# Git介绍及常见命令使用

## Git客户端安装

### 下载

下载地址：<https://git-scm.com/>

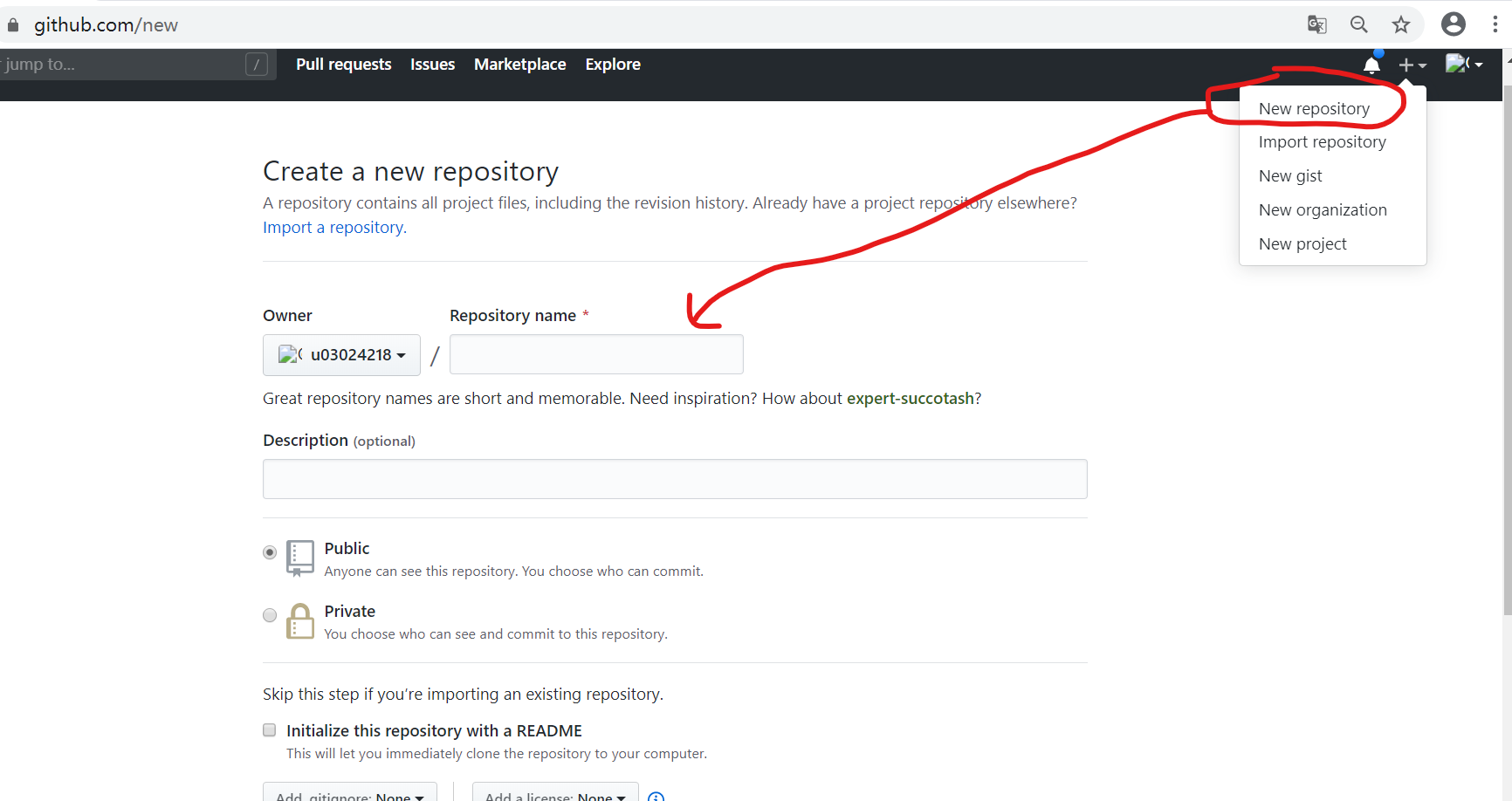
### 安装

<https://blog.csdn.net/q563573095/article/details/79558067>

## Git常见命令

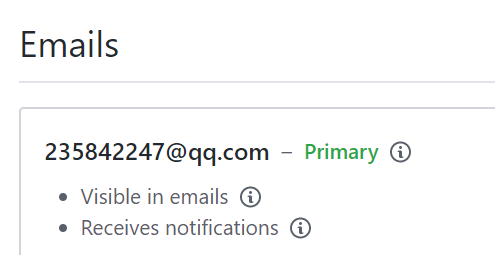
### 创建一个git项目

通过web页面进行git项目创建。

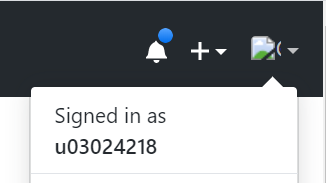


### 配置账户信息

git config --global user.email ["235842247@qq.com"](mailto:\"you@example.com\")



git config --global user.name "u03024218"



### 上传文件\文件夹

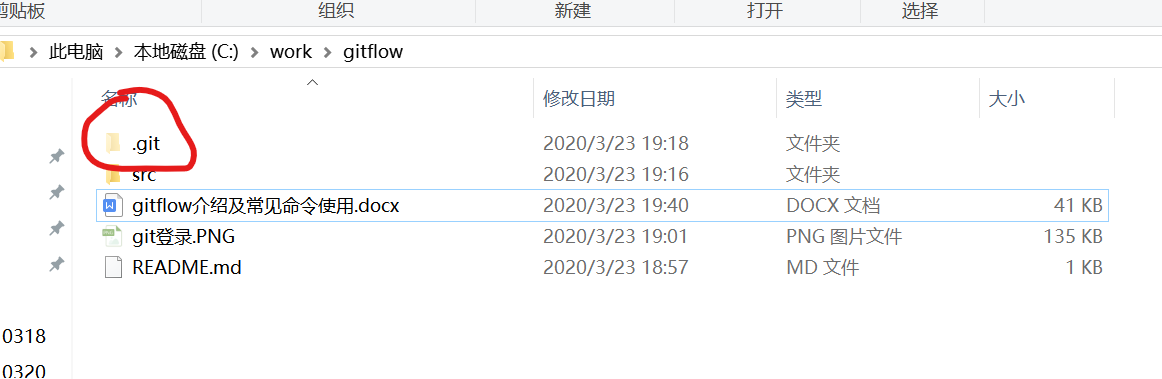
1. Git目录初始化（仅一次）

命令：git init

功能：进入到指定目录后，将目录初始化为git目录（初始化成功后，目录中会新 建一个.git的隐藏文件夹）

效果：

执行成功后系统提示：Initialized empty Git repository in C:/work/gitflow/.git/



1. 配置目标工程（仅一次）

命令：git remote add origin <https://github.com/u03024218/gitflow_test.git>

功能：绑定指定工程，用于后续上传文件或者文件夹

1. 新建文件\文件夹
2. 添加待上传文件\文件夹

git add .\README.md --上传文件

git add src --上传文件夹

1. 添加待上传文件\文件夹备注

git commit -m "first commit src" --设置注释

1. 执行上传

命令：git push -u origin master

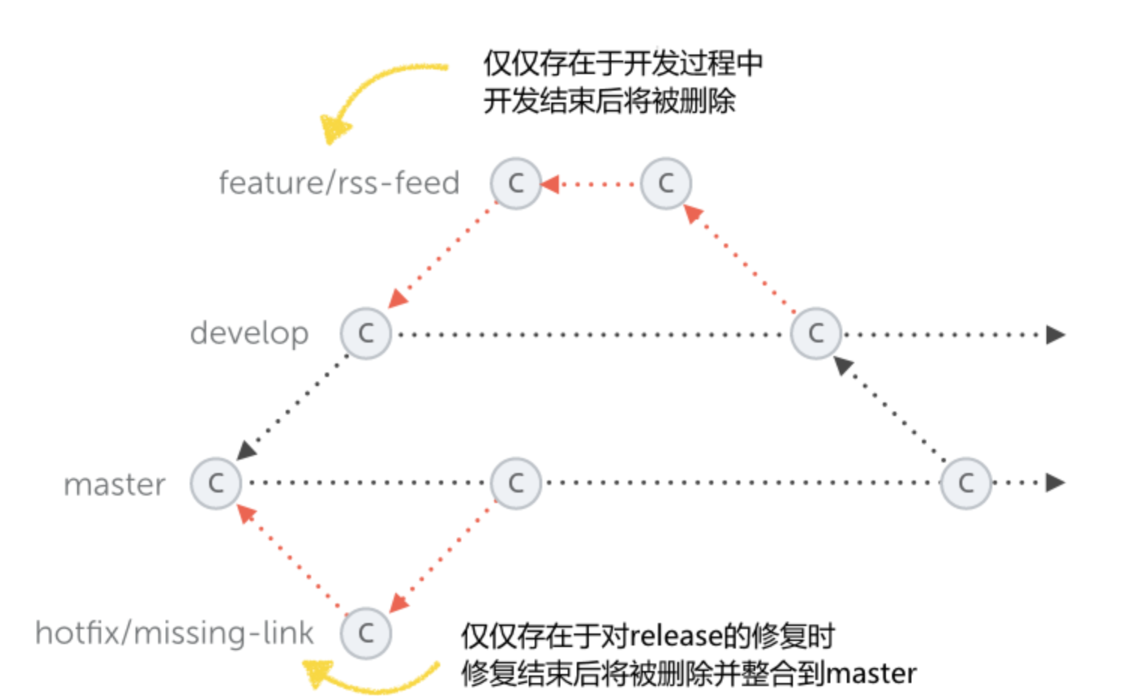
功能：执行上传操作

1. 修改文件

修改文件与新建操作一致

## Gitflow流程

### Gitflow流程与行内流程对比



对比行内研发阶段：

|  |  |  |
| --- | --- | --- |
| 行内 | Gitflow | 个人理解 |
| 开发库 | Develop | 行内开发库（开发环境）：主要是用于研发人员进行研发、冒烟环境。  Develop：gitflow中dev阶段属于 |
| 受控库 | Release | 行内受控库（测试环境）：主要用于技术测试，准备上线。  Release：项目上线前的一些全面测试以及上线准备（更多的测试投入在feature阶段） |
| 产品库 | Master | 行内产品库：个人理解为演示环境，供需求、研发、测试人员了解系统。  Master：项目的主要分支，可以把它理解成 稳定无bug发布版 |
|  | Featrure | 行内：暂无，因为不涉及该类研发模式。从目前现状看，行内每个项目（一般涉及1、2个功能点）可以算成1、2个feature。  Feature：每一个feature为一个功能点。Feature分支都是从develop中创建的。在每个feature测试完成后，会合并到develop分支中，最后进行全面回归测试。 |
|  | Hotfix | 行内：暂无，因为不涉及该类研发模式。从目前现状看，当线上出现严重、紧急问题时，研发人员重新打包上线后，将代码上传到svn的操作，算是一个hotfix。  Hotfix：用来修复紧急bug的。 |

### Git分支类别介绍

1. Master

项目的主要分支，可以把它理解成 稳定无bug发布版 。所以，git-flow 要求我们不能在master下做开发。

1. Develop

处于功能开发最前线的版本，查看develop分支就能知道下一个发布版有哪些功能了。  
develop一开始是从master里分出来的，并且定期会合并到master里，每一次合并到master，表示我们完成了一个阶段的开发，产生一个稳定版。  
同样的，develop下也不建议直接开发代码，develop代表的是已经开发好的功能

1. Feature

feature的作用是为每一个新功能从develop里创建出来的一个分支。  
 例如A和B分别做两个功能，一个为指纹登录，一个为转账。就可以创建两个分支，各自开发完以后，先后合并到develop里，这就叫做回归。  
 在这个过程里，A 和 B不需要任何的沟通，分别并行地开发，  
 git-flow能很好的处理好分支间并行开发的关系。  
 而develop，则会在适当的时候，由合适的人，合并到master，作为下一个稳定版本。

1. Hotfix  
   它是用来修复紧急bug的，而bug通常是来自线上的，  
   所以hotfix分支是从master里创建出来的，并且，在bug修改好以后，  
   要同时合并到master和develop。
2. Release

release更多倾向与版本发布，项目上线前的一些全面测试以及上线准备。