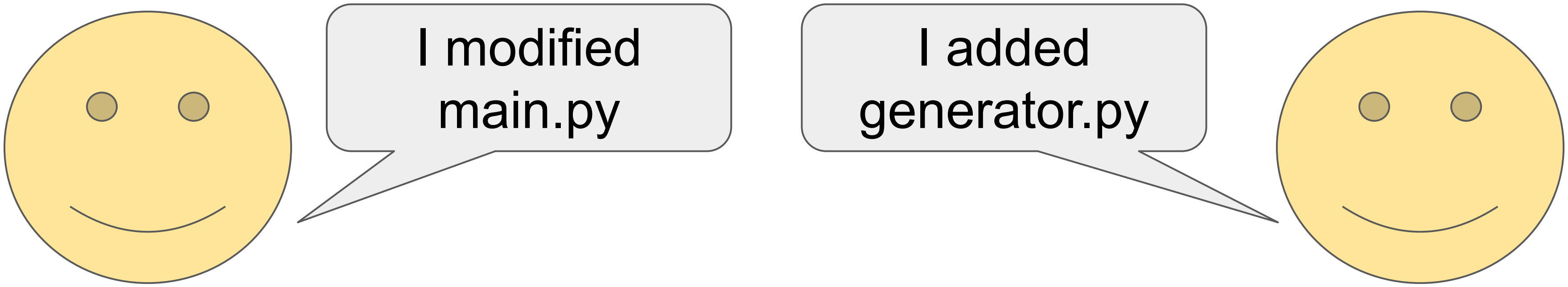
git - merge conflicts

Adapted from materials by Dr. Carrier

Bermix Studio - Unsplash

So far we’ve had no merge conflicts

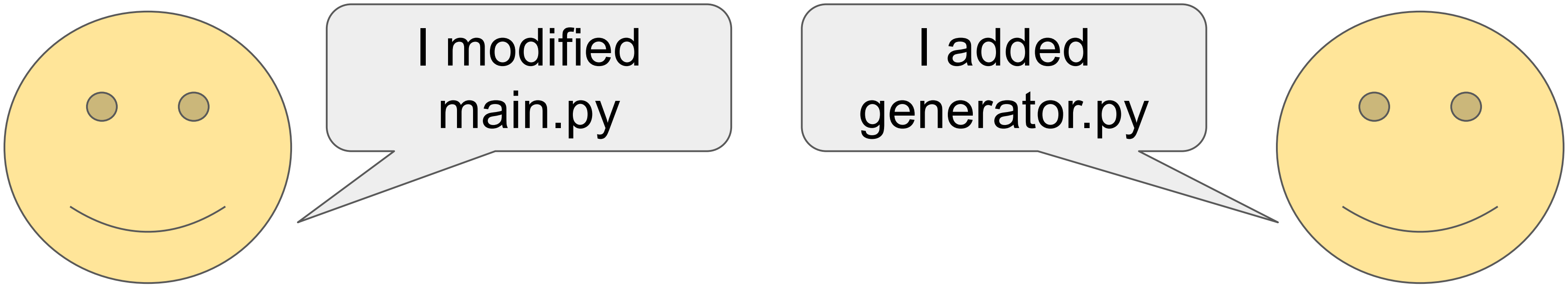
So far we’ve had no merge conflicts



I modified main.py

I added generator.py

So far we've had no merge conflicts

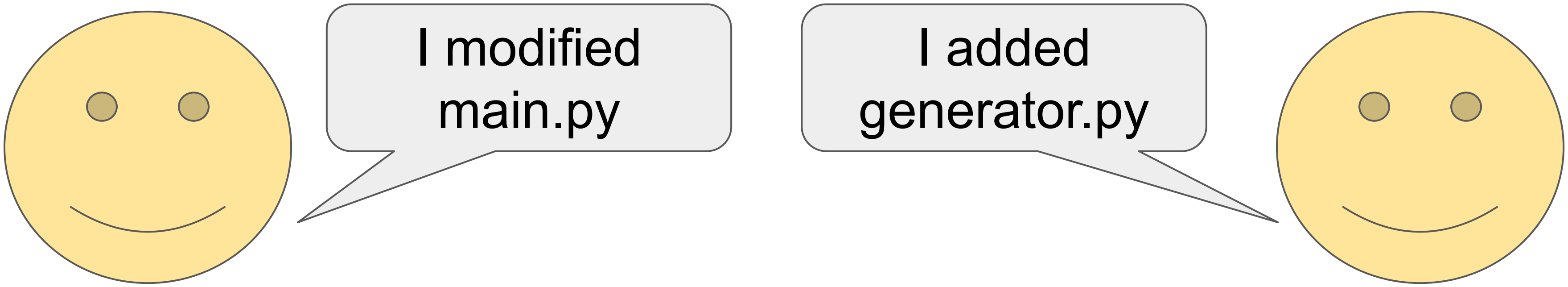


I modified main.py

I added generator.py

When might we have a merge conflict?

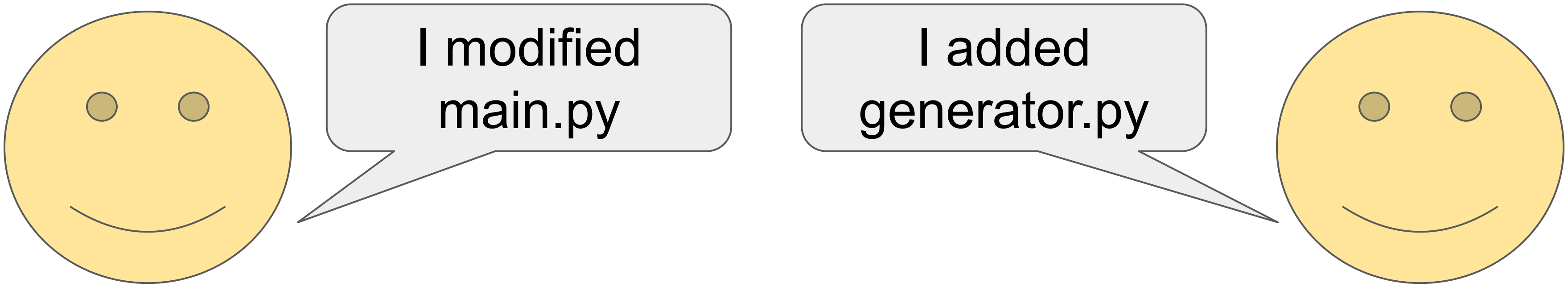
So far we've had no merge conflicts



When might we have a merge conflict?

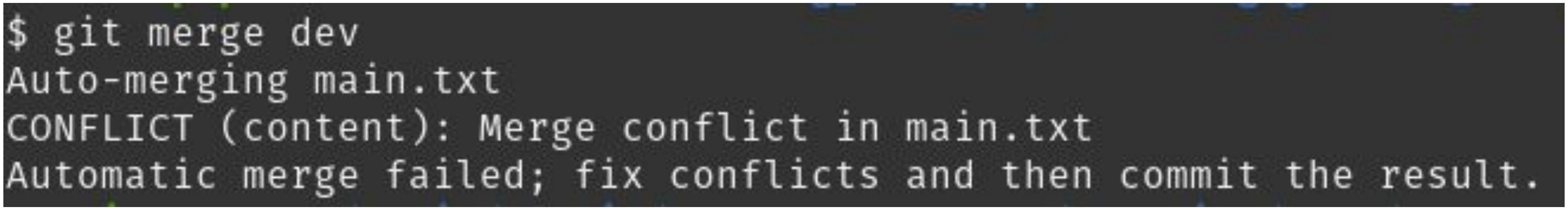
- People editing the same *line*
- File both edited and deleted

So far we’ve had no merge conflicts



When might we have a merge conflict?

- People editing the same *line*
- File both edited and deleted



```
$ git merge dev
Auto-merging main.txt
CONFLICT (content): Merge conflict in main.txt
Automatic merge failed; fix conflicts and then commit the result.
```

Resolving merge commits

We have to manually edit files with conflicts

Resolving merge commits

We have to manually edit files with conflicts

- We decide what to keep!

Resolving merge commits

We have to manually edit files with conflicts

- We decide what to keep!

Each conflict marked by markers:

Resolving merge commits

We have to manually edit files with conflicts

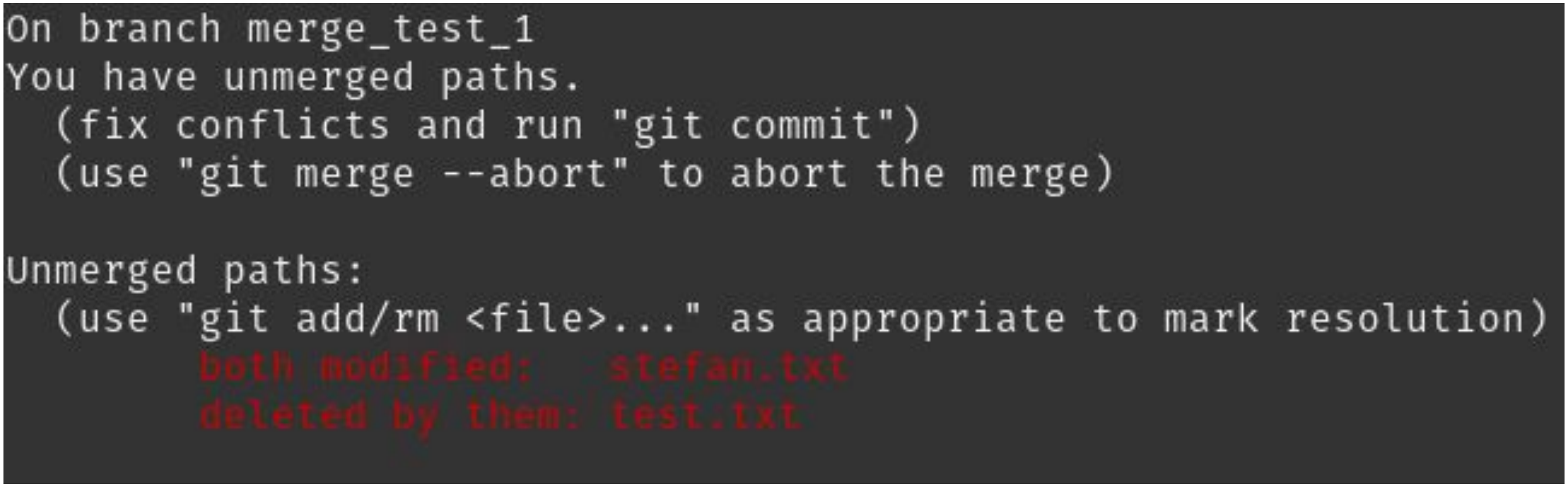
We decide what to keep!

Each conflict marked by markers:

```
# No conflict
<<<<<<<
This is from local branch (one we’re merging into)
=======
This is from remote (one being merged in)
>>>>>>>
# Conflict over
```

Resolving merge commits (part 1.5)

`git status` will show which files have conflicts



```
On branch merge_test_1
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Unmerged paths:
  (use "git add/rm <file>..." as appropriate to mark resolution)
        both modified:   stefan.txt
        deleted by them: test.txt
```

Resolving merge commits (part 2)

Resolving merge commits (part 2)

1. Decide what to keep

Resolving merge commits (part 2)

1. Decide what to keep
    a. Can be one version or a mixture

Resolving merge commits (part 2)

1. Decide what to keep
   1. Can be one version or a mixture
2. Edit the file to match

Resolving merge commits (part 2)

1. Decide what to keep
    a. Can be one version or a mixture
2. Edit the file to match
3. Remove markers

Resolving merge commits (part 2)

1. Decide what to keep
   a. Can be one version or a mixture
2. Edit the file to match
3. Remove markers
4. Repeat for all conflicts

Resolving merge commits (part 2)

1. Decide what to keep
   a. Can be one version or a mixture
2. Edit the file to match
3. Remove markers
4. Repeat for all conflicts
5. Stage and commit files

Resolving merge commits (part 3)

Shortcut for keeping one version:

Resolving merge commits (part 3)

Shortcut for keeping one version:

```
git restore --ours file
```

```
git restore --theirs file
```

Ours = branch being merged into, theirs is opposite

Resolving merge commits (part 3)

Shortcut for keeping one version:

```
git restore --ours file
```

```
git restore --theirs file
```

Ours = branch being merged into, theirs is opposite

What if we mess up?

Resolving merge commits (part 3)

Shortcut for keeping one version:

```
git restore --ours file
```

```
git restore --theirs file
```

Ours = branch being merged into, theirs is opposite

What if we mess up?

```
git merge --abort
```

Can delete changes in working directory!