1. commonJS：
   1. 导入：const module = require('url')
      1. 先从内置模块找，找不到再从node\_modules中找
   2. 导出: module.exports a
2. 安装
   1. npm install(i) <模块名字>
   2. npm uninstall
   3. npm update
3. 项目描述文件（package.json）: npm init
4. dependencies 依赖模块
5. 拿到项目先npm i

### 一些命令行参数的缩写方式：

**-g**

--global，缩写为-g，表示安装包时，视作全局的包。安装之后的包将位于系统预设的目录之下，一般来说

**-S**

--save，缩写为-S，表示安装的包将写入package.json里面的dependencies。

**-D**

--save-dev，缩写为-D，表示将安装的包将写入packege.json里面的devDependencies。

**-O**

--save-optional缩写为-O，表示将安装的包将写入packege.json里面的optionalDependencies。

**-E**

--save-exact缩写为-E，表示安装的包的版本是精确指定的。

**-B**

--save-bundle缩写为-B，表示将安装的包将写入packege.json里面的bundleDependencies。

使用淘宝 NPM 镜像

* 1. pm install -g cnpm --registry=https://registry.npm.taobao.org
  2. cnpm install [name]

### express

1. 路由

const express = *require*('express')  
  
const app = new express  
app.get('/',function (req, res) {  
 res.send('ok')  
})  
  
app.get('/list',function (req, res) {  
 res.send('list')  
})  
app.get('\*',function (req, res) {  
 res.send('这是一个404页面')  
})  
  
app.listen(412)

1. 中间件

app.use(function (req, res, next) {  
 console.log('处理操作')  
 next()//没有next会一直会卡在这里  
})

### node 升级

1. 先清除 npm cache npm cache clean -f
2. 安装n模块 npm install -g n
3. 然后就可以开始升级了：自动安装到最新的稳定版本命令是 n stable，自由选择版本安装是n 4.4.0（版本号）
4. 备注：升级前后可先使用node -v 查看版本。查看所有node版本是n ls
5. 也可以直接编译安装到最新的版本。

apt-get install gcc g++ make -y && wget https://nodejs.org/dist/v4.4.2/node-v4.4.2.tar.gz && tar zxf node-v4.4.2.tar.gz && cd node-v4.4.2 && ./configure && make && make install

1. 因为版本过低报错

[问题解决](http://kennytu.github.io/blog/2015/12/09/npm-weird-issue/)

1. 查看所有高级的npm moudles

npm list --depth=0

1. 查看所有全局安装的模块

npm list --depth=0 -global

1. npm root：查看当前包的安装路径
2. npm root -g：查看全局的包的安装路径
3. 查看某个模块的版本 npm 模块名 -v