Cách dùng git cho Ubuntu & Windows dòng lệnh.

Mình sẽ dùng giao thức SSH, không dùng giao thức HTTPS(Lười nhập acc & pass do lười nhập github).

|  |  |  |  |
| --- | --- | --- | --- |
|  | Task công việc | Ubuntu | Windows |
| 1 | Cài git | sudo apt-get install git | cài đặt git-scm từ trang :  git-scm.com/download/win |
| 2 | Gen SSH key (Nếu dùng giao thức HTTPS bỏ qua bước này). | 1.Ssh-keygen  2.hiện thông báo ghi vào file->enter luôn  3.nhập password.  4.nhập lại pass.  5.mở file id\_rsa.pub trong /home/[tên acc]/.ssh,cho vào SSH-KEY trong github tại https://github.com/setting/ssh | 0.Chạy git-bash lên.  1.Ssh-keygen  2.hiện thông báo ghi vào file->enter luôn  3.nhập password.  4.nhập lại pass.  5.mở file id\_rsa.pub trong /home/[tên acc]/.ssh,cho vào SSH-KEY trong github tại https://github.com/setting/ssh |
| 3.1 | Clone project dùng giao thức SSH | 1.vào thư mục htdocs của xampp, hoặc www của apache2 nếu không dùng xampp.  2.lệnh: git clone git@github.com:t02itjap/elearning → gõ pass là pass khi gen ssh-keygen | 0. chạy git-bash lên.  1.vào thư mục htdocs của xampp, hoặc www của apache2 nếu không dùng xampp.  2.lệnh: git clone git@github.com:t02itjap/elearning → gõ pass là pass khi gen ssh-keygen |
| 3.2 | Clone project dùng giao thức HTTPS | 1.vào thư mục htdocs của xampp, hoặc www của apache2 nếu không dùng xampp.  2.lệnh: git clone <https://github.com/t02itjap/elearning> → gõ id pass là id pass github. | 0.chạy git-bash lên.  1.vào thư mục htdocs của xampp, hoặc www của apache2 nếu không dùng xampp.  2.lệnh: git clone <https://github.com/t02itjap/elearning> → gõ id pass là id pass github. |
| 4 | Code cho project | -1. git checkout master.  0. git pull origin master.  1. git checkout [branch của mình]  Phi:phinm, Hoàn:hoanln, Khánh:khanhnd, Hưởng:huongcv,Thắng:thangvm, Thịnh:thinhnv, Tiến:tiendq.  → Dùng branch của từng người để code, tránh conflict.  2. git merge master : lấy dữ liệu từ project “chuẩn” về (nhánh master). Nếu có conflict nó sẽ liệt kê các file conflict. Cách sửa conflict :  <http://githowto.com/resolving_conflict>  rồi push lên nhánh của mình:git push origin [branch của mình].  2'. Nếu có thay đổi gì của nhánh của mình trên Repository thì pull về đã :  git pull origin [branch của mình].  3. gõ code.  4. git add -A : thêm các file đã code vào tracker.  5. git commit -m “comment nội dung thay đổi”  6. git push origin [branch của mình:ví dụ phinm]. | -1. git checkout master.  0. git pull origin master.  1. git checkout [branch của mình]  Phi:phinm, Hoàn:hoanln, Khánh:khanhnd, Hưởng:huongcv,Thắng:thangvm, Thịnh:thinhnv, Tiến:tiendq.  → Dùng branch của từng người để code, tránh conflict.  2. git merge master : lấy dữ liệu từ project “chuẩn” về (nhánh master). Nếu có conflict nó sẽ liệt kê các file conflict. Cách sửa conflict :  <http://githowto.com/resolving_conflict>  rồi push lên nhánh của mình:git push origin [branch của mình].  2'. Nếu có thay đổi gì của nhánh của mình trên Repository thì pull về đã :  git pull origin [branch của mình].  3. gõ code.  4. git add -A : thêm các file đã code vào tracker.  5. git commit -m “comment nội dung thay đổi”.  6. git push origin [branch của mình:ví dụ phinm]. |