Git basics

Tuesday 22 May 2018

14:55

1. Version control with Git

* Git is distributed version control system originally developed to manage Linux source codes.

* Using Git, you can easily view the revision history of your source code to track changes and easily go back in time to learn about the differences between different versions of your files.

* If the latest version of a file is on the shared repository, Git will prevent an unintentional overwrite by those maintaining an older version of the file.

BB BB BB3B 
index. txt 
index. txt 
BB3B 
point 
index. txt 
BB3B 
With Git, it is unnecessary to manually backup a copy every file 
version (ie_ saving multiple copies of your source directories). 

1. Repository to control history

* When you create a Git repository with the given files and directories, you can record their change history and start tracking their status and versions.

Ю0ИБоаеы 
ыиев 
s s щлч 

1. Remote repositories and local repositories

* There are two types of Git repositories namely a Remote repository and a Local repository.

* Remote repository: Repository that resides on a remote server that is shared among multiple team members.

* Local repository: Repository that resides on a local machine of an individual user.

* You can use all of Git's version control features on your local repository such as reverting changes, tracking changes, etc. However, when it comes to sharing your changes or pulling changes from your team members, that is where a remote repository comes in handy.

Pull 
Local repository 
html 
Push 
Remote repository 
I edit this! 
Push 
Pull 
Local repository 
html 
I edit this! 

1. Creating a repository

* There are two ways to create a local repository on your local machine.

* You can either create a brand new repository from scratch or by cloning an existing remote repository onto your local machine.

e10E2 e 00 
Lousode.l 
~ 0 so 
'MBN 