



Git和GitHub使用教程



MachinePlay 关注

1 2018.12.20 00:18:30 字数 4,360 阅读 30,823



github.png

内容提要

本文分为两个个部分

- 1.Git教程
- 2.Github教程

第一部分

1.什么是GitHub?

GitHub是世界上最大的软件远程仓库，是一个面向开源和私有软件项目的托管平台，使用Git做分布式版本控制。

简单的来说，GitHub就是全是界程序员和组织发布程序代码的平台之一，全世界的程序员讲自己写的代码上传到这里与大家分享。

当你需要完成某个轮子（开发某个东西）的时候，可以尝试先去GitHub借鉴一下别人已经开发过的。

或者你和同伴们打算开发一个小软件，就可以将代码放到GitHub上来实现合作开发。

推荐阅读

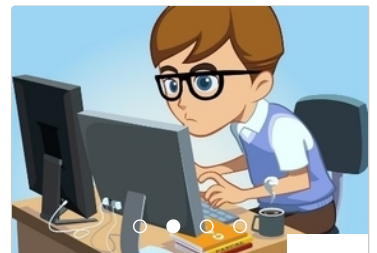
20200819 行人重识别算法比赛
阅读 512

原地哈希算法
阅读 1,027

docker安装和启动
阅读 91

将博客搬至CSDN
阅读 236

CatBoost算法 & XGBoost算法 &
Light GBM算法
阅读 21



程序员要学多久

写下你的评论...

评论7

赞54

...



1. Git是什么？



git.png

Git是实现分布式版本控制的一个工具，简单的来说就是实现文件历史管理的工具。
下面我举一个例子，说明Git的作用。

1.你是公司的一名程序员，现在你写了一段程序：

```
1 | print('Hello')
```

2.老板看到你的代码太差了，让你改掉，于是你改成了：

```
1 | print("你好")
```

3.老板看到以后觉得不行，让你把代码改回到上一次的状态

如果你从1改到2改了一大半内容，并且还没有留1的备份，是不是要哭了？

如果你在这个时候使用了Git做版本控制，就不会有这种问题了。

每一次你修改代码之后都做用Git一次记录，Git就会知道你每一次改了什么内容。

老板让你回到1，你只要输入 `git reset --hard 1`，就回到了当时的版本。

所以说，Git就是实现历史版本管理的工具。不论你改了多少内容，改了多少次，Git都能回到当时的版本。

同时Git也是合作开发的工具。

比如现在有100个人的团队一起开发某个软件，大家肯定不能用复制粘贴代码来合作。

我们可以指定所有人都把代码上传到一台服务器，然后大家下载服务器上的代码来修改，修改完了再上传回去。这时，Git记录每个人什么时候，改了什么内容。实现所有人同步，在必要时回到某些版本。

再比如，QQ每隔一段时间都要更新版本，忽然某个新版本不稳定，腾讯决定撤回到上个版本，这个时候就要让代码回到发布新版本之前。

推荐阅读

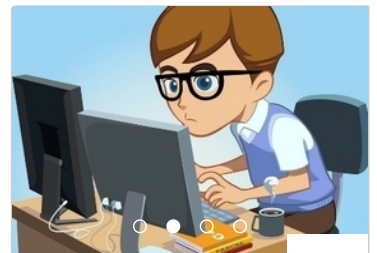
20200819 行人重识别算法比赛
阅读 512

原地哈希算法
阅读 1,027

docker安装和启动
阅读 91

将博客搬至CSDN
阅读 236

CatBoost算法 & XGBoost算法 &
Light GBM算法
阅读 21



程序员要学多久

2.使用Git

写下你的评论...

评论7

赞54

...

Git和GitHub使用教程



MachinePlay

关注

赞赏支持

去[Git官网](#)下载对应操作系统的Git。

windows和mac的安装过程简单，这里不做赘述，讲一下linux（ubuntu）的，其他linux系统的玩家水平应该都很高，不需要看教程~~

在终端输入

```
1 | sudo apt-get install git
```

输入密码安装即可。

设置Git

windows打开GitBash，macos和linux用户打开终端。

首先要设置自己的身份，比如git提交代码的时候要让别人知道什么人提交了代码，

设置身份内容有两条，一个是你的邮箱，另一个是你的称呼，以后你提交的代码都根据这个来确定是你

提交的的了。Your Name填你想让别人知道的名字，[email@example.com](#)换成自己的邮箱。

自此以后你写的bug就会被人认出来是你写的了。

输入：

```
1 | $ git config --global user.name "Your Name"
2 | $ git config --global user.email "email@example.com"
```

使用Git

从终端（cmd）进入你想要记录内容更改的文件夹里

例如我们进入 `gittest` 文件夹

输入：

```
1 | git init
```

这个文件夹以后的更改就会被记录了。（如果是空文件夹会提示 `Initialized empty Git repository in /home/yep/code/gittest/.git/`，告诉你文件夹为空）

现在我们在文件夹里新建一个文件hello.txt,内容是

```
1 | print("Hello World")
```

现在我们要把这个文件放入git仓库。

和把大象放到冰箱需要3步相比，把一个文件放到Git仓库只需要两步。

第一步：保存后，我们使用 `git add` 命令，告诉git我们把文件添加到仓库缓存区了，在终端输入

```
1 | git add hello.txt
```

推荐阅读

20200819 行人重识别算法比赛

阅读 512

原地哈希算法

阅读 1,027

docker安装和启动

阅读 91

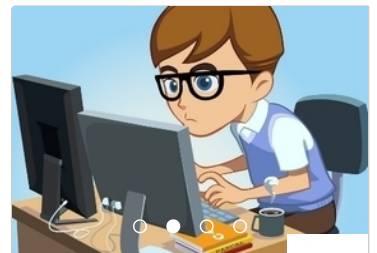
将博客搬至CSDN

阅读 236

CatBoost算法 & XGBoost算法 &

Light GBM算法

阅读 21



程序员要学多久

写下你的评论...

评论7

赞54

...

Git和GitHub使用教程



MachinePlay

关注

赞赏支持

```
1 | git commit -m "添加了hello.txt"
```

其中 -m 和后面引号内容是本次提交的说明，也就是描述你每次改了什么。

嫌麻烦不想输入 -m "xxx" 行不行？确实有办法可以这么干，但是强烈不建议你这么干，因为输入说明对自己对别人阅读都很重要。实在不想输入说明的童鞋请自行Google，我不告诉你这个参数。

```
1 | [master (root-commit) ec4652d] 添加了hello.txt
2 | 1 file changed, 1 insertion(+)
3 | create mode 100644 hello.txt
```

git commit命令执行成功后会告诉你，1 file changed：1个文件被改动（我们新添加的hello.txt文件）；1 insertions：插入了两行内容（hello.txt有一行内容）。

为什么Git添加文件需要 add，commit 一共两步呢？因为 commit 可以一次提交很多文件，所以你可以多次 add 不同的文件，比如：

```
1 | $ git add file1.txt
2 | $ git add file2.txt file3.txt
3 | $ git commit -m "add 3 files."
```

add 把文件放到了缓存区，然后 commit 正式提交到仓库。

实现版本回退

现在修改hello.txt里的内容：

```
1 | print("Hello World")
2 | print("老板是神经病")
```

因为对工作很不满，下班前添加了第二行，“老板是神经病”，然后提交

```
1 | git add hello.txt
2 | git commit -m "hello.txt里添加了一句话"
```

提示：

```
1 | [master 88d885c] 在hello.txt添加了一句话
2 | 1 file changed, 2 insertions(+), 1 deletion(-)
```

像这样，你不断对文件进行修改，然后不断提交修改到版本库里，就好比玩RPG游戏时，每通过一关就会自动把游戏状态存盘，如果某一关没过去，你还可以选择读取前一关的状态。有些时候，在打Boss之前，你会手动存盘，以便万一打Boss失败了，可以从最近的地方重新开始。Git也是一样，每当你觉得文件修改到一定程度的时候，就可以“保存一个快照”，这个快照在Git中被称为commit。一旦你把文件改乱了，或者误删了文件，还可以从最近的一个commit恢复，然后继续工作，而不是把几个月的工作成果全部丢失。

现在 我们回顾一下hello.txt文件一共有几个版本被提交到Git仓库里了。

推荐阅读

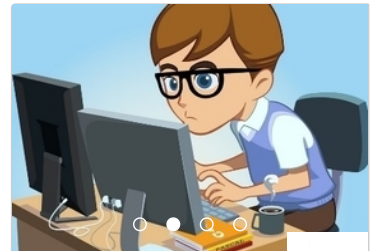
20200819 行人重识别算法比赛
阅读 512

原地哈希算法
阅读 1,027

docker安装和启动
阅读 91

将博客搬至CSDN
阅读 236

CatBoost算法 & XGBoost算法 &
Light GBM算法
阅读 21



程序员要学多久

写下你的评论...

评论7

赞54

...

Git和GitHub使用教程



MachinePlay

关注

赞赏支持

版本2：在hellotxt添加了几句话

到了第二天早上你后悔了，想回到昨天下班前代码的状态

当然了，在实际工作中，我们脑子里怎么可能记得一个几千行的文件每次都改了什么内容，不然要版本控制系统干什么。版本控制系统肯定有某个命令可以告诉我们历史记录，在Git中，我们用git log命令查看：

输入 `git log` 命令查看版本情况

输入：

```
1 | git log
```

显示了过去所有的修改时间、修改人、修改内容：

```
1 | commit 88d885c21216cbdacb1692e08d51afa6d4e32a7 (HEAD -> master)
2 | Author: yepdlpc <mattbaisteins@gmail.com>
3 | Date: Wed Dec 19 20:13:22 2018 +0800
4 |
5 | 在hello.txt添加了一句话
6 |
7 | commit ec4652d5d0b8662fc8730d64b42341d1c363a442
8 | Author: yepdlpc <mattbaisteins@gmail.com>
9 | Date: Wed Dec 19 20:11:42 2018 +0800
10 |
11 | 添加了hello.txt
```

yepdlpc和邮箱都是我的。。我们可以看到，最新的提交在最上面，并按时间有近到远。并且每一次提交修改都生成了一个commit id，我们可以认为这个id是当时的这个版本的版本号，这个commit id是我们找回当时版本的 **唯一凭据**。

我们只有两次提交，只需要回到 `ec645.....` 这个commit id时的版本就行。

Git中使用HEAD表示当前版本，也就是 `commit 88d885c21216cbdacb1692e08d51afa6d4e32a7`，

`HEAD^` 表示上一个版本，`HEAD^^` 表示上上一个版本，当然往上100个版本写100个`^` 比较容易数不过来，所以写成 `HEAD~100`。

现在，我们要把当前版本回退到上一个版本，就可以使用 `git reset` 命令：

```
1 | git reset --hard HEAD^
```

`--hard` 参数有啥意义？这个后面再讲，现在你先放心使用。

此时我们可以看到hello.txt的文件内容变回了：

```
1 | print("Hello World")
```

果然被还原了。

还可以继续回退到上一个版本，不过且慢，然我们用 `git log` 再看看现在版本库的状态：

```
1 | Author: yepdlpc <mattbaisteins@gmail.com>
2 | Date: Wed Dec 19 20:11:42 2018 +0800
```

推荐阅读

20200819 行人重识别算法比赛

阅读 512

原地哈希算法

阅读 1,027

docker安装和启动

阅读 91

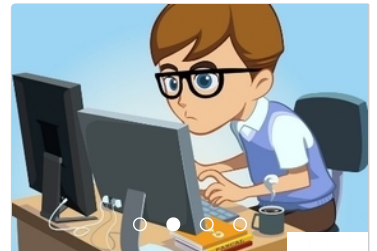
将博客搬至CSDN

阅读 236

CatBoost算法 & XGBoost算法 &

Light GBM算法

阅读 21



程序员要学多久

写下你的评论...

评论7

赞54

...

Git和GitHub使用教程

穿梭机来到了19世纪，想再回去已经回不去了，肿么办？

办法其实还是有的，只要上面的命令行窗口还没有被关掉，你就可以顺着往上找啊找啊，找到那个commit id是 88d885c21216cbdacb1692e08d51afa6d4e32a7，于是就可以指定回到未来的某个版本：

```
1 | git reset --hard 88d885
```

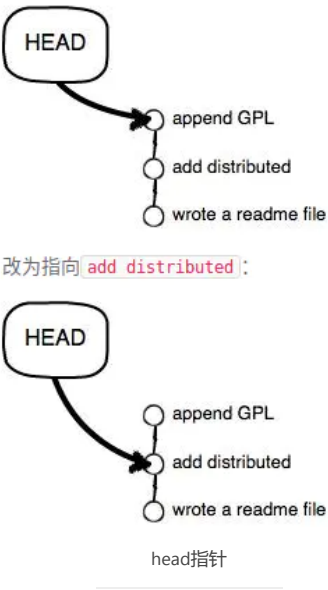
版本号没必要写全，前几位就可以了，Git会自动去找。当然也不能只写前一两位，因为Git可能会找到多个版本号，就无法确定是哪一个了。

我们再看hello.txt的内容：

```
1 | print("Hello World!")
2 | print("老板是神经病")
```

果然，我胡汉三又回来了。

Git的版本回退速度非常快，因为Git在内部有个指向当前版本的HEAD指针，当你回退版本的时候，Git仅仅是把HEAD从指向某个版本：



现在，你回退到了某个版本，关掉了电脑，第二天早上就后悔了，想恢复到新版本怎么办？找不到新版本的 commit id 怎么办？

在Git中，总是有后悔药可以吃的。当你用 `$ git reset --hard HEAD^` 回退到旧版本时，再想恢复到新版本，就必须找到新版本的的commit id。

Git提供了一个命令 `git reflog` 用来记录你的每一次命令：

```
1 | $ git reflog
2 | 88d885c (HEAD -> master) HEAD@{0}: reset: moving to 88d885c21216cbdacb1692e08d51afa6d4e32a7
3 | ec4652d HEAD@{1}: reset: moving to HEAD^
4 | 88d885c (HEAD -> master) HEAD@{2}: commit: 在hello.txt添加了一句话
5 | ec4652d HEAD@{3}: commit (initial): 添加了hello.txt
```

推荐阅读

20200819 行人重识别算法比赛
阅读 512

原地哈希算法
阅读 1,027

docker安装和启动
阅读 91

将博客搬至CSDN
阅读 236

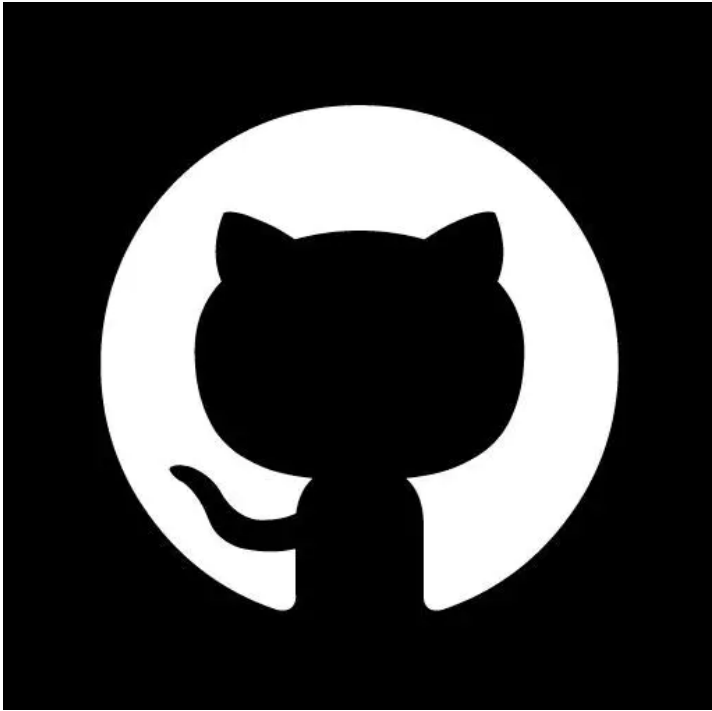
CatBoost算法 & XGBoost算法 &
Light GBM算法
阅读 21



程序员要学多久

写下你的评论...

评论7 赞54 ...



github

推荐阅读

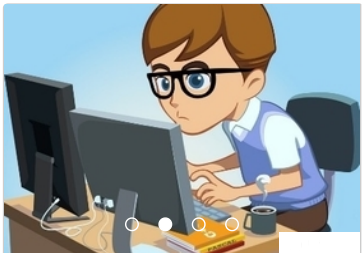
20200819 行人重识别算法比赛
阅读 512

原地哈希算法
阅读 1,027

docker安装和启动
阅读 91

将博客搬至CSDN
阅读 236

CatBoost算法 & XGBoost算法 &
Light GBM算法
阅读 21



程序员要学多久

为什么用GitHub?

学会使用Git后我们的团队合作能力大大提升，我们可以设置一台代码仓库服务器，本地提交完成（commit）后将代码与仓库同步，就能实现分布式版本控制了。

然而，为一个小项目单独设立仓库服务器成本有些高，另外，自己搭建的物理主机服务器难免会有故障，我们也很难保障网络（独立ip很贵），于是，大胆的想法就出现了！

没错！就是上云啦！GitHub就是这样一家提供免费代码仓库的公司，它是由Git之父、linux之父linus成立的（git其实是linus为了维护linux开发的）为广大程序员提供远程代码仓库的伟大公司
~~

前不久被微软收购了。

页面大概是这样：



Git和GitHub使用教程

每个用户有一个类似博客的主页，和很多仓库（一个项目一个仓库），程序员把自己写的代码上传到GitHub来托管，因为是大公司在管，你不必担心它的服务器会挂掉，在全球任何有网络的地方你都可以开始工作！
没错，很多公司也使用GitHub管理他们的项目

推荐阅读

20200819 行人重识别算法比赛
阅读 512

原地哈希算法
阅读 1,027

docker安装和启动
阅读 91

将博客搬至CSDN
阅读 236

CatBoost算法 & XGBoost算法 &
Light GBM算法
阅读 21



google.png

大概有三种用途：

1.合作开发

如果有多个人一起开发某项目，把仓库设在GitHub，大家在各自笔记本写代码、修改使用git上传、同步，避免了复制粘贴代码，而且还能实现版本控制，谁修改了什么内容一清二楚，连谁写了多少行改了多少行代码都能统计清楚。摸鱼是不可能的。

2.软件仓库

GitHub提供两种仓库私有仓库和公开仓库。

GitHub免费为所有用户提供公开仓库空间，公开仓库向网络公开，所有人都能访问，但只有所有者和授权用户才能修改。

私有仓库不对外公开，但要向GitHub付费。

3.代码公开

这一条应该是最有用的了，你可以在这里找到全世界程序员的劳动成果，上到各种算法实现、下到各种app源码。只要遵循开源协议，你都可以copy下来用。

使用GitHub

第一步：注册账号[GitHub官网](#)



Git和GitHub使用教程



MachinePlay

关注

赞赏支持

```
1 | ssh-keygen -t rsa -C "youremail@example.com"
```

你需要把邮件地址换成你自己的邮件地址，然后一路回车，使用默认值即可，由于这个Key也不是用于军事目的，所以也无需设置密码。

如果一切顺利的话，可以在用户主目录里找到.ssh目录，里面有 `id_rsa` 和 `id_rsa.pub` 两个文件，这两个就是SSH Key的密钥对，`id_rsa` 是私钥，不能泄露出去，`id_rsa.pub` 是公钥，可以放心地告诉任何人。

打开id_rsa.pub,复制里面的内容。

第三步，登陆GitHub，打开“Account settings”，“SSH Keys and GPG keys”页面：

然后，点“New SSH Key”，填上任意Title，在Key文本框里粘贴id_rsa.pub文件的内容即可。

推荐阅读

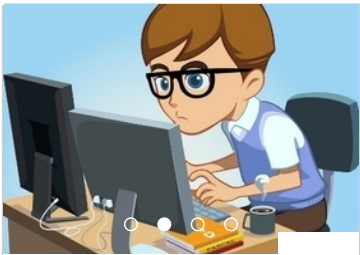
20200819 行人重识别算法比赛
阅读 512

原地哈希算法
阅读 1,027

docker安装和启动
阅读 91

将博客搬至CSDN
阅读 236

CatBoost算法 & XGBoost算法 & Light GBM算法
阅读 21



程序员要学多久

添加sshkey

点“Add Key”，你就应该看到已经添加的Key。

为什么GitHub需要SSH Key呢？因为GitHub需要识别出你推送的提交确实是你推送的，而不是别人冒充的，而Git支持SSH协议，所以，GitHub只要知道了你的公钥，就可以确认只有你自己才能推送。

当然，GitHub允许你添加多个Key。假定你有若干电脑，你一会儿在公司提交，一会儿在家里提交，只要把每台电脑的Key都添加到GitHub，就可以在每台电脑上往GitHub推送了。

最后友情提示，在GitHub上免费托管的Git仓库，任何人都可以看到喔（但只有你自己才能改）。所以，不要把敏感信息放进去。

创建仓库

然后你可以创建仓库了

首先，登陆GitHub，然后，在右上角找到“Create a new repo”按钮，创建一个新的仓库



写下你的评论...

评论7

赞54



Git和GitHub使用教程

系统启动 | 教程 | 项目 | 资源



MachinePlay

关注

赞赏支持

推荐阅读

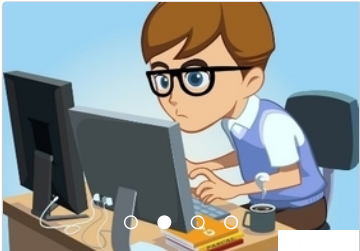
20200819 行人重识别算法比赛
阅读 512

原地哈希算法
阅读 1,027

docker安装和启动
阅读 91

将博客搬至CSDN
阅读 236

CatBoost算法 & XGBoost算法 &
Light GBM算法
阅读 21



程序员要学多久

test.png

来到这个项目的主页

主页

可以看到主页提供了两种和本地仓库关联起来的方式
方法一：
把本地已有的同名Git仓库和GitHub上的仓库关联起来
我们在本地新建了一个名为Gittest的文件夹

1 | git init

写下你的评论...

评论7

赞54

...

Git和GitHub使用教程



MachinePlay

关注

赞赏支持

```
1 | git add hello.txt
2 | git commit -m "first commit"
```

到此为止已经提交到了本地仓库

接下来我们把本地仓库和远程仓库联系起来

```
1 | git remote add origin git@github.com:MachinePlay/Gittest.git
```

请千万注意，把上面的MachinePlay替换成你自己的GitHub账户名，否则，你在本地关联的就是我的远程库，关联没有问题，但是你以后推送是推不上去的，因为你的SSH Key公钥不在我的账户列表中。

添加后，远程库的名字就是origin，这是Git默认的叫法，也可以改成别的，但是origin这个名字一看就知道是远程库

下一步，就可以把本地库的所有内容推送到远程库上

```
1 | git push -u origin master
```

由于远程库是空的，我们第一次推送master分支时，加上了-u参数，Git不但会把本地的master分支内容推送的远程新的master分支，还会把本地的master分支和远程的master分支关联起来，在以后的推送或者拉取时就可以简化命令。

此后 `git add` `git commit-m` 之后 就可以使用

```
1 | git push origin master
```

就可以把自己的代码上传到远程仓库了。

方法二：

使用 `Git clone` 直接从远程仓库克隆下来

前面我们讲了先有本地库，后有远程库的时候，如何关联远程库。

现在，假设我们从零开发，那么最好的方式是先创建远程库，然后，从远程库克隆。

首先，登陆GitHub，创建一个新的仓库，名字叫Gittest

每个仓库都有一个地址：

推荐阅读

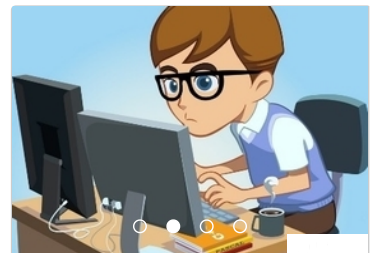
20200819 行人重识别算法比赛
阅读 512

原地哈希算法
阅读 1,027

docker安装和启动
阅读 91

将博客搬至CSDN
阅读 236

CatBoost算法 & XGBoost算法 &
Light GBM算法
阅读 21



程序员要学多久

写下你的评论...

评论7

赞54

...

Git和GitHub使用教程

MachinePlay

关注

赞赏支持

我们在本地使用 `git clone` 直接把远程仓库克隆下来。

```
1 | git clone git@github.com:MachinePlay/Gittest.git
```

看到这里大家就懂了，不仅仅是自己的仓库可以克隆，也可以克隆其他用户的公开仓库。

例如现在随便找一个安卓阅读app的仓库

推荐阅读

- 20200819 行人重识别算法比赛
阅读 512
- 原地哈希算法
阅读 1,027
- docker安装和启动
阅读 91
- 将博客搬至CSDN
阅读 236
- CatBoost算法 & XGBoost算法 & Light GBM算法
阅读 21



image.png

```
1 | git clone git@github.com:smuyyh/BookReader.git
```

就把这位用户仓库里一个完整的阅读app源码全部下载到本地了。

至于有了全部源码如何使用？可以学习别人的code结构、或者找成熟的中间模块使用。

```
1 | 有点晚了，我过段时间继续更新使用Github合作开发内容
```

54人点赞>



闲杂技术



"喜欢就赞赏我吧！"

赞赏支持

 共2人赞赏



MachinePlay 一只R&D社畜，兴趣驱动，学习、分享、沉淀。筚路蓝缕，以启...
总资产36 (约2.57元) 共写了21.7W字 获得301个赞 共174个粉丝

关注



Git和GitHub使用教程



MachinePlay

关注

赞赏支持



被以下专题收入，发现更多相似内容



Git



工具使用



开发工具分享

推荐阅读

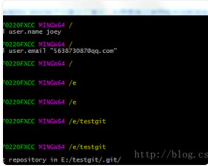
更多精彩内容>

git的安装与基本使用方法

1.git的安装 1.1 在Windows上安装Git msysgit是Windows版的Git，从https:/...



落魂... 阅读 8,359 评论 4 赞 49



【转】使用git和github管理自己的项目---基础操作学习

本文转载自：使用git和github管理自己的项目---基础操作学习，进行了重新排版。二次转载请注明原作出处。我...



mac在路上 阅读 1,496 评论 0 赞 23

夹克的童话|中国单身过两亿了，你还不会偶遇吗？

安德鲁拖着疲惫的身体走出会场，经过三天的谈判，他终于拿下了这个案子，无疑，这个案子会让他升职加薪。虽然早上要乘早班...



韩小日 阅读 151 评论 0 赞 4



她不停买买买，只因为心里缺失幸福感

马云大咖设立双十一购物节后，他身后的女人们深陷买买买的魔咒不能自拔。网上甚至出了老婆等老公睡着后，偷偷牵着老公的手...



夕颜悦风 阅读 293 评论 3 赞 2



爱

爱是不变的星辰，爱是一世的牵挂！



田园闲客 阅读 46 评论 0 赞 0

推荐阅读

20200819 行人重识别算法比赛

阅读 512

原地哈希算法

阅读 1,027

docker安装和启动

阅读 91

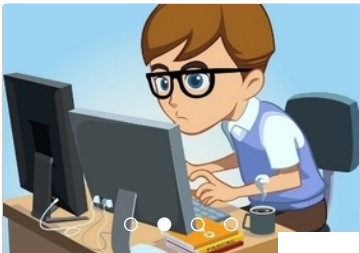
将博客搬至CSDN

阅读 236

CatBoost算法 & XGBoost算法 &

Light GBM算法

阅读 21



程序员要学多久



写下你的评论...



评论7



赞54

...