

Git & GitHub intro QUIZ for 2nd
day Brazilian Portuges version

1. O que é verdade sobre Git / GitHub?

- a. GitHub é um software para controle de versões
- b. Git é um website que guarda os backups de projetos-git
- c. GitHub é um website para backups and collaborations
- d. Git e GitHub são a mesma coisa

2. Para iniciar um novo projeto-git eu uso:

- a. `git add 'nome_projeto'`
- b. `git init`
- c. `git commit -m 'descrição do projeto'`
- d. `git remote add`

3. Para salvar uma versão do meu projeto no git eu faço:

- a. `git add` depois `git commit -m "msg"` depois `git push`
- b. `git add` depois `git commit -m "msg"`
- c. `git commit` depois `git push`
- d. `git save --version`

4. O que eu devo considerar quando for escrever a mensagem do `commit`?

- a. Eu escrevo uma referência para explicar para mim mesmo o que foi modificado.
- b. Eu escrevo algumas considerações sobre porquê e como as mudanças foram feitas.
- c. Eu escrevo sobre possíveis efeitos e limitações das mudanças realizadas.
- d. Eu escrevo sobre porque, como, efeitos e limitações das alterações feitas.

5. Qual das opções representa a melhor mensagem para um `commit`?

- a. .
- b. Atualizações
- c. Atualização do arquivo zzz.py na segunda-feira
- d. Reescrevi a sentença com significado ambíguo na Introdução

6. Se eu iniciar um novo repositório-Git em um subdiretório de um repositório pre-existente, o que eu devo fazer?

- a. Deletar ambos repositórios, é a única forma.
- b. Deletar a nova pasta completamente.
- c. Deletar o arquivo `'.git'` do novo repositório.
- d. Simplesmente faça o commit 2x, uma em cada repositório.
 - a. Eu me desespero e choro!

```
1 folder/  
2   .git/  
3   subfolder/  
4     important_data.txt  
5     .git/
```

7. Eu quero conferir o quê já foi feito
O `commit` ou não. Para isso devo usar:

- a. `git status`
- b. `git check`
- c. `git commit --list`
- d. `git summary`

8. Eu acredito que a minha versão anterior está melhor que a atual, como posso conferir?

- a. `git compare`
- b. `git diff`
- c. `git show`
- d. `git gud`

9. Eu quero recuperar a minha última versão porque confirmei que está melhor, eu devo usar:

- a. Não é possível.
- b. `git commit --undo`
- c. `git rebase`
- d. `git revert`
 - a. A vida é só daqui pra frente.

10. Eu comecei uma colaboração com um amigo, como eu coloco uma cópia do repositório dele no meu computador?

- a. `git push`
- b. `git clone`
- c. `git pull`
- d. Ele só precisa me enviar um convite de colaboração.
 - a. Eu não sei porque eu comecei uma colaboração, quero desistir.

11. Para compartilhar as modificações do nosso projeto como meu amigo nós (Eu // Meu amigo) precisamos fazer:

- a. Eu faço commit depois push // ele(a) faz pull
- b. Eu faço commit depois pull // ele(a) faz clone
- c. Eu faço pull depois clone // ele(a) faz push
- d. Eu push // Ele(a) faz clone depois pull

12. Por que o Git usa o conceito de Staging area?

- a. A vida é complexa, assim como são os softwares
- b. Para que seja possível agregar atualizações em uma unidade lógica
- c. Eu gosto de fazer compras e colocar coisas no carrinho
- d. Eu só sei que eu sou obrigado a fazer isso!