Using Git with Github and Term

elias julian marko garcia

February 6, 2017

Outline

1. Why You Should Use Version Control
2. Basics
3. Contributing to a Repository

Version control is your friend

Version control is your friend

FOR ALL THAT IS HOLY, YOU SHOULD ALREADY BE USING VERSION CONTROL

Version control is your friend

FOR ALL THAT IS HOLY, YOU SHOULD ALREADY BE USING VERSION CONTROL

- You never know what might happen to your stuff.

Version control is your friend

FOR ALL THAT IS HOLY, YOU SHOULD ALREADY BE USING VERSION CONTROL

- You never know what might happen to your stuff.
- If your laptop was destroyed in the next ten minutes, would you be able to recover by tomorrow?

Version control is your friend

FOR ALL THAT IS HOLY, YOU SHOULD ALREADY BE USING VERSION CONTROL

- You never know what might happen to your stuff.
- If your laptop was destroyed in the next ten minutes, would you be able to recover by tomorrow?
- More extreme: what if your apartment, dorm, etc., caught fire?

Version control is a no brainer

Version control is a no brainer

This equally applies to all other work that you have, whether it is class papers or family photos or code.

Version control is a no brainer

This equally applies to all other work that you have, whether it is class papers or family photos or code.

- It should be agreeable that you need *something*

Version control is a no brainer

This equally applies to all other work that you have, whether it is class papers or family photos or code.

- It should be agreeable that you need *something*
- Your code is no exception

Version control is more than just a backup

It allows you to do many things:

Version control is more than just a backup

It allows you to do many things:

1. Track and organize your work

Version control is more than just a backup

It allows you to do many things:

1. Track and organize your work
    1.1 This means being able to switch between versions and revert regressive changes

Version control is more than just a backup

It allows you to do many things:

1. Track and organize your work
    1.1 This means being able to switch between versions and revert regressive changes
        1.1.1 You can have a main branch, a development branch, a bleeding edge branch, patch branches, and nth feature branches all simultaneously in existence

Version control is more than just a backup

It allows you to do many things:

1. Track and organize your work
    - 1.1 This means being able to switch between versions and revert regressive changes
        - 1.1.1 You can have a main branch, a development branch, a bleeding edge branch, patch branches, and nth feature branches all simultaneously in existence
    - 1.2 It becomes a (near) one stop hub for planning and organizing your project

Version control is more than just a backup

It allows you to do many things:

1. Track and organize your work

    1.1 This means being able to switch between versions and revert regressive changes

        1.1.1 You can have a main branch, a development branch, a bleeding edge branch, patch branches, and nth feature branches all simultaneously in existence

    1.2 It becomes a (near) one stop hub for planning and organizing your project

        1.2.1 Especially so with services like github

Version control is more than just a backup

It allows you to do many things:

1. Track and organize your work
   - 1.1 This means being able to switch between versions and revert regressive changes
     - 1.1.1 You can have a main branch, a development branch, a bleeding edge branch, patch branches, and nth feature branches all simultaneously in existence
   - 1.2 It becomes a (near) one stop hub for planning and organizing your project
     - 1.2.1 Especially so with services like github
     - 1.2.2 This is even for non-code/programming related things, i.e. academic research

Version control is more than just a backup

It allows you to do many things:

1. Track and organize your work
   - 1.1 This means being able to switch between versions and revert regressive changes
     - 1.1.1 You can have a main branch, a development branch, a bleeding edge branch, patch branches, and nth feature branches all simultaneously in existence
   - 1.2 It becomes a (near) one stop hub for planning and organizing your project
     - 1.2.1 Especially so with services like github
     - 1.2.2 This is even for non-code/programming related things, i.e. academic research
2. It allows you to show your work to others

Version control is more than just a backup

It allows you to do many things:

1. Track and organize your work
   1.1 This means being able to switch between versions and revert regressive changes
      1.1.1 You can have a main branch, a development branch, a bleeding edge branch, patch branches, and n$^{\text{th}}$ feature branches all simultaneously in existence
   1.2 It becomes a (near) one stop hub for planning and organizing your project
      1.2.1 Especially so with services like github
      1.2.2 This is even for non-code/programming related things, i.e. academic research
2. It allows you to show your work to others
   2.1 this is a huge plus for employers

Version control is more than just a backup

It allows you to do many things:

1. Track and organize your work
   - 1.1 This means being able to switch between versions and revert regressive changes
     - 1.1.1 You can have a main branch, a development branch, a bleeding edge branch, patch branches, and nth feature branches all simultaneously in existence
   - 1.2 It becomes a (near) one stop hub for planning and organizing your project
     - 1.2.1 Especially so with services like github
     - 1.2.2 This is even for non-code/programming related things, i.e. academic research
2. It allows you to show your work to others
   - 2.1 this is a huge plus for employers
   - 2.2 shows your ability (specialization and breadth of knowledge)

Version control is more than just a backup

It allows you to do many things:

1. Track and organize your work
    - 1.1 This means being able to switch between versions and revert regressive changes
        - 1.1.1 You can have a main branch, a development branch, a bleeding edge branch, patch branches, and nth feature branches all simultaneously in existence
    - 1.2 It becomes a (near) one stop hub for planning and organizing your project
        - 1.2.1 Especially so with services like github
        - 1.2.2 This is even for non-code/programming related things, i.e. academic research
2. It allows you to show your work to others
    - 2.1 this is a huge plus for employers
    - 2.2 shows your ability (specialization and breadth of knowledge)
    - 2.3 . . . your commitment (how long have you been programming?)

Version control is more than just a backup

It allows you to do many things:

1. Track and organize your work
   - 1.1 This means being able to switch between versions and revert regressive changes
     - 1.1.1 You can have a main branch, a development branch, a bleeding edge branch, patch branches, and n^{th} feature branches all simultaneously in existence
   - 1.2 It becomes a (near) one stop hub for planning and organizing your project
     - 1.2.1 Especially so with services like github
     - 1.2.2 This is even for non-code/programming related things, i.e. academic research
2. It allows you to show your work to others
   - 2.1 this is a huge plus for employers
   - 2.2 shows your ability (specialization and breadth of knowledge)
   - 2.3 . . . your commitment (how long have you been programming?)
   - 2.4 . . . your creativity (interesting, funny, useful projects)

Version control is more than just a backup cont.

It allows you to do many things:

1. Track and organize your work
2. It allows you to show your work to others

Version control is more than just a backup cont.

It allows you to do many things:

1. Track and organize your work
2. It allows you to show your work to others
3. Contribute to FOSS

Version control is more than just a backup cont.

It allows you to do many things:

1. Track and organize your work
2. It allows you to show your work to others
3. Contribute to FOSS
   - 3.1 Linux's kernal has been on Github for years.

Version control is more than just a backup cont.

It allows you to do many things:

1. Track and organize your work
2. It allows you to show your work to others
3. Contribute to FOSS
   - 3.1 Linux's kernal has been on Github for years.
   - 3.2 Google, Facebook, and etc. put many of their projects online and open for use, contribution

Version control is more than just a backup cont.

It allows you to do many things:

1. Track and organize your work
2. It allows you to show your work to others
3. Contribute to FOSS
   3.1 Linux's kernal has been on Github for years.
   3.2 Google, Facebook, and etc. put many of their projects online and open for use, contribution
   3.3 V.C. has been a key part to this process, before Github and especially after.

Version control is more than just a backup cont.

It allows you to do many things:

1. Track and organize your work
2. It allows you to show your work to others
3. Contribute to FOSS
   3.1 Linux's kernal has been on Github for years.
   3.2 Google, Facebook, and etc. put many of their projects online and open for use, contribution
   3.3 V.C. has been a key part to this process, before Github and especially after.
4. Cloud based solutions, like Github, ensure you stuff is always safe.

Version control is more than just a backup cont.

It allows you to do many things:

1. Track and organize your work
2. It allows you to show your work to others
3. Contribute to FOSS
   3.1 Linux's kernal has been on Github for years.
   3.2 Google, Facebook, and etc. put many of their projects online and open for use, contribution
   3.3 V.C. has been a key part to this process, before Github and especially after.
4. Cloud based solutions, like Github, ensure you stuff is always safe.
   4.1 Or, at least reasonably safer than you, a non corporate entity with non-corporate backup mechanisms, can guarantee.

There is no excuse not to use version control

There is no excuse not to use version control

Git is available for all platforms, as is services like Github

There is no excuse not to use version control

Git is available for all platforms, as is services like Github

If you are really concerned about sharing your code for x reason, still no excuses

There is no excuse not to use version control

Git is available for all platforms, as is services like Github

If you are really concerned about sharing your code for x reason, still no excuses

- Github has a student account program that gives you the paid benefits of Github for free

There is no excuse not to use version control

Git is available for all platforms, as is services like Github

If you are really concerned about sharing your code for x reason, still no excuses

- Github has a student account program that gives you the paid benefits of Github for free
    - i.e. private repositories that no one else can see unless you invite them to see it

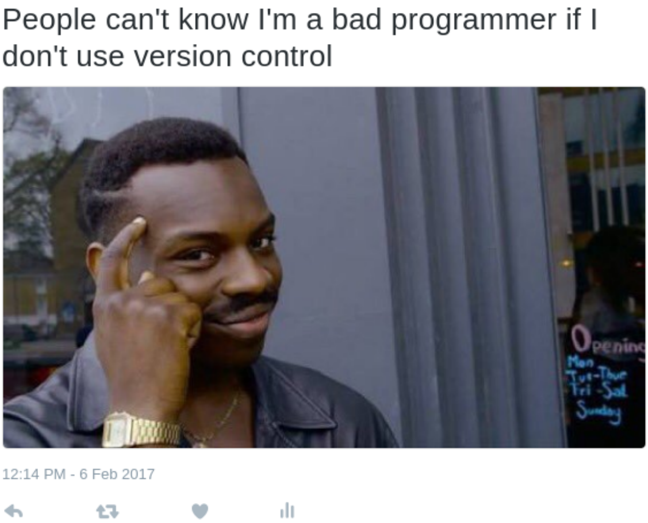
There is no excuse not to use version control

Git is available for all platforms, as is services like Github

If you are really concerned about sharing your code for x reason, still no excuses

- Github has a student account program that gives you the paid benefits of Github for free
  - i.e. private repositories that no one else can see unless you invite them to see it
  - tons of other benefits: free domain hosting, free hosting credits, payment processing waiver, database access, email infrastructure credits, free access to the unreal game engine, build integration tools, and a few more

"I don't need version control"



People can't know I'm a bad programmer if I don't use version control

12:14 PM - 6 Feb 2017

Preface

This is a hands on tutorial

Preface

This is a hands on tutorial

- I am not a git professional, but neither are most programmers

Preface

This is a hands on tutorial

- I am not a git professional, but neither are most programmers
- If you are a project manager, things obviously change.

Preface

This is a hands on tutorial

- I am not a git professional, but neither are most programmers
- If you are a project manager, things obviously change.
- Follow along if you can with your machine, if not, try to write notes.

Preface

This is a hands on tutorial

- I am not a git professional, but neither are most programmers
- If you are a project manager, things obviously change.
- Follow along if you can with your machine, if not, try to write notes.
  - Will go over your head otherwise because that is just how git is.

Assumptions

You have the following installed for your system:

Assumptions

You have the following installed for your system:

1. Git

Assumptions

You have the following installed for your system:

1. Git
2. Python 3.5

Assumptions

You have the following installed for your system:

1. Git
2. Python 3.5
3. Some kind of text editor (won't cover)

Assumptions

You have the following installed for your system:

1. Git
2. Python 3.5
3. Some kind of text editor (won't cover)
4. Some kind of shell (won't cover)

Assumptions

You have the following installed for your system:

1. Git
2. Python 3.5
3. Some kind of text editor (won't cover)
4. Some kind of shell (won't cover)
    - If you can do 1-2 without shell access, then you are probably using Windows. Boo.

Assumptions

You have the following installed for your system:

1. Git
2. Python 3.5
3. Some kind of text editor (won't cover)
4. Some kind of shell (won't cover)
    - If you can do 1-2 without shell access, then you are probably using Windows. Boo.
5. Github account (won't cover)

1. Install Git on your machine (Reference)

Linux

Tons of ways to do it, tar method works on all. Use system package manager when possible, just a lot easier.

```
# for ubuntu
sudo apt install git #16.04
sudo apt-get install git #14.04
```

Mac

1. Binary Installer via Git
2. Macports

   ```
   sudo port install git +svn +doc +bash_completion +gitweb
   ```

3. Homebrew

   ```
   brew install git
   ```

1. Install Git on your machine (Reference)

Windows

1. Install Git on your machine (Reference)

Windows

- Personally, I have zero experience with this. That said, Git says it's easy using the msysGit

2. Install Python on your machine (Reference)

2. Install Python on your machine (Reference)

Linux and Mac

2. Install Python on your machine (Reference)

Linux and Mac

- Again, tons of ways.

2. Install Python on your machine (Reference)

Linux and Mac

- Again, tons of ways.
- Personally recommend ‘Pyenv‘, fantastic Python version and virtualenv manager.

2. Install Python on your machine (Reference)

Linux and Mac

- Again, tons of ways.
- Personally recommend ‘Pyenv‘, fantastic Python version and virtualenv manager.

Windows

2. Install Python on your machine (Reference)

Linux and Mac

- Again, tons of ways.
- Personally recommend ‘Pyenv‘, fantastic Python version and virtualenv manager.

Windows

- Hahaha

2. Install Python on your machine (Reference)

Linux and Mac

- Again, tons of ways.
- Personally recommend ‘Pyenv‘, fantastic Python version and virtualenv manager.

Windows

- Hahaha
- Anaconda is a very good package that basically does what Pyenv does.

2. Install Python on your machine (Reference)

Linux and Mac

- Again, tons of ways.
- Personally recommend ‘Pyenv‘, fantastic Python version and virtualenv manager.

Windows

- Hahaha
- Anaconda is a very good package that basically does what Pyenv does.
  - Anaconda also works for Linux and Mac

Frame 1

Thanks to Eric Fraga for the first viable Beamer setup in Org

This is test text blah blah

Frame 1

Thanks to Eric Fraga for the first viable Beamer setup in Org

This is test text blah blah

Frame 1

Thanks to Eric Fraga for the first viable Beamer setup in Org

This is test text blah blah

Thanks to everyone else for
contributing to the discussion

Frame 2 (where we will not use columns)

Request

Please test this stuff!

Frame 2 (where we will not use columns)

Request

Please test this stuff!

Also, potatoes!