**Git**是一个分布式的版本控制系统，最初由Linus Torvalds编写，用作Linux内核代码的管理。在推出后，Git在其它项目中也取得了很大成功，尤其是在Ruby社区中。目前，包括 Rubinius和Merb在内的很多知名项目都使用了Git。Git同样可以被诸如Capistrano和Vlad the Deployer这样的部署工具所使用。

如前所述，作为一个分布式的版本控制系统，在Git中并不存在主库这样的概念，每一份[复制](http://baike.baidu.com/view/548940.htm)出的库都可以独立使用，任何两个库之间的不一致之处都可以进行合并。

**GitHub**可以托管各种[git](http://baike.baidu.com/view/1531489.htm)库，并提供一个web界面，但与其它像 [SourceForge](http://baike.baidu.com/view/1091461.htm)或[Google Code](http://baike.baidu.com/view/2252816.htm)这 样的服务不同，**GitHub**的独特卖点在于从另外一个项目进行分支的简易性。为一个项目贡献代码非常简单：首先点击项目站点的“fork”的按钮，然后将 代码检出并将修改加入到刚才分出的代码库中，最后通过内建的“pull request”机制向项目负责人申请代码合并。已经有人将GitHub称为代码玩家的MySpace。

今天我们就来学习github的使用，我们将用它来管理我们的代码，你会发现它的好处的，当然是要在本系列**教程**全部完成之后，所以请紧跟站长的步伐，今天是第一天，我们来学习如何在git上建立自己的版本仓库，并将代码上传到仓库中。

教程【一】

一、注册一个免费git帐号，可以无限建立访问权限为公共的版本仓库，注册地址：<https://github.com/signup/free>

二、依次填写，用户名【昵称】、邮箱地址【用来登录】、密码，点击“creat an account”进行注册

三、点击右上方的”sign in”使用注册邮箱登录到git

四、点击右上方，你的用户名之后的图标，新建一个版本仓库

五、在“Repository name”一栏里填写版本仓库的名称，我们假设为”test”,Description栏是描述，可填可不填。

默认访问权限为公共，我们点击”Creating Repository”按钮，创建我们的第一个版本仓库。

接下来，我们需要在本地安装**git客户端**。

六、以[windows](http://www.woiweb.net/tag/windows)为例，下载客户端：<http://github-windows.s3.amazonaws.com/GitHubSetup.exe>，安装完成后，会在桌面上出现两个图标，Git Shell和GitHub。

七、那两个图标分别是命令行工具和图形界面，我们先来学习使用shell。

八、双击打开Git Shell,【win7用户使用管理员权限打开】

九、输入git config --global user.name "d1php",这里的d1php是站长注册的用户名，你得改成你自己的

十、输入git config --global user.email "studyphp@yahoo.cn",这里的邮箱也是是站长注册的邮箱，你得改成你自己的

十一、上面两步在客户端配置你的git帐户，下面我们在本地建立自己的版本仓库，站长把版本仓库建立在D盘。输入cd d:，切换到D盘目录。

十二、输入mkdir test，建立一个test目录，**注意本地仓库名要和git中建立的仓库名一样**

十三、输入cd test，进入test目录

十四、输入git init,初始化版本仓库

十五、输入touch README,建立一个README文件

十六、用文本编辑器打开README文件，随便输入一点文件，比如：this is a test!

十七、输入git add README,将文件添加到上传队列

十八、输入git commit -m 'test commit'，缓存提交，引号中的是对本次提交的描述，必须填写，不能为空

十九、输入git remote add origin https://github.com/*d1php*/test.git,注意url格式，d1php是你注册时的昵称，test是你刚才建立的版本仓库

二十、输入git push origin master，将README文件上传到版本库。这一步会一次让你填写你的git注册邮箱和密码，注意不要填错了

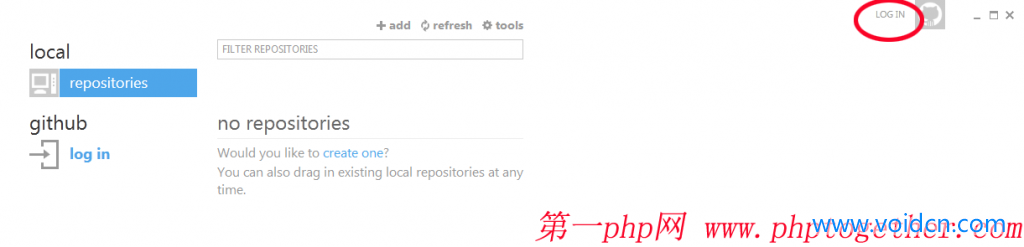
好了，到这一步，我们就完成了在线上和线下建立git版本库并上传文件的全部步骤。下节课我们来学习GitHub的使用，图形界面大家肯定更容易接受。

教程【二】

上一节课我们学习了[git shell的初步使用](http://www.phptogether.com/archives/12762)，今天我们来学习github的图形界面。

一、双击桌面上的GitHub图标

二、在打开的界面中，点击Log In,图下图所示：

[](http://www.phptogether.com/wp-content/uploads/2012/07/11.png)

三、在界面中输入用户名和密码，如下图所示：

[](http://www.phptogether.com/wp-content/uploads/2012/07/21.png)

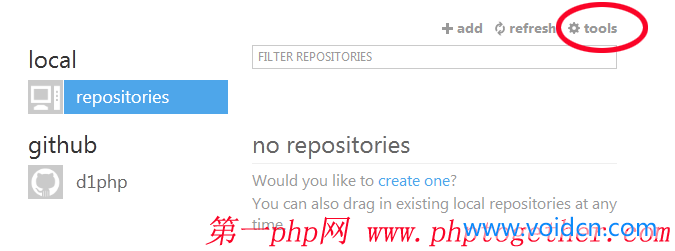
四、点击LOG IN，进入下图所示界面：

[](http://www.phptogether.com/wp-content/uploads/2012/07/3.png)

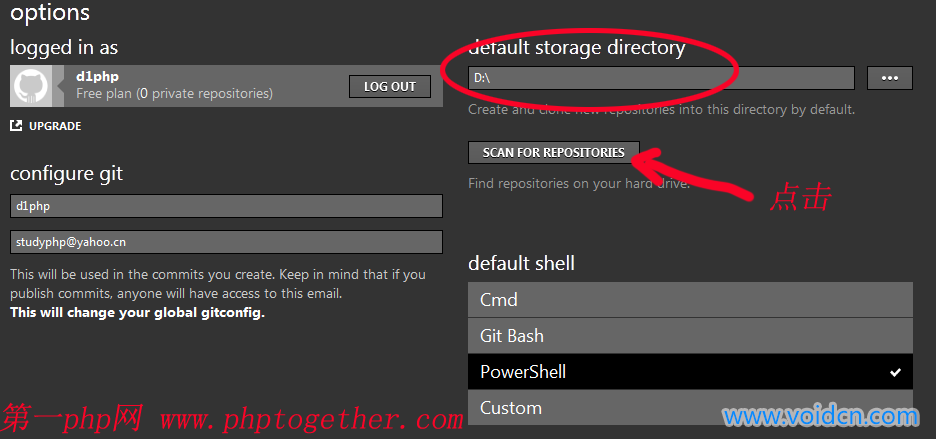
五、点击’continue’，进入下图所示界面：

[](http://www.phptogether.com/wp-content/uploads/2012/07/4.png)

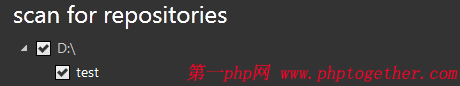
六、提示我们没有找到本地版本仓库，这是因为git客户端默认是在C盘的git目录中查找版本库，而我们在上节课中建立的版本仓库在D盘，所以才找不到，没关系我们自己来查找，点击SKIP,进入下图所示界面：

[](http://www.phptogether.com/wp-content/uploads/2012/07/5.png)

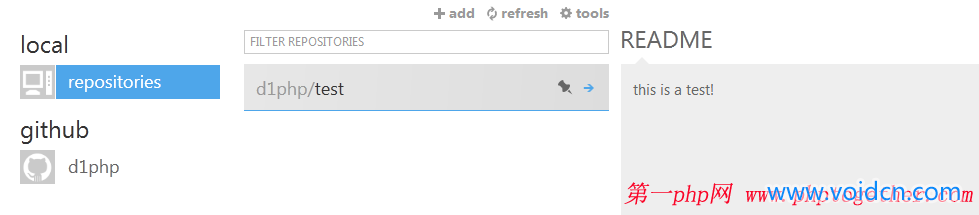
七、点击’tools’,在下拉菜单中选择’options’，进入下图所示界面：

[](http://www.phptogether.com/wp-content/uploads/2012/07/6.png)

八、定位到D盘目录，然后点击SCAN FOR REPOSITORIE，扫描本地仓库,如下图所示：

[](http://www.phptogether.com/wp-content/uploads/2012/07/7.png)

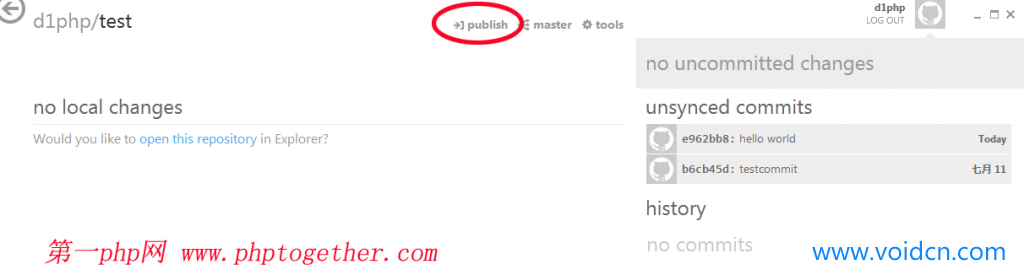
九、很快就找到了我们上节课建立的版本仓库，test，点击左下方的add，再点击update,进入下图所示界面：

[](http://www.phptogether.com/wp-content/uploads/2012/07/8.png)

十、进入D:/test目录，新建一个名为hello的txt文件，在其中输入hello world!,然后点击上图中的箭头打开版本库，进入下图所示界面：

[](http://www.phptogether.com/wp-content/uploads/2012/07/9.png)

十一、左边栏会列出最新有改动的文件，右边第一栏填写对本次提交的描述，我们填’hello world’;第二栏是详细描述，可以不填。填完后点击’commit’提交,进入下图所示界面：

[](http://www.phptogether.com/wp-content/uploads/2012/07/10.png)

十二、在右边我们可以看见刚才新建的提交:hello world,点击publish，将我们的改动上传到线上的git仓库中。

十三、稍等片刻，即可完成上传，如下图所示：

[](http://www.phptogether.com/wp-content/uploads/2012/07/111.png)

上传成功后，我们可以在左边看见刚才完成的提交。然后我们访问线上的仓库，检查一下：<https://github.com/d1php/test>

是的，线上的test仓库又多了2个文件。

好了，到这里我们就已经使用GitHub图形界面完成了一次文件上传，下节课我们将回到[gitshell](http://www.woiweb.net/tag/gitshell)的学习。我们将使用gitshell下载线上的版本仓库到本地，如果对本教程有疑惑可以联系站长哦。

教程【三】

今天我们来学习如何使用gitshell将线上的版本仓库下载到本地。我们为什么要把线上的仓库下载到本地，通常情况下是我们想要参与别人的项目。

github上面有一个项目：<https://github.com/octocat/Spoon-Knife>，点击访问它。

现在我们想要参与这个项目，需要进行以下步骤：

一、访问上面的链接，点击fork按钮，

二、现在spoon-knife项目已经添加到你的线上版本仓库中了，我们要下载它。打开gitshell，输入https://github.com/d1php/Spoon-Knife.git，注意这里的d1php应该是你注册时的昵称，在此使用了站长的帐号。

三、前两节我们把版本仓库建立在了D盘根目录，因此这里默认是在D盘目录只想的shell，进入D盘，会发现多了一个spoon-knife目录，线上的源码已经下载下来了。

四、输入cd spoon-knife

五、输入git remote add upstream https://github.com/octocat/Spoon-Knife.git，这里我们使用的是原项目的地址，确保我们可以随时跟踪到原项目

六、输入git fetch upstream，获取原项目最新的与你本地版本库不同的修改，但是不修改你的本地文件

好了，如何将线上仓库下载到本地就学习到这里，下节课我们学习如何建立版本分支。

教程【四】

学习，贵在持之以恒！欢迎大家跟站长继续学习github的使用，今天我们将学习如何使用 gitshell建立版本分支。如果你想要在原项目的基础上尝试一些新的思路或者开发新功能、修改等等，你可以建立原项目的一个分支，这样当进行修改时原 来的代码是不会受到影响的。我们前几节课建立一个test版本库，今天我们来为它建立 一个分支。

一、打开gitshell，输入 cd test，进入本地的test版本库

二、输入git branch testbranch

三、输入git checkout testbranch，将testbranch分支设置为当前工作分支

四、输入git add ./，将test目录的所有文件添加至上传队列

五、输入git commit -m 'newbranch'，缓存提交

六、输入git remote add origin https://github.com/d1php/test.git，配置上传目标仓库

七、输入git push -u origin testbranch，将文件上传到分支

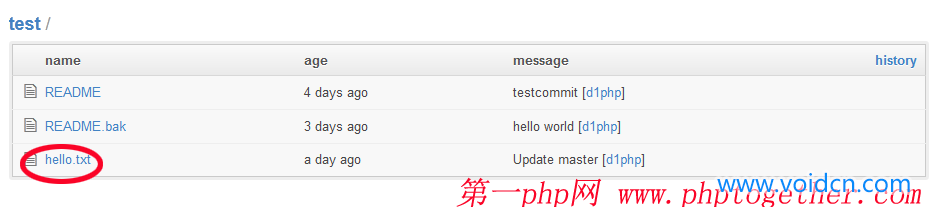
然后访问：<https://github.com/d1php/test/branches>，会发现多出了一个分支testbranch！

怎么样，今天的学习很轻松吧。明天我们将学习如何使用github图像界面将线上仓库中有改动的文件下载到本地，或者称之为“版本同步”。

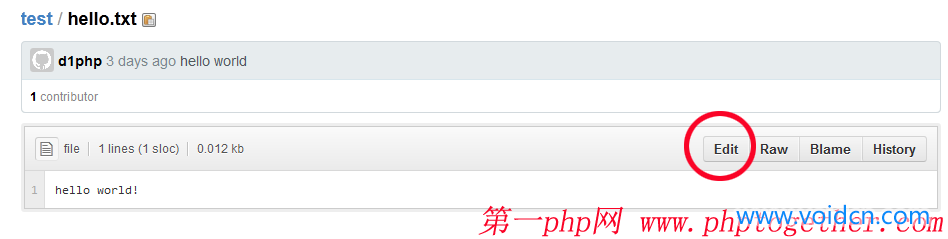
教程【五】

今天我们学习如何使用github图形界面来让我们本地的版本仓库和线上的仓库保持一致，或者说如何把线上仓库中有改动的文件下载到本地。

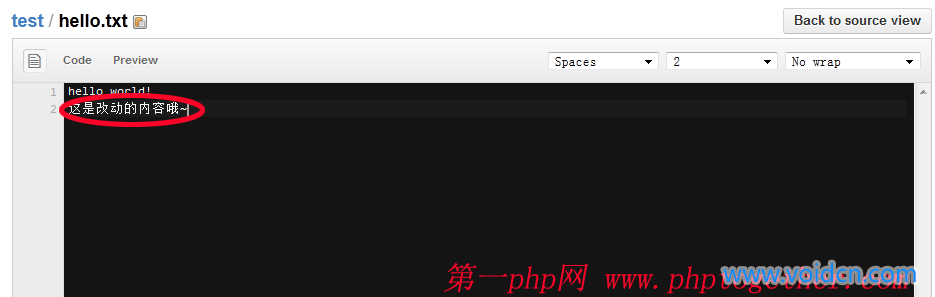
一、首先，我们访问<https://github.com/d1php/test>。还记得我们在第一节课建立的hello.txt文件吗？如下图所示：

[](http://www.phptogether.com/wp-content/uploads/2012/07/12.png)

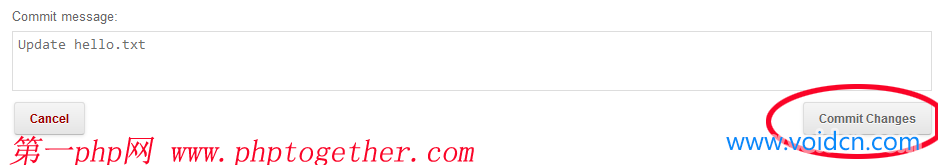
二、双击该文件，来到下图所示的界面：

[](http://www.phptogether.com/wp-content/uploads/2012/07/22.png)

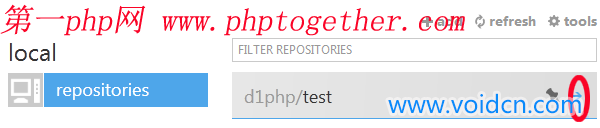
三、点击edit按钮，并对文本进行一些修改，如下图所示:

[](http://www.phptogether.com/wp-content/uploads/2012/07/31.png)

四、编辑完之后，点击页面最下方的commit changes按钮：

[](http://www.phptogether.com/wp-content/uploads/2012/07/41.png)

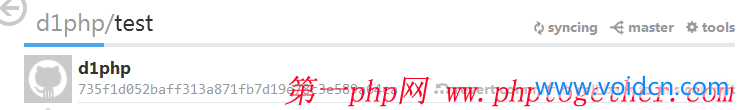
五、打开github,如下图所示：

[](http://www.phptogether.com/wp-content/uploads/2012/07/51.png)

六、点击右边的箭头，打开本地仓库，如下图所示：

[](http://www.phptogether.com/wp-content/uploads/2012/07/61.png)

七、github会自动读出最新的修改，并且上端会出现一个蓝色sync【sync是同步的意思】按钮。如果你没有看见蓝色的图标，可以点击F5刷新界面。现在我们点击sync按钮，这样github会将线上有改动的文件下载到本地，如下图所示：

[](http://www.phptogether.com/wp-content/uploads/2012/07/71.png)

下载玩之后，蓝色的sync会变成灰色的in sync，这说明最新的文件已经下载到本地的仓库了，去检查一下吧！怎么样，没错吧。

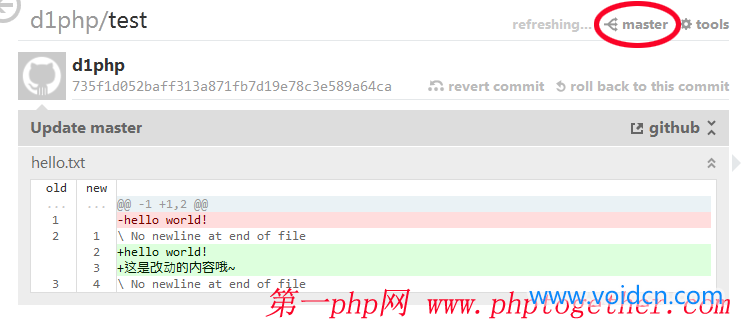
感谢大家跟着站长进行了5天的学习，明天我们将继续学习github图形界面的使用，我们将学习如何使用github图形界面来建立一个版本库的分支。

教程【六】

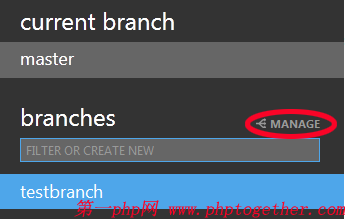
学习，有时候就是一个漫长而枯燥的过程，但是当你走到终点时，回过头来再看看这段路，则会露出满不在乎的表情！今天我们学习如何使用gitgui来建立版本分支。

一、打开github图形界面,点击蓝色箭头打开我们之前建立的test仓库，如下图所示：

二、在打开的界面中，点击master按钮，如下图所示：

[](http://www.phptogether.com/wp-content/uploads/2012/07/23.png)

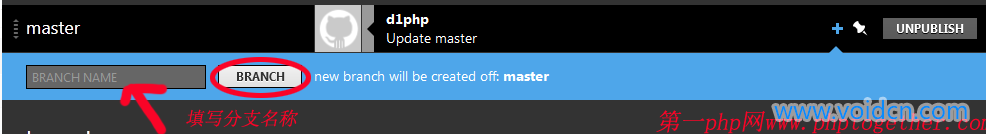
三、在弹出的窗口中，点击manage按钮，如下图所示：

[](http://www.phptogether.com/wp-content/uploads/2012/07/32.png)

四、进入分支界面，把鼠标移动到master分支上，并点击加号，如下图所示：

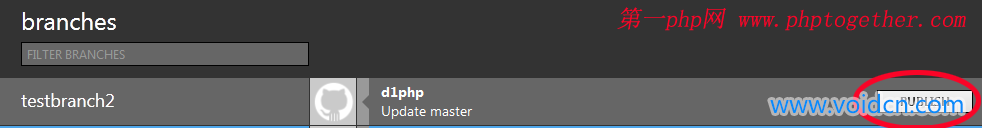
这会以master分支的文件为蓝本建立一个新的分支。

五、点击加号后，会弹出输入框，输入新分支的名称，并点击BRANCH按钮创建分支，如下图所示：

[](http://www.phptogether.com/wp-content/uploads/2012/07/52.png)

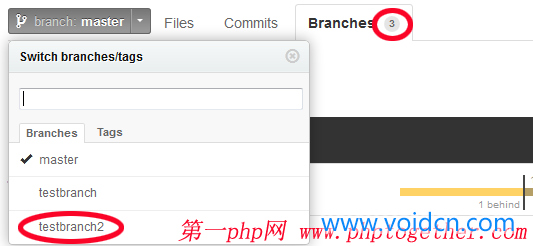
站长输入是testbranch2，因为之前用gitshell建立一个testbranch分支。

六、创建完之后，会在当前页面显示新分支，如下图所示：

[](http://www.phptogether.com/wp-content/uploads/2012/07/62.png)

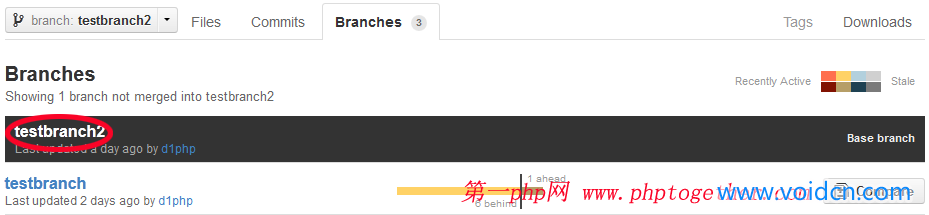
点击publish按钮，将新分支发布到线上的git仓库中。发布完之后，publish按钮会变成unpublish，说明分支已经上传到线上的仓库中了。

七、访问<https://github.com/d1php/test/branches>，查看分支情况，如下图所示：

[](http://www.phptogether.com/wp-content/uploads/2012/07/72.png)

我们会发现，尽管显示有3个分支，但是在页面上只呈现了2个，新建立的没有出现，这是因为默认是读取master分支以及没有与master整合的 分支，而我们刚才创建的testbranch2分支是以master为蓝本，因此它们之间没有任何不同，所以才没有显示。我们点击branch下拉按钮， 可以看见我们刚才建立的testbranch2分支。

八、点击上图中的testbranch2按钮，进入下图所示界面：

[](http://www.phptogether.com/wp-content/uploads/2012/07/81.png)

现在，我们可以在线上的版本仓库中看见我们刚才建立的分支了。

九、现在我们回到第六步的github图形界面，把鼠标移动到新建分支testbranch2上面，如下图所示：

[](http://www.phptogether.com/wp-content/uploads/2012/07/91.png)

上图中第一个红圈包围的图标是用来删除分支的，它将同时删除本地和线上仓库的分支；第二个红圈仅仅删除线上仓库中的分支，本地分支会被保留。现在我们，点击第一个红圈包围的图标，将本地分支和线上分支同时删除。

十、访问<https://github.com/d1php/test/branches>，我们会发现刚才建立的testbranch2分支已经被删除了，gitgui中也不再显示该分支了。

ok，到此我们就学会了如何使用github图形界面创建、发布、删除分支，下节课我们将学习如何使用github图形界面对分支版本进行操作。