

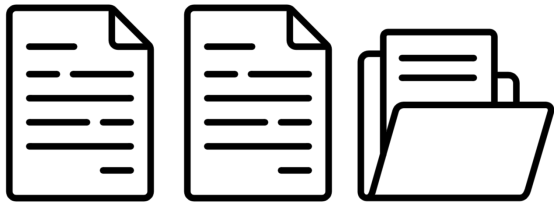
Git

In Practice

Undo working tree and
staging area



Three Undo Scenarios



Working Tree

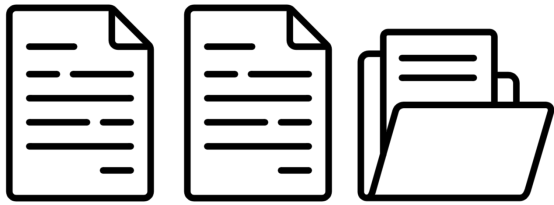


Staging Area (index file)



Commits

1 – Undo Working Tree Changes



Working Tree

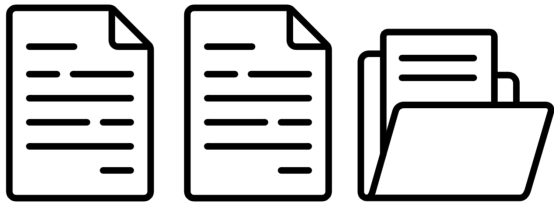


Staging Area (index file)



Commits

2 – Undo Staging Area Changes



Working Tree



Staging Area (index file)



Commits

2 – Undo Staging Area Changes



Working Tree



Staging Area (index file)



Commits

3 – Undo Commits History



Working Tree

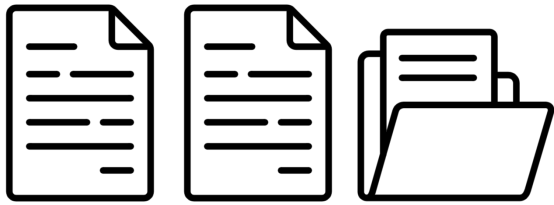


Staging Area (index file)



Commits

3 – Undo Commits History



Working Tree

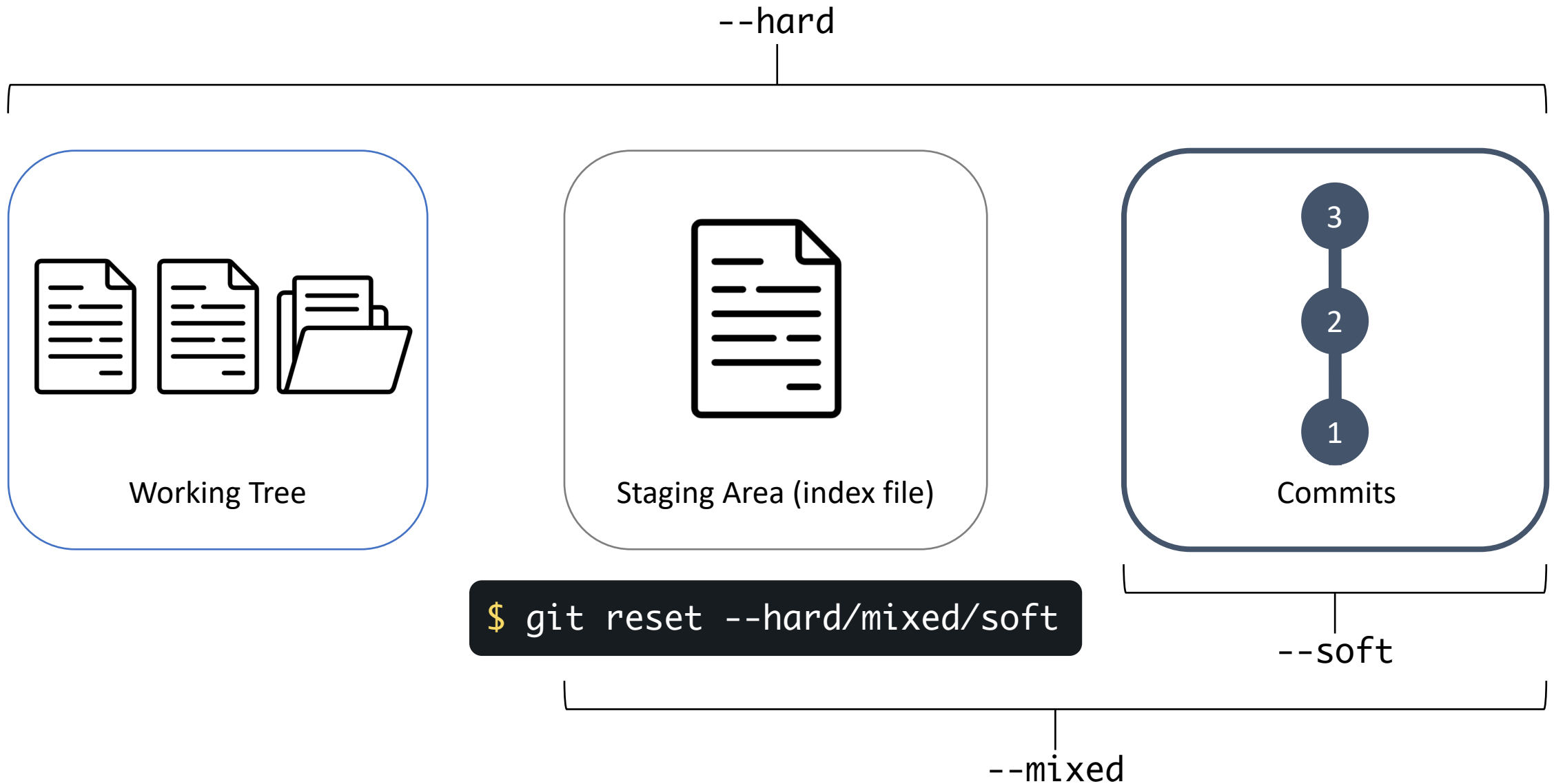


Staging Area (index file)

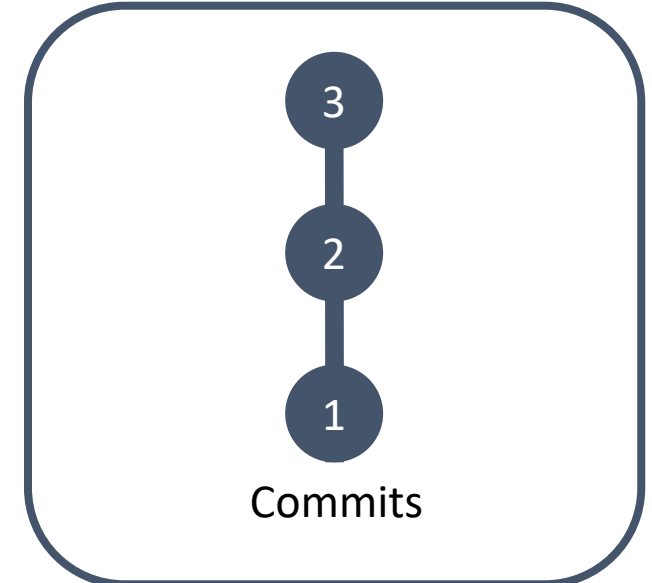
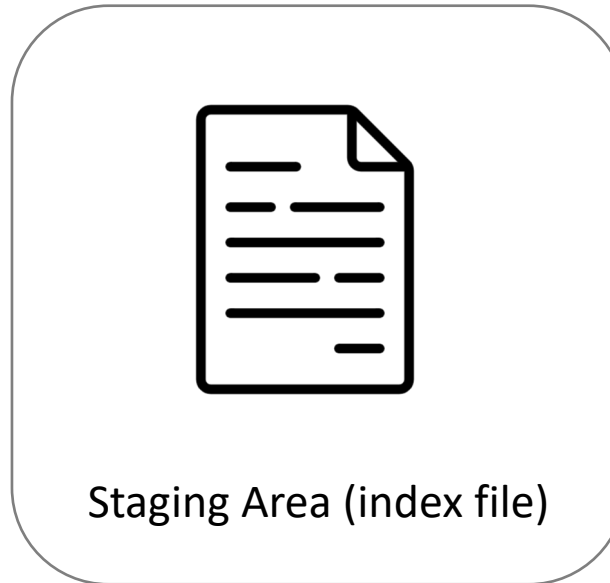


Commits

3 – Undo Commits History



3 – Undo Commits History



```
$ git reset --hard/mixed/soft
```

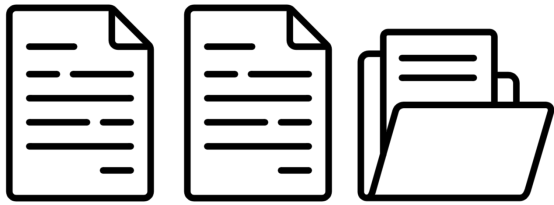
--hard

--soft

--mixed

1 – Undo Working Tree Changes

1 – Undo Working Tree Changes



Working Tree



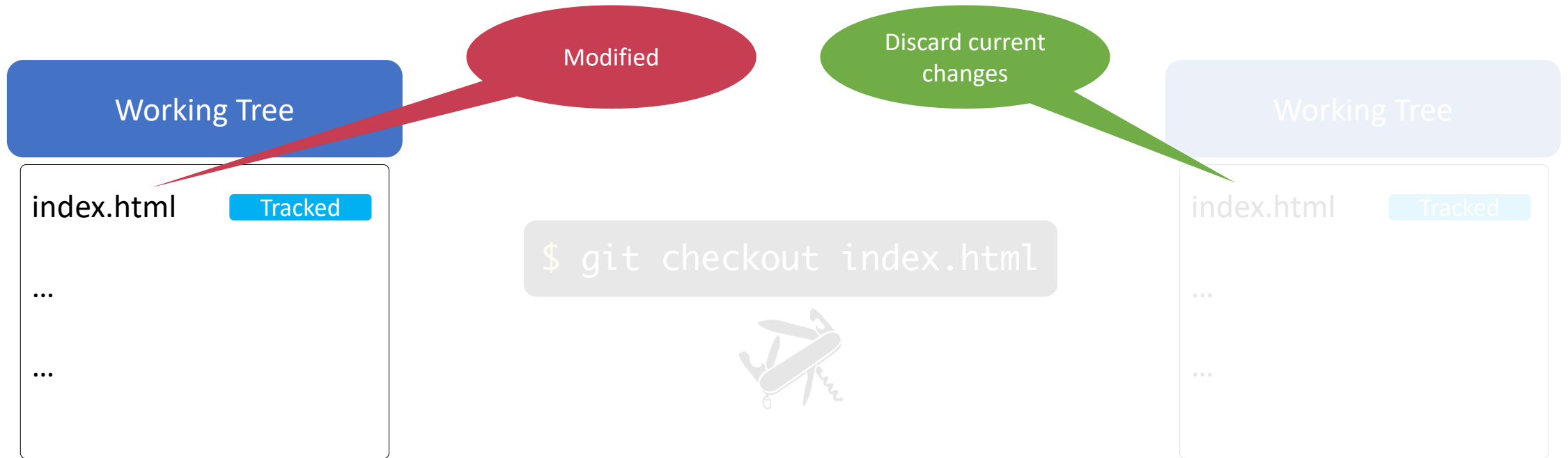
Staging Area (index file)



Commits

1 – Undo Working Tree Changes

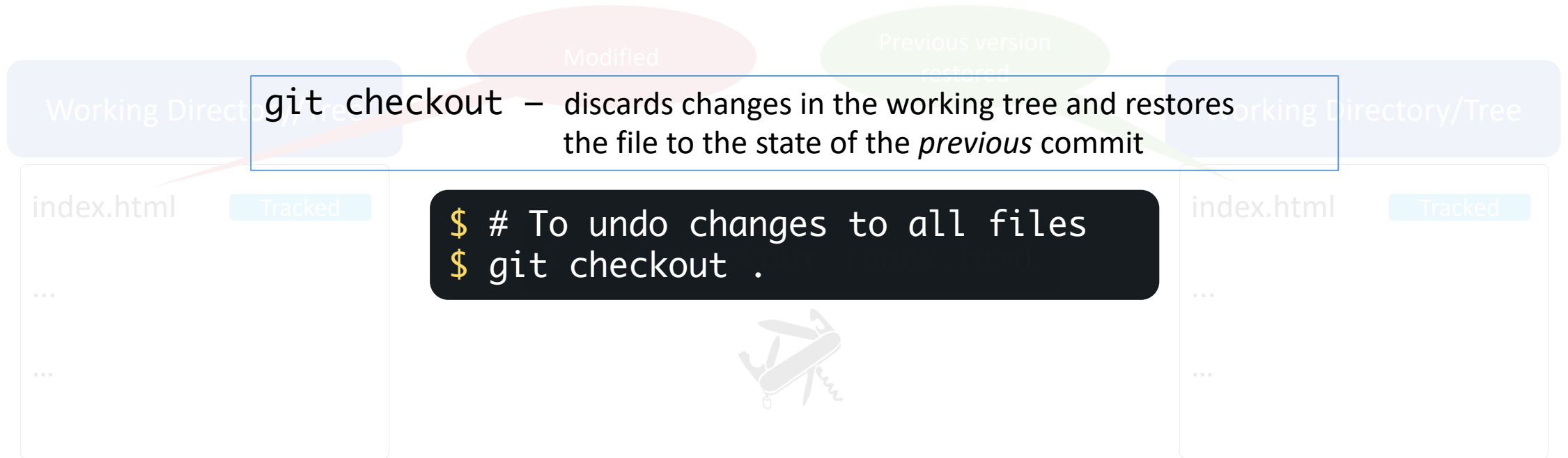
Unstaged



`git checkout` – discards changes in the working tree and restores the file to the state of the *previous* commit

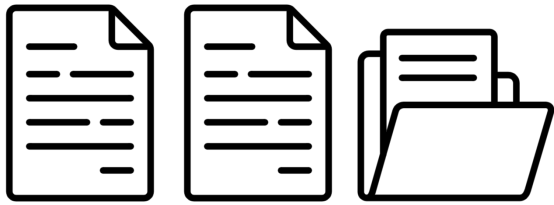
Unstaged

1 – Undo Working Tree Changes



Let us practice

2 – Undo Staging Area Changes



Working Tree

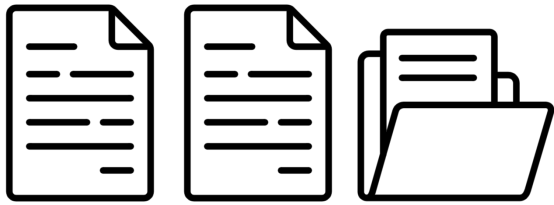


Staging Area (index file)



Commits

2 – Undo Staging Area Changes



Working Tree



Staging Area (index file)



Commits

Staged

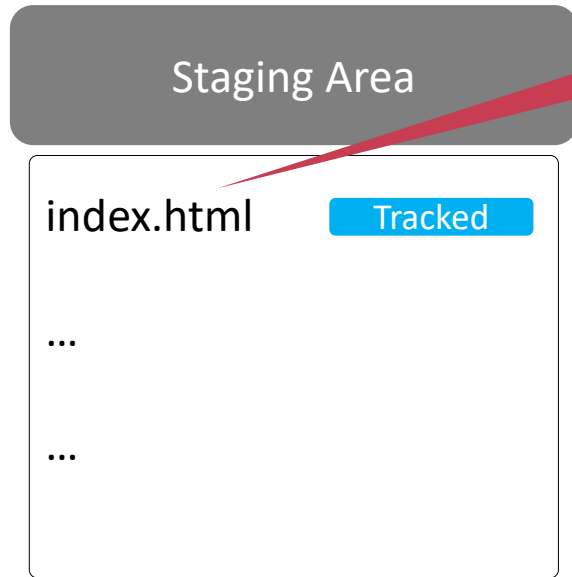
2 – Undo Staging Area Changes

- ❑ Unstaging staged changes is a two-step process
 - ❑ Step 1 – Unstage file from the staging area
 - ❑ Step 2 – Discard changes from the working tree
- ❑ New command in git v2.23+
 - ❑ `git restore --staged` and `git restore`

Step 1 – Undo Staging Area

Staged

Using restore



Newly Added or Modified

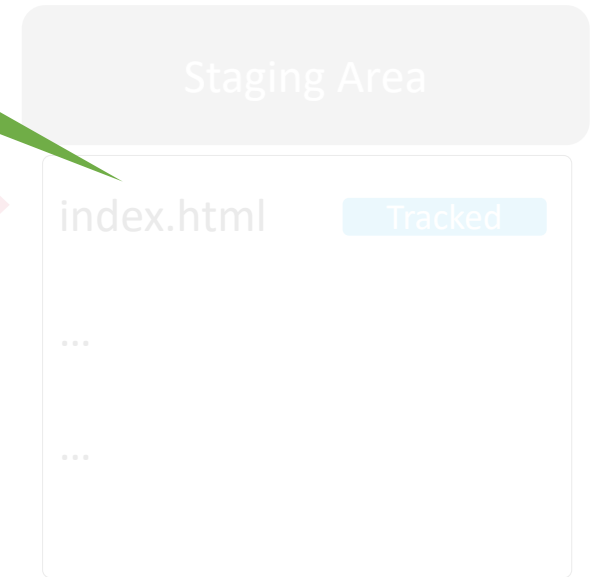
Unstages from staging area

`$ git restore --staged index.html`

`$ git rm --cached index.html`

`$ git reset index.html`

2.23



`git restore --staged` – unstages the recent git add/rm of tracked files from the staging area.

Step 2 – Undo Working Tree

Staged

Using restore

Working Tree

index.html

Tracked

...

...

Modified

Discard current
changes

Working Tree

index.html

Tracked

...

...

\$ git restore index.html

\$ git checkout index.html

2.23

`git restore` – discards changes in the working tree
and restores the file to the state of the *previous* commit

Let us practice

Scenarios – Summary

Undo Staging Area

```
$ git restore --staged <filename>
```

2.23

```
$ git rm --cached <filename>
```

```
$ git reset <filename>
```

Undo Working Tree

```
$ git restore <filename>
```

2.23

```
$ git checkout <filename>
```



Undo Commits History

```
$ git reset --hard/mixed/soft
```

Takeaways

✓ Commands

Start a working area

1. `git init`

Work on the current change

2. `git add`

3. `git rm`

4. `git rm --cached`

5. `git restore -staged`

6. `git restore`

7. `git checkout`

Grow, mark and tweak your common history

8. `git commit`

9. `git reset`

Show current status and commit history

10. `git status`

11. `git ls-files`

12. `git log`