ForGoodCode

GIT EXPLAINED TO A 6 YEAR-OLD

Simplest way possible

Imagine you get some paper and crayons for your birthday.

Abi Bouhmaïda
@forgoodcode

You use them to draw yourself a nice beach scene. Then you go have a nice sandwich for lunch.


Abi Bouhmaïda
@forgoodcode

When you come back, you start to draw some seagulls- but, shit, the food coma hits and they look terrible!

Abi Bouhmaïda
@forgoodcode

So, you crumple up the paper in anger, throw it on the floor for mommy to clean up, and you start over from scratch, **re-drawing** the beach scene, then trying the seagulls once more, after you finish crying, of course.

Abi Bouhmaïda
@forgoodcode

Wouldn't it be nice if, instead, mommy could reach into the closet and pull out an exact copy of the beach scene before you went to lunch?

That way, you can just get right back to the seagulls and not have to re-do all the work before that point.

Abi Bouhmaïda
@forgoodcode

That's essentially Git

(except that YOU can go into the closet yourself and mommy can continue to lay on the couch enjoying her coffee- that's how everyone's childhood was, right?!)


Abi Bouhmaïda
@forgoodcode

With Git though, it's code.
You create a file.
You edit it a bit.
Then, you "**commit**" it to Git.

Abi Bouhmaïda
@forgoodcode

Without getting into the inner workings, there is essentially, from your perspective anyway, a copy of the file exactly as it was at the time you commit it.

Make some more changes, commit it again, now there's two copies (two "**versions**") that look like the file at the time of each commit.

Abi Bouhmaïda
@forgoodcode

These copies are the file's "**history**". You can compare two different versions to see what changed, and you can "**revert**" your own copy to a previous version, just like our bratty young artist.