一、Git基础-安装并配置Git

配置用户信息

用户名：baokw

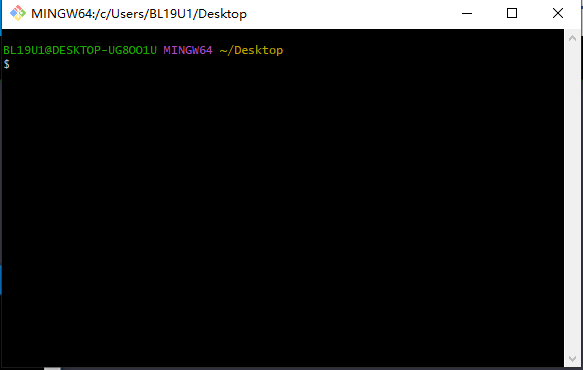
邮件地址：baokw@sari.ac.cn

1.git config --global user.name “itheima”

2.git config --global user.email [itheima@itcast.cn](mailto:itheima@itcast.cn)

注意：如果使用了 –global选项，那么该命令只需要运行一次，即可永久生效。

1.打开git窗口



2.输入命令

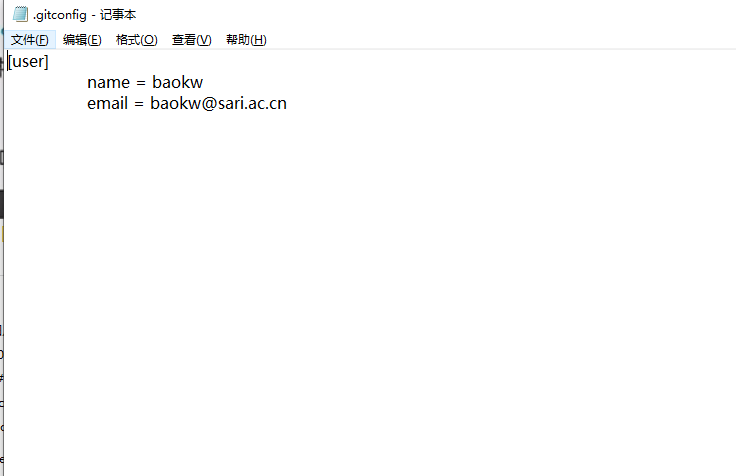
git config --global user.name “baokw”

git config --global user.email “baokw@sari.ac.cn”

二、Git的全局配置文件

通过git config –global user.name和git config –global user.email配置的用户名和邮箱地址，会被写入到C：/Users/用户名文件夹/.gitconfig文件中。这个文件是Git的全局配置文件，配置一次即可永久生效。

可以使用记事本打开此文件夹，从而查看自己曾经对Git做了哪些全局性的配置。



三、检查配置信息

除了使用记事本查看全局的配置信息之外，还可以运行如下的终端命令，快速的查看Git的全局配置信息：

1. #查看所有的全局配置项

2. git config –list –global

3. # 查看指定的全局配置项

4. git config user.name

5. git config user.email

四、获取帮助信息

可以使用git help<verb>命令，无需联网即可在浏览器中打开帮助手册，例如

1.#要想打开git config 命令的帮助手册

2.git help config

如果不想查看完整的手册，那么可以用-h选项获得更简明的“help” 输出:

1.#想要获取git config 命令的快速参考

2.git config -h

五、Git的基本操作

1.获取Git仓库的两种方式

* 将尚未进行版本控制的本地目录转换为Git仓库
* 从其他服务器克隆一个已存在的Git仓库

以上两种方式都能够在自己的电脑上得到一个可用的Git仓库

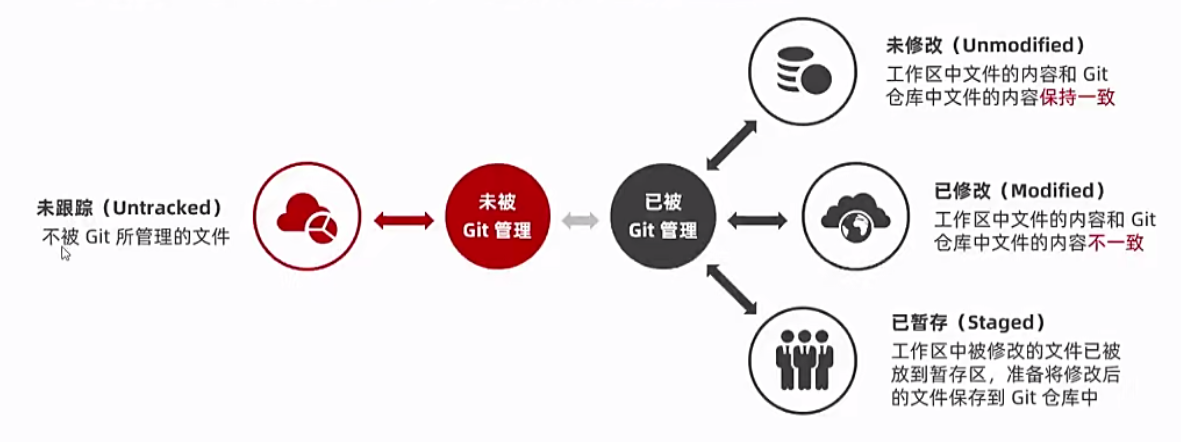
2.在现有目录中初始化仓库

如果自己有一个尚未进行版本控制的项目目录，想要用Git来控制它，需要执行如下两个步骤：

* 在项目目录中，通过鼠标右键打开“Git Bash”
* 执行git init 命令将当前的目录转化为Git仓库

git init命令会常见一个名为.git的隐藏目录，这个.git目录就是当前项目的Git仓库，里面包含了初始的必要文件，这些文件是Git仓库的必要组成部分。

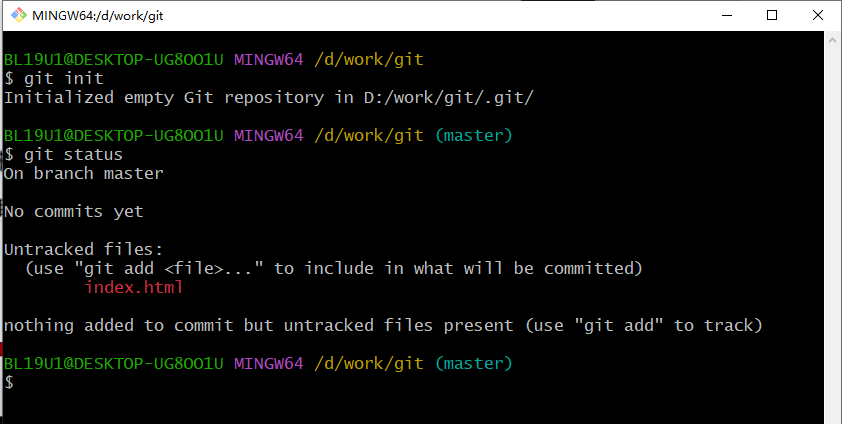
3.工作区中文件的4种状态



Git操作的终极结果：让工作区中的文件都处于“未修改”的状态。

4.检查文件的状态

可以使用git status命令查看文件处于什么状态，例如：



在状态报告中可以看到新建的index.html文件出现在Untracked files（未跟踪的文件）下面。

未跟踪的文件意味着Git在之前的快照（提交）中没有这些文件；Git不会自动将之纳入跟踪范围，除非明确地告诉它“我需要使用Git跟踪管理该文件”。

5.以精简的方式显示文件状态

使用git status输出的状态报告很详细，但有些繁琐。如果希望以精简的方式显示文件的状态，可以使用如下两条完成等价的命令，其中-s是—shrot的简写形式：

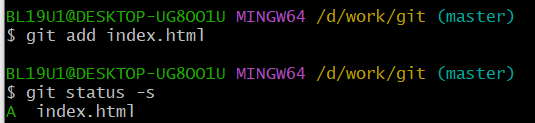


未跟踪文件前面有红色的??标记，例如:

6.跟踪新文件

使用命令git add开始跟踪一个文件。所以，要跟踪index.html文件，运行如下命令即可：

* git add index.html



此时再运行git status命令，会看到index.html文件在Changes to be committed这行的下面，说明已被跟踪，并处于暂存状态：



以精简的方式显示文件的状态:

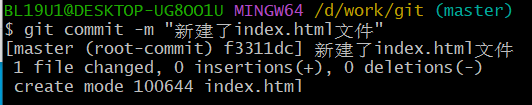
新添加到暂存区中的文件前面有绿色的A标记

7.提交更新

现在暂存区中有一个index.html文件等待被提交到Git仓库中进行保存。可以执行git commit命令进行提交，其中-m选项后面是本次的提交消息，用来对提交的内容做进一步的描述。

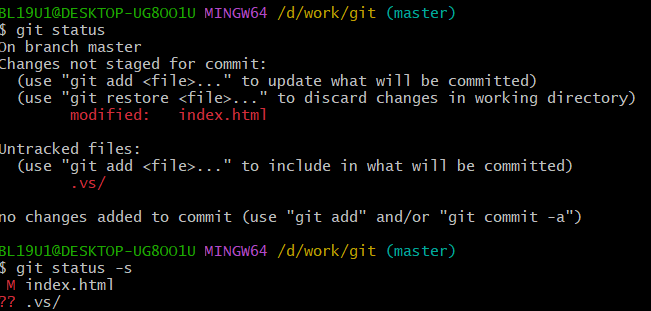
* git commit -m“新建了index.html文件”

提交成功之后，会显示如下消息



8.对已提交的文件进行修改

目前，index.html文件已经被Git跟踪，并且工作区和Git仓库中的index.html文件内容保持一致。当我们修改了工作区中index.html的内容之后，再次运行git status 和git status -s命令，会看到如下内容：



文件index.html出现在Changes not staged for commit这行下面，说明已跟踪文件的内容发生了变化，但还没有放到暂存区。

注意：修改过的、没有放入暂存区的文件前面有红色的M标记。

9.暂存已修改的文件

目前，工作区中的index.html文件已被修改，如果要暂存这次修改，需要再次运行git add命令，这个命令是个多功能的命令，主要有如下3个功效：

* 可以用它开始跟踪新文件
* 把已跟踪的、且已修改的文件放到暂存区git add index.html
* 把有冲突的文件标记为已解决状态

10.提交已暂存的文件