# Welcome to Git-Flow Super-Skill

- We will go through the commands that a team use to come out with an updated and correct code.
- In this super-skill there is no assessments and no checkpoint, but after each Skill, you will have to open terminal and practise some code to be familiar with the commands.
- This Super-Skill is very important when you will work in a team.
- Enjoy typing the commands and take your time .

## Using SSH over HTTPS

When you push or pull code from GitHub, it's essential that GitHub can successfully authenticate you before letting you run these commands. This means that every time you try to `git push` or `git pull`, GitHub will ask you for your email and password, and if they are correct the command will execute. Without authentication, GitHub would be chaos: anyone could push whatever code they wanted to any repository at any time!

So, being able to authenticate people when they push or pull is critical. But it also gets tedious pretty quickly. When you are pushing and pulling frequently, typing your credentials each time is bi of a hassle. Thankfully, if you use the SSH protocol and configure everything properly, you will be able to authenticate without having to put your username and password each time. To do this we need to create an SSH key locally, add it to GitHub, and make sure we can authenticate successfully.

## Creating an SSH Key

This can be a bit intimidating, but if you follow the steps carefully, you should be able to get this to work. (If you get stuck, GitHub also provides instructions on configuring SSH.)

1. Open up Terminal.
2. Anywhere in Terminal paste the following `ssh-keygen -t rsa -b 4096 -C "PUT YOUR EMAIL HERE"` and replace "PUT YOUR EMAIL HERE" with the email you used to sign up with GitHub.
3. Once you are prompted with `Enter a file in which to save the key (/Users/you/.ssh/id_rsa): [Press enter]` press enter.
4. You will then be prompted to enter a passphrase. **Just press enter here**
5. You will then be prompted to enter a passphrase again. **Just press enter here as well**
6. Paste the following in terminal: `eval "$(ssh-agent -s)"`. If you do not see a `pid` number, start from the first step again.
7. Paste the following in terminal: `ssh-add ~/.ssh/id_rsa`. if you see an error message, start from the first step again.
8. Paste the following in terminal `pbcopy < ~/.ssh/id_rsa.pub`.
9. Head over to your GitHub account (make sure you sign in).
10. In the top right corner of any page, click your profile photo, then click Settings.
11. In the user settings sidebar, click SSH and GPG keys.
12. Click New SSH key or Add SSH key.
13. In the "Title" field, add a descriptive label for the new key. For example, if you're using a personal Mac, you might call this key "Personal MacBook Air".
14. Paste your key into the "Key" field. (you can just right click and click paste or use a keyboard shortcut. The previous command `pbcopy` did the copying for you).
15. Click Add SSH key.
16. If prompted, confirm your GitHub password.
17. Anywhere in Terminal, type `ssh -T git@github.com` and if you see "Successfully authenticated" (ignore the rest of the message) you are good to go! If you do not see that, start from the beginning again.

This might seem like a pain, but thankfully you only need to do it once. Once the connection is established, you'll be able to push and pull without having to authenticate every single time.

‹ Previous

next ›