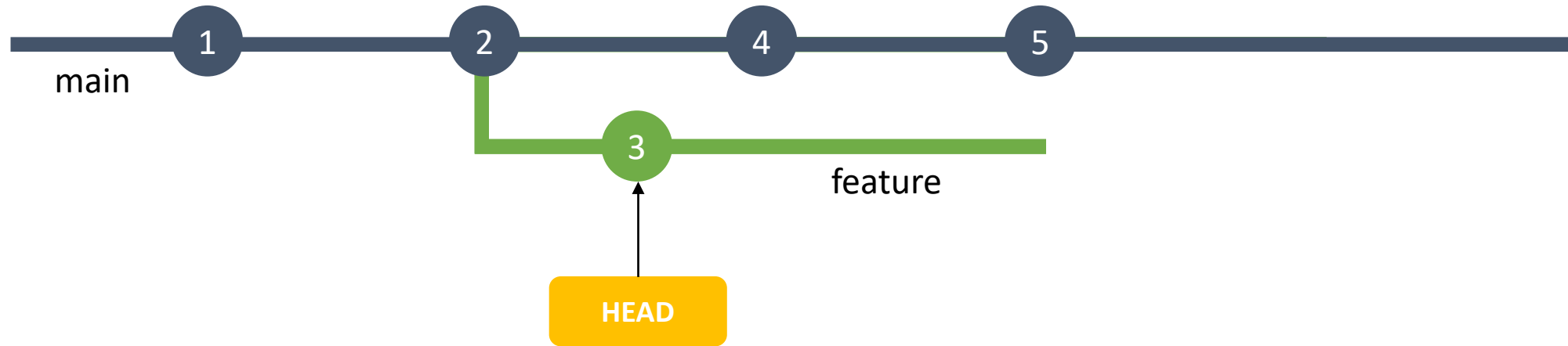
New commit in feature





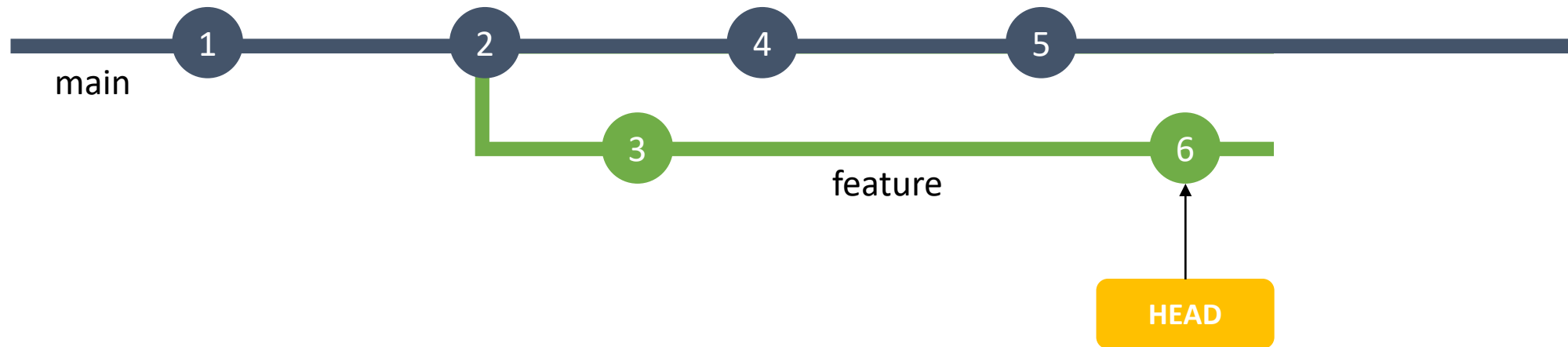
# What is a merge?

More commits in main



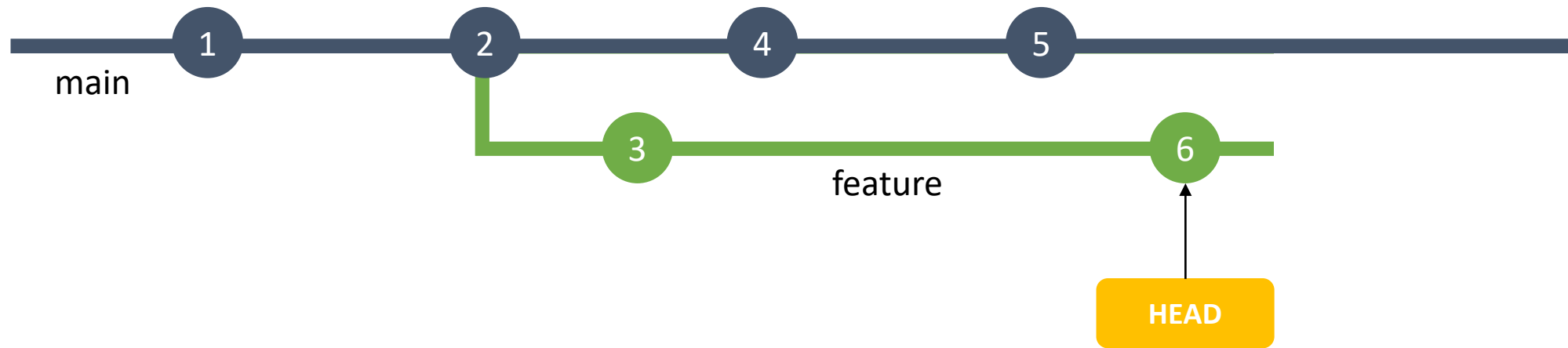
# What is a merge?

One more commit in feature

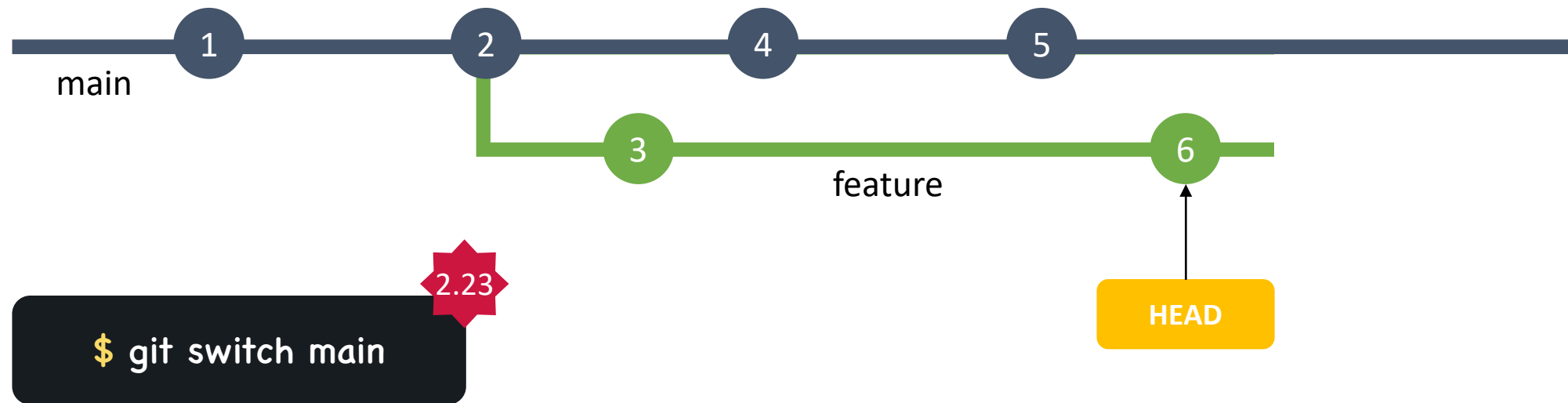


# What is a merge?

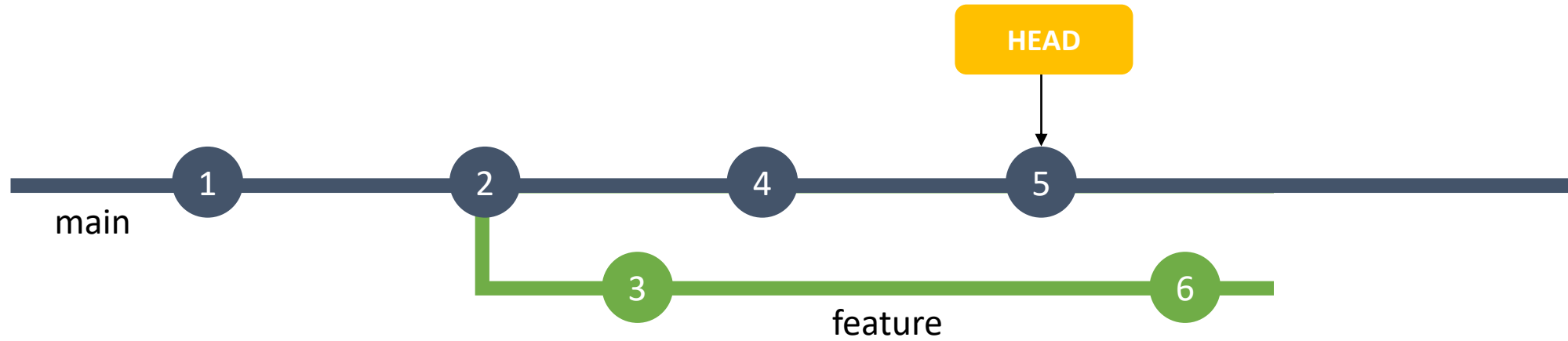
Merge develop to main



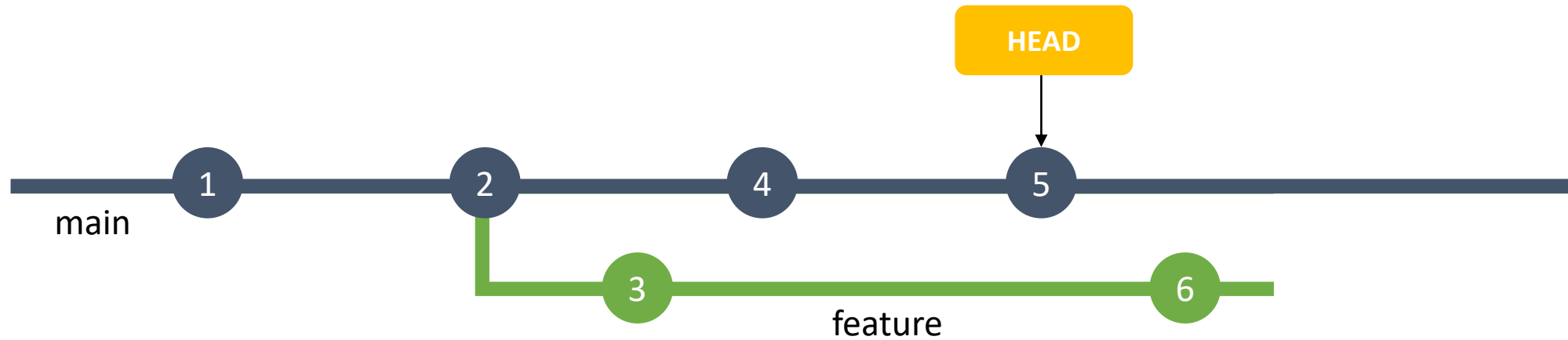
# What is a merge?



# What is a merge?

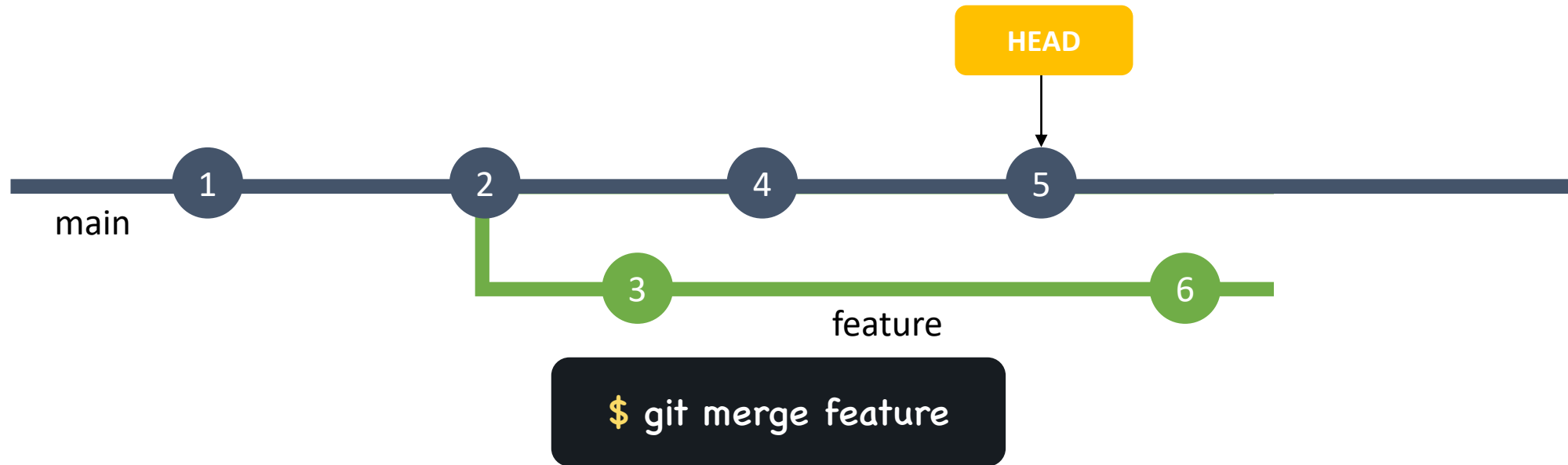


# What is a merge?

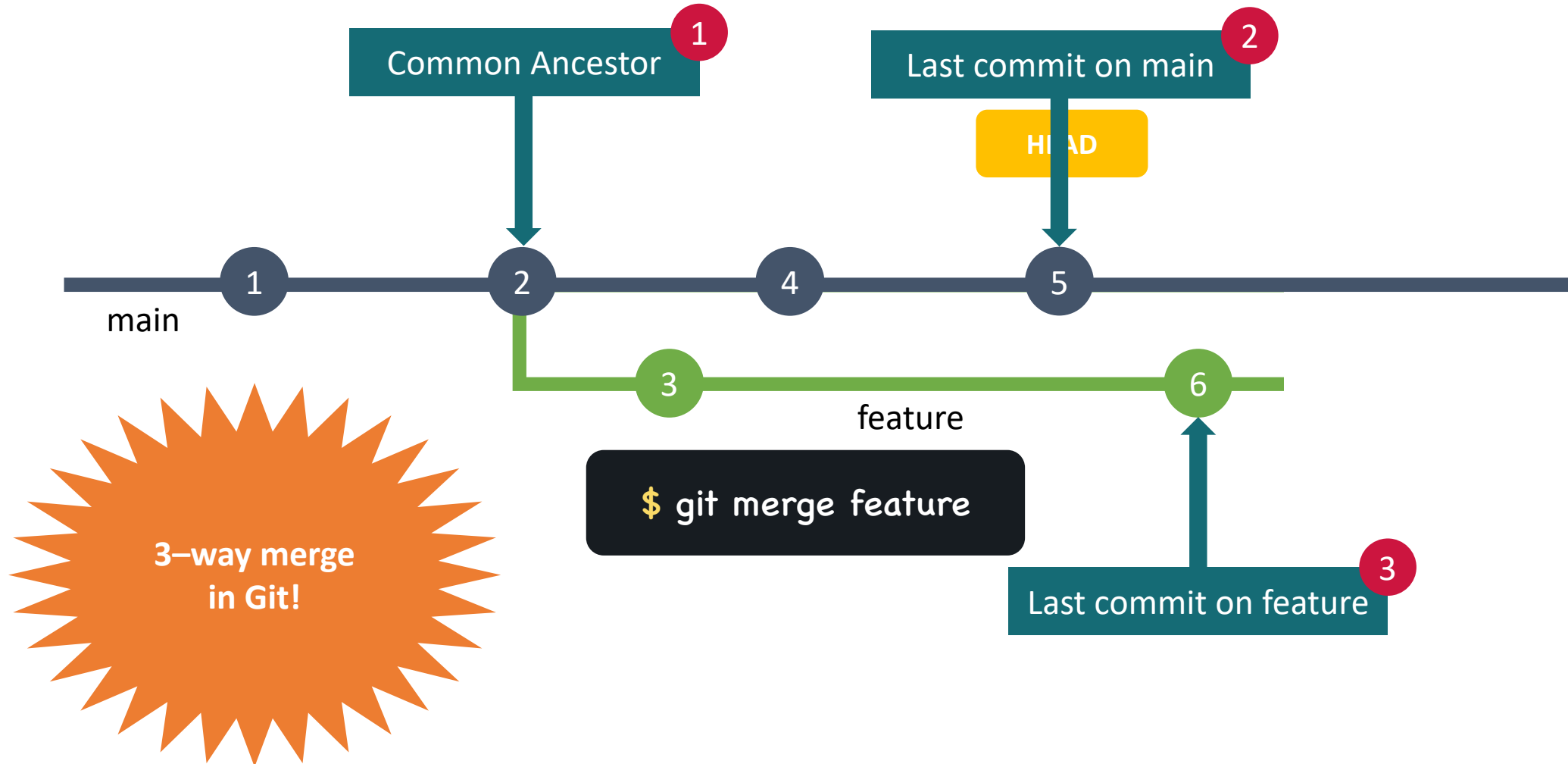


`$ git fetch # explained in later videos`

# What is a merge?

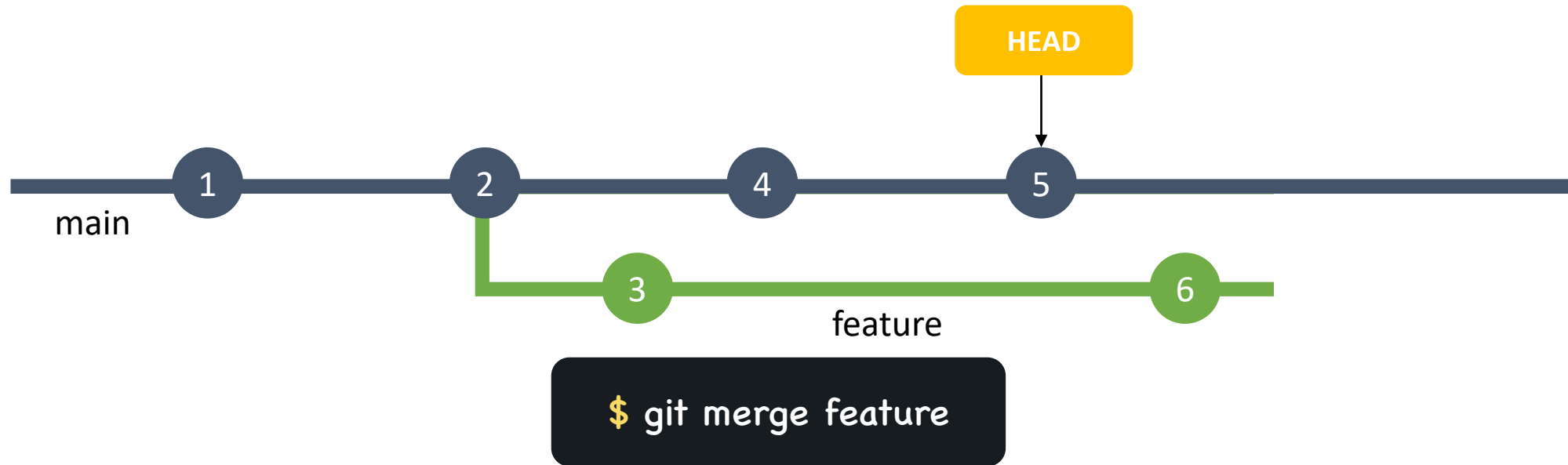


# What is a merge?





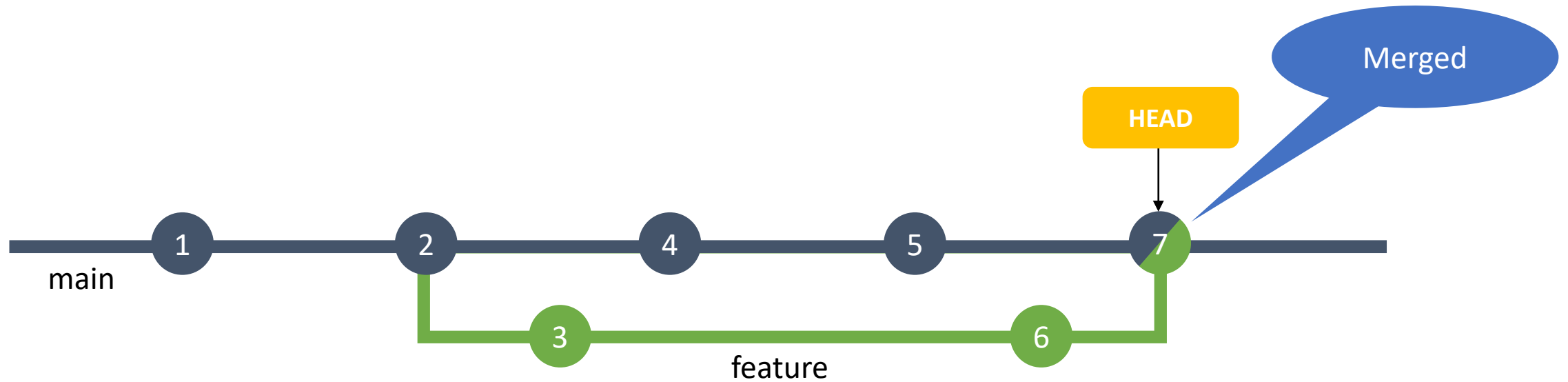
# What is a merge?



# What is a merge?



# What is a merge?



```
$ git branch -d feature
```

# What is a merge?



# What is a merge?

## *Takeaway*

*Git supports merge into feature!*

Always switch to the branch you want to merge into first, before merging

# Types Of Merge Strategies

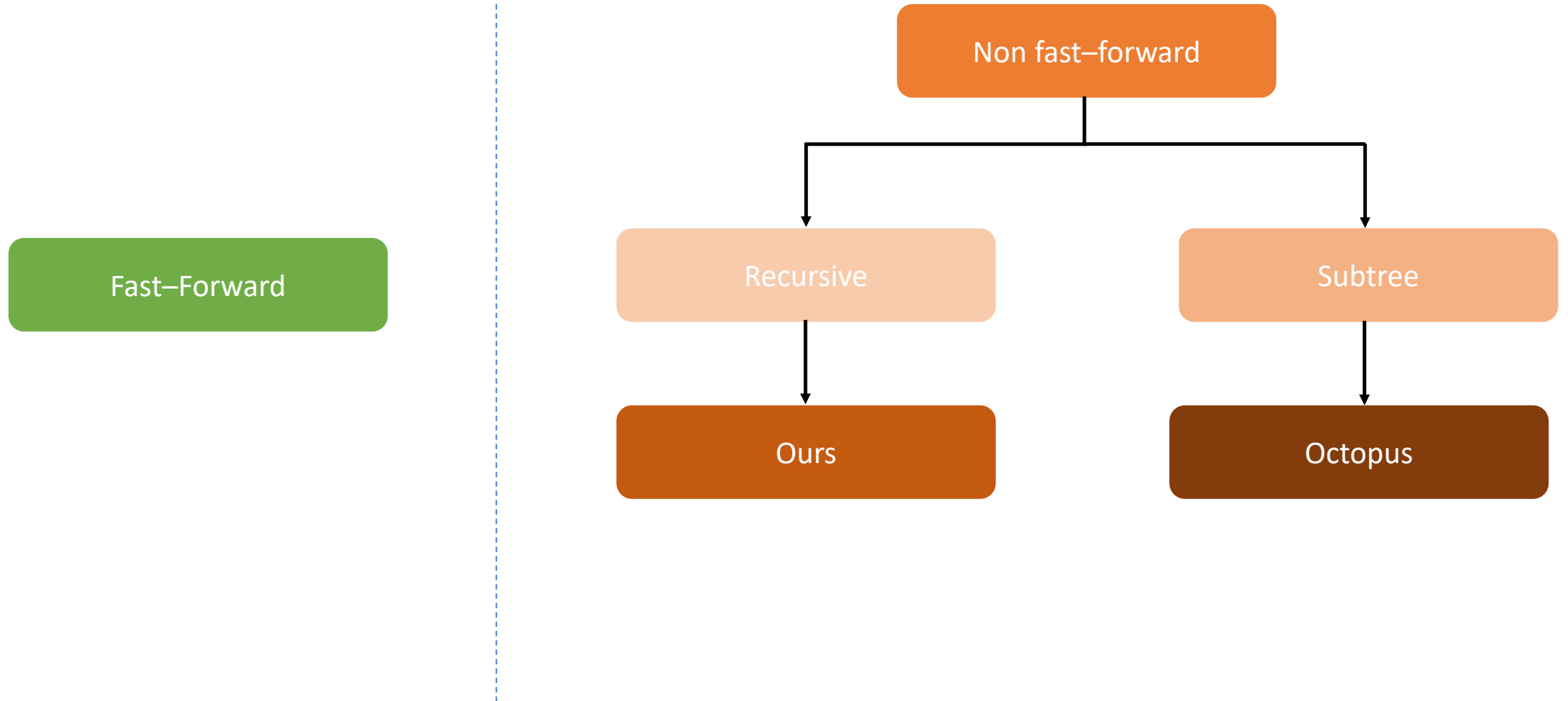


Fast-Forward

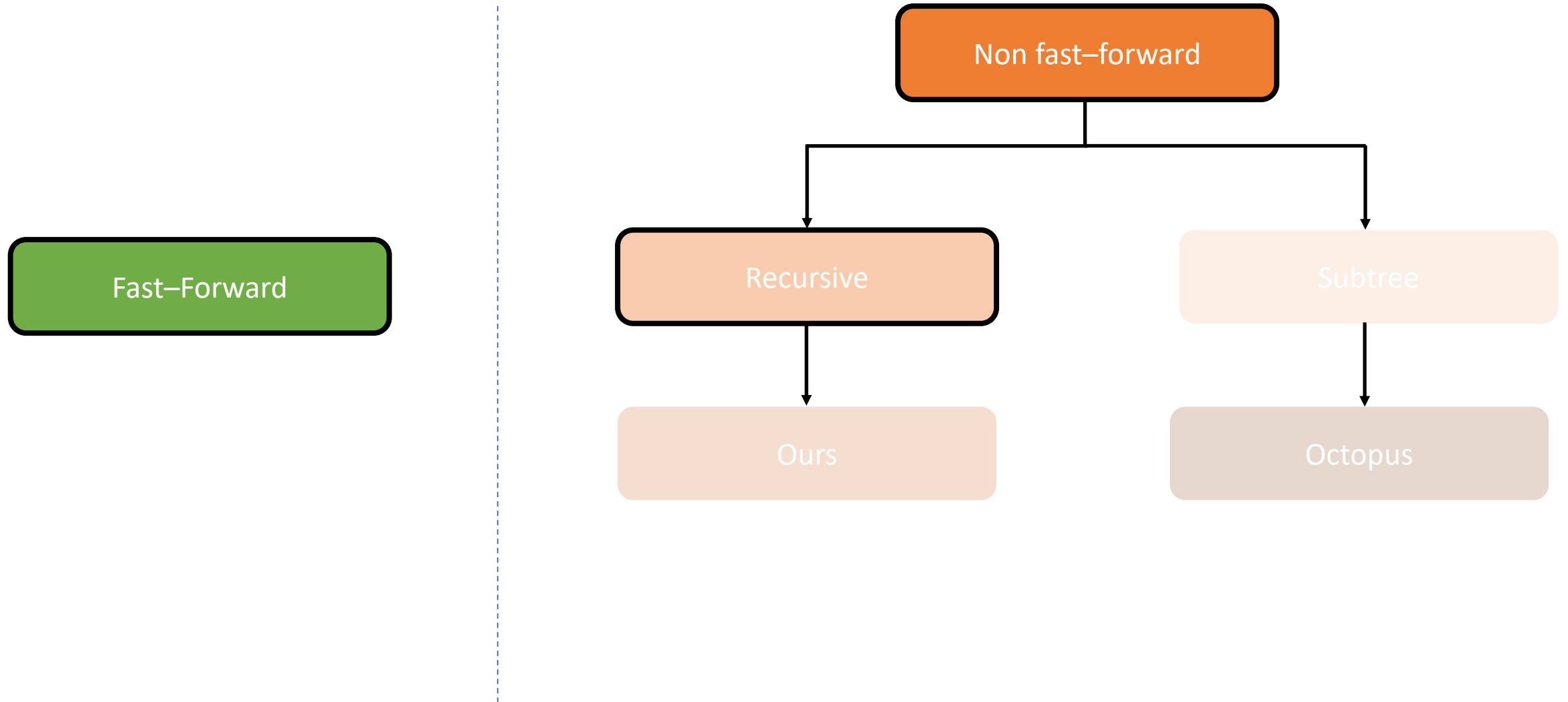
The diagram consists of a vertical dashed blue line that divides the space into two columns. To the left of this line is a green rounded rectangle containing the text 'Fast-Forward'. To the right of the line is an orange rounded rectangle containing the text 'Non fast-forward'.

Non fast-forward

# Types Of Merge Strategies



# The Two Common Merge Strategies





# The Two Common Merge Strategies

1. Fast Forward Merges – Merges that don't create a new snapshot
2. Merge commits – Merges that create a new snapshot
  - Merge without conflict
  - Merge with conflict

# Takeaways

1. Git supports **merge into** feature and not merge from
2. Switch to the branch you want to merge into first, before merging
3. Commands
  - a. `git switch <branchname>` – to switch between branches
  - b. `git merge <branchname>`