Version Control with Git

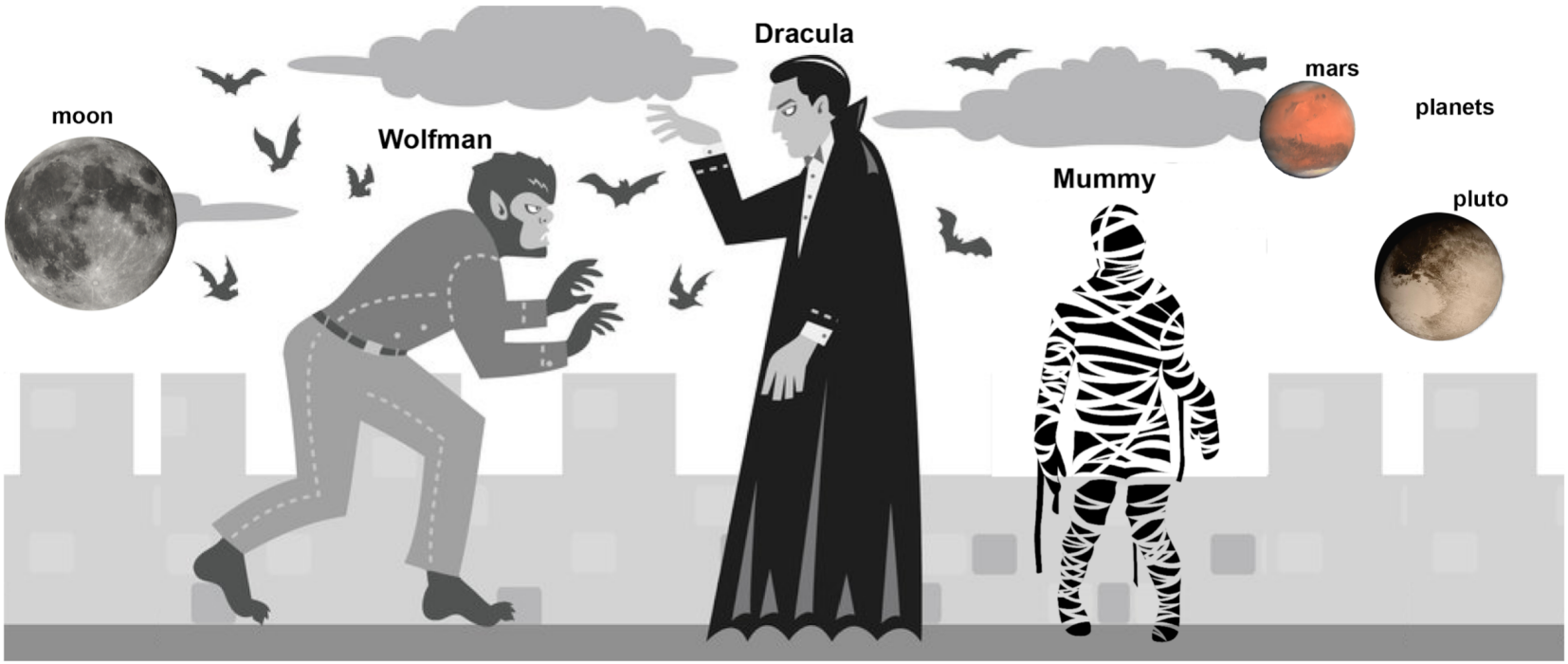
Koen Leuveld

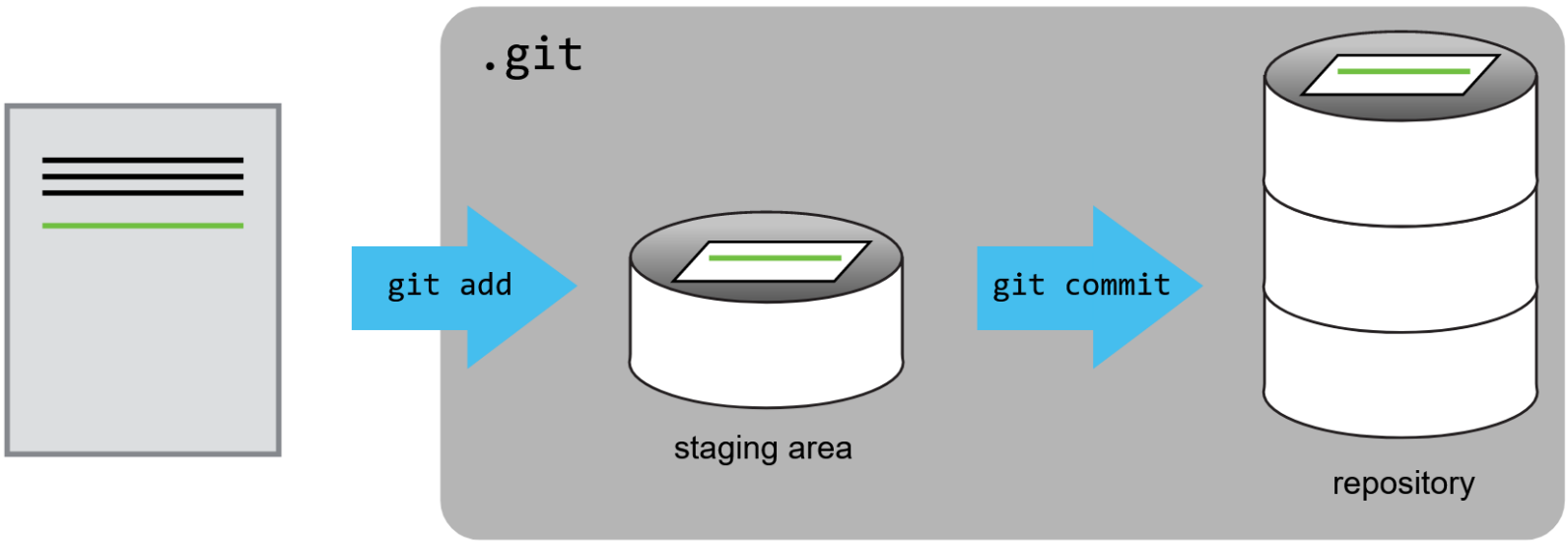
June 2, 2023

Contents

"FINAL".doc

Universal Missions




.git
git add
staging area
git commit
repository

Episode 5: Recovering Older Versions of a File

Jennifer has made changes to the Python script that she has been working on for weeks, and the modifications she made this morning "broke" the script and it no longer runs. She has spent ~ 1hr trying to fix it, with no luck...

Luckily, she has been keeping track of her project's versions using Git! Which commands below will let her recover the last committed version of her Python script called data_cruncher.py?

1. `$ git checkout HEAD`
2. `$ git checkout HEAD data_cruncher.py`
3. `$ git checkout HEAD~1 data_cruncher.py`
4. `$ git checkout <unique ID of last commit> data_cruncher.py`

Episode 5: Reverting a Commit

Jennifer wants to undo a change she made to a public repository. The command git revert [erroneous commit ID] will create a new commit that reverses the erroneous commit.

The command git revert is different from git checkout [commit ID] because git checkout returns the files not yet committed within the local repository to a previous state, whereas git revert reverses changes committed to the local and project repositories.

Below are the right steps and explanations for Jennifer to use git revert, what is the missing command?

1. `_____ # Look at the git history of the project to find the commit ID`
2. Copy the ID (the first few characters of the ID, e.g. 0b1d055).
3. `git revert [commit ID]`
4. Type in the new commit message.
5. Save and close

Episode 5: Understanding Workflow and History

What is the output of the last command in:

```
$ cd planets

$ echo "Venus is beautiful and full of love" >
venus.txt

$ git add venus.txt

$ echo "Venus is too hot to be suitable as a base" >>
venus.txt

$ git commit -m "Comment on Venus as an unsuitable
base"

$ git checkout HEAD venus.txt

$ cat venus.txt #this will print the contents of
venus.txt to the screen
```