[NPM笔记 1](#_Toc513884487)

[Nginx笔记 2](#_Toc513884488)

[Gulp笔记 3](#_Toc513884489)

[Linux笔记 4](#_Toc513884490)

[Git笔记 8](#_Toc513884491)

[Bower笔记 9](#_Toc513884492)

# NPM笔记

npm config get prefix 查看npm全局仓库路径

npm config set prefix 设置npm全局仓库路径

npm view jquery versions npm查看模块全部版本：(jquery)

npm install gulp-sourcemaps --save 安装gulp-sourcemaps模块并写入package.json

package.json版本(^和~区别)

波浪符号（~）：他会更新到当前minor version（也就是中间的那位数字）中最新的版本

插入符号（^）：将会把当前库的版本更新到当前major version（也就是第一位数字）中最新的版本

# Nginx笔记

**CentOS 7上安装Nginx**

1.添加CentOS 7 EPEL仓库

sudo yum install epel-release

2.安装Nginx

sudo yum install nginx

3.启动Nginx

sudo systemctl start nginx

4.如果您正在运行防火墙，请运行以下命令以允许HTTP和HTTPS通信：

sudo firewall-cmd --permanent --zone=public --add-service=http

sudo firewall-cmd --permanent --zone=public --add-service=https

sudo firewall-cmd --reload

5.如果想在系统启动时启用Nginx。请输入以下命令：

sudo systemctl enable nginx

★ nginx 更改配置文件后需要重启，重启命令

nginx –s reload **→**重启命令

nginx –t **→**检查配置文件是否正确

# Gulp笔记

gulp.src(globs[, options])

★ gulp.src (['./src/\*\*', '**!**\*\*/node\_modules/\*\*']) **→**感叹号表示不包括第二个参数代表的内容

★ gulp.src ('./src/\*\*/\*.js', {base:'./src/'}) **→**base作用：'./src/\*\*/\*.js'路径去除'./src/'得到的路径加到gulp.dest里的路径后面得出最终的目的路径

# Linux笔记

systemctl poweroff 系统关机

systemctl reboot 重新启动

systemctl suspend 进入暂停模式

systemctl hibernate 进入休眠模式

systemctl rescue 强制进入救援模式

systemctl emergency 强制进入紧急救援模式

systemctl set-default graphical.target

rpm -q < package name> **→**查询一个包是否被安装

rpm -qa | sort **→**查询系统安装的所有软件包

rpm -ql < package name> **→**列出软件包安装的文件**和位置**

startx

init 5

进入桌面环境

cd .. **→**返回上级目录

.表示当前目录；..表示上级目录；./aaa/ 表明的就是当前目录下面的aaa目录

rm -rf /home/test **→**删除test目录下所有文件，包括test目录

rm -rf /home/test/\* **→**删除test目录下所有文件，**保留**test目录

-f：强制删除; -r：递归的删除目录下面的文件以及子目录下文件

rm –rf !(a|b) **→**删除除a和b外所有文件，必须在当前目录下执行

如果提示“-bash: !: event not found ” shopt -s extglob 来解决

find a/\* [-not] –name b文件名 | xargs rm –rf

**→**可选参数[-not]存在时，删除a文件下所有除b外的所有其他文件(b可为文件夹名，若b为文件夹时，b底下的文件也会被删除)

find /usr/sam/\* -path /usr/sam/dir1 -prune -o –print

**→**在/usr/sam目录下查找不在dir1子目录之内的所有文件

mkdir 文件夹名称 **→**新建文件夹

vi 文件名称 **→**修改文件，按i键进入编辑模式，按esc退出编辑模式；在浏览模式中输入:qw 保存退出，:q直接退出。

vi 新文件名 **→**创建文本文件

vncserver 启动vncserver服务

vncpasswd 修改vnc远程登录密码

systemctl list-unit-files | grep vnc

查看vnc服务

Systemctl enable vncserver@serivce 启用服务

sudo 以系统管理者的身份执行指令

netstat –lntp

<https://bbs.aliyun.com/read/308745.html?utm_campaign=wenzhang&utm_medium=article&utm_source=QQ-qun&utm_content=m_12510>

http://blog.csdn.net/leveretz/article/details/75210101

bash环境下通配符：



bash环境下特殊符号：







 管线命令仅会处理 standard output，对于 standard error output 会予以忽略

 管线命令必须要能够接受来自前一个指令的数据成为 standard input 继续处理才行。

仅观察自己的 bash 相关进程： ps –l

观察系统所有进程： ps aux

# Git笔记

git config --global http.sslVerify false 设置全局变量http.sslVerify为false

从远程仓库更新文件到本地 pull ,需指明Remote Branch

git pull = git fetch + git merge



# Bower笔记

bower install bootstrap –save

安装twitter bootstrap并更新bower.json文件

bower install jquery --save 添加到dependencies

bower install angular --save-dev 添加到devDependencies

{

"name":"", //必须，如果需要注册包，则该包名唯一。

"description":"", //可选，包描述

"main":[], //可选，入口文件，bower本身不使用，供第三方构建工具会使用,每种文件类型只能有一个。

"ignore":[], //可选，文件或目录列表。bower安装的时候将忽略该列表中的文件,bower是从git仓库或压缩包下载一个包，里面的文件并不一定全部需要。

"dependencies":[], //依赖包，name:value，value可以是包的semver range(版本号范围)，也可以直接是一个包的git地址或压缩包地址。

"devDependencies":[], //开发依赖包，仅仅在开发过程中测试或者编译文档用，部署生产环境是不需要。

"resolutions":[], //包引用冲突自动使用该模块指定的包版本,格式和dependencies 相同

"overrides" :[ //这个也很关键，可以覆盖一个包中的默认设置，比如main里面设定的入口文件

"package-name":{ //这样可以根据需要，让第三方工具只打包需要的文件。

"main":[]

}

],

"moduleType":"", //可选，指定包采用那种模块化方式(globals,amd,node,es6,yui)

"private":Boolean, //是否公开发布当前包,如果只是使用bower来管理项目的包，设置为true.

"license":"", //授权方式(GPL-3.0,CC-BY-4.0.....)

"keywords":[], //可选，方便注册后容易被其他人搜索到。

"authors":[], //作者列表

"homepage":[], //主页，包介绍页

"repository":{ //包所在仓库。

"type": "git",

"url": "git://github.com/foo/bar.git"

},

｝