**There are 3 levels of git configuration:**

* --system: for your whole machine, and any user in this computer.
* --global: your user settings for any project.
* --local: are the settings for just this specific project here.

**Edit global configuration:**

* git config --global –edit
* git config --global core.editor code (allows to configure the editor to edit: visual code in this case)

**In Git config**: (.gitconfig is a file that can be handy if you want to customize your git commands)

[core]

    editor = code --wait

--wait allows for git to load all the information in this document, before we opened the eye viewer.

**Git special commands:**

* git log --pretty=format:'%cn' (blame commit)
* git log --pretty=format:'%cr' (tell me at what time a commit was done)
* git log --pretty=format:'%H' (shows full git id)
* git log --pretty=format:'%h' (shows reduced git id)
* git log --pretty=format:"%h%x09%an%x09%ad%x09%s" (%h = abbreviated commit hash, %x09 = tab, %an = author name, %ad = author date, %s = subject)
* git log --pretty=format:"%C(blue)%h%x09%an%x09%ad%x09%s" (%C(blue) colors the output)
* git log --pretty=format:"%C(blue)%h%x09%C(yellow)%an%x09%C(green)%ad%x09%C(white)%s" (nicely color formatted)
* git commit --amend --no-edit (amend the last commit and merges this changes to that commit, so that you don’t create a new one)

**conventional commits**

Commits need to follow a certain pattern which is conventional and understandable by humans and machines. To force this format you can use: <https://www.npmjs.com/package/git-commit-msg-linter>

**Javascript syntax patters:** <https://standardjs.com/rules.html>

**Husky**: allows you to correct certain syntax anti patterns using git hooks: https://www.npmjs.com/package/husky