Table of Contents

Introduction

Do not expect this guide to work to solve all your problems, but to give you a first steps to improve your developers skill! If you have any comments, improvements or even a new chapter to add, feel free to fork or clone this repository and send us a message!

> “ 'A person is a person through other people' strikes an affirmation of one’s humanity through recognition of an ‘other’ in his or her uniqueness and difference. It is a demand for a creative intersubjective formation in which the ‘other’ becomes a mirror (but only a mirror) for my subjectivity. This idealism suggests to us that humanity is not embedded in my person solely as an individual; my humanity is co-substantively bestowed upon the other and me. Humanity is a quality we owe to each other. We create each other and need to sustain this otherness creation. And if we belong to each other, we participate in our creations: we are because you are, and since you are, definitely I am. The ‘I am’ is not a rigid subject, but a dynamic self-constitution dependent on this otherness creation of relation and distance”. -Michael Onyebuchi Eze, Ubuntu explanation

Writing markdown

Markdown is an excelent and lightweight markup language. With it you can make documentation like this, adding math notation, code blocks of different types of language, add image and even more.

To learn how to write markdown go here

Markdown package for Atom

Atom have nice Package to work with markdown:

In Atom go to settings ['ctrl + , '] > Install and then search for the following packages:

- Markdown-preview

  this package allow you to live preview your markdown text using the command ["ctrl + shift + m"]

- Markdown-pdf

  By selecting your markdown code and using the command ['Ctrl + shift + c'], this package convert a **single** file in a .PDF file.

Create a documentation book like this:

1. The official tutorial can be found here:
2. Install Gitbook

   ```
   npm install -g gitbook-cli
   ```

3. Prepare environment for gitbook

   ```
   gitbook init
   ```

4. Create PDF file

   ```
   gitbook pdf .\ .\book.pdf
   ```

Install and use virtual environment

Install virtualenv via pip:

1. Make sure you have python and pip installed

   ```
   pip install virtualenv
   ```

   If your using version control, dont forget to add your venv to your gitignore

   ```
   touch .gitignore
   ```

   open file and write **venv/**

2. Create a virtual environment

   ```
   virtualenv venv
   ```

3. Activate virtual environment

   ```
   source venv/Scripts/activate
   ```

4. Deactivate your virtual environment

   ```
   deactivate
   ```

5. Freeze your environment so you can create a file with all the libs you have installed in your virtual environment

   ```
   pip freeze > requirements.txt
   ```

6. Install requirements

   ```
   pip install -r requirements.txt
   ```

Installing and using version control git

Git is a version control system (VCS) for code. It is used to keep track of revisions and allow a developer or dev team to work together on a project through branches.

Installation and Set up

1. Download git.

2. Create an account on bitbucket or github:

   - GitHub for public repositories
   - Bitbucket for private repositories for free.

1. **Create an Online Git repository**

   Create a repository with a chosen name :

   - GitHub
   - Bitbucket

   Repository is a way to to share your code with your team or public.

2. Configurate SSH Key

   For security reasons SSH key is a measure to give permission to the user to modify the source code remotely.

   Every time you are in a different network, you will have to configure it.

   Follow instructions here for Bitbucket or here for GitHub.

Learn the basics

how to create a project repository

After you create your repository in GitHub or Bitbucket:

1. Create a Local Project and enter the folder:

   ```
   mkdir project-local
   cd project-local
   ```

2. To prepare your project to be a Git repository do :

   ```
   git init
   ```

   This will creater a ".git" directory with files to version control your file.

3. Configure your global Git account

   ```
   git config --global user.name "Firstname Lastname"
   git config --global user.email username@email.com
   ```

4. To **link** your local directory with the repository we made at github.com or bitbucket, for example:

   ```
   git remote add origin https://github.com/your_username/project_name
   ```

5. Check the status of your local repository.

   ```
   git status
   ```

6. Add ALL the files to the repo :

   ```
   git add .
   #or individually
   git add filename1 filename2 filename3 ...
   ```

7. Commit tracked files to the master branch

   Commit is a common command that after you update a change in your code you will *record* it changes and take a note to what have you done:

   ```
   git commit -am "Type here changes you've made on your work"
   ```

   In this section, you should type changes you've made on your code so you can track them later.

8. Now you send the files to the Git repo at github.com or bitbucket:

   ```
   git push origin master
   ```

New Branches

If you have one issue or a **BRAND NEW FEATURE THAT WILL REVOLUTIONIZE THE WORLD** and don't want to mess with the main code (MASTER), is good to create a branch for this code:

1. Create new branch

   ```
   git branch example
   ```

2. Start working on your new branch

   ```
   git checkout example
   ```

3. [*On the new branch*] to commit the branches:

   ```
   git push origin example
   ```

Merging Branchs

After your tasks on branch are over, you will have to merge the changes on the **master** branch, to do so":

1. checkout to master

   ```
   git checkout master
   ```

2. merge branch on the master:

   ```
   git merge example
   ```

3. push your change to the repository:

   ```
   git push origin master
   ```

Cloning a git repository

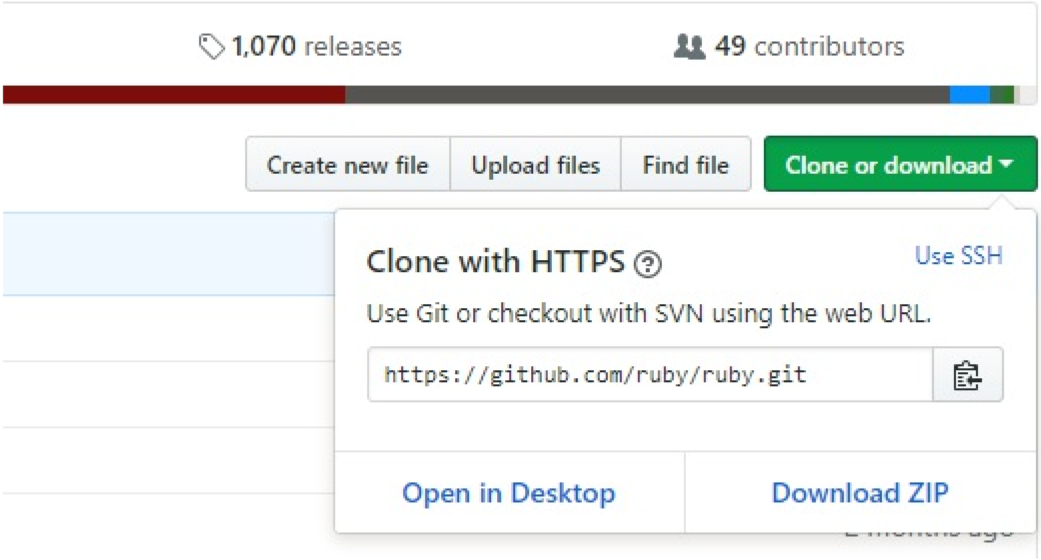
Sometimes you will have to clone a repository, i.e get the repository locally:

1. define your repository path:

   ```
   cd path/to/repository
   ```

2. in the github or bitbucket copy the https path protocol:

   - GitHub

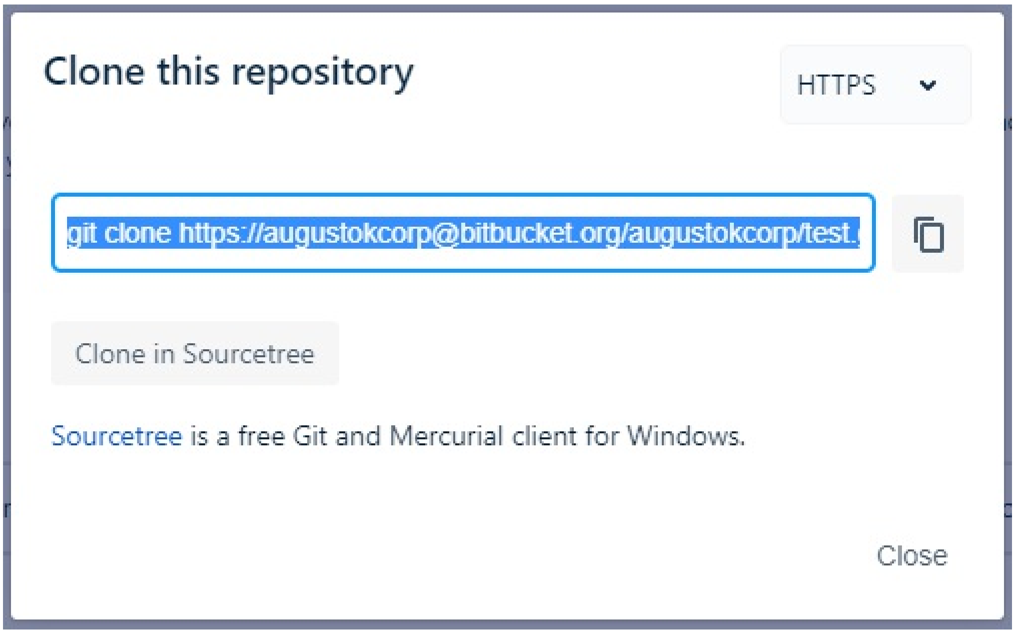


Clone with HTTPS

Use Git or checkout with SVN using the web URL.

```
https://github.com/ruby/ruby.git
```

   - bitbucket



Clone this repository

HTTPS

```
git clone https://augustokcorp@bitbucket.org/augustokcorp/test.
```

Sourcetree is a free Git and Mercurial client for Windows.

3. now just clone it:

```
git clone git://github.com/schacon/grit.git
```

---

bibliography

- GIT Documentation
- Bitbucket
- Github