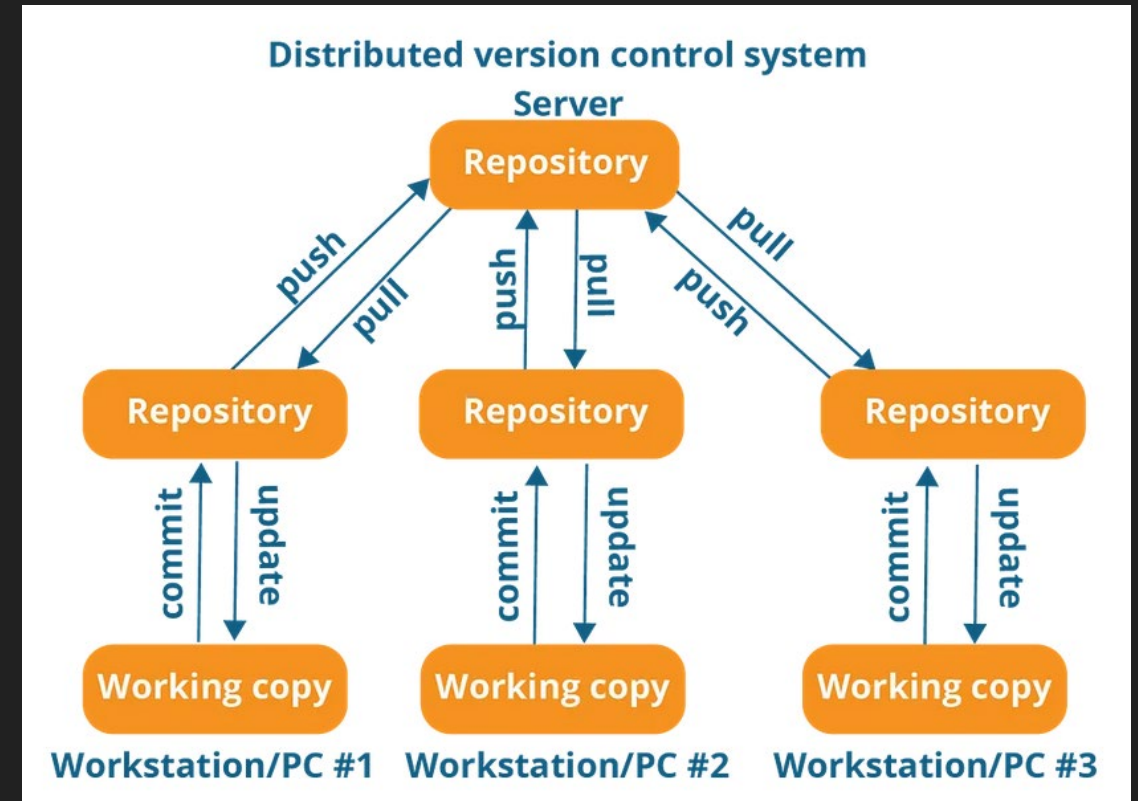


GIT



What is Git?

- distributed version control system (DVCS)
- Git was developed by Linus Torvalds in 2005
- Version control
- Collaboration
- Backup
- Branching and merging
- Industry standard



Git workflow

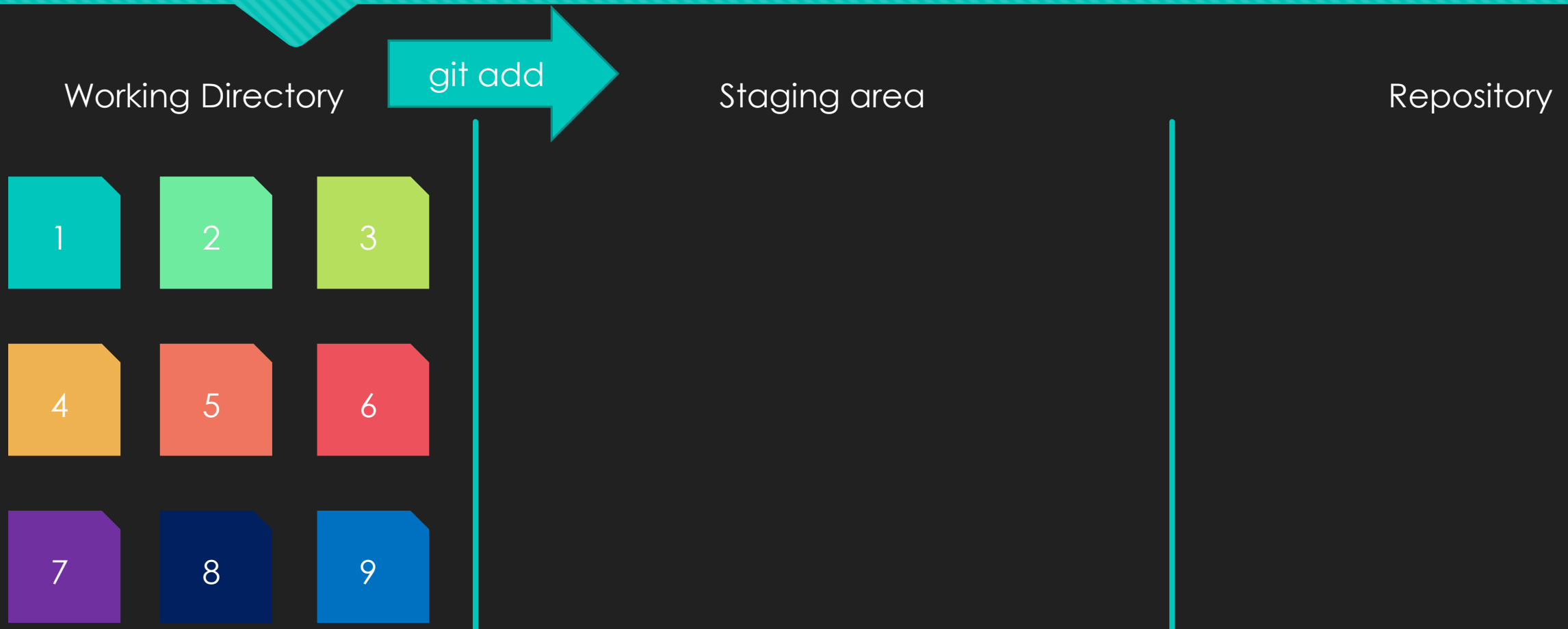
Working Directory



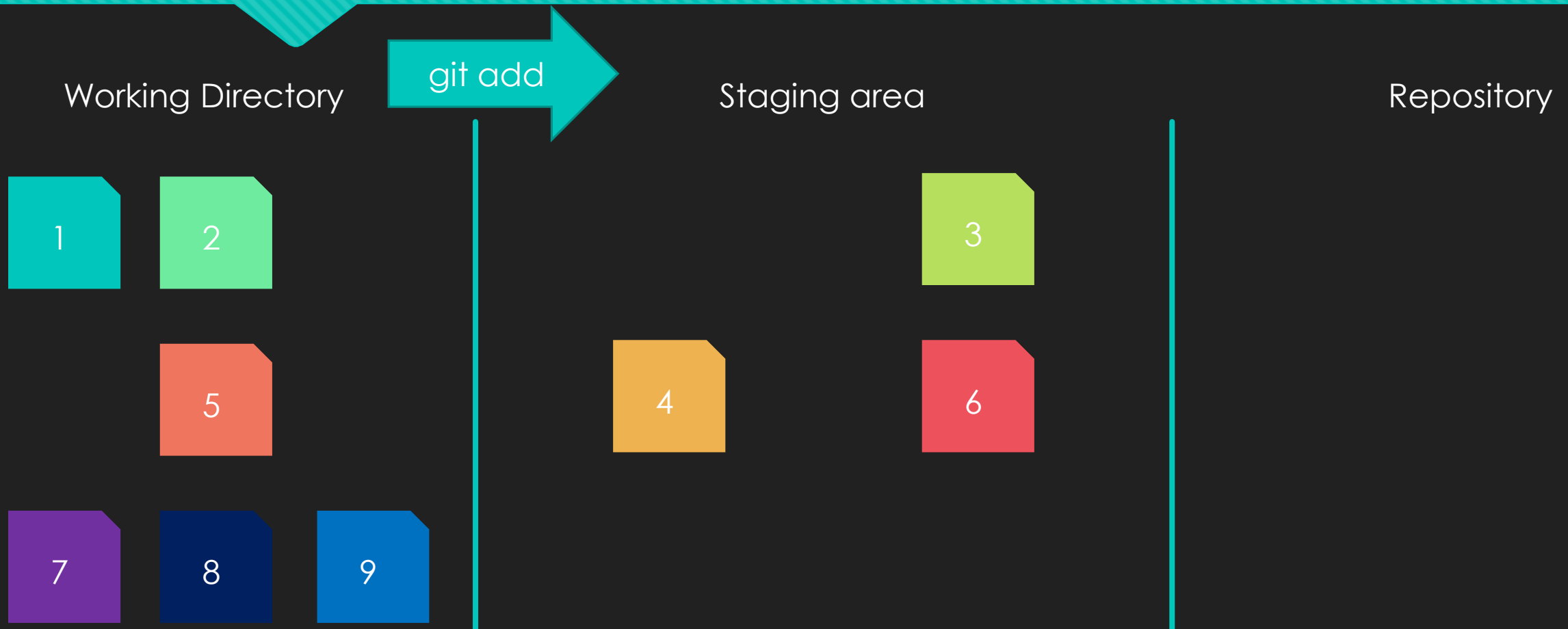
Staging area

Repository

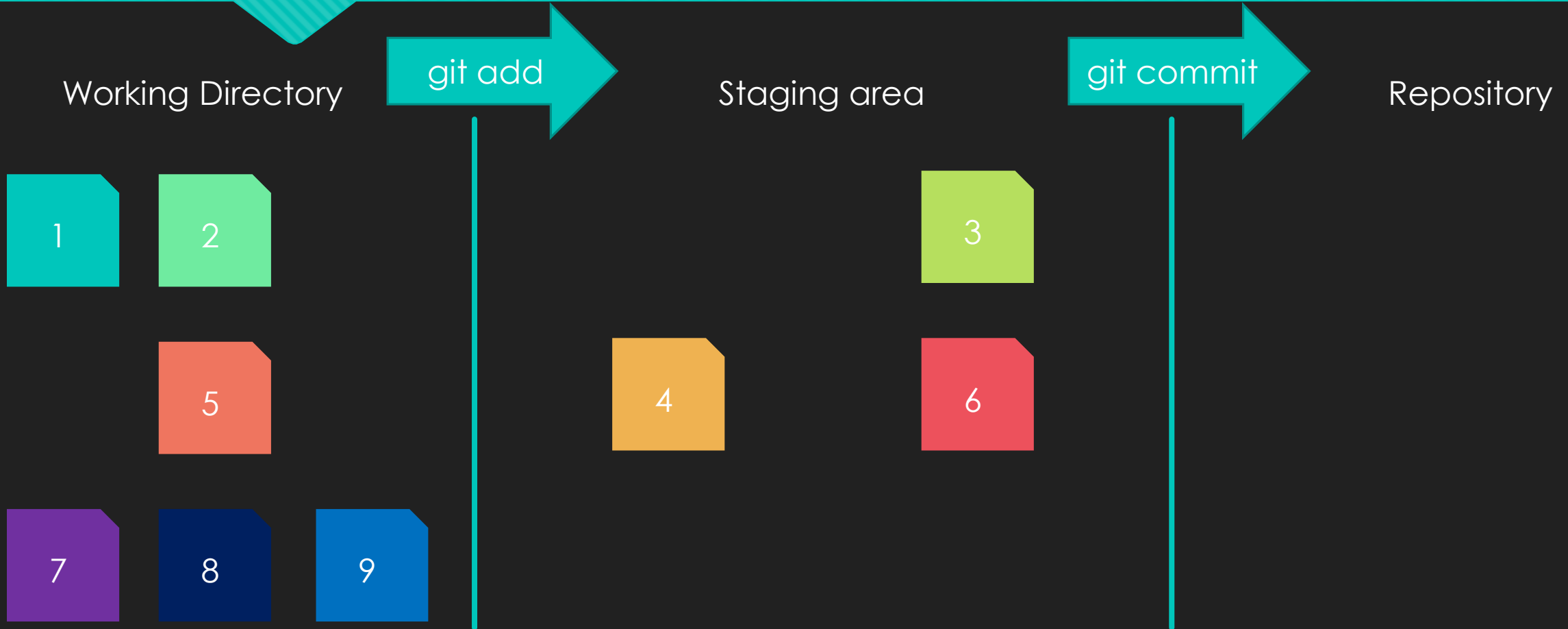
Git workflow



Git workflow



Git workflow



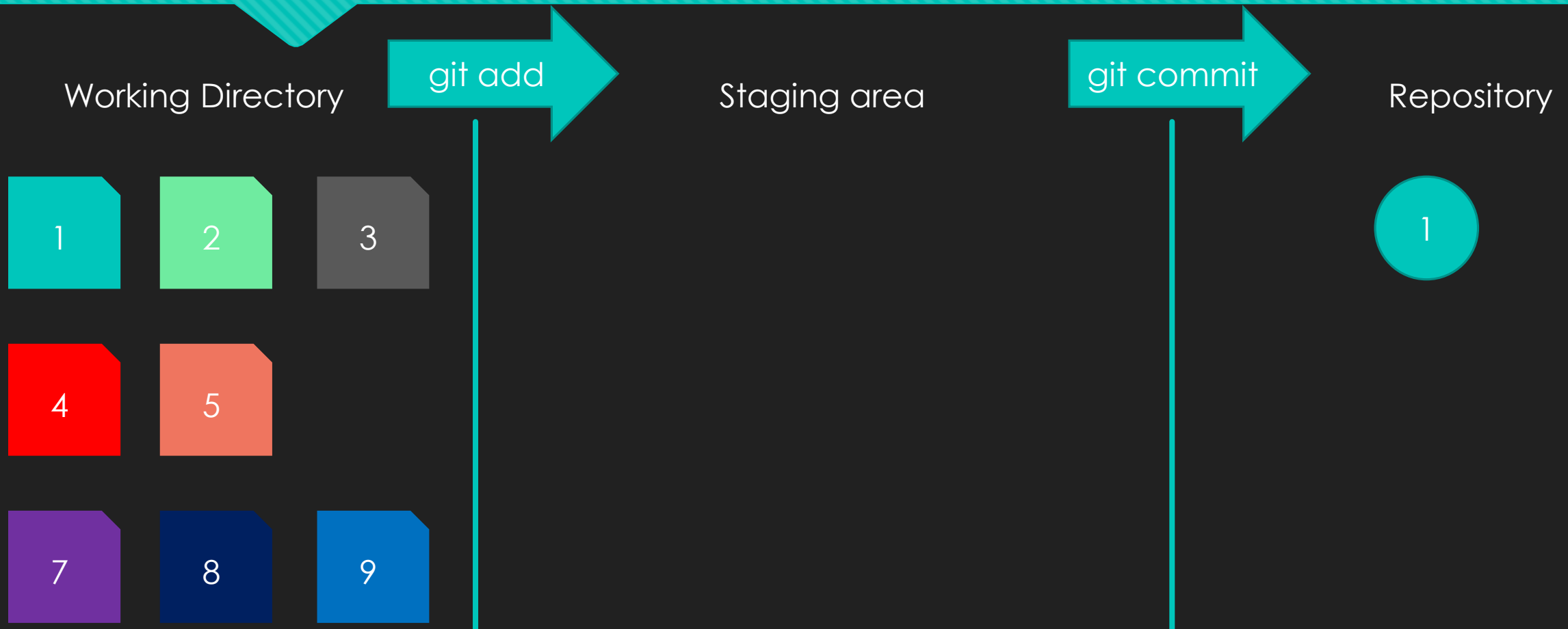
Git workflow



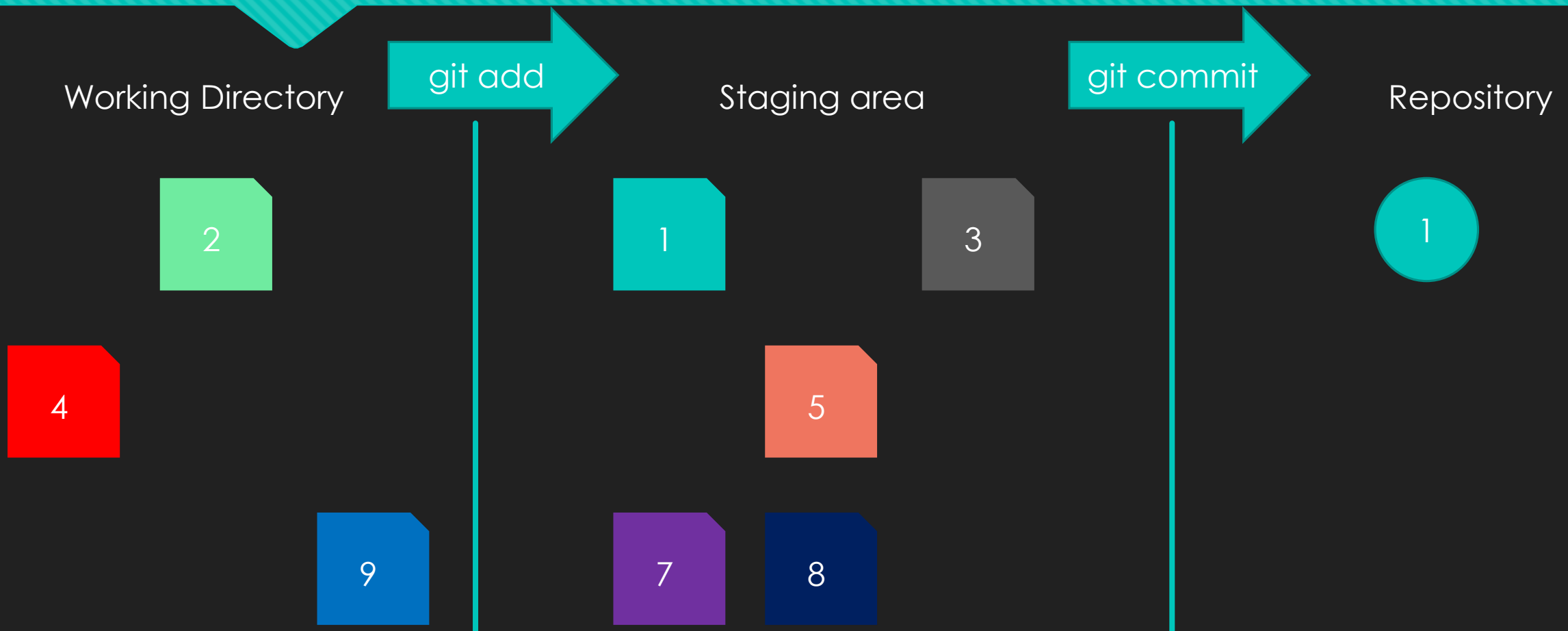
Git workflow



Git workflow



Git workflow



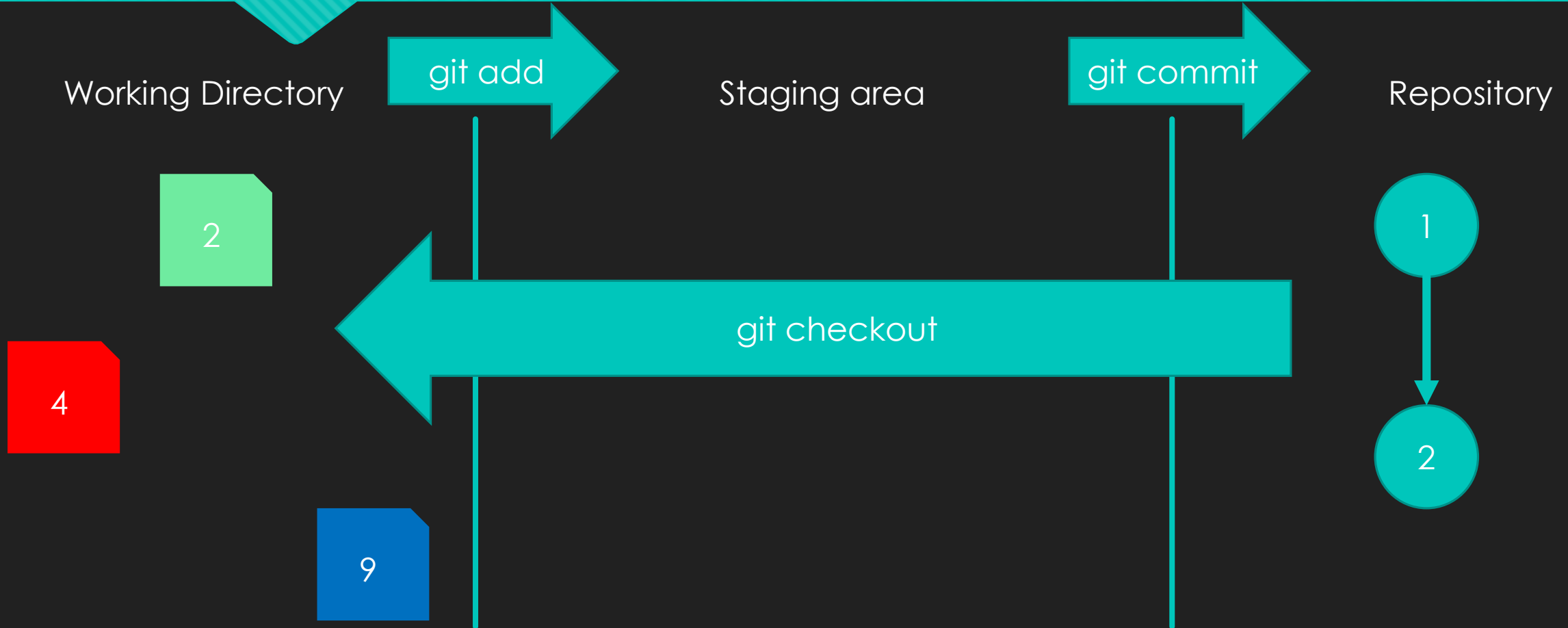
Git workflow



Git workflow



Git workflow



Git workflow



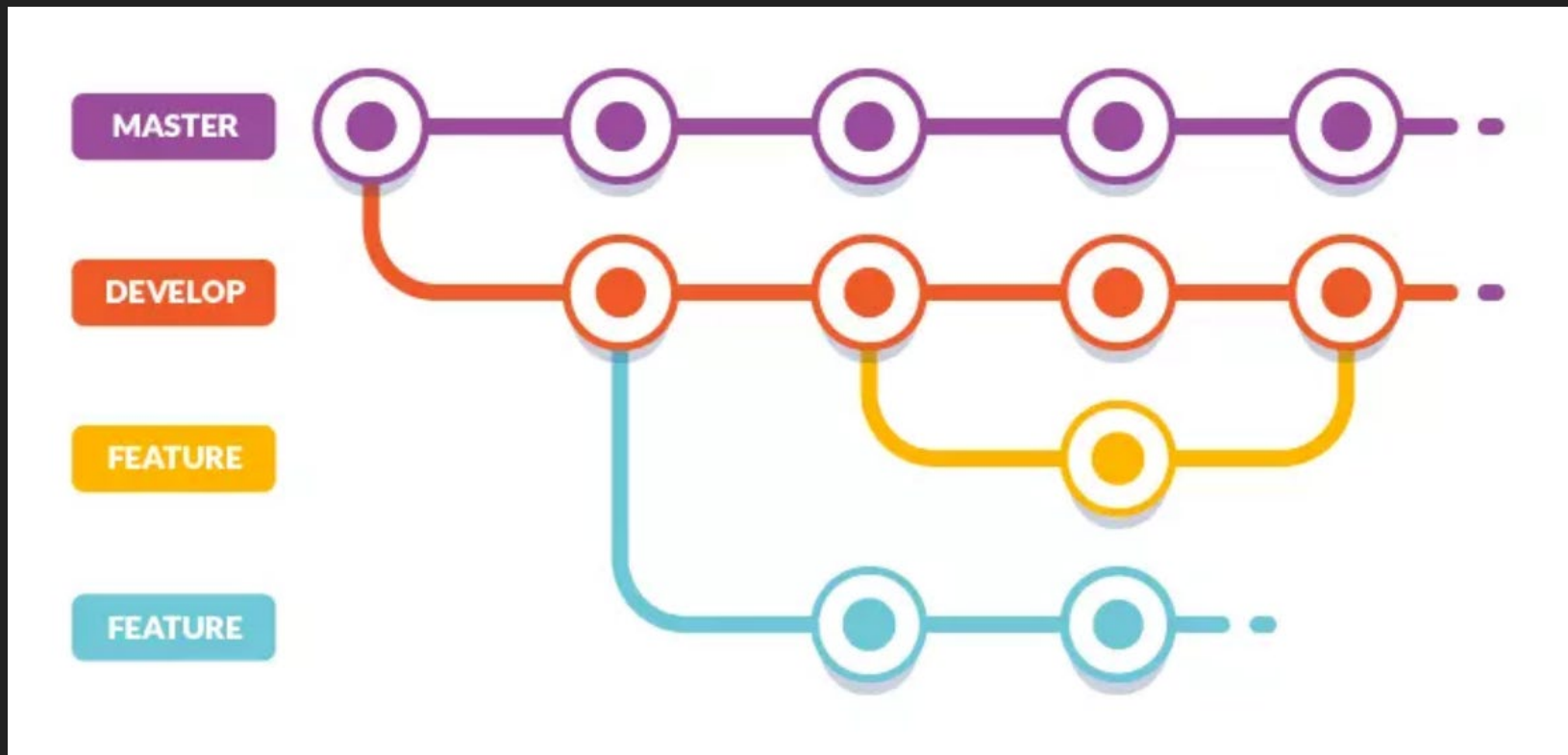
Git workflow



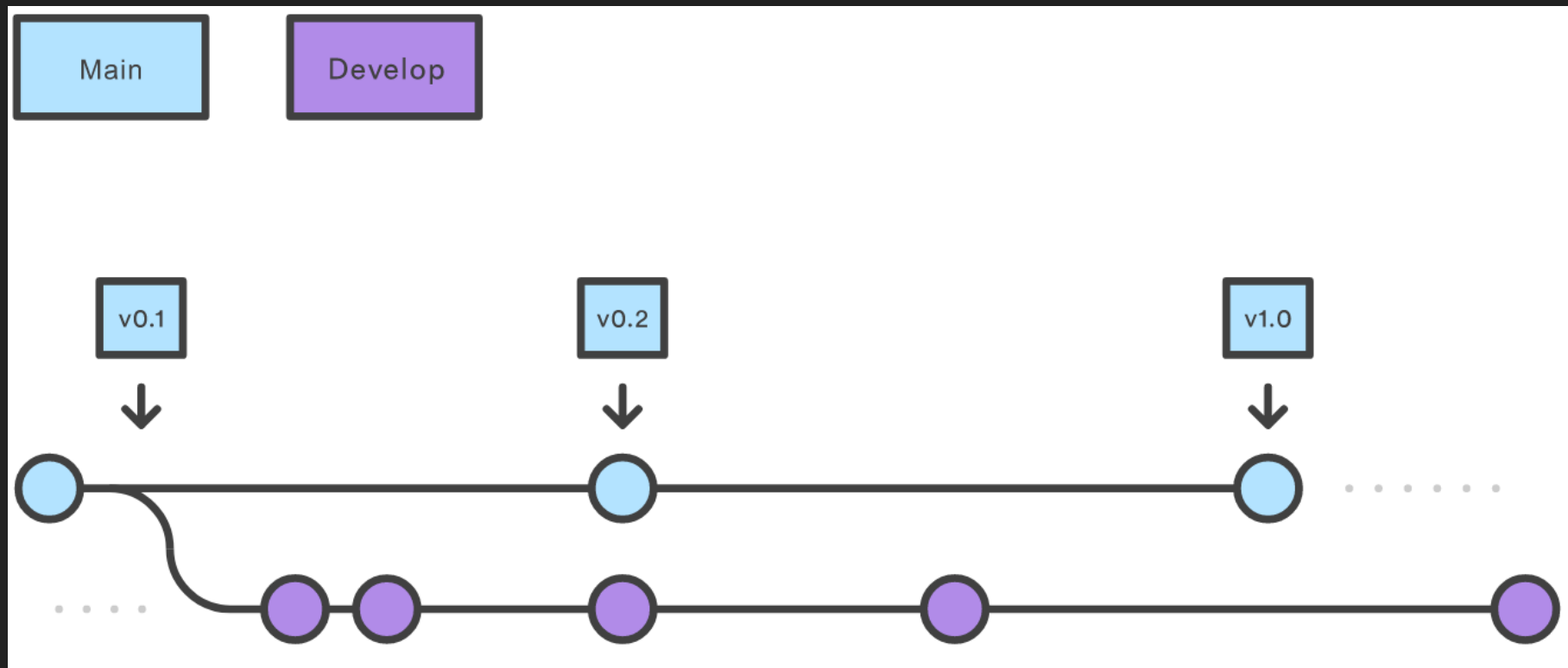
Git



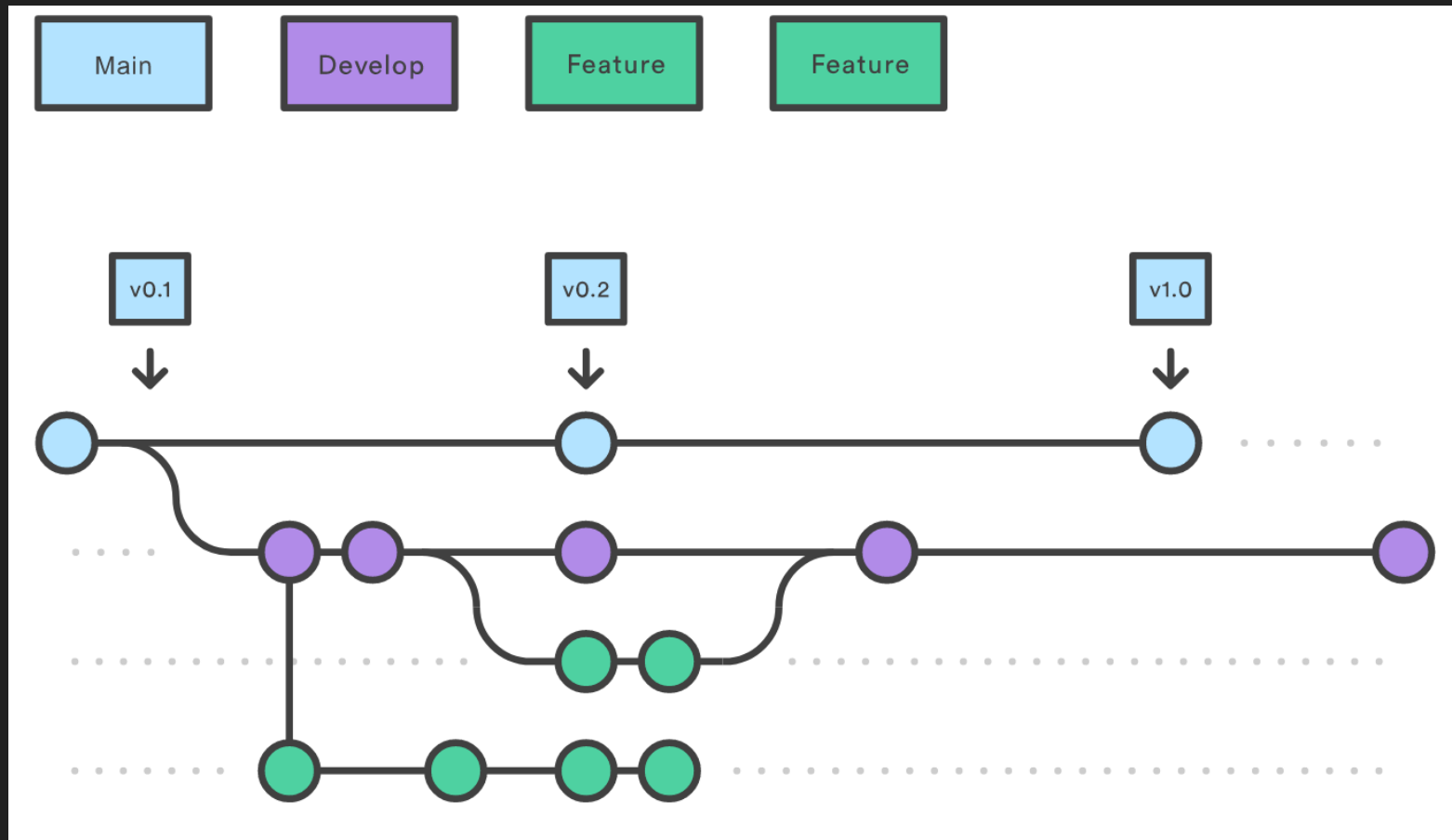
Git



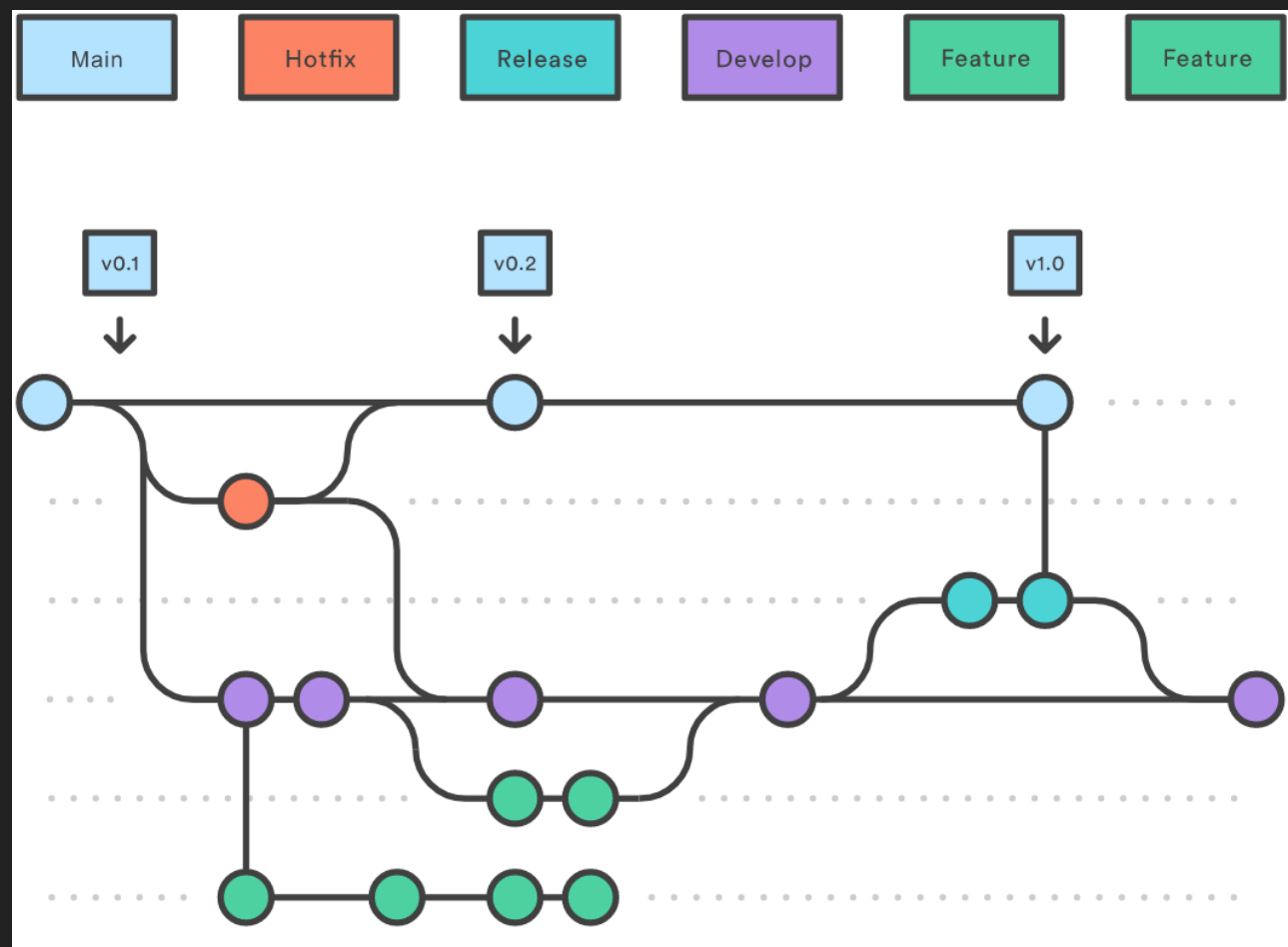
Example in action



Example in action



Example in action



Getting Started

- Install git
- Set name and email

```
git config --global user.name "your name here"
```

```
git config --global user.email "your email here"
```

Git new repository

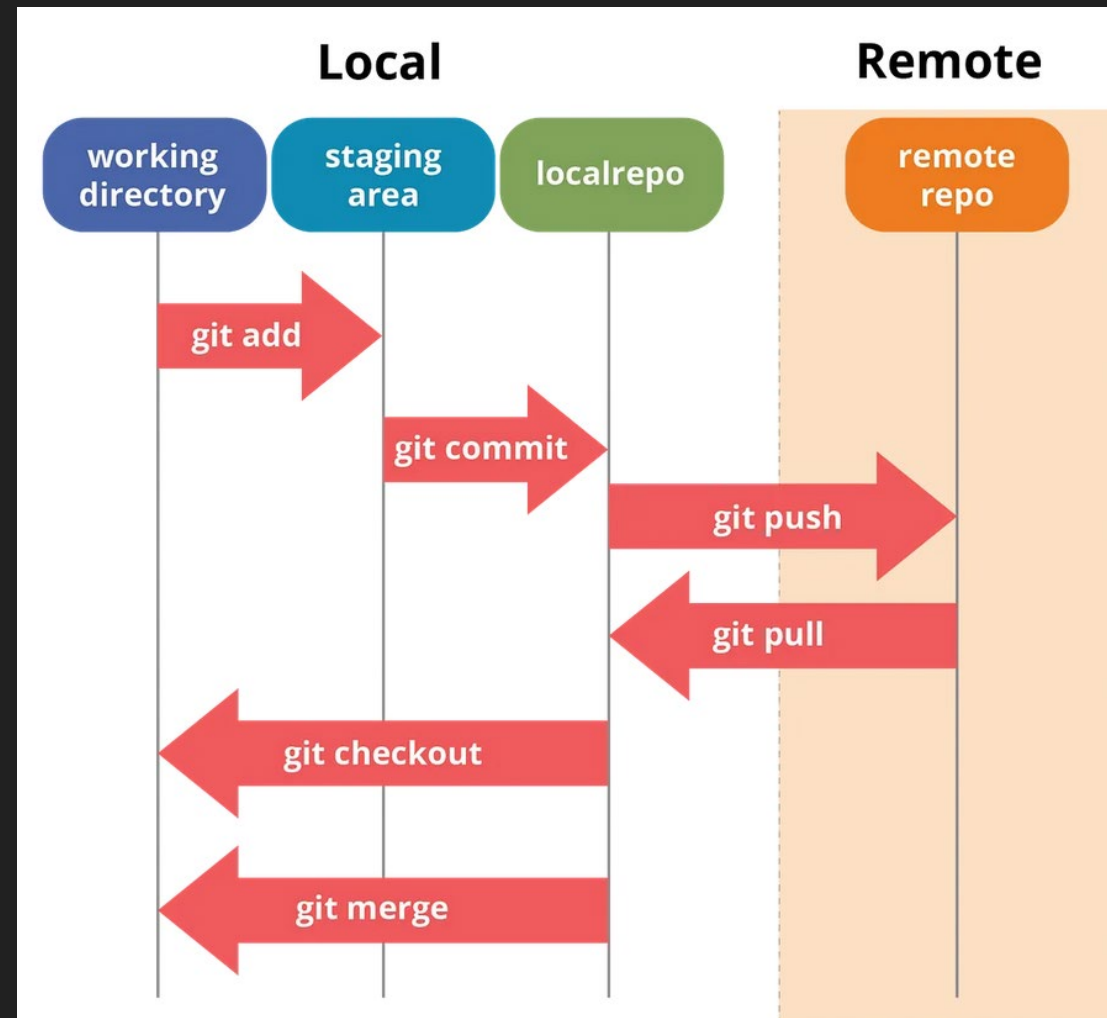
- Start a new repository in your local folder

```
git init
```

download a repository from github / gitlab into your computer

```
git clone <repo>
```

Git workflow

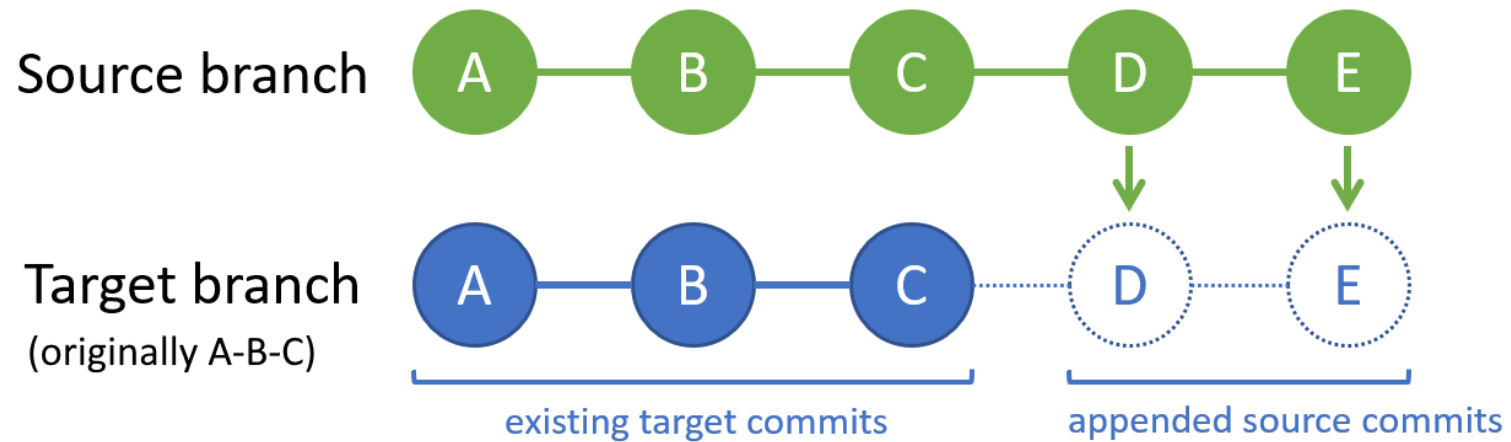


Git basic operations

- initialize
- add
- commit
- pull
- push
- branching
- merging
- rebasing

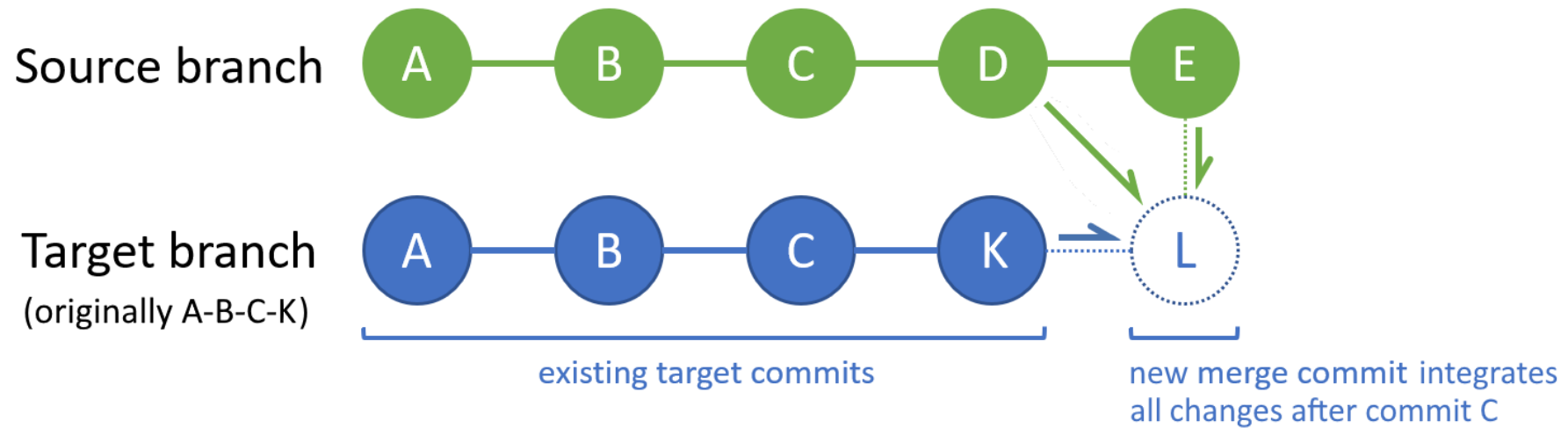
git merge

Merge (fast-forward)



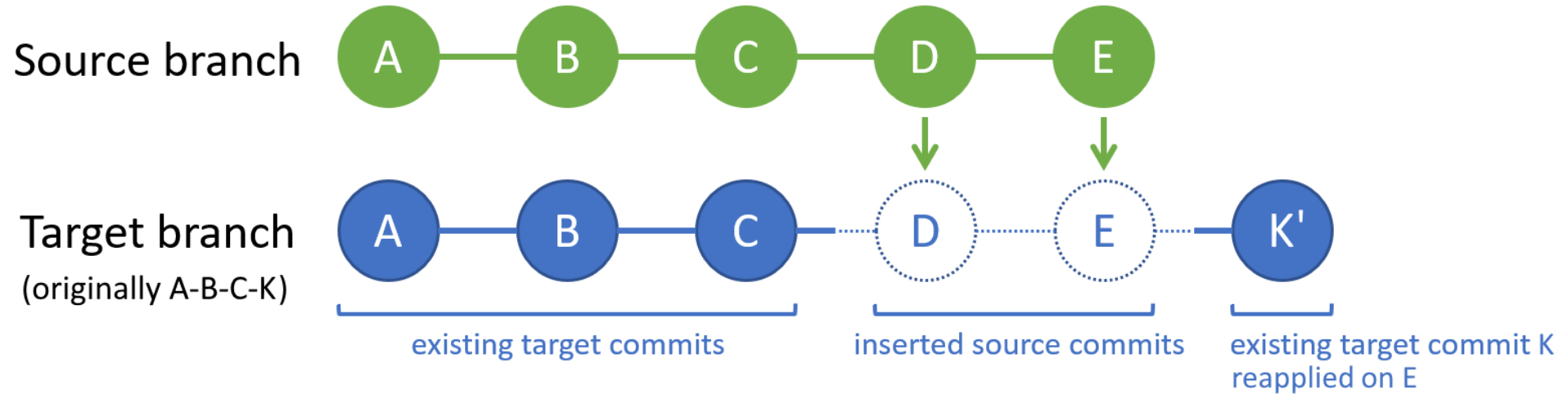
git merge

Merge (no-fast-forward)

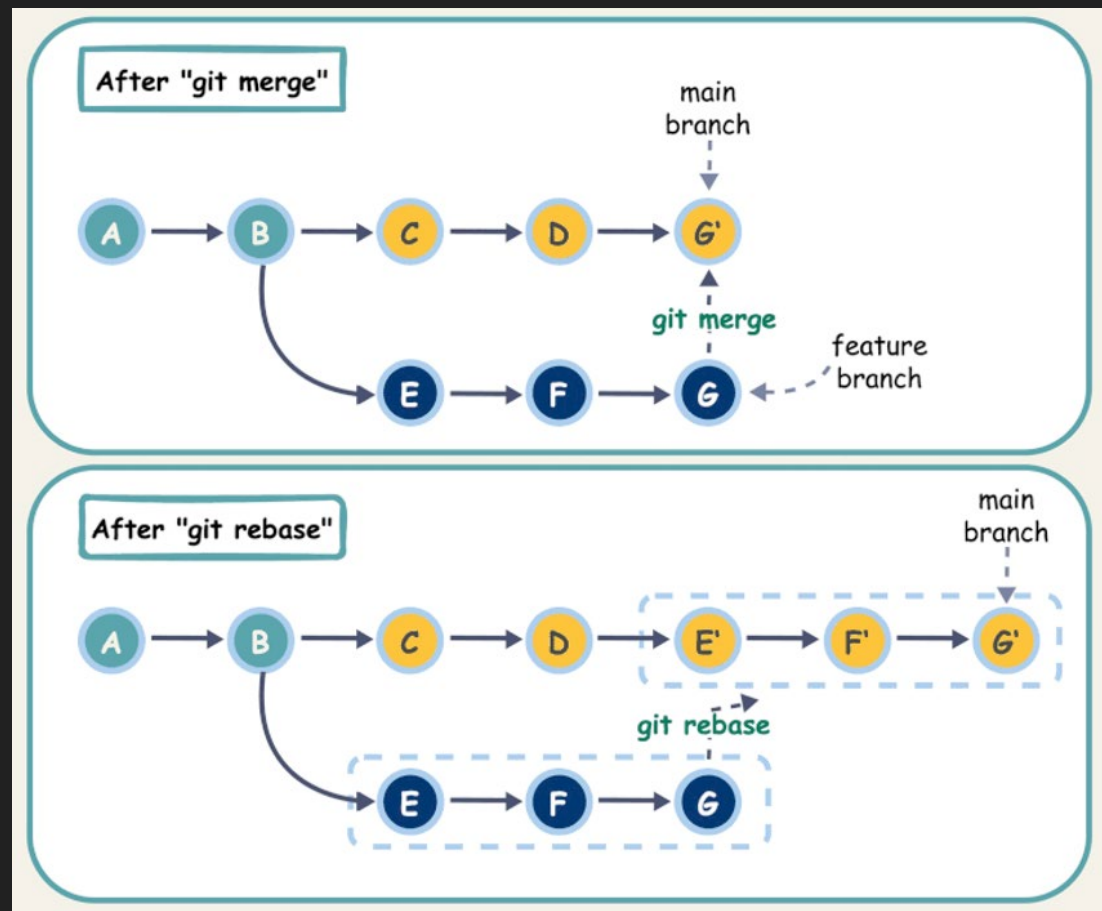


git merge

Rebase



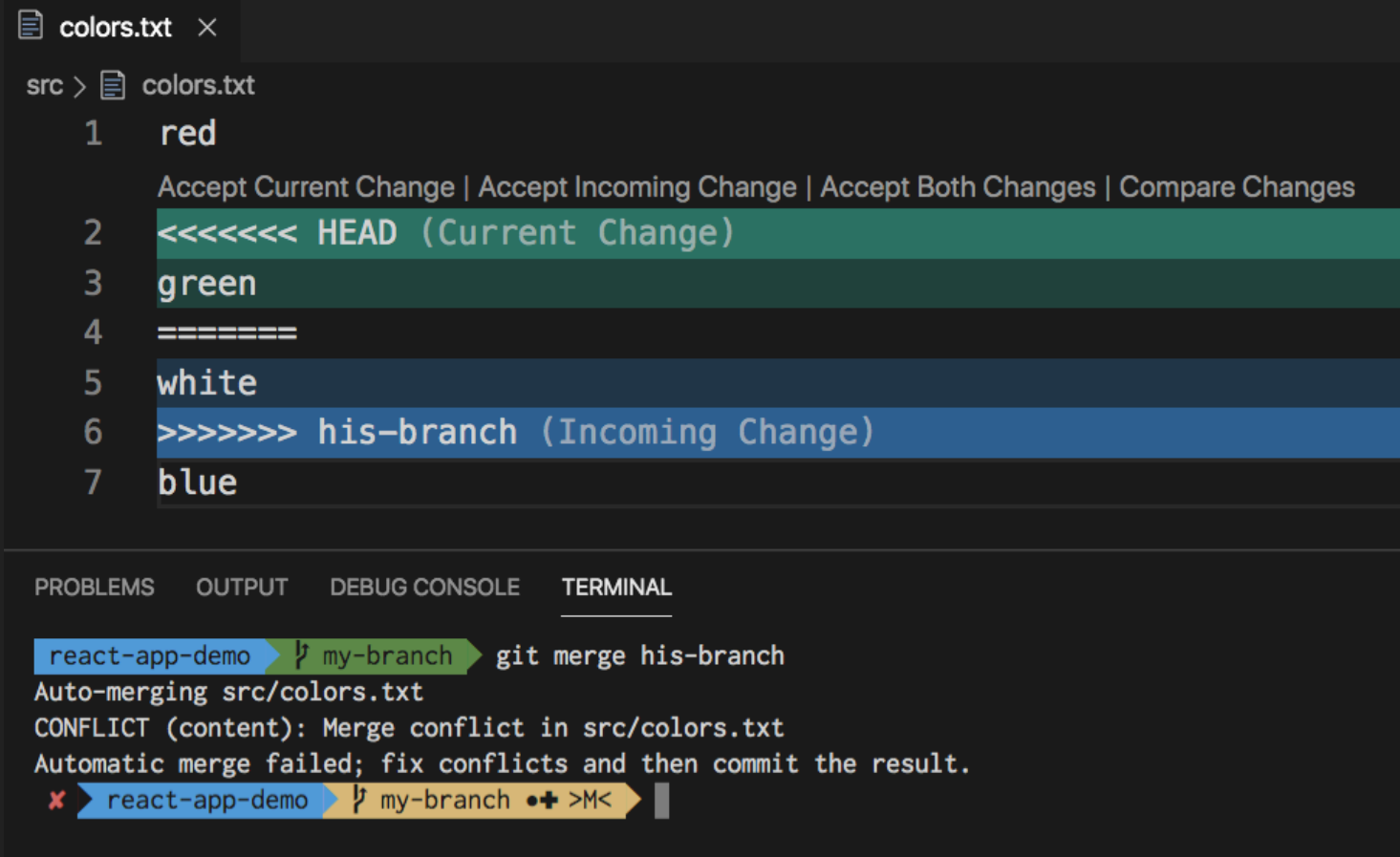
git merge



git merge conflict

```
01 // colors.txt
02
03
04 // in master, you have both created a branch from here
05 red
06 yellow
07 blue
08
09 // commit in your branch
10 red
11 green
12 blue
13
14 // commit in his branch
15 red
16 white
```

git merge conflict



The image shows a code editor with a file named `colors.txt` open. The editor displays a merge conflict between the `HEAD` (Current Change) and `his-branch` (Incoming Change). The conflict is resolved by accepting both changes, resulting in the following content:

```
src > colors.txt
1  red
   Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
2  <<<<<< HEAD (Current Change)
3  green
4  =====
5  white
6  >>>>>> his-branch (Incoming Change)
7  blue
```

The bottom panel of the editor shows the `TERMINAL` tab with the following output:

```
react-app-demo my-branch git merge his-branch
Auto-merging src/colors.txt
CONFLICT (content): Merge conflict in src/colors.txt
Automatic merge failed; fix conflicts and then commit the result.
react-app-demo my-branch
```

git merge conflict

```
01 // resolved colors.txt
02
03 // Accept current change
04 red
05 green
06 blue
07
08 // Accept incoming change
09 red
10 white
11 blue
12
13 // Accept both changes
14 red
15 green
16 white
17
```