**Git**

git clone +地址 克隆

git status 查看状态

git add \_ 将文件提交到暂存区

git commit -m “注释：说明”

git push origin master 推送

git pull origin master 拉取(保证和远程的版本一致)

**推送服务器:**

1. 前提(本地与服务器创建联系) git remote add origin 地址

创建分支：

git branch dev

查看分支：

git branch

切换分支：

git checkout branch

切换主分支：

git checkout master

合并分支：

git merage \_\_\_\_

**表示没有修改内容（内容没有改变）**

$ git status

On branch master

Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean

**引用**

长引用 <blockquote><blockquote>

短饮用 <q></q>

引用标题 <cite> 《》</cite> 引用部分部分内容斜体

标注文字 <mark> </mark> 标注部分文字背景颜色为黄色

文字方向 <bdo dir=”rtl”></bdo> right to left

删除线 <del>原件</del> 原件删除线

详细信息 <details></details> 点击显示内容(目前只有谷歌浏览器支持)



<details>

<summary>标题信息</summary>

<p>被显示和隐藏的内容</p>

</details>

进度条标签

<meter min="0" max="100" low="30" optimum="50" high="90" value="60"></meter>

<progress></progress>

**Github**

1. 创建一个新的秘钥

ssh-keygen -t rse -C “邮箱”

**正则**

**new RegExp(表达式，模式修正符)**

使用 **//** 定义正则表达式

\d 匹配任意一个0-9之间的数字

\D 匹配任意一个除了0-9之间的数字

\W(大写) 匹配任意一个除了字母、下划线、数字以外的字符

\w 匹配任意一个字母、下划线、数字中的字符

\s(小写) 匹配任意一个空白符

\S(大写) 匹配任意一个除了空白符之外的字符

. 匹配除了换行符意外的任意一个字符

？ 修饰其前原子的数量为1个或者0个

+ 修饰其前原子出现1个或者0个

\* 修饰其前原子出现0次或者1次

/a?b/ 相当于[ab | b]

^ 抑扬符 表示字符串开始的位置

$ 表示字符串结束的位置

**元字符：**

{ n } 修饰其前原子恰好出现n次

{ n , } 修饰其前原子至少出现n次

{ n, m } 修饰其前原子至少出现n次，最多m次

\b 表示单词的边界位置

\B 表示单词的非边界位置