## 常见包管理工具

JavaScript

服务器端 [**JavaScript**](http://lib.csdn.net/base/javascript) 通常使用[NPM](http://npmjs.org/)作为依赖管理工具。

* 通过NPM提供的npm命令来进行依赖的下载、升级和移除。
* 通过package.json来定义软件包的元信息、开发依赖（开发或测试需要）、部署依赖（运行时需要）。
* 依赖递归地存储在node\_modules中。
* 依赖在项目之间是隔离的，全局安装（-g）会使它成为命令行工具而不是全局依赖。
* > 递归的依赖下载风格使得NPM的缓存及其重要。缓存位于~/.npm下，这里保存着.tgz格式的包文件。

[**JavaScript**](http://lib.csdn.net/base/javascript) 通常使用 [Grunt](http://gruntjs.com/) 进行构建。

* Grunt通过插件来完成任务，每个插件相当于Makefile的一个命令。
* Grunt任务定义在Gruntfile.js中。
* NPM提供了众多的Grunt插件，当然你也可以手写。
* Grunt任务继承了JavaScript的异步特性。

前端 lib

前端 lib 可以通过 [Bower](http://bower.io/) 来下载。Bower 不仅可以下载已注册的软件包，还可以下载Github Repo，甚至是一个文件的URL。

* 通过bower命令进行依赖管理。
* bower.json定义了软件包的元信息与依赖。
* 依赖所在路径可以在bower.json中进行设置。
* Bower只是一个命令行工具，你需要在正确的路径执行Bower命令。

Bower可以灵活地下载各种依赖，但它的缺点也是明显的：未注册的软件包往往包含冗余的非生产环境的代码，有时甚至需要手动构建。

Java

[Maven](https://maven.apache.org/)是基于项目对象模型(POM)的项目管理工具，通过文本文件来描述项目的版本、URL、License、依赖等信息。

* 通过pom.xml来描述Maven项目。
* Maven是项目管理工具，其功能包括依赖管理、软件构建。
* Maven通过.properties文件和命令行参数读取系统特性。

[Ant](http://ant.apache.org/)是由[**Java**](http://lib.csdn.net/base/javase)写成的编译、[**测试**](http://lib.csdn.net/base/softwaretest)和部署[**Java**](http://lib.csdn.net/base/javase)应用的命令行工具。

* 通过build.xml来描述构建过程。
* build.xml的逻辑类似Makefile，每个target由命令和参数构成。
* 通过命令行参数来读取系统特性。

[Gradle](https://zh.wikipedia.org/wiki/Gradle)是一个基于Apache Ant和Apache Maven概念的项目自动化建构工具。它使用一种基于Groovy的特定领域语言来声明项目设置，而不是传统的XML。

Python

[**Python**](http://lib.csdn.net/base/python)的包管理工具能讲一整天的故事：distribute、setuptools、distutils、easy\_install、pip、distutils2、packaging、eggs、wheels、pyvenv、virtualenv……

这也是我为什么讨厌安装[**Python**](http://lib.csdn.net/base/python)软件的原因，与此同时，旧版本的python2比新的python3更加流行以及两者不兼容也常常给[**Linux**](http://lib.csdn.net/base/linux)包管理造成麻烦。

现在开始讲故事：

1. distutils是python的标准库；
2. setuptools试图完成distutils缺少的特性而开始开发；
3. easy\_install是setuptools的命令行接口，有更多的特性;
4. 在setuptools的开发过程中产生了分歧，于是出现了distribute，它fork自setuptools，并在2013年取得和解并重新merge到setuptools 0.7版本；
5. 相比于上述工具，[pip](http://pypi.python.org/pypi/pip)是一个更加高阶的接口，**pip成为事实上的python软件的安装工具**；
6. **eggs和wheels成为事实上的二进制python软件的安装工具**。

venv和virtualenv则是用来创建python虚拟环境来实现应用隔离的。

* [venv](https://docs.python.org/3/library/venv.html)在python3.3中引入，用于在自己的目录下创建轻量级的虚拟环境，也可以孤立于系统环境。
* [virtualenv](http://virtualenv.readthedocs.org/)则是一个python软件包，用于创建孤立的python环境。

Ruby

Ruby的软件包单元为[RubyGem](https://rubygems.org/)。

* Gem由.gemspec文件描述。
* Gem的构建过程由Rakefile描述。
* [Rake](https://rubygems.org/gems/rake)是Gem的构建工具，它与Make类似，用以完成自动化测试和代码生成。
* [Bundle](http://bundler.io/)则是Ruby的包管理工具，用来跟踪和下载正确版本的Gem。

除非注明，本博客文章均为原创，转载请以链接形式标明本文地址： <http://harttle.com/2015/05/29/pkg-manager.html>