# 一、Node

## 1、环境配置

|  |  |
| --- | --- |
| NODE\_PATH | .; |
| Path | F:\Node\node\node\_global;F:\Node\node; |

# 二、NPM

## 1.环境配置

|  |  |
| --- | --- |
| 设置全局模块 | npm config set prefix "E:\node\node\_globals" |
| 设置加载缓存 | npm config set cache "E:\node\node\_cache" |

## 2.常用命令

|  |  |
| --- | --- |
| npm init | 会引导你创建一个package.json文件 |
| npm list | 查看当前目录下已安装的node包 |
| npm list –g | 查看所有全局安装的模块 |
| npm ls | 查看当前目录下已安装的node包 |
| npm ls -g | 查看所有全局安装的模块 |
| npm remove <package> | 移除 |
| npm help | 查看帮助命令 |
| npm help <command> | 查看某个命令的帮助文档 |
| npm install <package> | 本地模块安装 |
| npm install <package>@<version> | 本地安装指定版本的模块 |
| npm install <package> -g | 全局安装模块 |
| npm install <package>@<version> -g | 全局安装指定版本模块 |
| npm install <package> --save | 安装的同时，将信息写入package.json中 |
| npm uninstall <package> | 卸载本地模块 |
| npm cache clear | 清空NPM本地缓存 |
| npm search <package> | 搜索模块 |
| npm update <package> | 更新模块 |
| npm update <package> -g | 把全局安装的对应命令行程序更新至最新版 |
| npm publish | 发布模块 |
| npm unpublish <package>@<version> | 撤销发布自己发布过的某个版本代码 |
| npm view <package> | 查看node模块的package.json文件夹 |
| npm help folder | 查看npm使用的所有文件夹 |
| npm view <package> dependencies | 查看包的依赖关系 |
| npm view <package> repository.url | 查看包的源文件地址 |
| npm root | 查看当前包的安装路径 |
| npm root -g | 查看全局的包的安装路径 |
| npm –v | 查看当前npm版本号 |

# 三、Bower

## 1.环境搭建

①设置项目模块目录

新建一个文件 .bowerrc并写入以下内容：

{

"directory" : "js/lib"

}

## 2.常用命令

|  |  |
| --- | --- |
| bower init | 生成bower.json |
| bower list | 查看已安装的模块 |
| bower help | 查看帮助文档 |
| bower install package --save | 安装包 |
| bower install package#version --save | 安装指定版本的包 |
| bower info package | 查看包信息 |
| bower update package | 更新包 |
| bower search package | 查找包 |
| bower uninstall package | 卸载包 |
| bower cache list | 列出所有缓存文件 |
| bower cache list package | 列出指定包缓存 |
| bower install package --offline | 离线安装本地缓存包 |
| bower cache clean | 清除所有缓存 |
| bower cache clean package | 清除指定包缓存 |
| bower cache clean package#version | 清楚指定版本包缓存 |

# 四、Git

## 1.环境搭建

正常安装后请配置path路径

## 2.常用命令

|  |  |
| --- | --- |
| mkdir dirName | 创建一个文件夹 |
| cd dirName | 切换目录 |
| pwd | 显示当前目录 |
| ls –ah | 查看目录文件 |
| cat <file> | 把文件内容打印到屏幕 |
| cat > <file> | 向文件写入内容，按 Ctrl + D 结束输入 |
| git init | 把目录变成Git可以管理的仓库 |
| git add <file> | 反复多次使用，添加多个文件 |
| git rm <file> | 移除版本库中的文件 |
| git commit | 提交内容 |
| git commit - -m “comment” | 提交内容并添加注释 |
| git status | 显示仓库当前的状态，是否有文件被修改 |
| git diff | 查看修改内容 |
| git diff HEAD -- <file> | 对比版本库最新文件与本地文件 |
| git log | 显示提交日志 |
| git log –pretty=oneline | 单行显示提交日志 |
| git reset - -hard HEAD^ | 回退到上一个版本 |
| git reset - -hard HEAD^^ | 回退到上上一个版本 |
| git reset - -hard id | 回退到指定的id版本（git log可以查看） |
| git reflog | 查看命令历史，以便确定要回到未来的哪个版本 |
| git checkout -- file | 检出文件并覆盖 |
| ssh-keygen –t rsa -C "mail@domain" | SSH key生成，在用户文件夹的.ssh下面。 |
| git remote add origin < path > | 关联远程库  path：git@github.com:MangoNiki/mangojs.git |
| git remote add origin < gitPath > | gitPath：git@server-name:path/repo-name.git |
| git push -u origin master | 把本地库的所有内容推送到远程库上 |
| git push origin master | 推送最新修改 |
| git branch | 查看分支 |
| git branch <name> | 创建分支 |
| git checkout <name> | 切换分支 |
| git checkout -b <name> | 创建 + 新建分支 |
| git merge <name> | 合并分支到当前分支 |
| git branch -d <name> | 删除分支 |
|  |  |

