

Module 2



Init



Add & Commit



Logs



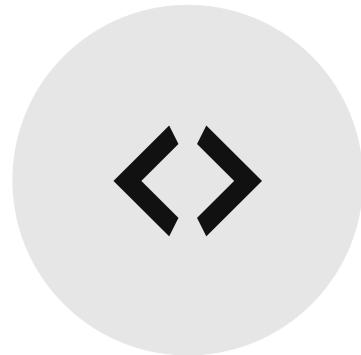
Checkout



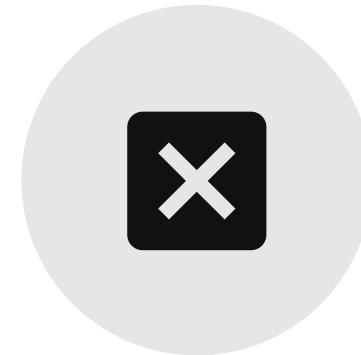
Restore & Reset



Git Ignore & Git Keep



Git Aliases



Removing Git

Initializing Git

Initializing Git

- Initializing Git lets us track changes.

`git init`

Checking Status

Checking Status

- Checking the status in Git shows us what has changed, what's ready to be saved, and what's not tracked yet.
- By default, all the files are untracked by Git.

`git status`

Adding to Staging Area

Adding to Staging Area

- Adding to the staging area in Git is like making a checklist before saving changes.
- It helps us to pick which updates to include in the next save (commit, version).

git add filename

git add .

Committing

Committing

- Committing in Git is like saving a version of our project so we can go back to it later if needed.

```
git commit -m "message"
```

Logs

Logs

- Git logs shows a history of all the changes (commits) we've made, like a timeline of our project.

`git log`

`git log --oneline`

```
ashis@DESKTOP-N09PGF5 MINGW64 /d/Documents/Documents/Git-GitHub-Session/git (master)
● $ git log
commit efdda6e5ece4b32a313f8ab467bd41ea91076653 (HEAD -> master)
Author:
Date:
initial commit
```



Commit Hash

```
ashis@DESKTOP-N09PGF5 MINGW64 /d/Documents/Documents/Git-GitHub-Session/git (master)
● $ git log
commit efdda6e5ece4b32a313f8ab467bd41ea91076653 (HEAD -> master)
Author:
Date:

initial commit
```



Current Position

```
ashis@DESKTOP-N09PGF5 MINGW64 /d/Documents/Documents/Git-GitHub-Session/git (master)
● $ git log
commit efdda6e5ece4b32a313f8ab467bd41ea91076653 (HEAD -> master)
Author:
Date:

initial commit
```



Current Branch

Warmup 3 🌶

Warmup 3 ⚡

- Create a new directory called **wup3** and navigate into it.
- Initialize a Git repository within the directory.
- Create two Python files: **src/main.py** and **app.py**.
- Add and commit them with the message “***initial commit***”.

Contd..

Warmup 3 ⚡

- Write `print("line 1")` in both `src/main.py` and `app.py`, then add and commit these files to Git with the message “**add line 1**”.
- Write `print("line 2")` in both files, then add and commit these changes with the message “**add line 2**”.
- Again, write `print("line 3")` in both files, then add and commit these changes with the message “**add line 3**”.
- View the commit history.

Appending Commit

Appending Commit

- We can append our commit with the previous commit using amend.

```
git commit -m "message" --amend
```

Warmup 4 ↗

Warmup 4 ⚡

- Create a new directory called **wup4** and navigate into it.
- Initialize a Git repository within the directory.
- Create two Python files: **src/main.py** and **app.py**.
- Add **app.py** file and commit with the message “***initial commit***”.
- Now, add **main.py** file and commit with the message “***initial commit with main.py***”. Note, it should not create new commit. Append the commit with previous commit.
- View the commit history.

Unstaging Changes

Unstaging Changes

- We can unstage the changes to remove files from staging area.

`git reset filename`

`git reset folder`

`git reset .`

Warmup 5 🌶

Warmup 5 ↴

- Create a new directory called **wup5** and navigate into it.
- Initialize a Git repository within the directory.
- Create two Python files: **src/main.py** and **app.py**.
- Add and commit them with the message “***initial commit***”.
- Create a new file named **temp.py** and add it to staging area.
- Check the status.
- Now, unstage the **temp.py** from the staging area and again check the status.

Viewing Previous Version

Viewing Previous Version

- We can view our previous version of the files, like going back in time.

```
git checkout <commit-hash>
```

```
git checkout master
```

Viewing Future Commits

- Checkout hides the commits that are ahead of current HEAD.

```
git log --all
```

Warmup 6 🌶

Warmup 6 ↴

- Create a new directory called **wup6** and navigate into it.
- Initialize a Git repository within the directory.
- Create a text file: **friends.txt**.
- Add and commit with the message “***initial commit***”.

Contd..

Warmup 6 ↴

- Write friend name in ***friends.txt***, then add and commit file to Git.
- Write another friend name in ***friends.txt***, then add and commit these changes.
- Again, write another friend name in ***friends.txt***, then add and commit these changes.
- View the commit history.
- Then, view the version that contains first friend name.

Restoring Previous Version

Restoring Previous Version

- While we can view our previous version of the files, we can also restore them back in our directory.

git checkout <commit-hash> filename

git checkout <commit-hash> folder

git checkout <commit-hash> .

Warmup 7 ↩

Warmup 7 ↪

- Create a new directory called **wup7** and navigate into it.
- Initialize a Git repository within the directory.
- Create a Python file: **fruits.py**.
- Add and commit with the message “***initial commit***”.

Contd..

Warmup 7 ↪

- Write `print("apple")` in `fruits.py`, then add and commit file to Git with the message “**add apple**”.
- Overwrite `fruits.py` with `print("banana")`, then add and commit these changes with the message “**add banana**”.
- Again, overwrite `fruits.py` with `print("cherry")` in file, then add and commit these changes with the message “**add cherry**”.
- View the commit history.

Contd..

Warmup 7 ⚡

- Restore the apple in ***fruits.py***, then add and commit file to Git with the message “***restore apple***”.

Reset

Reset

- Mixed Reset
- Soft Reset
- Hard Reset

Mixed Reset

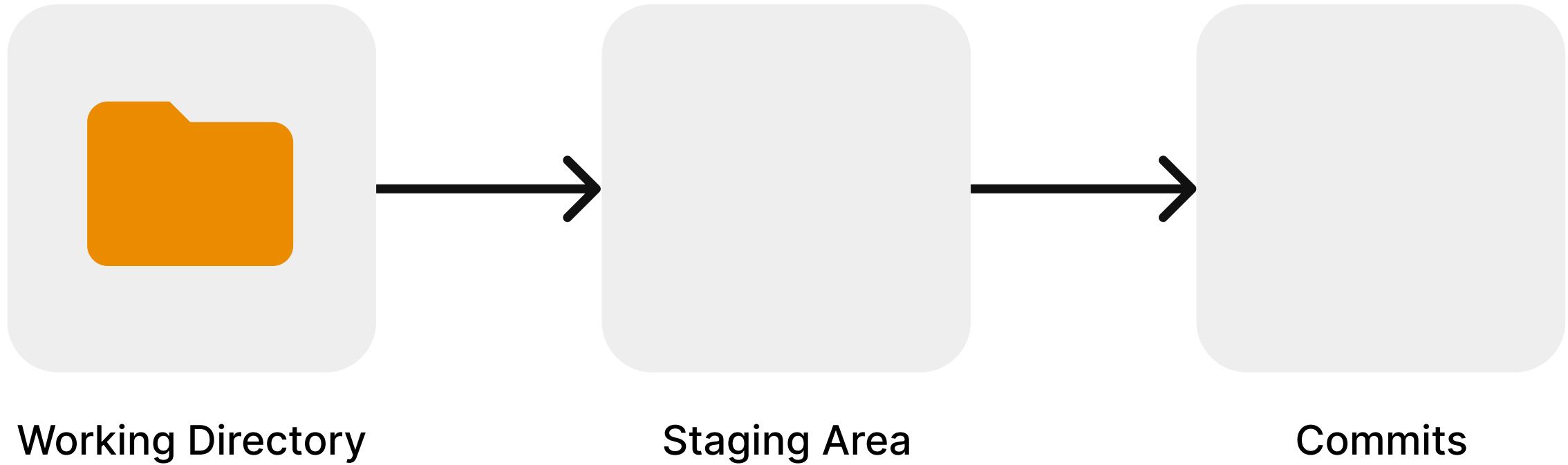
Mixed Reset

- You committed something, but you want to undo the commit and unstage changes.
- Mixed Reset will not discard the changes.

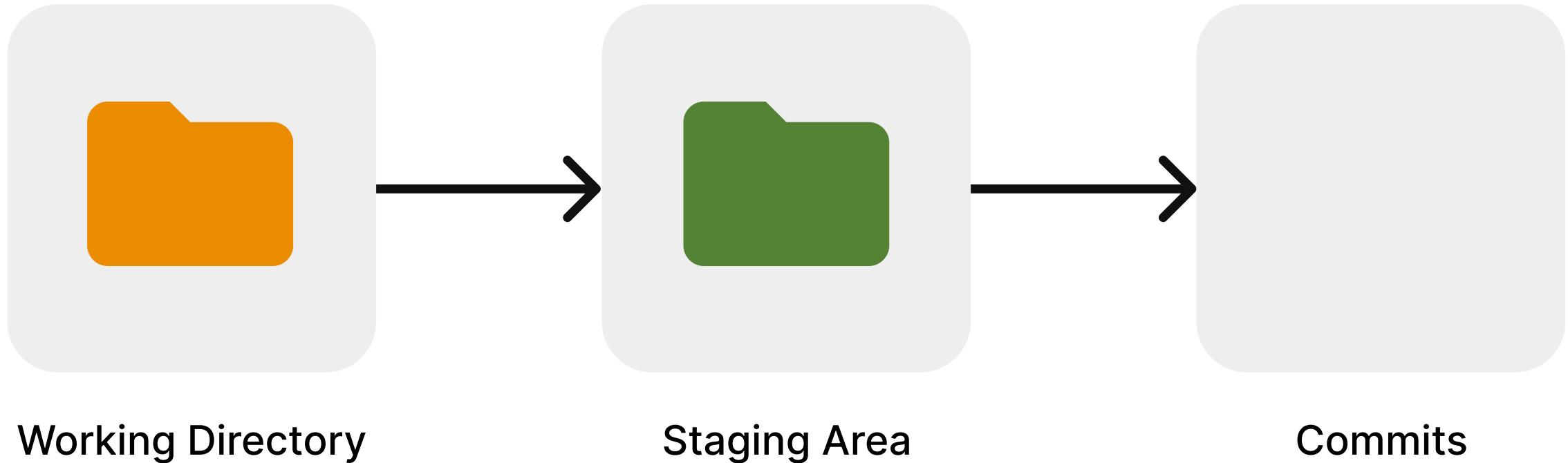
```
git reset --mixed HEAD~n
```

```
git reset --mixed <commit-hash>
```

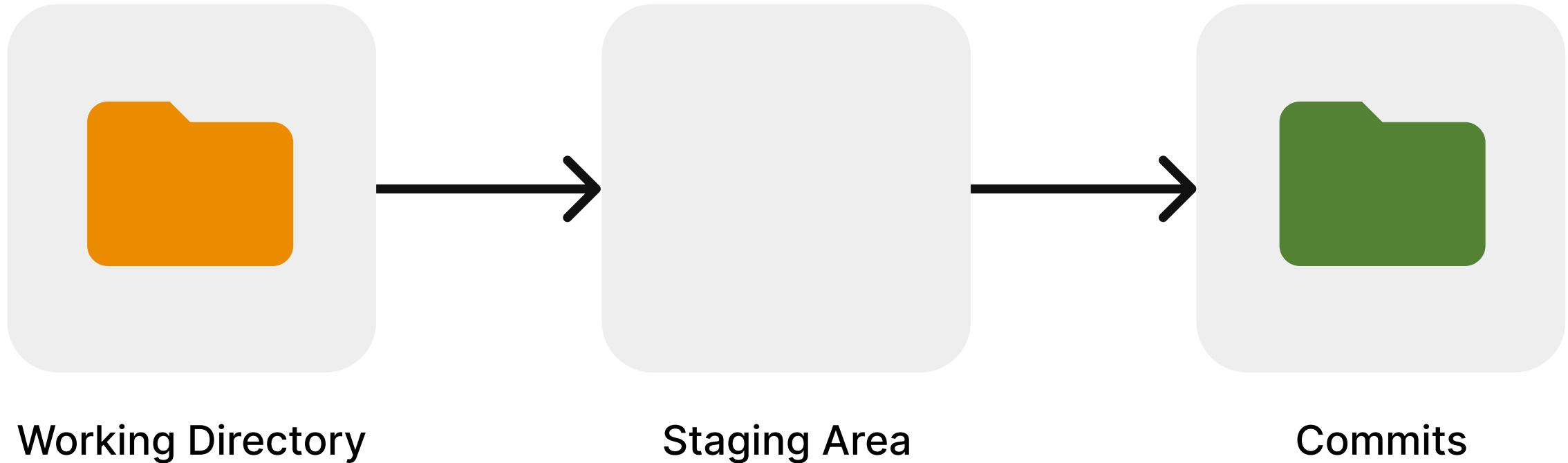
Example 1



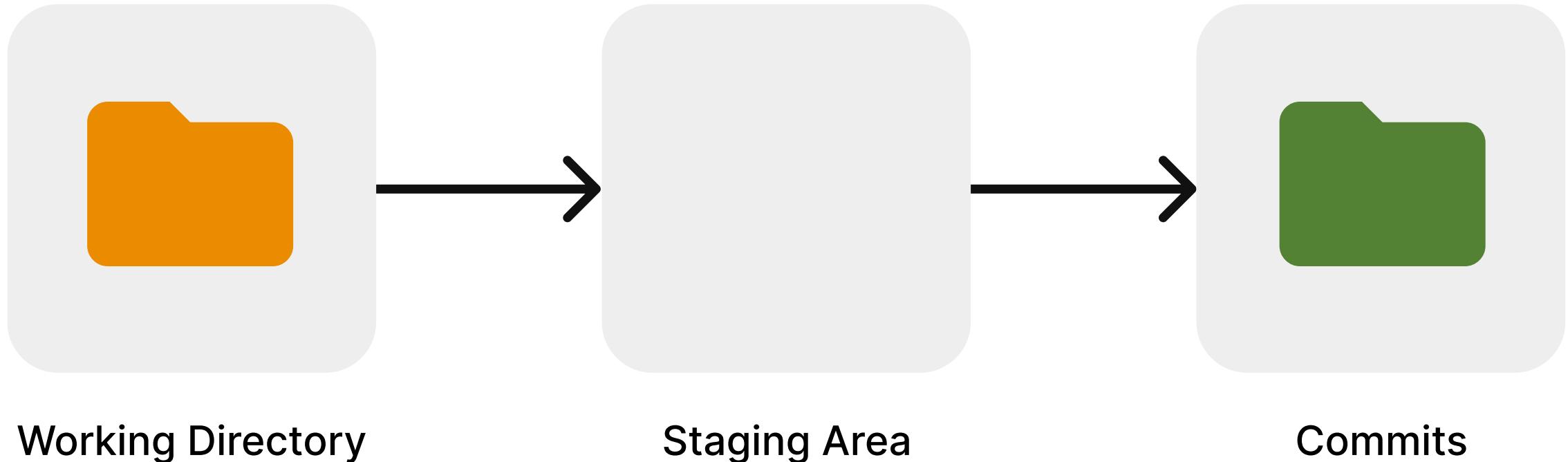
Example 1



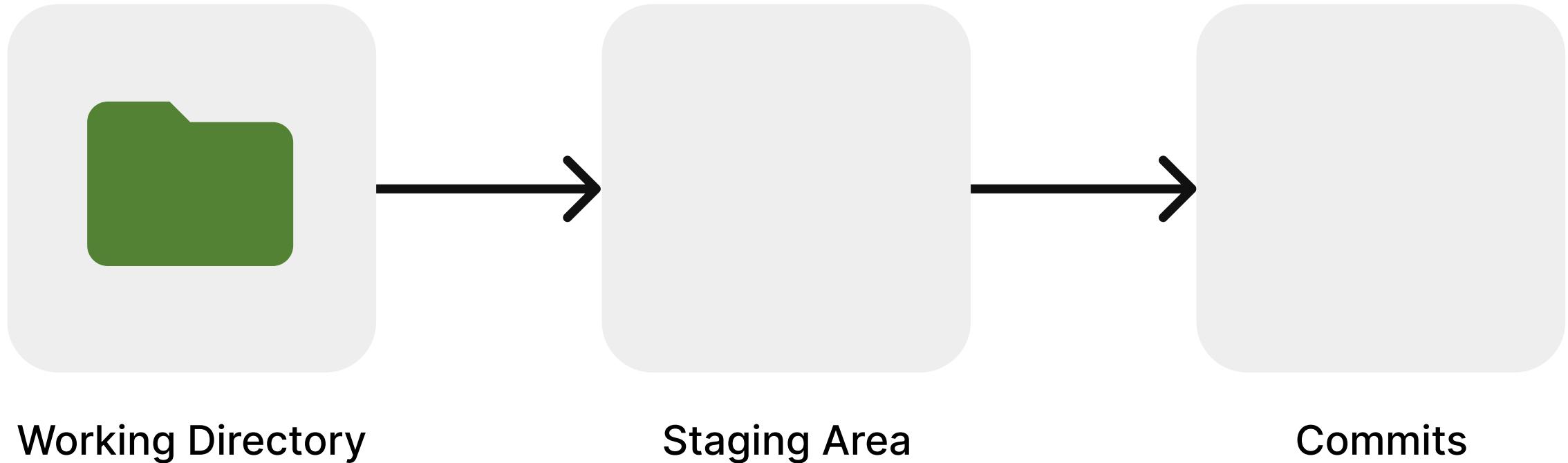
Example 1



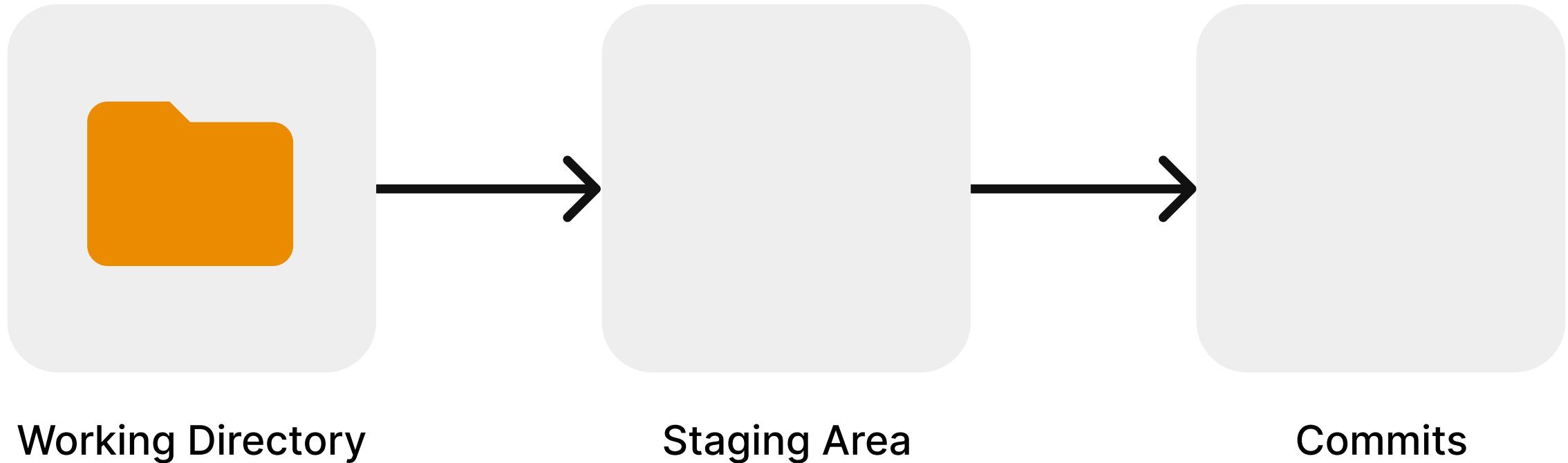
Mixed Reset



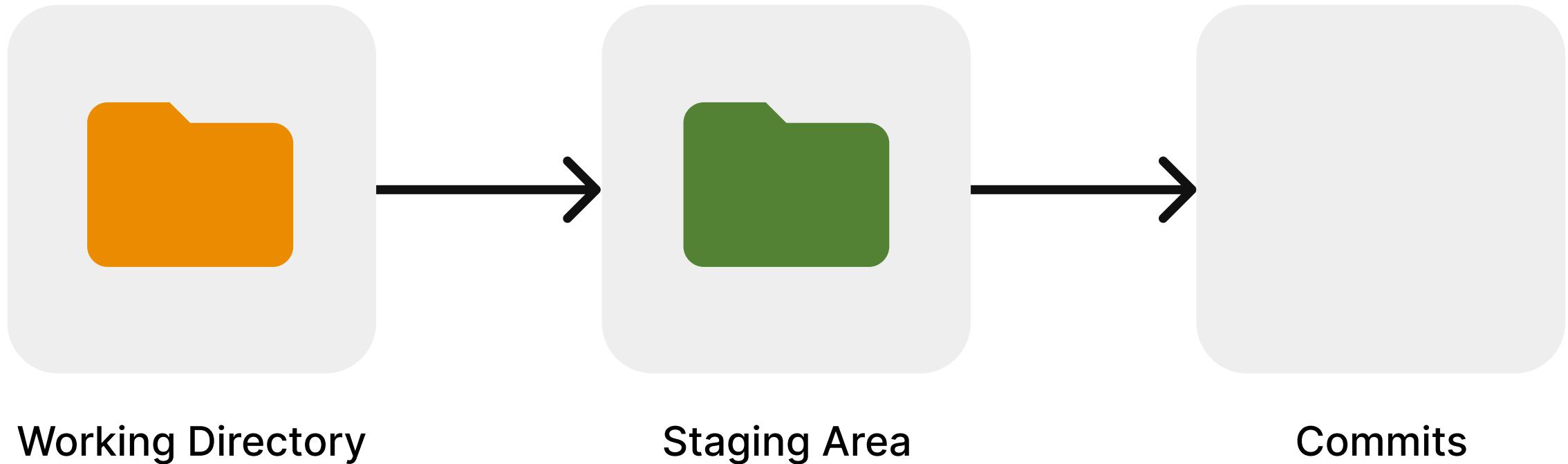
Mixed Reset



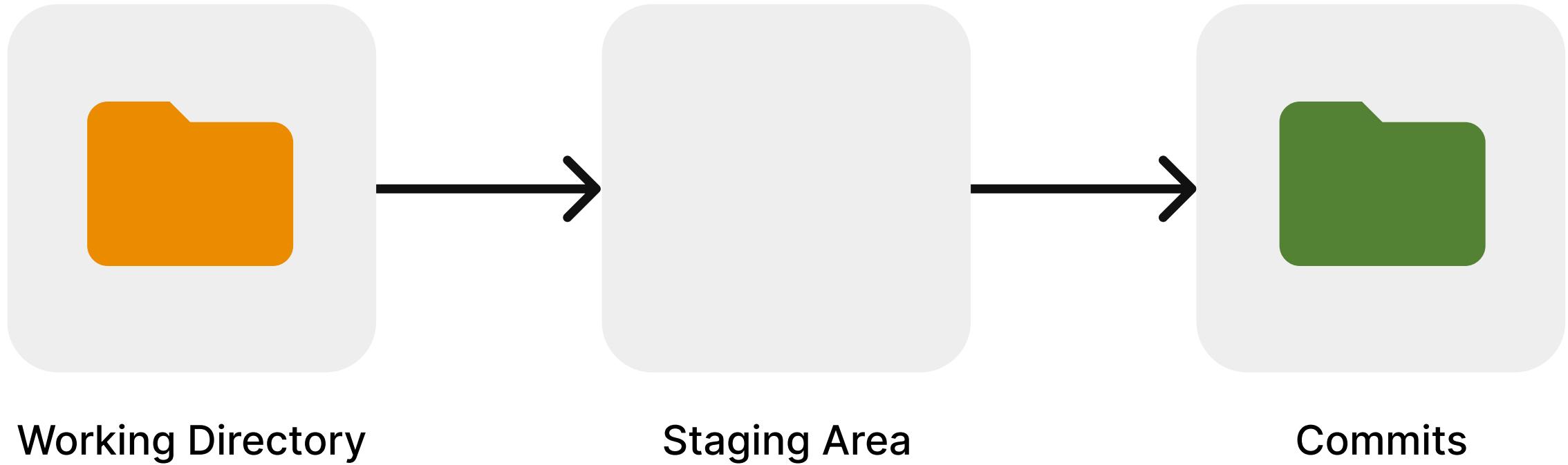
Example 2



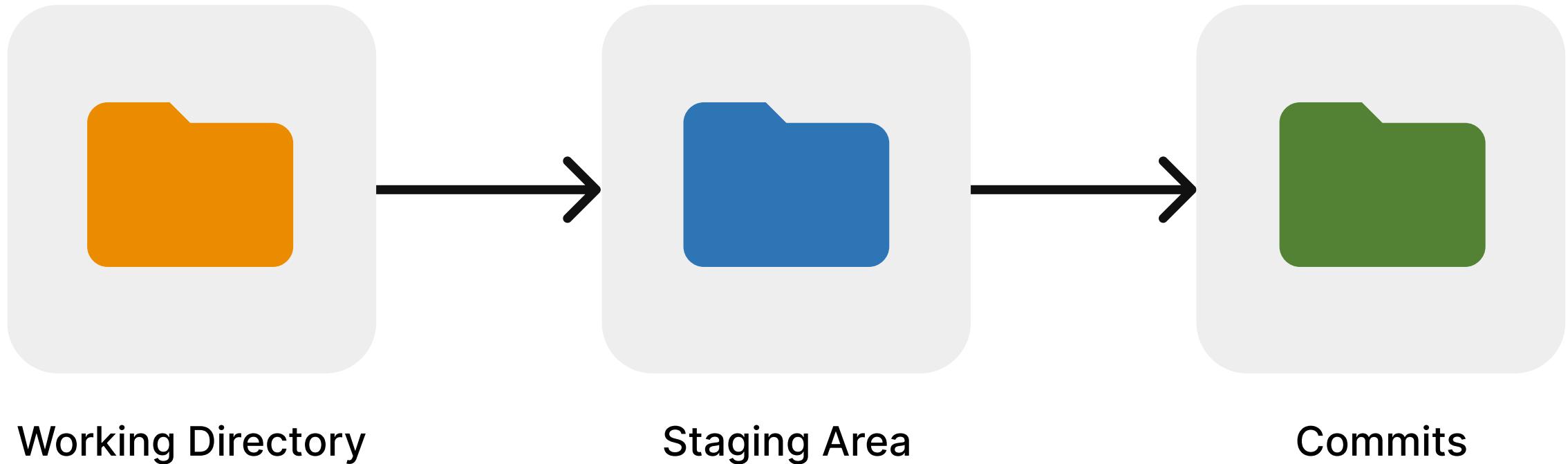
Example 2



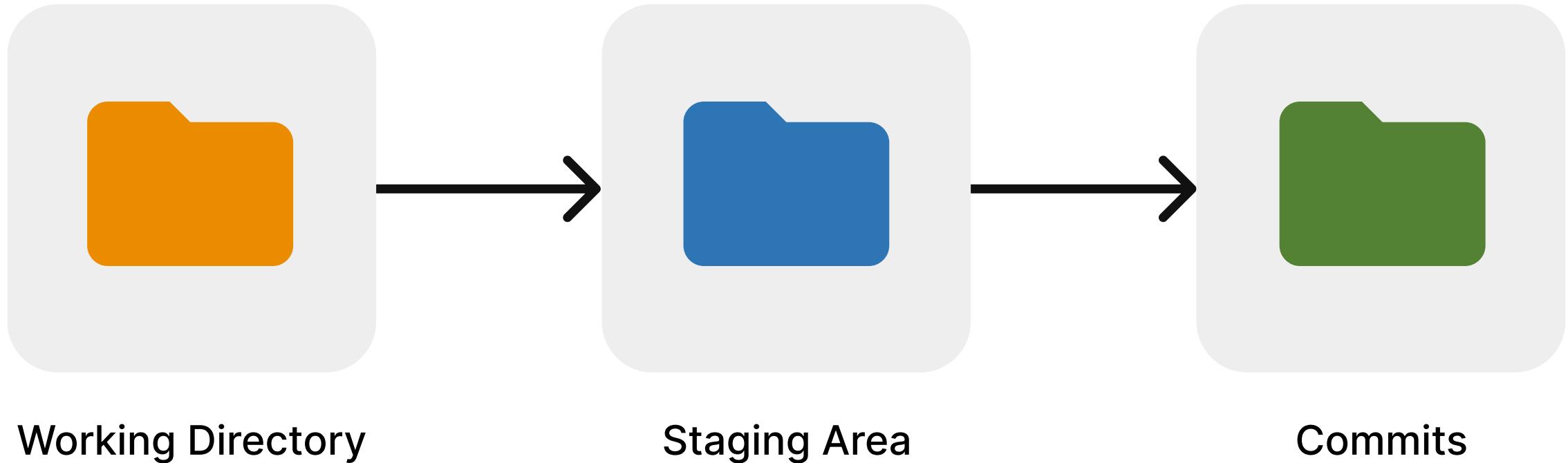
Example 2



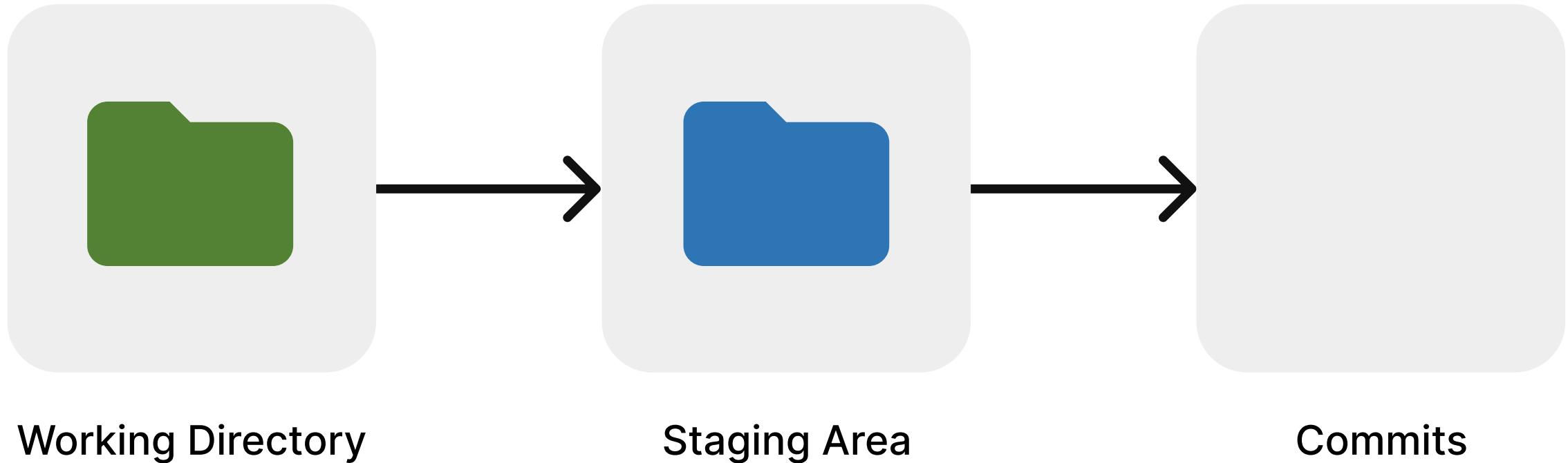
Example 2



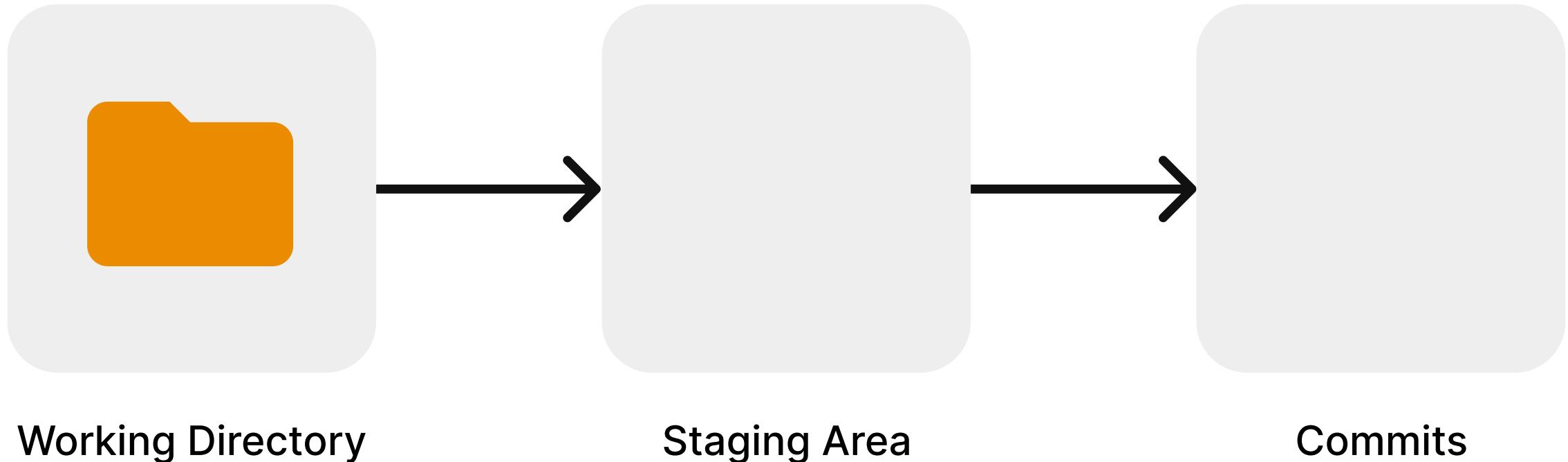
Mixed Reset



Mixed Reset



Mixed Reset



Warmup 8 

Warmup 8 ↴

- Create a new directory called **wup8** and navigate into it.
- Initialize a Git repository within the directory.
- Create a Python file: **mix.py**.
- Add and commit with the message “***initial commit***”.

Contd..

Warmup 8 ↩

- Write **`print("soft one")`** in **`mix.py`**, then add and commit the file to Git with the message "**`first mix`**".
- Write **`print("soft two")`** in file, then add and commit these changes with the message "**`second mix`**".
- Again, write **`print("soft 3")`** in file, then add and commit these changes with the message "**`third mix`**".
- View the commit history.

Contd..

Warmup 8 ↩

- Again, write **`print("remix")`** in file, then add to staging area.
- Now, perform a mixed reset to undo the commit “***third mix***”.
- ***Check the status.***
- Again, commit the file with message “***removed third mix and add remix***”.
- View the commit history.

Soft Reset

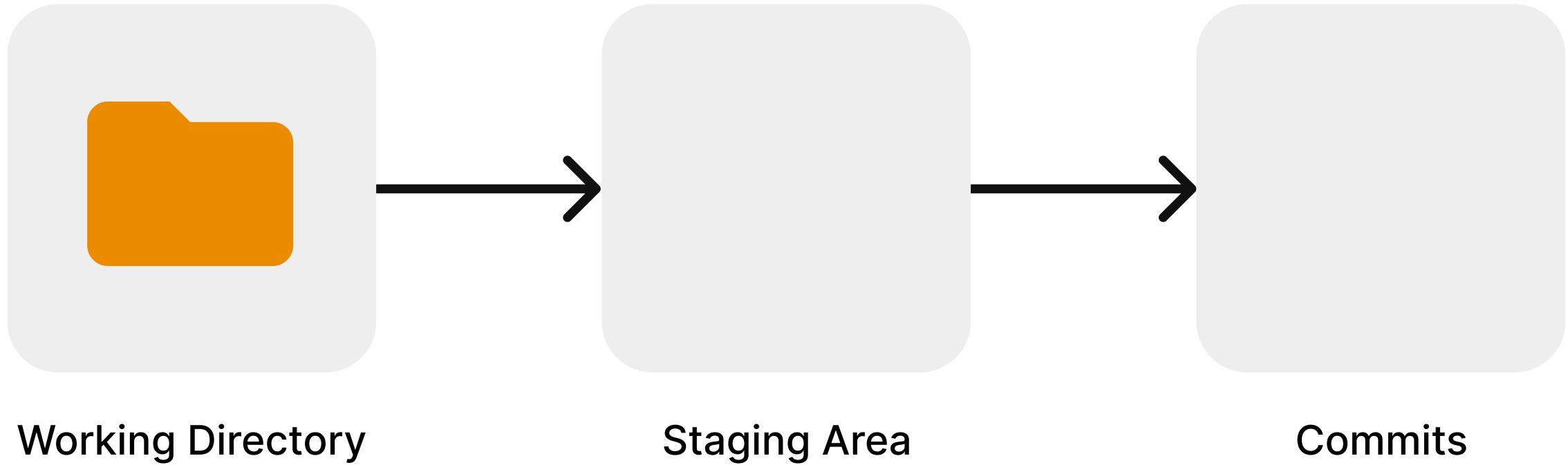
Soft Reset

- You committed something, but you want to undo the commit and stage changes.
- Soft Reset will not discard the changes.

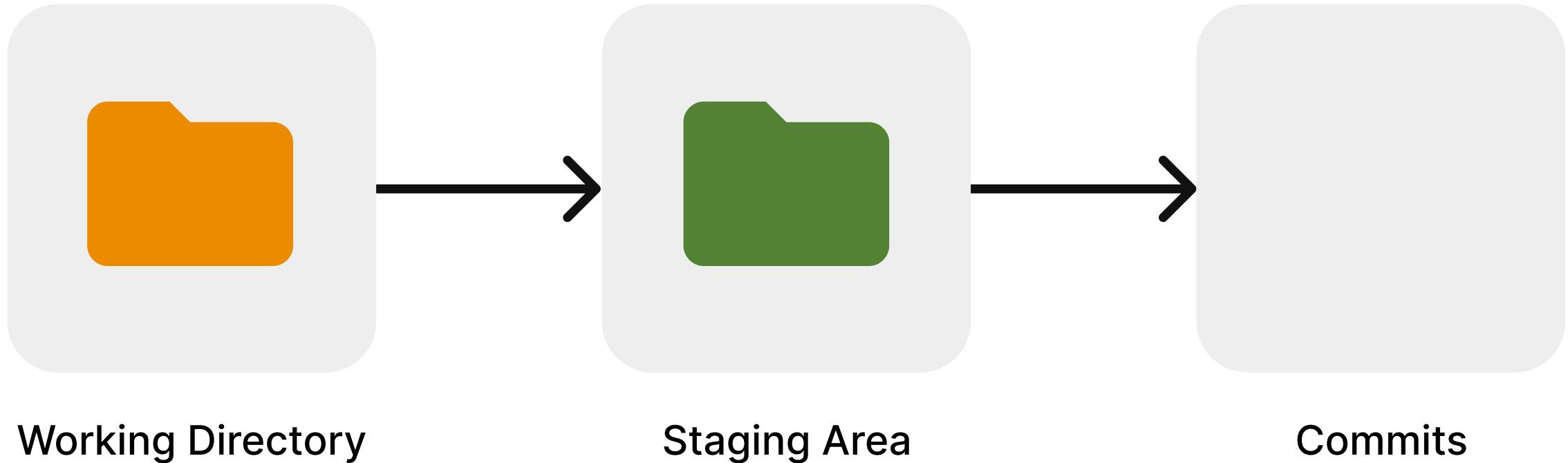
```
git reset --soft HEAD~n
```

```
git reset --soft <commit-hash>
```

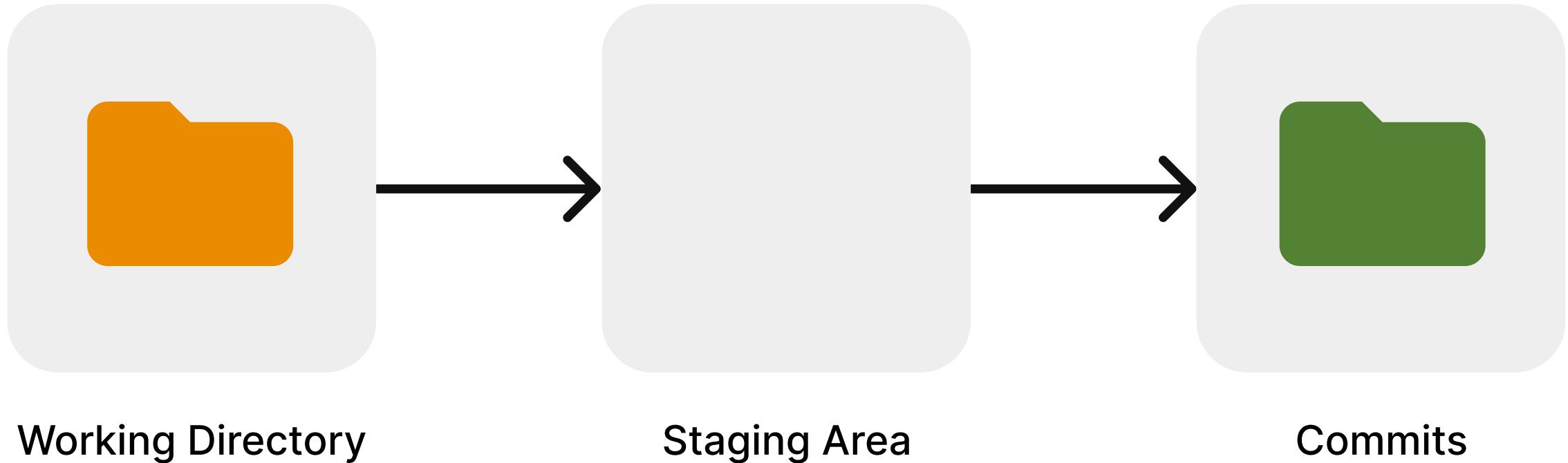
Example 1



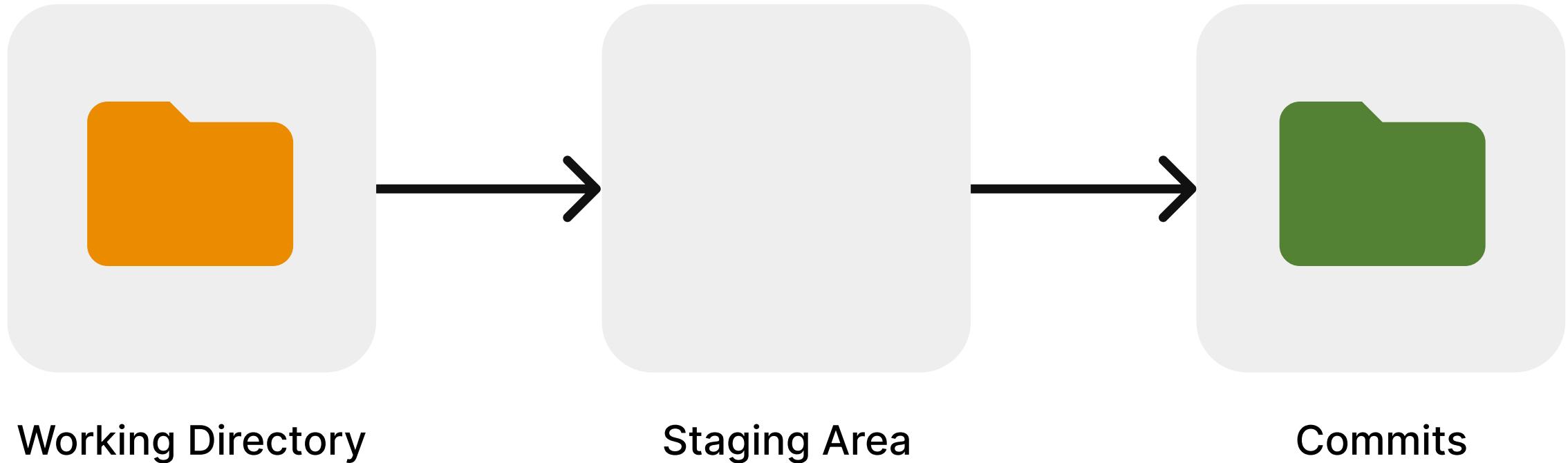
Example 1



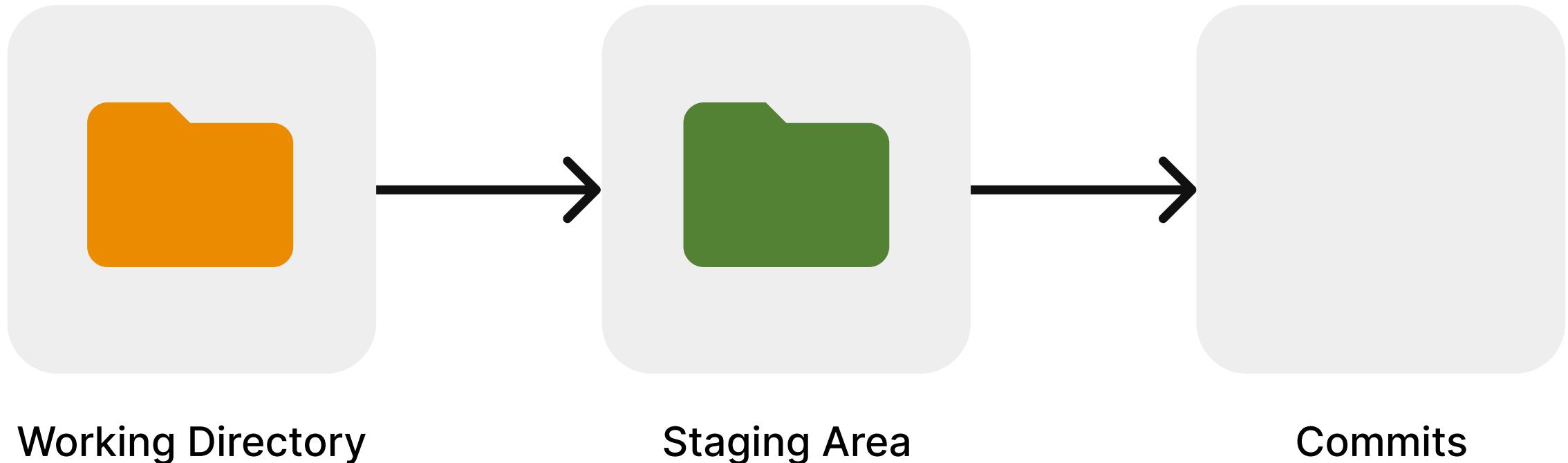
Example 1



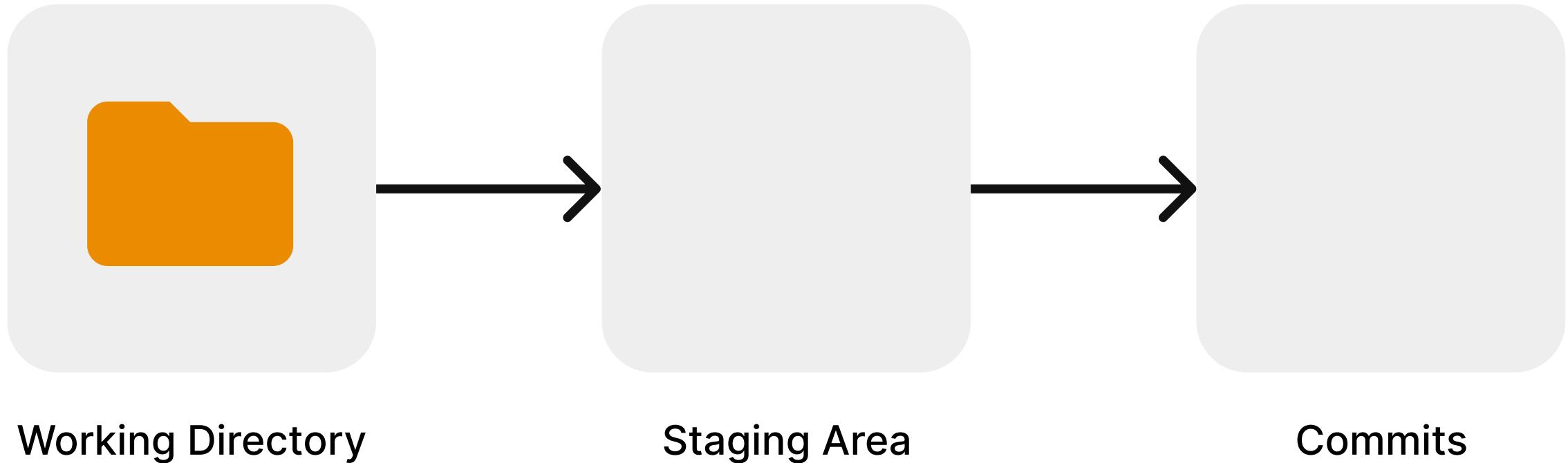
Soft Reset



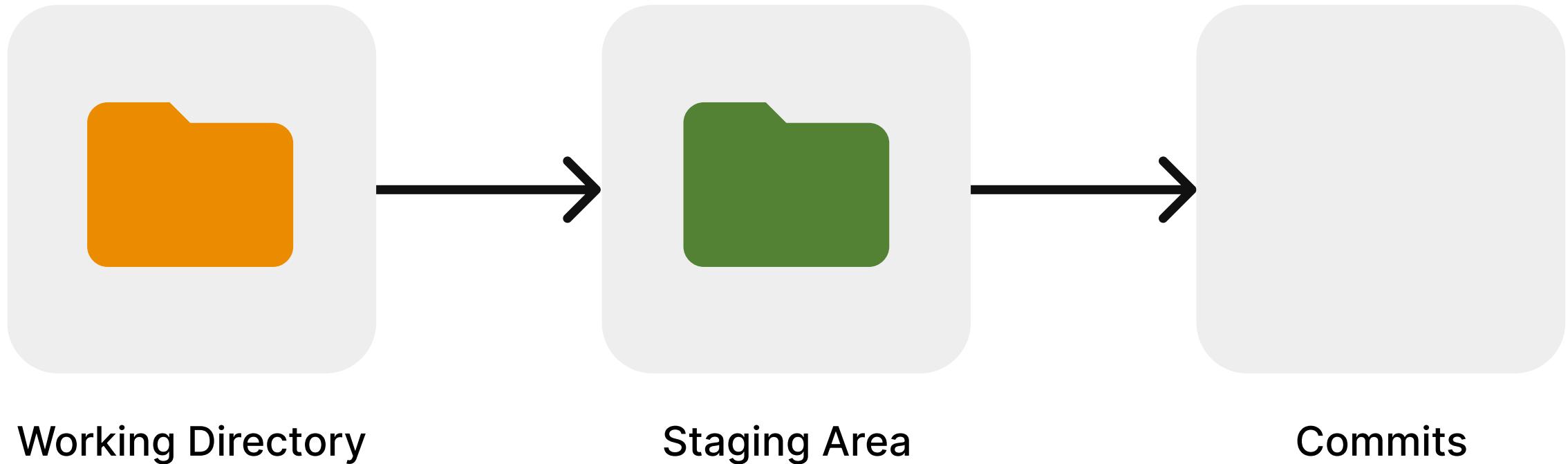
Soft Reset



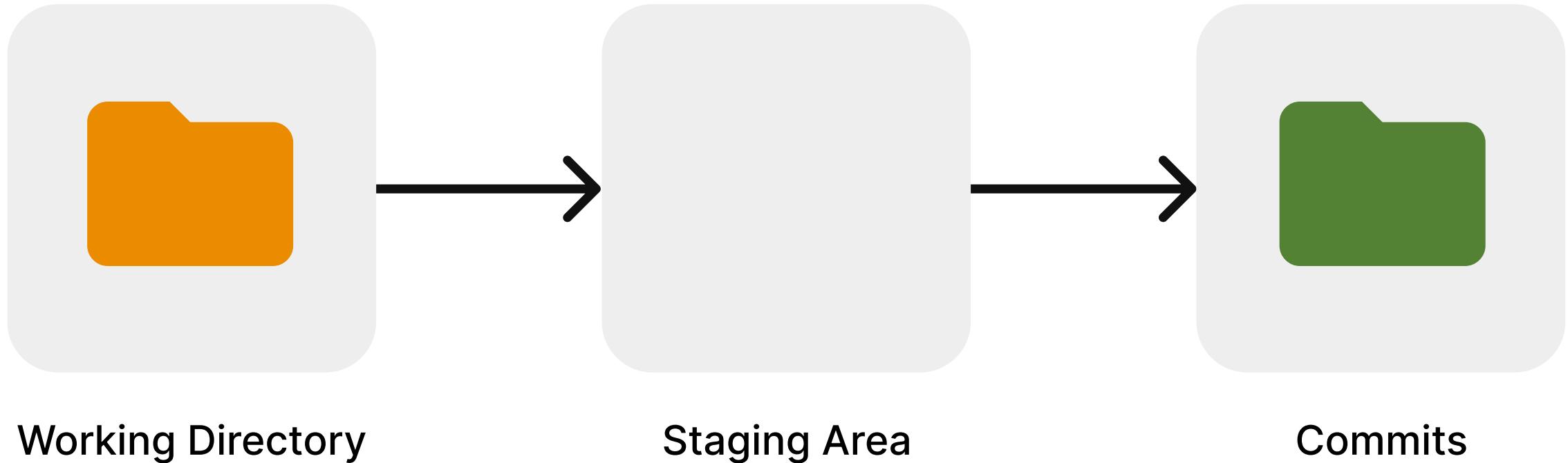
Example 2



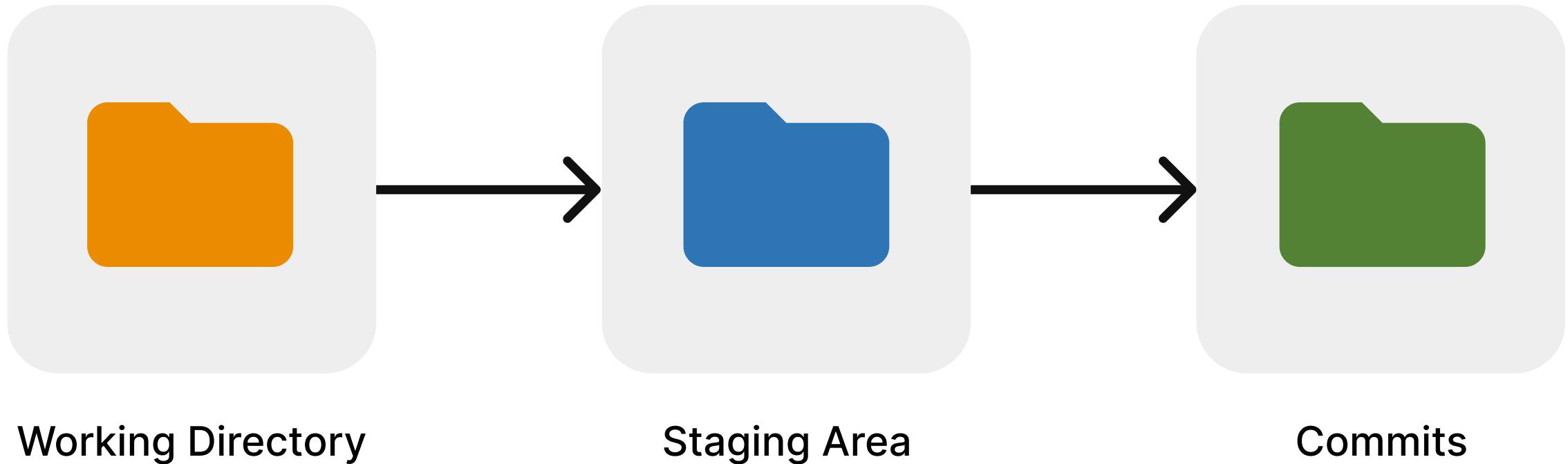
Example 2



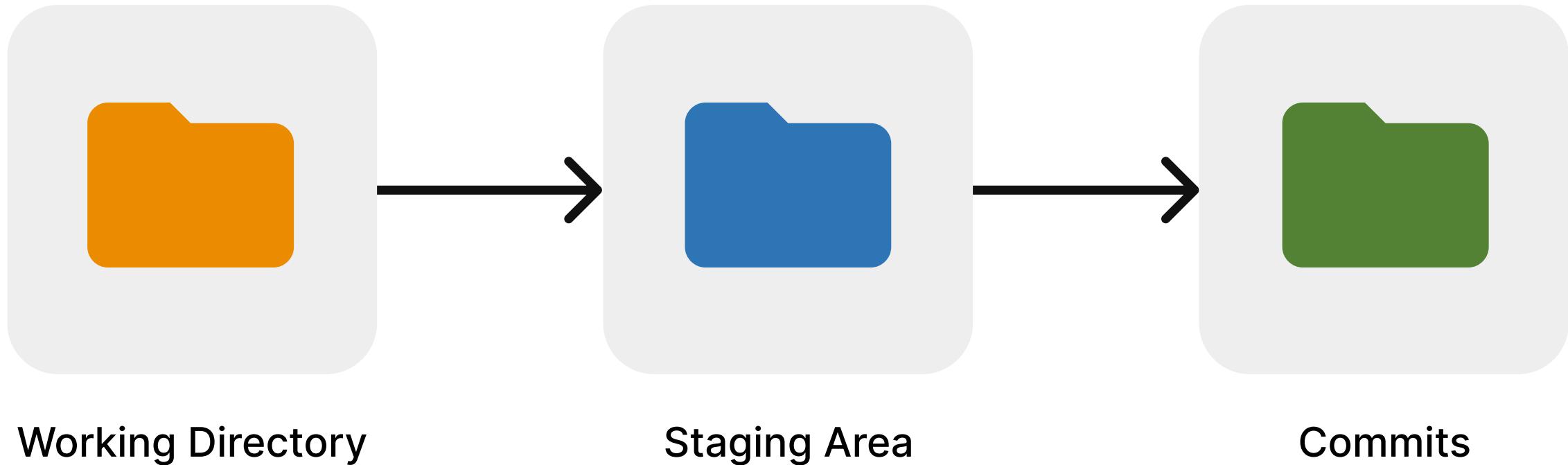
Example 2



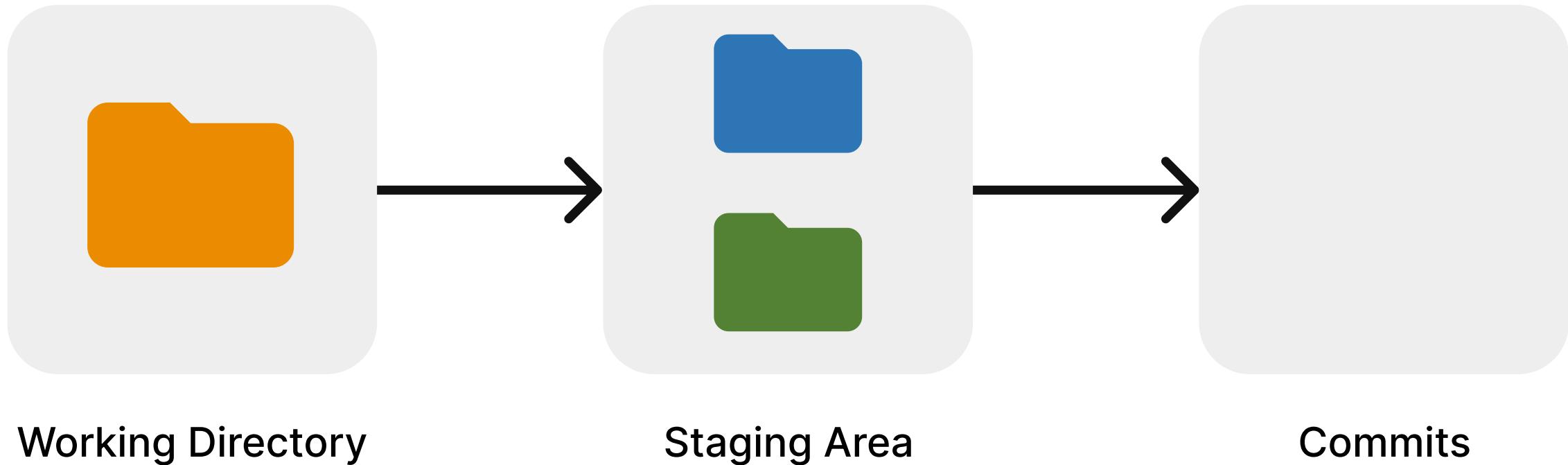
Example 2



Soft Reset



Soft Reset



Warmup 9 ↗

Warmup 9 ↴

- Create a new directory called **wup9** and navigate into it.
- Initialize a Git repository within the directory.
- Create two files: **chai.py** and **bill.txt**.
- Add and commit with the message “***initial commit***”.

Contd..

Warmup 9 ↴

- Write `print("black chai")` in `chai.py`, then add and commit the file to Git with the message "**add black chai**".
- Write `print("lemon chai")` in `chai.py`, then add and commit these changes with the message "**add lemon chai**".
- Again, write `print("milk chai")` in `chai.py`, then add and commit these changes with the message "**add milk chai**".
- View the commit history.

Contd..

Warmup 9 ↴

- Again, write `print("ginger chai")` in `chai.py`, then add to staging area.
- Now, perform a soft reset to undo the commit “**add milk chai**”.
- **Check the status.**
- Again, commit the `chai.py` with message “**replace milk with ginger chai**”. Write “**ginger chai -Rs.25**” in `bill.txt`, then add and commit file to Git with the message “**pay for ginger chai**”.
- View the commit history.

Hard Reset

A promotional image for Spider-Man: No Way Home. It features two versions of Spider-Man in a dark, space-like environment. On the left, Peter Parker is shown from the chest up, looking towards the right. On the right, Tony Stark is shown from the chest up, also looking towards the right. Both characters have their signature spider webs visible around them. The background is dark with some glowing particles.

#SPIDERMANNOWAYHOME

WITH GREAT POWER...

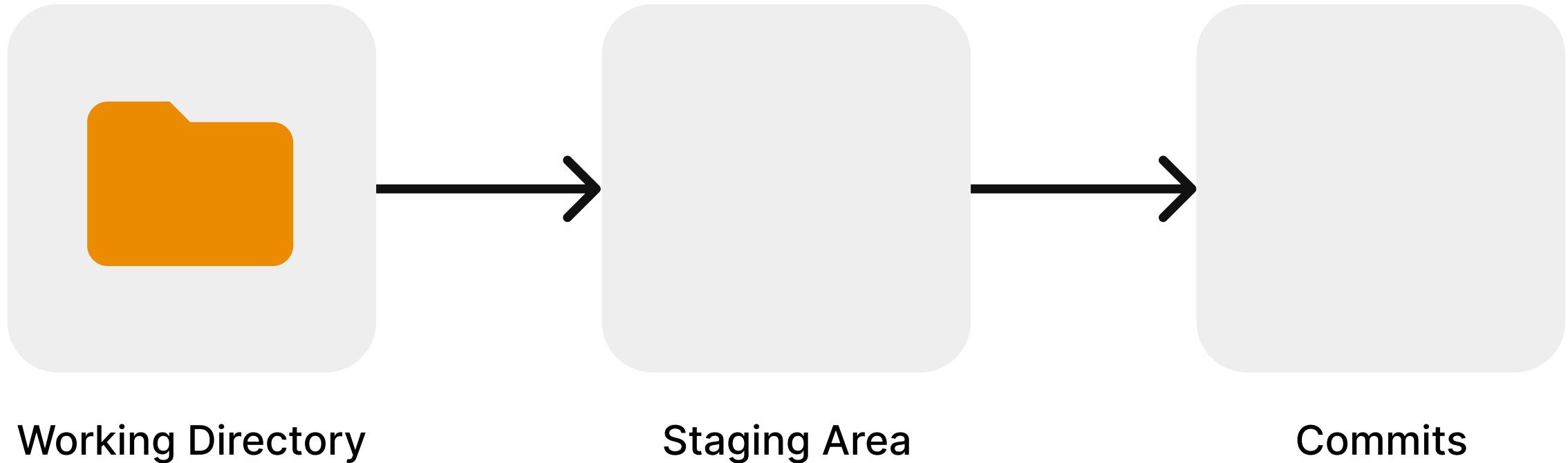
Hard Reset

- You committed something, but you want to undo the commit and remove the changes completely.

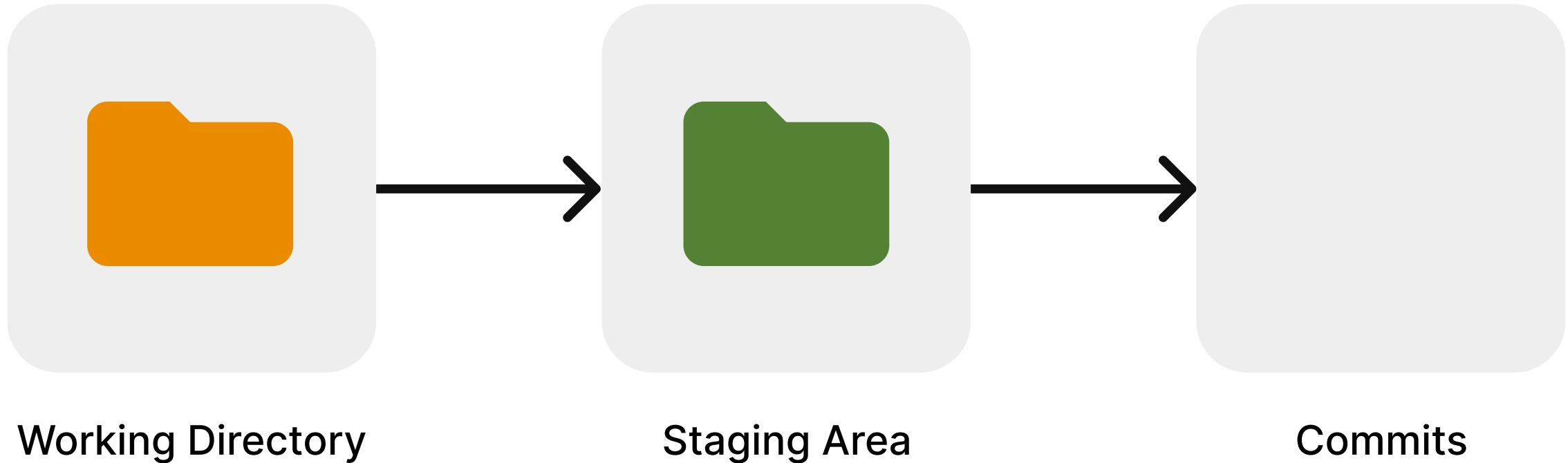
```
git reset --hard HEAD~n
```

```
git reset --hard <commit-hash>
```

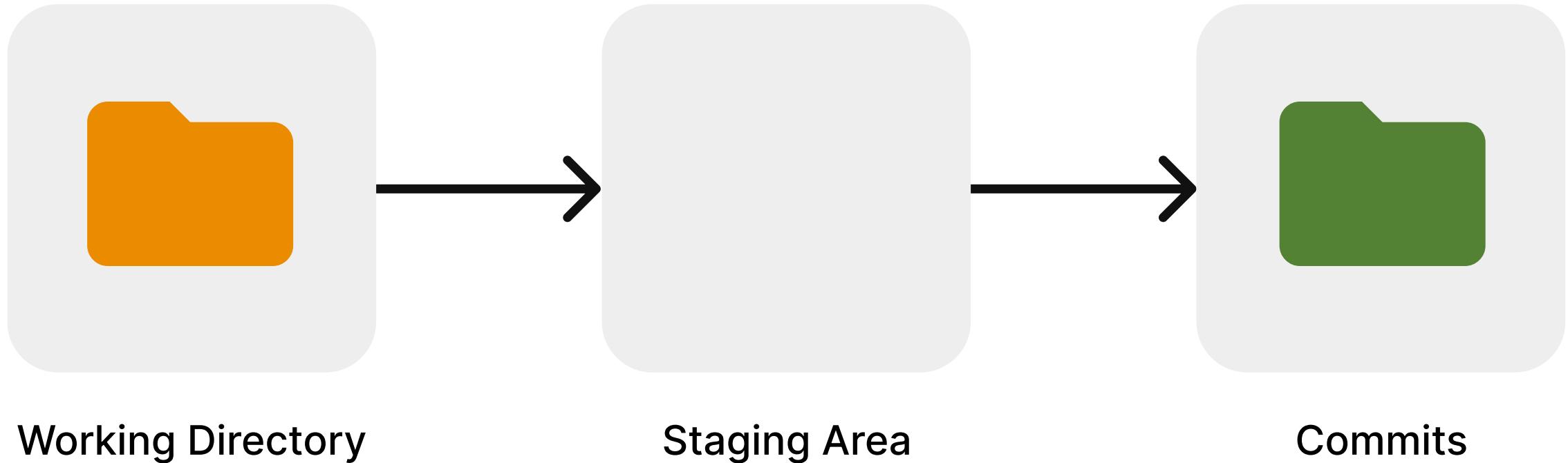
Example 1



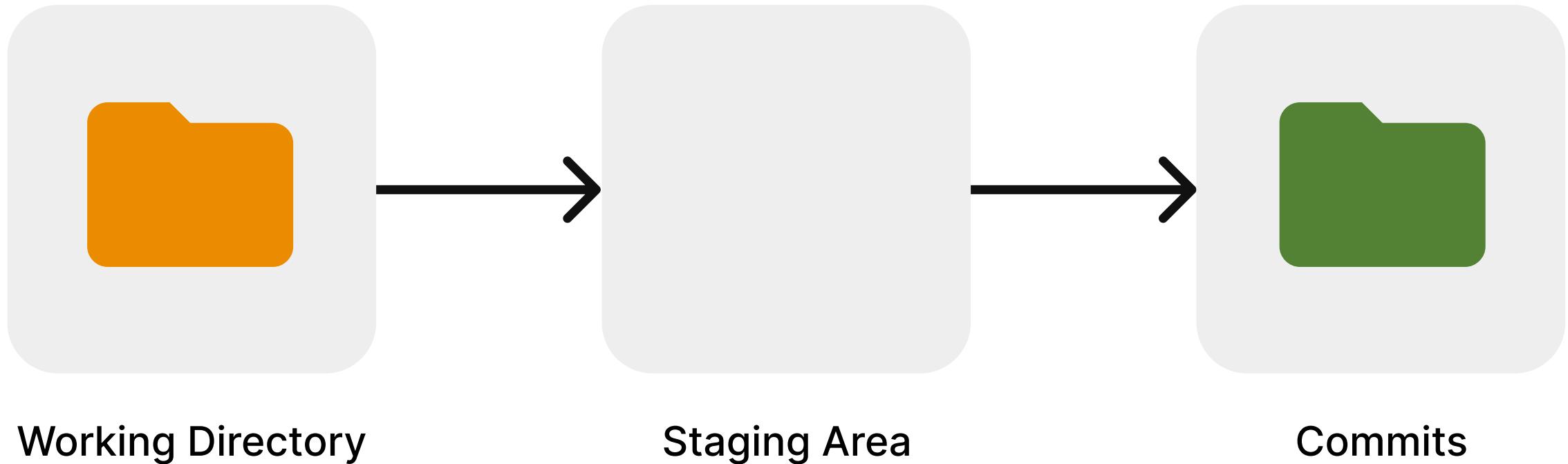
Example 1



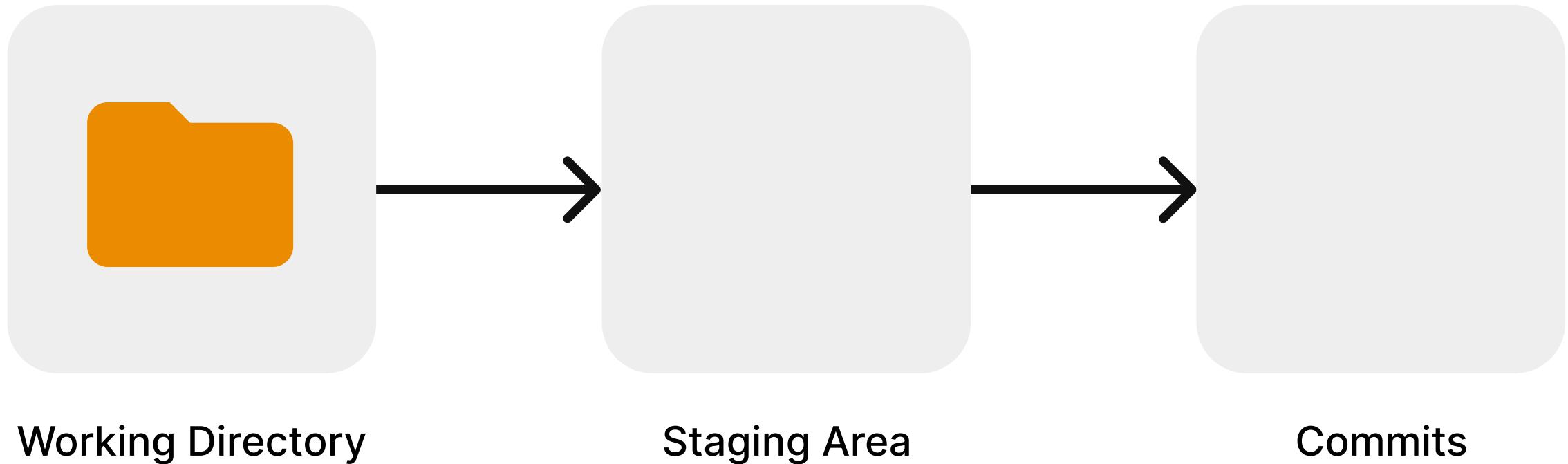
Example 1



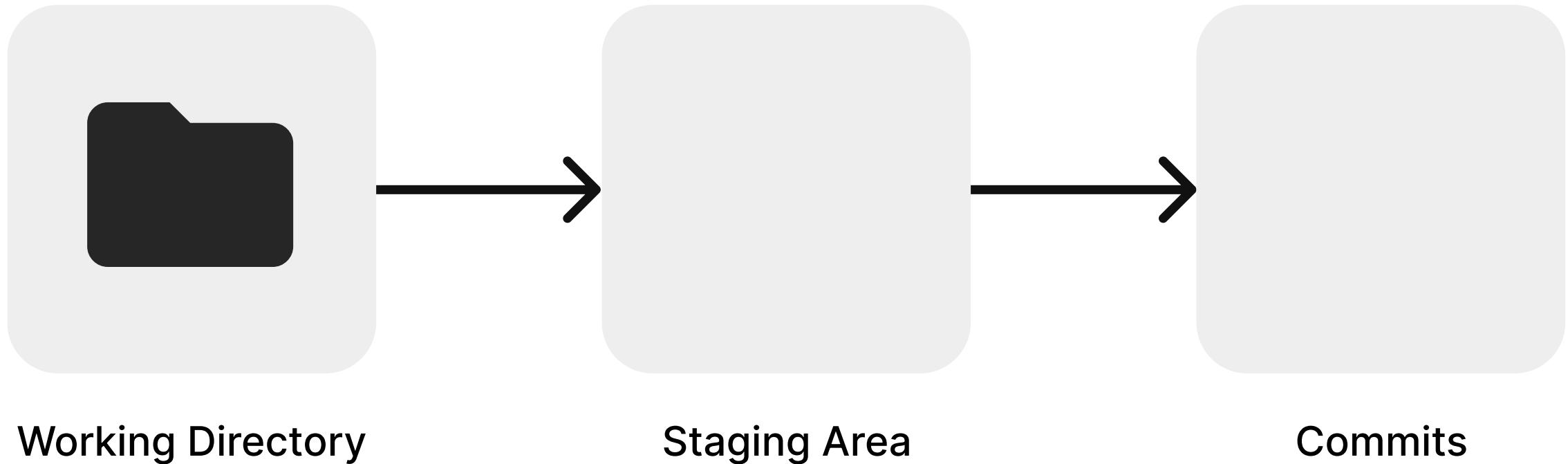
Hard Reset



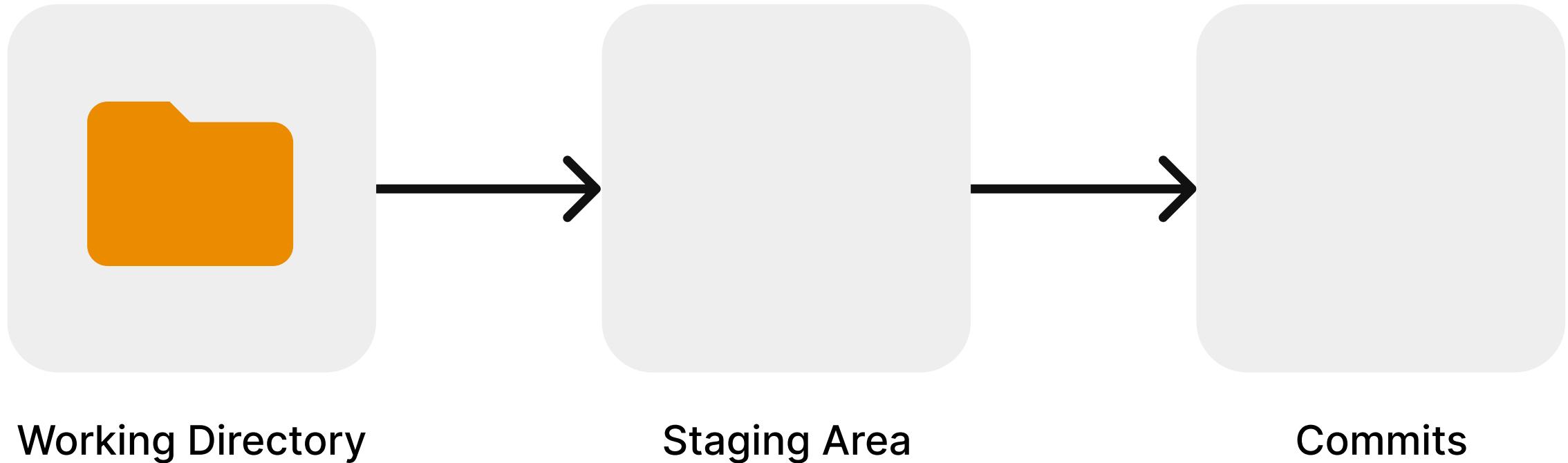
Hard Reset



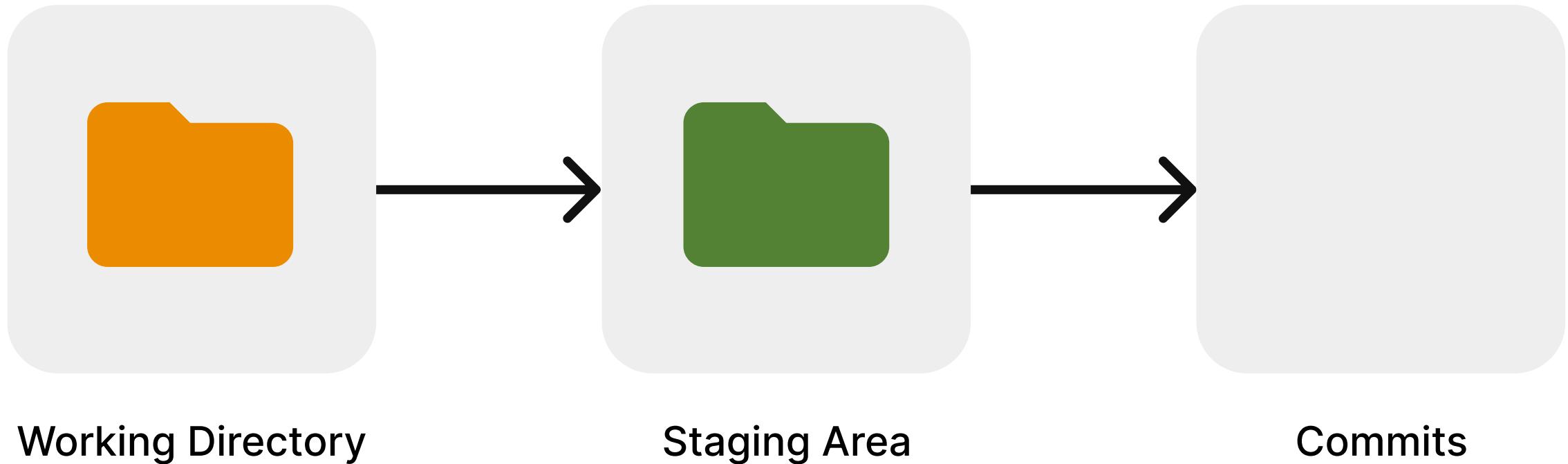
Hard Reset



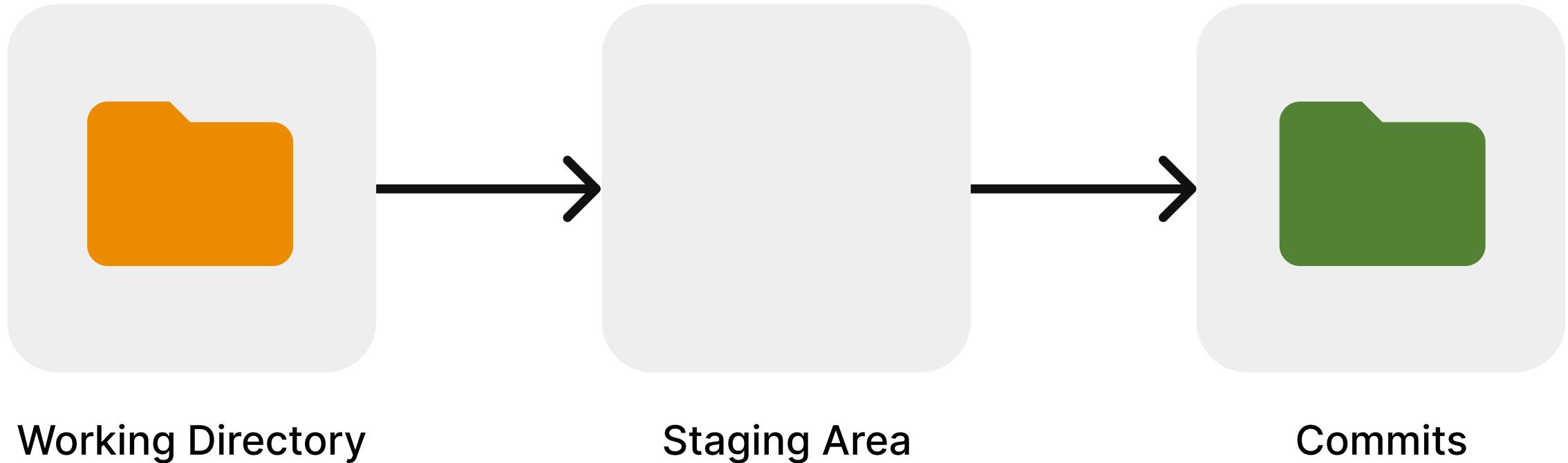
Example 2



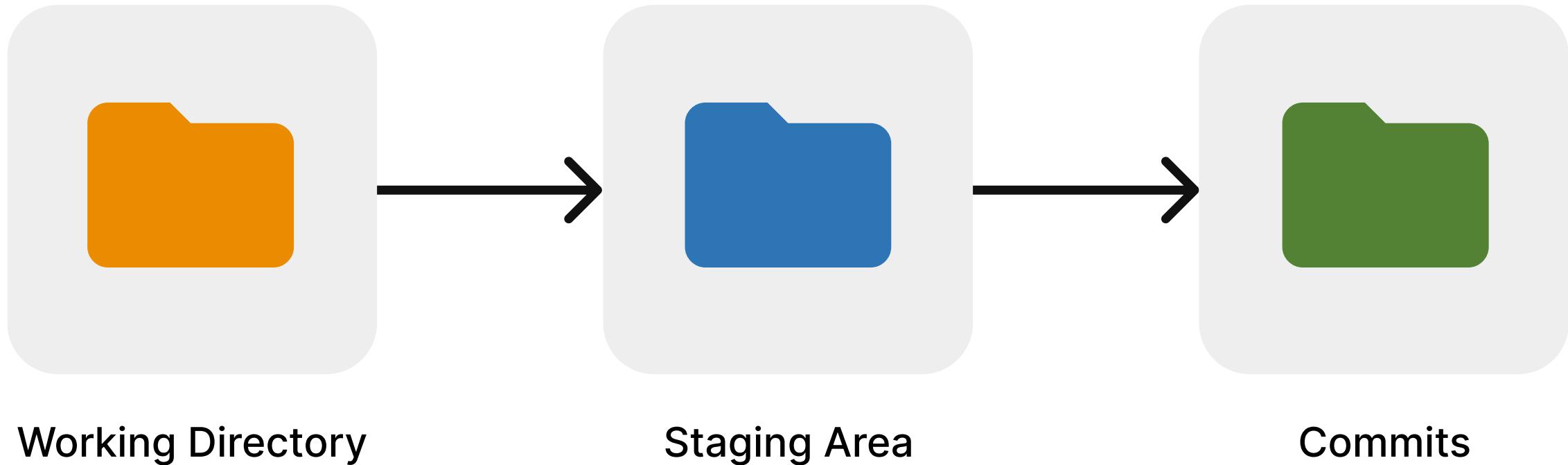
Example 2



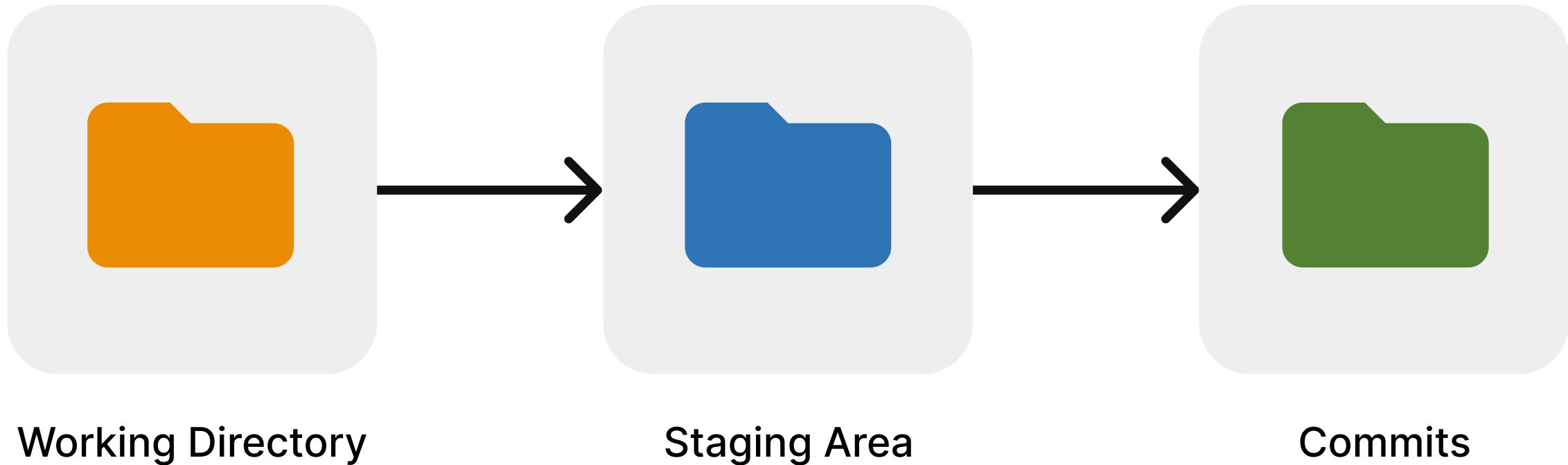
Example 2



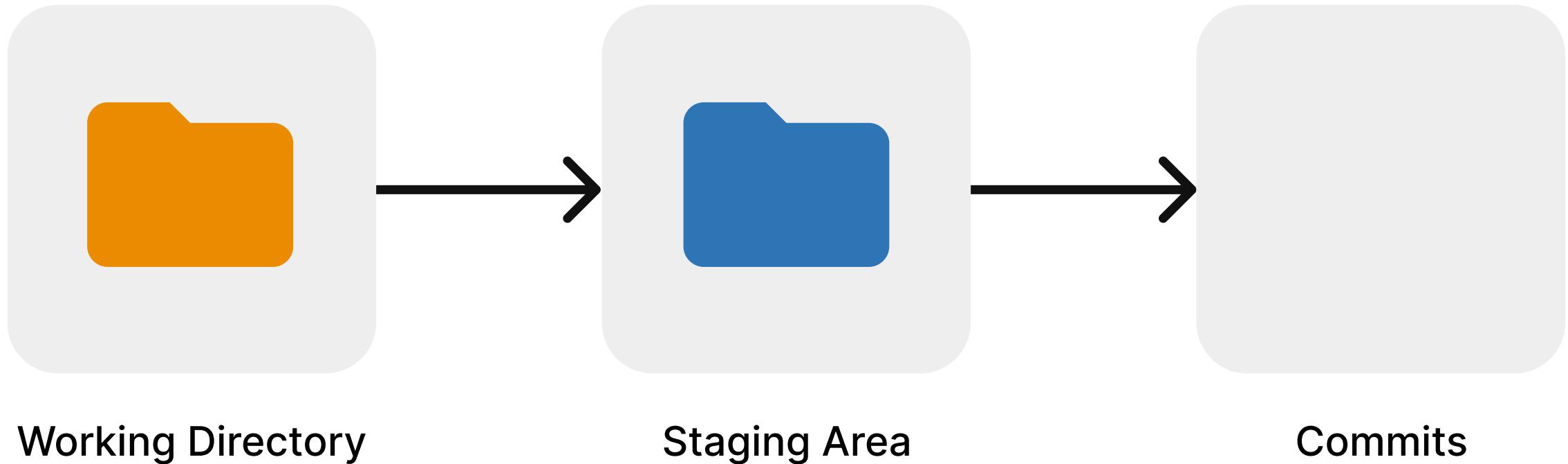
Example 2



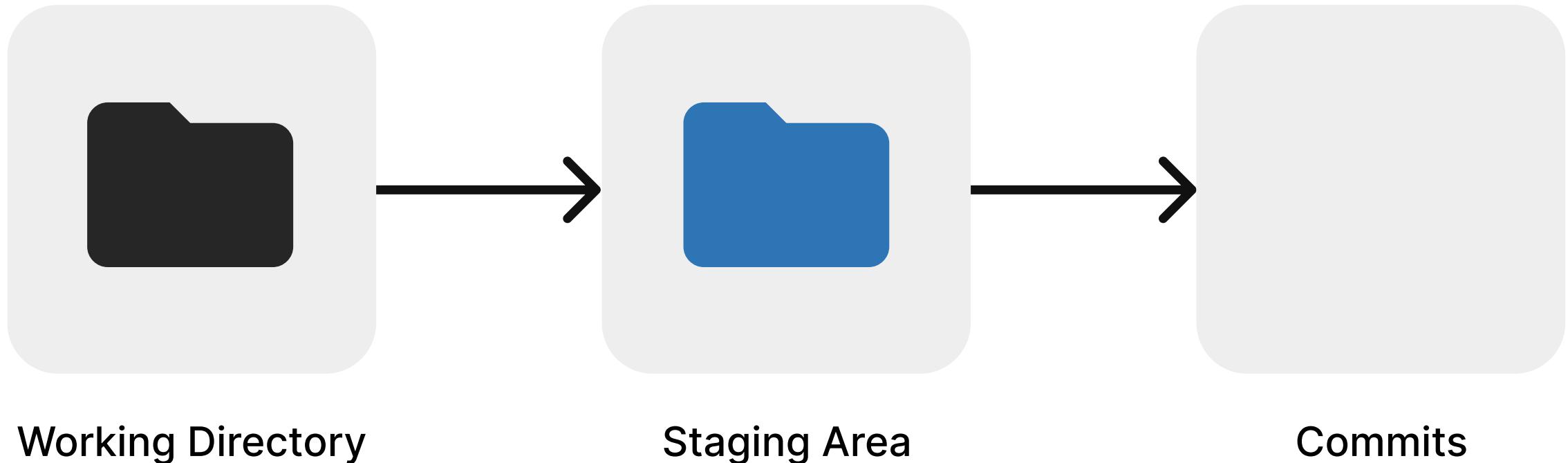
Hard Reset



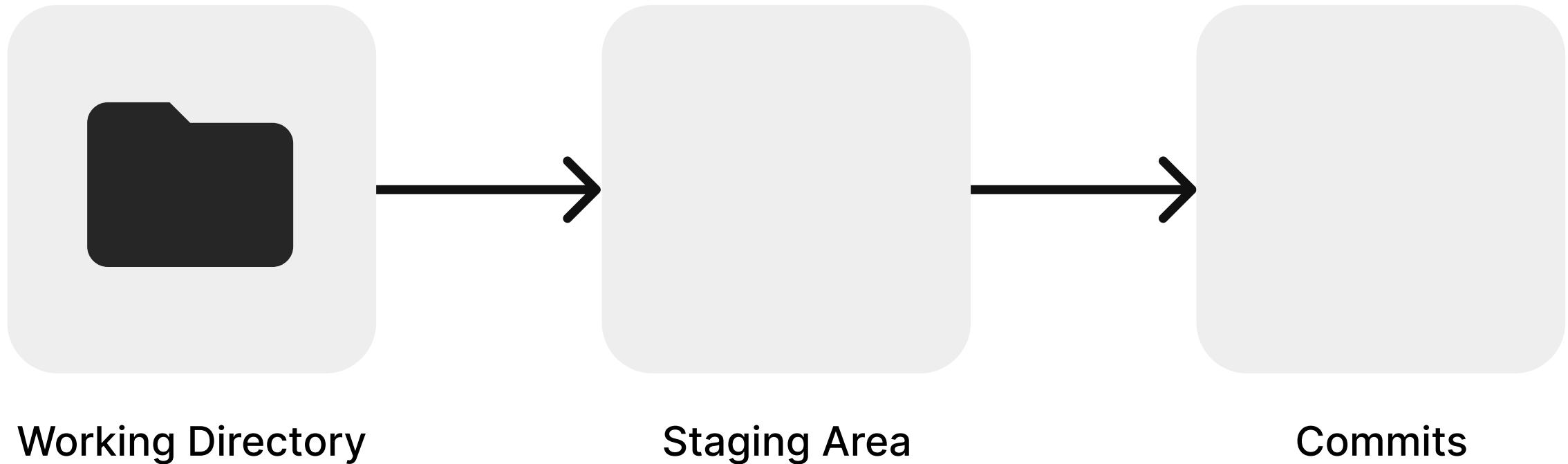
Hard Reset



Hard Reset



Hard Reset



Warmup 10 🌶

Warmup 10 ↴

- Create a new directory called **wup10** and navigate into it.
- Initialize a Git repository within the directory.
- Create a Python file: **dates.py**.
- Add and commit with the message “***initial commit***”.

Contd..

Warmup 10 ↴

- Write `print("Magh 24")` in `dates.py`, then add and commit file to Git with the message “**add yesterday**”.
- Overwrite `dates.py` with `print("Magh 25")`, then add and commit these changes with the message “**add today**”.
- Again, overwrite `dates.py` with `print("Magh 26")` in file, then add and commit these changes with the message “**add tomorrow**”.
- Again, overwrite `dates.py` with `print("Magh 27")` in file, then add and commit these changes with the message “**add day after tomorrow**”.
- View the commit history.

Contd..

Warmup 10 ↴

- Reset ***dates.py*** to today's date.
- View the commit history.

Revert

Revert

- Revert undoes a specific commit by creating a new commit without changing the commit history.

```
git revert <commit-hash>
```

Git Ignore

**What is the heaviest object
in the universe?**



node_modules

Git Ignore

- We can tell Git which files or folders to ignore, so they don't get tracked or included in commits.
- We can create a special Git ignore file for that.

`.gitignore`

Ignore a Specific File or Folder

filename

folder

folder/filename

Ignore All Files of a Type

* .tmp

* .log

Ignore All Except a Certain File

```
*  
! * .py  
config  
! config/settings.json
```

Warmup 11 🌶

Warmup 11 ⚡

- Create a new directory called **wup11** and navigate into it.
- Initialize a Git repository within the directory.
- Create a two files: **greet.py** and **secrets.txt**.
- Ignore the **secrets.txt**.
- Add and commit with the message “***initial commit***”.

Git Keep

Git Keep

- Normally, Git doesn't track empty directories because it only tracks files.
- We can create a special Git keep file for that.

.gitkeep

Warmup 12 ♪

Warmup 12 ↧

- Create a new directory called **wup12** and navigate into it.
- Initialize a Git repository within the directory.
- Create following files:
app.py
models
views
controllers
- Add, ***app.py*** and commit with the message “***initial commit***”.
- Check the status.
- Add ***models***, ***views*** and ***controllers*** folders and commit with the message “***add mvc***”.

Git Aliases

Git Aliases

- Git aliases are shortcuts for longer Git commands.
- Instead of typing out a long command every time, we can create a shorter, easier-to-remember version of it.

`.git config --global alias.name "command"`

Warmup 13 ↩

Warmup 13 ↗

- Create a new directory called **wup13** and navigate into it.
- Initialize a Git repository within the directory.
- Create a file: alias.txt
- Create the alias for **commit -m** as **cm**.
- Write “**alias in git**” to **alias.txt** file, add and commit to Git using the created alias with the message “**commit using alias**”.

Removing Git

Removing Git

- Similar to initiating Git on our project, we can also remove Git from our project folder but keep our files.

```
rm -rf .git
```

Warmup 14 🌶

Warmup 14 ↴

- Create a new directory called **wup14** and navigate into it.
- Initialize a Git repository within the directory.
- Create a file: **temp.txt**
- Add and commit to Git with the message “**add temp file**”.
- View the commit history.
- Remove the Git from current directory.



Module 2 Completed