Git是一款免费、开源的分布式版本控制系统，用于敏捷高效地处理任何或小或大的项目。学习前端，离不开Git的使用，面试时也是一个常考的话题，在日常开发中，也困扰我们许久，下面就让我们一起走进它~

我将从以下几点进行介绍,准备好走进Git的世界了嘛~

* 高频词汇的含义
* git常用指令
* 几种常见git操作流程

高频词汇的介绍

git操作博大精深，细节很多，对于日常开发的我们而言，熟识几条常用的操作便可解决日常问题。对于初学者而言了解git基本词汇的含义和git的工作原理还是很有必要的，这样对解决git操作中的“bug”也非常有帮助。

* remote：指存放在远程的一个计算机，在一般指GitHub服务器。
* clone：指克隆，将github上的项目克隆（也可以说是复制）到自己的电脑上。
* origin：指远程的代码仓库（自己的仓库），即源文件。
* upstream：指上游，fork别人项目的地址，别人的项目就是你的上游，即上一层源文件。
* master：指主分支。
* pull：指拉代码，把你Github上的远程仓库拉到本地计算机上。
* push：指提交代码，把你本机上的项目提交到Github上的远程仓库。

**特别注意：本地的 .git文件是指本地仓库，.git目录下存放着所有文件的版本和关联信息，该目录在默认条件下是隐藏的。**

git常用指令的用法

**1.源文件：**

* git remote -v ：指查看远程关联的仓库，包括源文件和上一层源文件。
* git remote add origin + 网址：指添加一个源文件。
* git remote add upstream + 网址：指添加一个上一层源文件，即fork项目的源文件。
* git remote remove origin/upstream + 网址：指删除一个源文件或上一层源文件。

**2.初始化：**

* git clone + 网址：指把网址中对应的项目克隆到自己本机上。
* git init：指初始化.git文件，里面没有链接信息，初始化后要重新建立与远程仓库的关联。
* git pull upstream master：指把获得原项目更新。

**3.提交：**

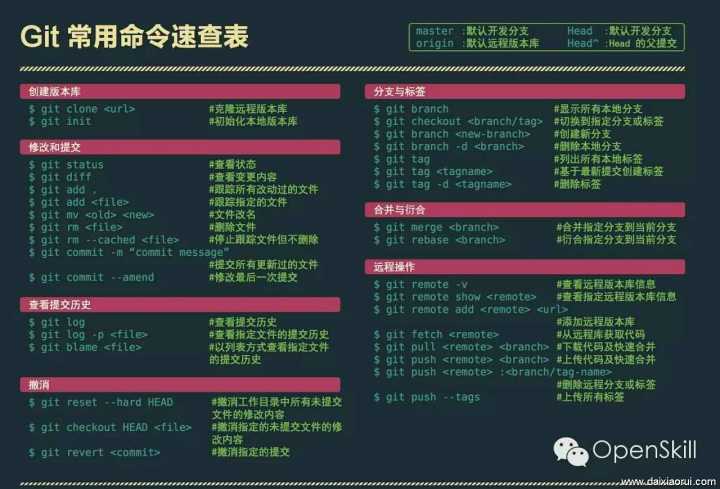
* git add + 文件名：指把修改的某个文件添加到暂存区（指.git仓库）；
* git add . :指把所有的修改文件添加到暂存区；
* git commit -m "认真写为什么要做这次提交"：便于日后查阅和他人理解；
* git push -u origin master:指把暂存区的文件提交到github中。
* git log：指查看提交的历史记录。
* dir：查看该文件的子目录。

**4.分支：**

* git status：查看所有情况；
* git branch：指查看分支信息；
* git branch + 分支名：指创建新的分支；
* git checkout + 分支名：指切换分支。

看到这里大家是不是已经有点晕呢

下面用一张图对git的常用命令总结一下：

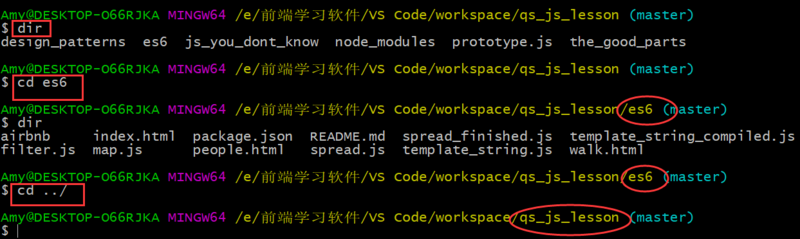


几种常见的git流程

**前提：只有进入到相应的文件目录后才能对git进行相关的操作。**

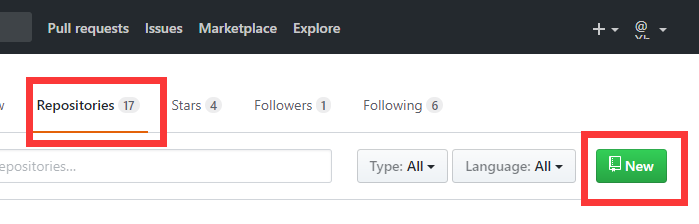
相关文件夹命令行操作：

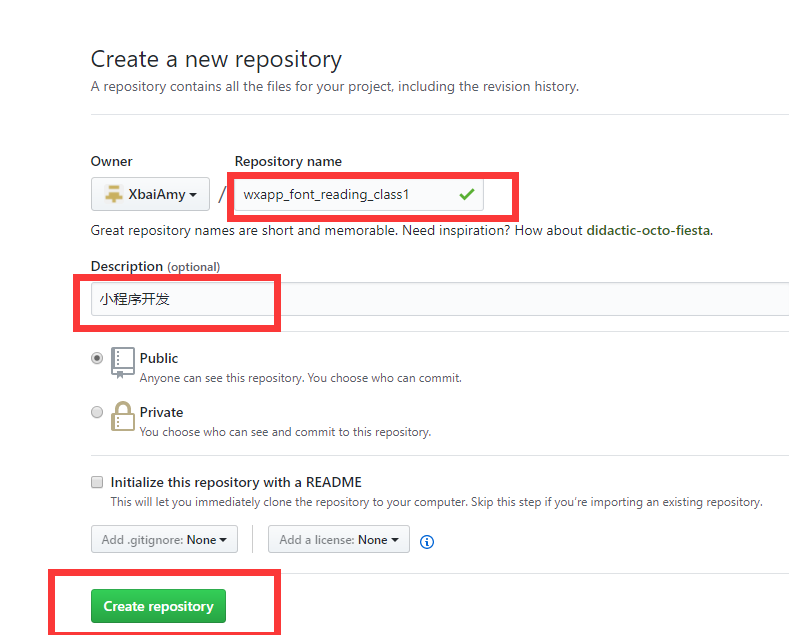
* cd + 文件名：进入该文件目录下
* cd ../ :返回到上级目录
* dir：查看当前文件下的所有子文件



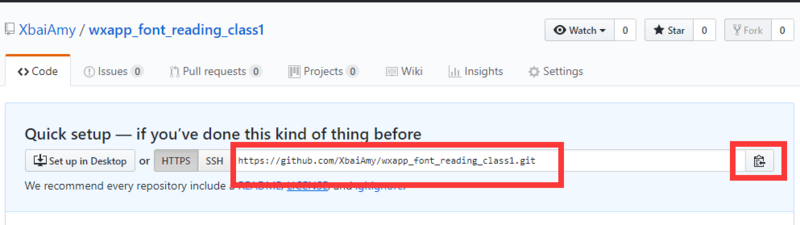
**情况一：新建一个项目并建立关联**

1.先在github上创建一个新仓库

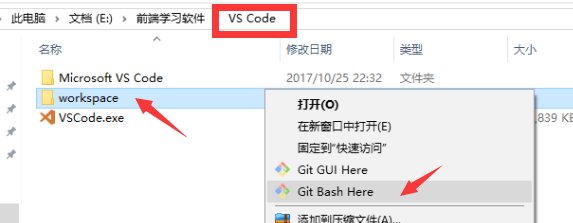


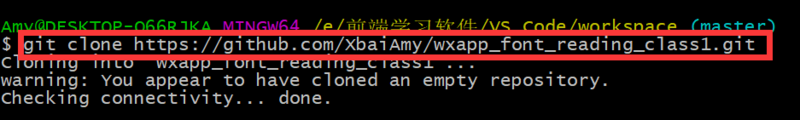


点右边红框里的按钮复制仓库地址：



2.在本地VSCode的workplace下点击右键Git Bash Here打开

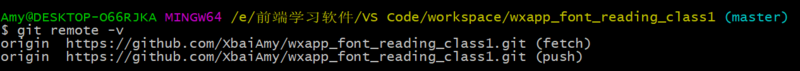


3.git clone + 仓库地址，把仓库里的文件克隆到本机上。  


此时在本地就新建了一个wxapp\_font\_reading\_class1文件夹

https://segmentfault.com/img/bVZAV7?w=399&h=57

4.git remote -v 查看文件关联



**情况二：fork别人的项目到本机上**

1. 在workplace下新建一个新的文件夹；
2. 右键该文件夹，点击Git Bash Here 打开；
3. 依次输入：
4. （1）git init（初始化该文件夹）
5. （2）git remote **add** upstream + fork文件地址（与fork的项目关联）
6. （3）git pull upstream master（把fork的项目拉到本机上）
7. 在新建文件夹下创建了fork的文件。

**情况三：如何和fork的项目保持同步更新**

* 与fork的项目建立了关联

输入"git pull upstream master"就可实现代码更新

* 未与fork的项目建立关联
* 1）先建立项目关联
* git remote **add** upstream + fork文件地址
* git remote -v 查看项目关联
* 2）再实现代码更新
* git pull upstream master 把项目拉到本机上

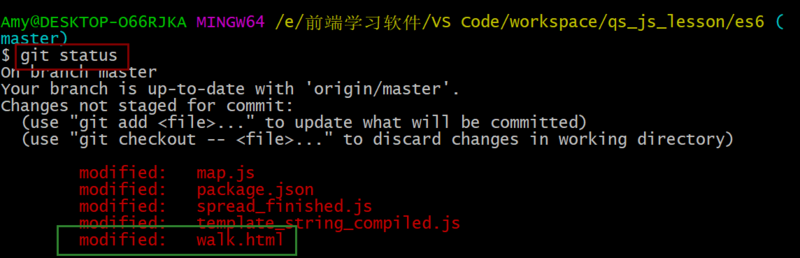
**情况四：git别人的项目，但是不通过fork形式的流程**

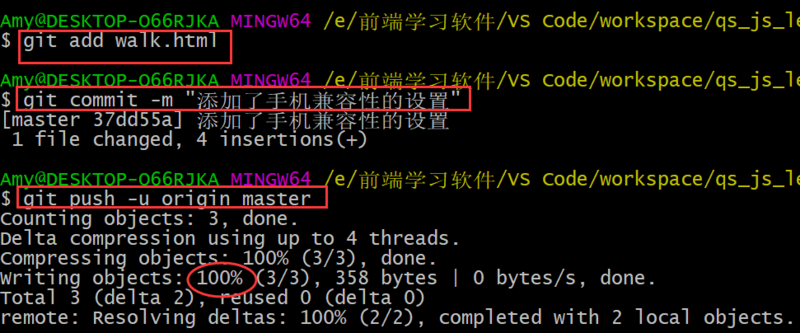
以增加wxapp-starter文件的实例来说明:

* 在github上新建一个"wxapp-starter"仓库
* git clone + 别人项目的链接
* git remote remove origin （删除别人origin的链接）
* remove .git（删除wxapp-starter文件里的.git 文件）
* git init （初始化 .git文件）
* git remote add origin + 自己仓库的地址（在 .git 中增加自己的origin的链接）
* git pull origin master（与自己的项目同步）

**情况五：git提交的通用过程（每次修改代码后都要进行的操作）**

1. git status （查看当前文件的修改状态）；
2. git add 某文件（添加某个修改文件）/ git add .(添加文件所有修改文件)；
3. git commit -m "注释"（提交文件并备注）；
4. git push -u origin master（把暂存区的文件发送到github上）

以es6文件为例进行提交：  




关于git操作的分享就告一段落啦，后续会有更多其他内容，欢迎大家一起交流学习哟(＾Ｕ＾)ノ