Summarizer 0110

Palindromes

Version Control Systems

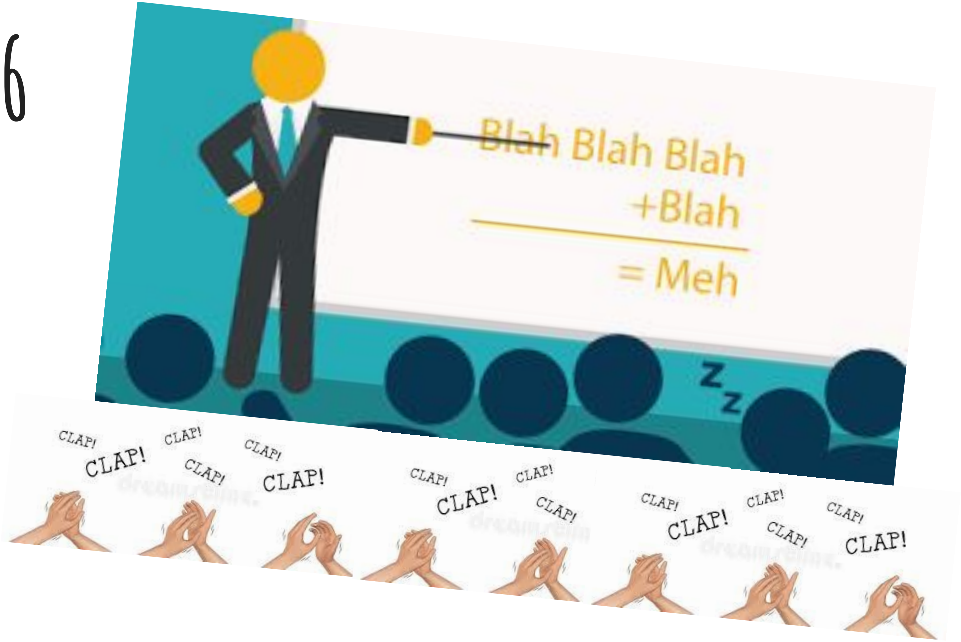
Git

ACADEMIA DE CÓDIGO

<andreneri@bootcamp16\Fundão\09-05-2018_>

Previously on BootCamp #16

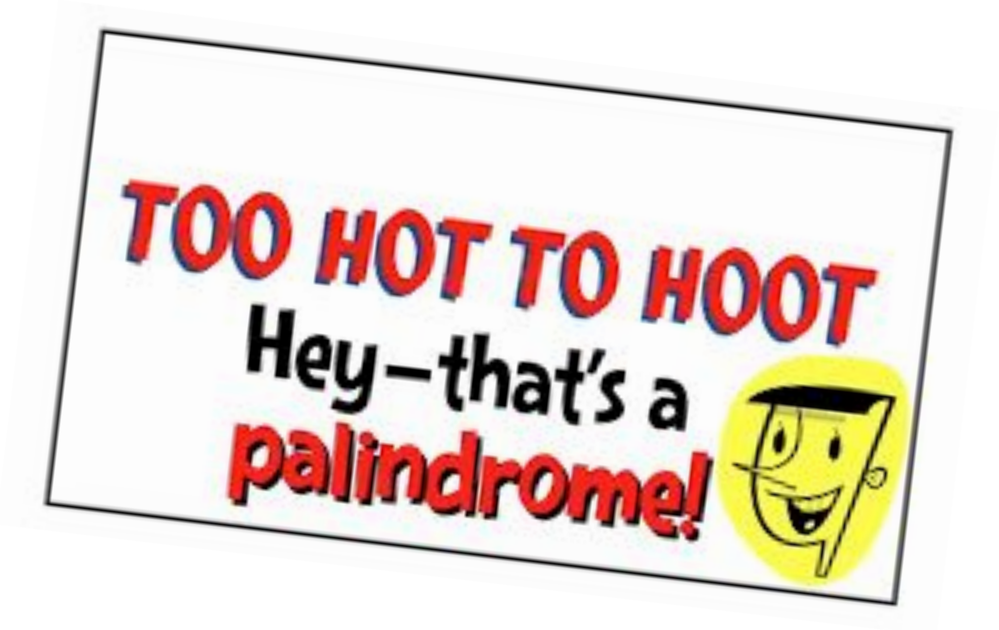
- We heard Pedro’s presentation about Linus Torvalds (great guy btw)
- Maria’s presentation about Margaret Hamilton (wrote 1,60m+ of code in assembly; like… whaaaaaat?!)
- André Silva’s summarizer on data structures and types of algorithms



Followed by the typical coffee break

PALINDROME

WHAT IS IT?



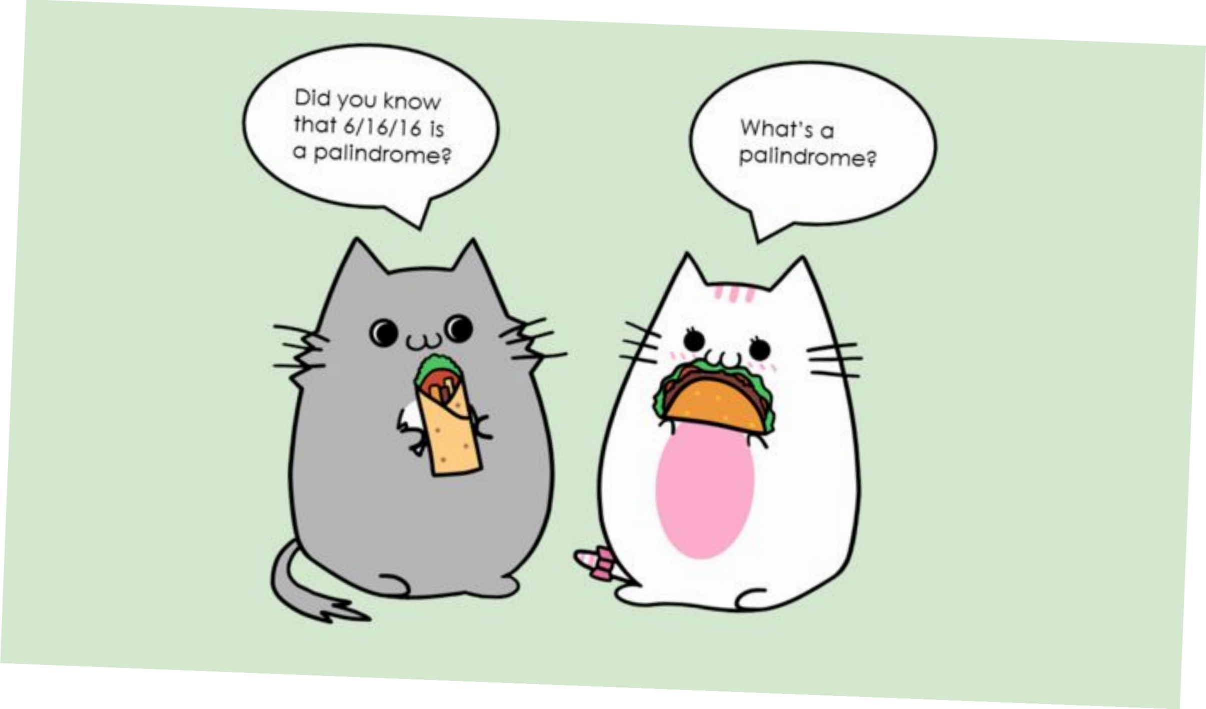
A word, phrase or sentence that reads the same either forward or backwards

pretty straight forward, right?

How would we create a program in order to test if a certain word or expression is a palindrome?

In other Words

How do we explain to the machine what the hell is a palindrome??



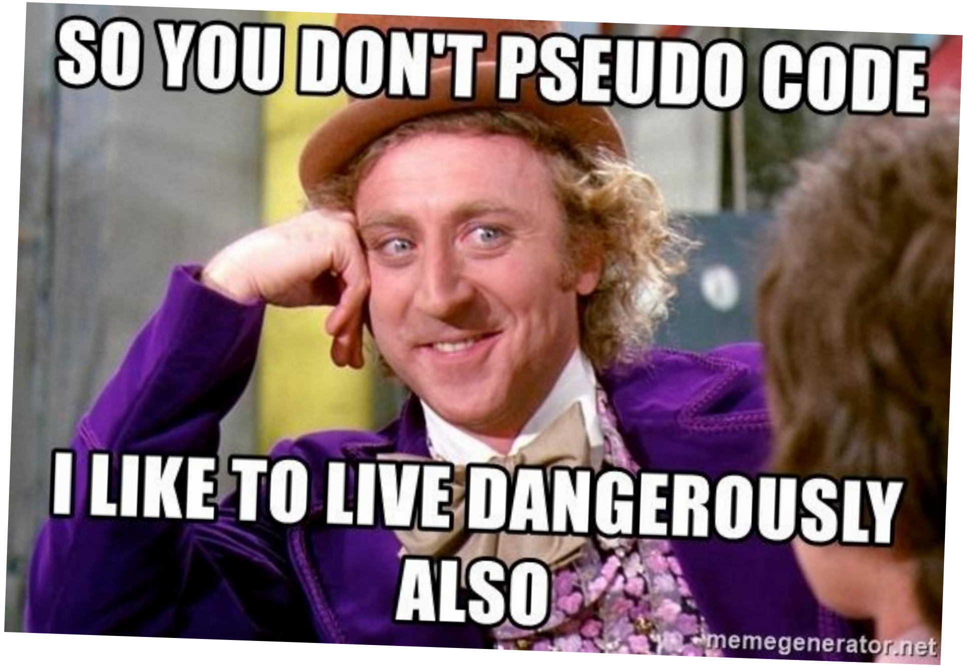
The main goal

- Understanding that machines, although powerful, are stupid
- They need a series of logic steps to understand what to do, much like a child, although a child is much smarter, but you get the idea…
- We need to break a BIG problem into smaller achievable goals

It doesn’t matter which language you’re writing your code with, the most important thing is logic. If you understand how to solve a specific problem, you can code it in any language

We need...

To create pseudocode in order to better understand the steps we need to take to implement our code. It's basically a sketch of what our code will look like in human language.



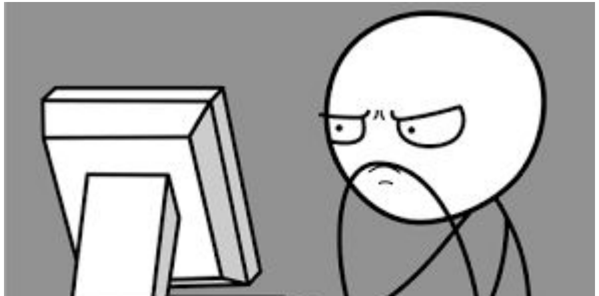
Just a tip

```
var remember = “Think first, code later”;

// it’s so much easier
```

THE PSEUDO CODE

```
step 1: define 2 variables → currentInitialPosition = 0 (cIP) && currentFinalPosition = word length - 1 (cFP)
step 2: compare characters at cIP && cFP
step 3: if char cIP !== char cFP → STOP because it’s not a palindrome
step 4: if char cIP == char cFP → cIP++ && cFP--
step 5: repeat step 2 while cIP < cFP
```



and time to dance

VCS (Version Control System)

What is it ?

Software that manages changes on documents.

No VCS

Centralized Version Control

Based on the idea that there is a single "central" copy of your project somewhere (on a central repository), and programmers will *"commit"* their changes to this central copy.

Needs internet connection

commit: creating a new version of that project

Explanation on how it works

Pull down any changes other people have made from the central server (update)

Make your changes, and make sure they work properly

Commit your changes to the central server, so other programmers can see them

Distributed Version Control

-These systems do not necessarily rely on a central server to store all the versions of a project's files.

-Instead, every developer "clones" a copy of a repository and has the **full** history of the project on their own hard drive. This copy (or "clone") has *all* of the metadata of the original.

pull: getting new changes from a repository

push: moving your own changes to a repository

Explanation on how it works

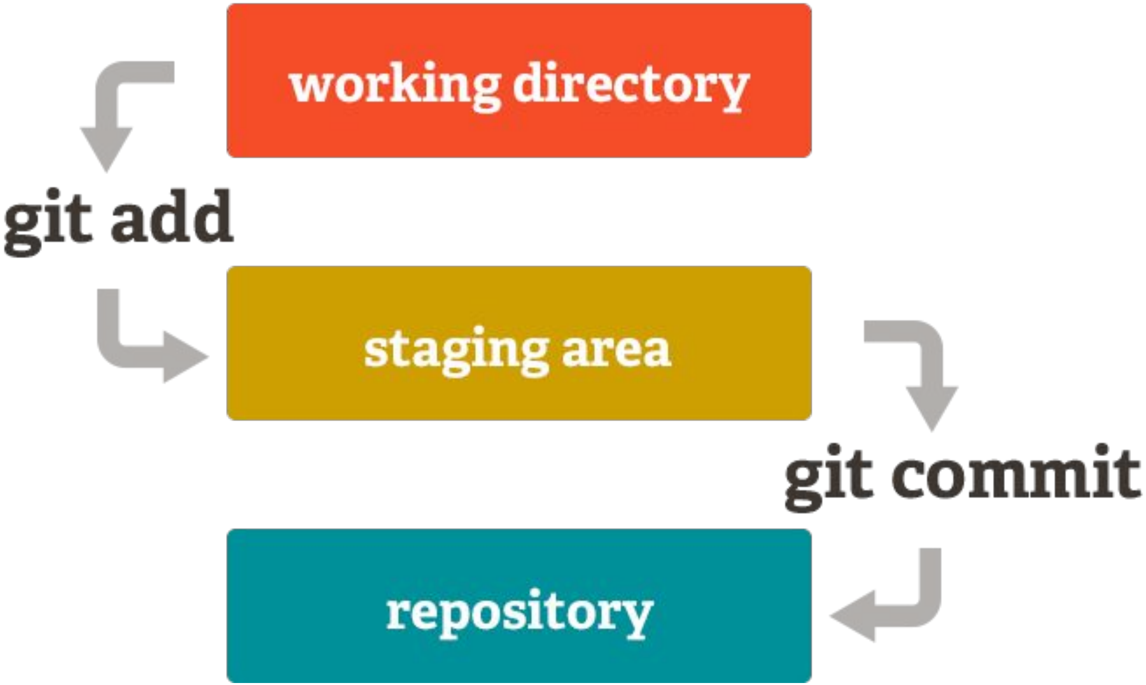
One common misconception about distributed version control systems is that there *cannot* be a central project repository. This is simply not true.

Instead of a central repository being *required* by the tools you use, it is now optional and purely a social issue.

GIT

Well, that’s legit…
// ifyouknowwhatimean

How it works



working directory

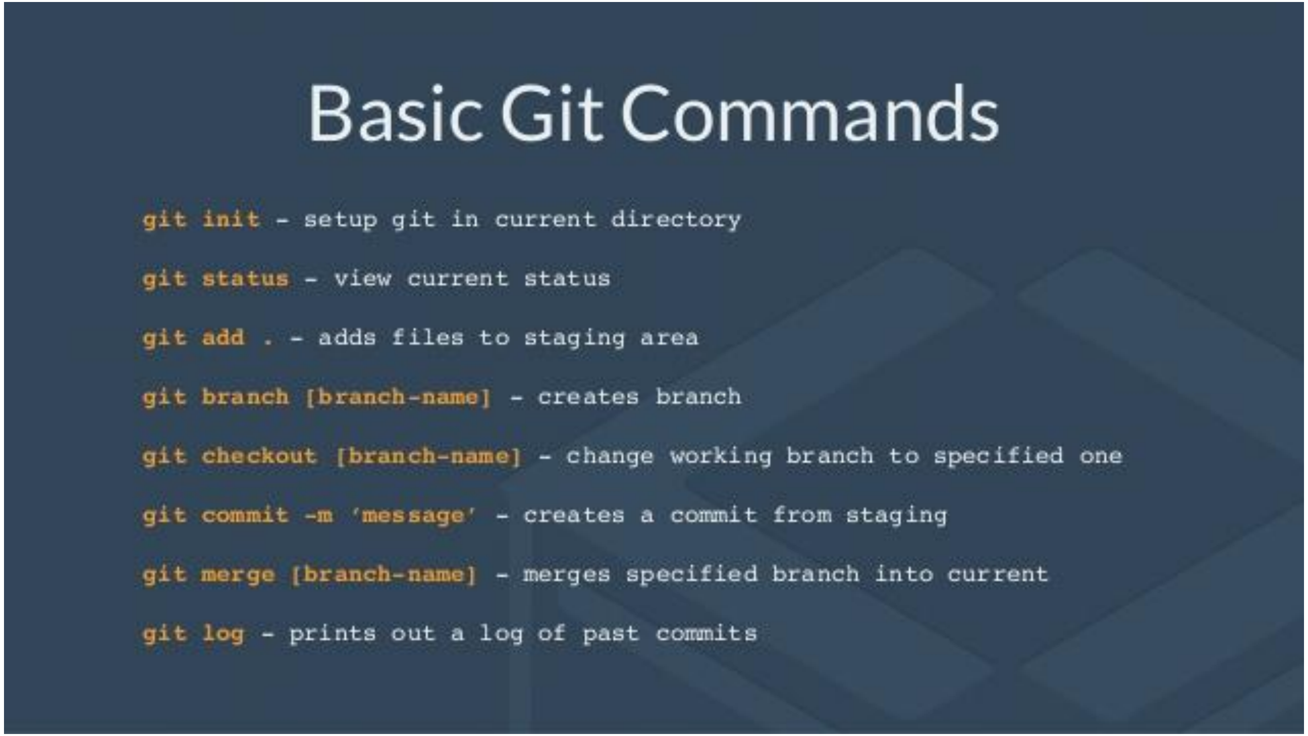
git add

staging area

git commit

repository

Some Useful git commands



Basic Git Commands

```
git init - setup git in current directory

git status - view current status

git add . - adds files to staging area

git branch [branch-name] - creates branch

git checkout [branch-name] - change working branch to specified one

git commit -m 'message' - creates a commit from staging

git merge [branch-name] - merges specified branch into current

git log - prints out a log of past commits
```

The End