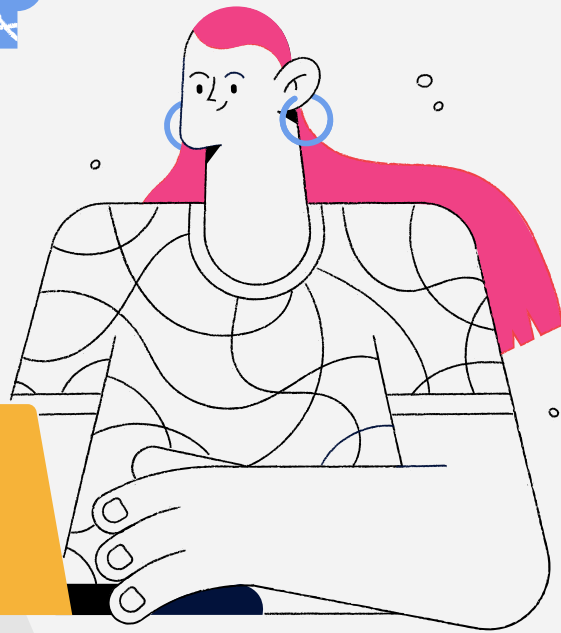
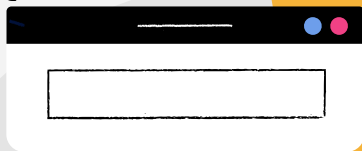


## 2.4 Git Basics: Undoing Things

# Amend

You can't always get what you want, but if you try...



# The old way...



End of a work session, going to commit...

```
$ git add my_file.md
```

```
$ git commit -m 'Finished work on slide presentation'
```

Thinks to self... "Shoot I forgot to save that file!" Grr\$#!@ ... saves file

```
$ git add my_file.md
```

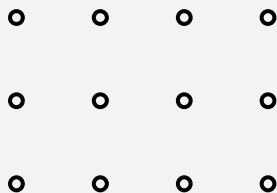
```
$ git commit -m 'Same commit, just forgot to save'
```

## ...cluttered history



×

# Cluttered History





# \$ git commit --amend

This replaces the last commit entirely.

Use git log to check the hash and verify.



# The new way...



End of a work session, going to commit...

```
$ git add my_file.md
```

```
$ git commit -m 'Finished work on slide presentation'
```

Thinks to self... "Shoot I forgot to save my files first!" Grr\$#!@ ... saves files

Then thinks... "But wait! **I need only amend my last commit**"

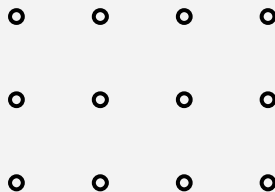
```
$ git add my_file.md
```


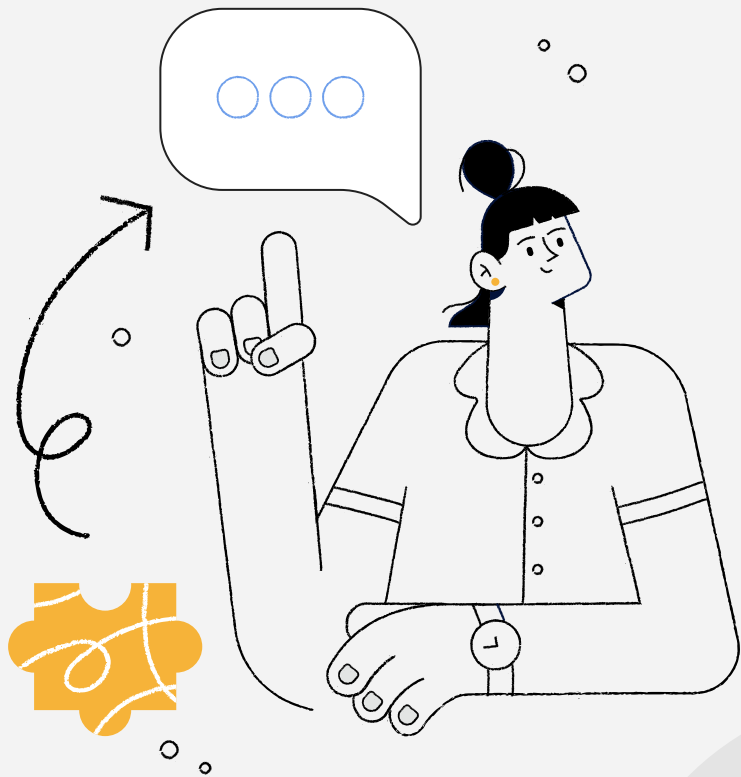
```
$ git commit --amend
```

## Clean history.

×

# Clean History





# Potential Dangers...

## Rebasing

When using a remote repository and working within a team environment, beware of using `--amend` after you have pushed your commit.

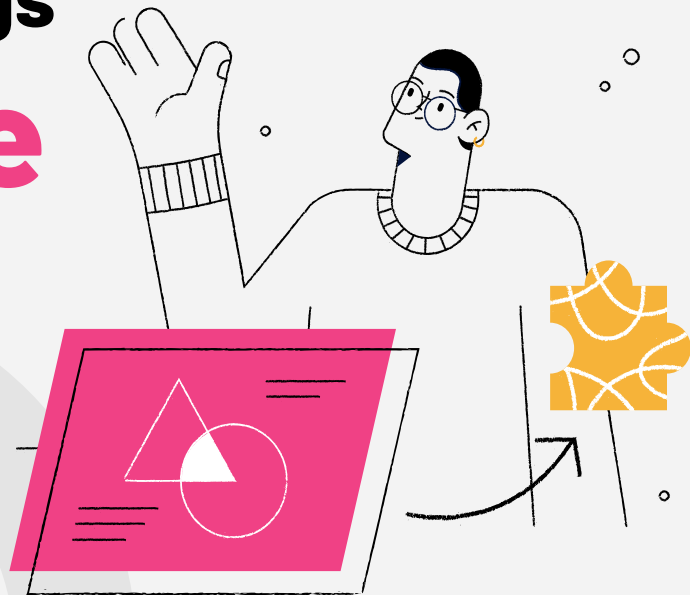




## 2.4 Git Basics: Undoing Things

# Unstaging a file

You can't always get what you want, but if you try...





**Problem:** Staged file...

**Solution:** Unstage the file...



**\$ git reset HEAD file\_to\_unstage.md**



**Hint:** git status



**git HEAD**

**git reset vs git restore**

