Git subtree 和submodule的区别

Git subtree 和submodule的区别主要在于submodule 的合并是引用的方式，占内存小，并且子模块和父项目是相对独立的。subtree 的合并是复制，以子文件夹的形式，占内存大，

虽然子模块也是相对父模块独立的，但是不能克隆，克隆后子模块就失去了独立，变成父项目的一个文件夹。

1添加子项目

Submodule

git submodule add 远程仓库地址 路径名

Subtree

git remote add 项目别名 远程仓库地址

git subtree add --prefix=路径名 项目别名 master（分支，）

添加项目后，需要push 到远程仓库，

Submodule 是引用的方式，只是添加一个标识，速度很快。

Subtree 是复制的方式，所以要将子项目上传到父项目的远程仓库，会很慢。

2 pull 的使用

Submodule

1、进入子项目目录，执行pull ,

2、在父项目内通过 git submodule foreach git pull ,同时操做多个子项目

子项目更新完以后需要在父项目中保存，提交到远程仓库，已更新远程中父项目的引用。

子模块pull 之后最好再提交一条语句 git submodule update

以保证能将更新顺利添加到父项目的远程仓库。

Subtree

1通过语句 git subtree pull --prefix=路径名 项目别名 master（分支）

2进入子项目单独pull

子项目更新完以后需要在父项目中保存，提交到远程仓库，已更新远程中父项目的引用。

1. Push 的使用

Submodule

1. git submodule foreach git push

2进入子项目 puhs

Subtree

1. git subtree push --prefix=路径名 项目别名 master（分支）
2. 进入子项目 puhs

Push 后都需要在父项目进行更新。 将子项目的更新提交到远程仓库。

1. clone

Submodule

1， 直接克隆， 这时子项目只有一个文件夹，没有内容，需要在父项目内执行

git submodule init

git submodule update

2, git clone --recursive git链接 ; 采用递归参数，可以一次clone全部。

Subtree

直接克隆， 并且克隆后项目中就没有了子项目的远程地址，变成了一个项目。

