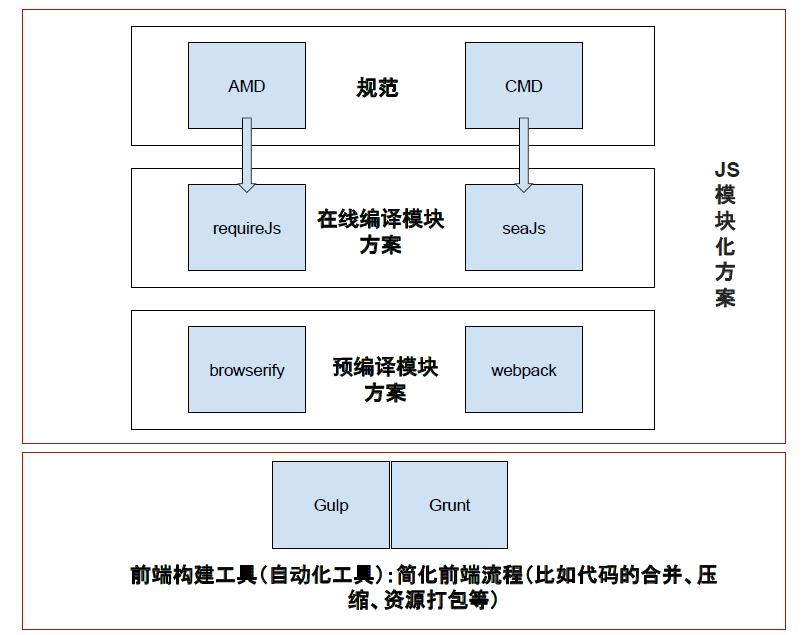
1. glup、grunt、webpack的使用场景，优缺点？

你的工程模块依赖很简单，不需要把js或各种资源打包，只需要简单的合并、压缩，在页面中引用就好了。那就不需要Browserify、Webpack。Gulp就够用了。

webpack是静态资源打包工具，grunt和gulp是自动化构建工具，grunt和gulp二选一就可以，用gulp的比较多，gulp可以和webpack搭配使用。



Gulp / Grunt 是一种工具，能够优化前端工作流程。比如自动刷新页面、combo、压缩css、js、编译less等等。简单来说，就是使用Gulp/Grunt，然后配置你需要的插件，就可以把以前需要手工做的事情让它帮你做了。

browserify / webpack : 是一个预编译模块的方案，相比于上面 ，这个方案更加智能。没用过browserify，这里以webpack为例。首先，它是预编译的，不需要在[浏览器](https://www.baidu.com/s?wd=%E6%B5%8F%E8%A7%88%E5%99%A8&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1dbm19BP1wBP1cknvFWmHDY0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPWD4nWnYn1TkPjbsnj6Ln1nz)中加载解释器。另外，你在本地直接写JS，不管是 AMD / CMD / ES6 风格的[模块化](https://www.baidu.com/s?wd=%E6%A8%A1%E5%9D%97%E5%8C%96&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1dbm19BP1wBP1cknvFWmHDY0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPWD4nWnYn1TkPjbsnj6Ln1nz)，它都能认识，并且编译成[浏览器](https://www.baidu.com/s?wd=%E6%B5%8F%E8%A7%88%E5%99%A8&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1dbm19BP1wBP1cknvFWmHDY0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPWD4nWnYn1TkPjbsnj6Ln1nz)认识的JS。  
这样就知道，Gulp是一个工具，而webpack等等是模块化方案。Gulp也可以配置seajs、requirejs甚至webpack的插件。

Webpack适合spa 如果你的工程庞大，页面中使用了很多库（SPA很容易出现这种情况），那就可以选择某种模块化方案。感觉主要再结合react、vue使用，继续研究

Webpack提供一个小巧的基于express的开发服务器。支持自动刷新、模块热替换。还有代理。具体如何配置在[这里](http://webpack.github.io/docs/webpack-dev-server.html)。

代理（proxy）在开发是还是很有用的。你可以将动态请求映射到后端的开发机，方便联调

**webpack 在设计之初主要还是冲着浏览器端（尤其是 SPA）去的，牛逼在人家对依赖的理解上，也因此才打出 Unversial Bundler + Code Splitting + Load on demand 这种组合技；**如果只是在 server 端用做工具流工具就毫无优势了，强依赖静态分析的做法在灵活性上被 gulp 与 browserify 这种小而美的工具完爆

**webpack 在设计之初主要还是冲着浏览器端（尤其是 SPA）去的，牛逼在人家对依赖的理解上，也因此才打出 Unversial Bundler + Code Splitting + Load on demand 这种组合技；**如果只是在 server 端用做工具流工具就毫无优势了，强依赖静态分析的做法在灵活性上被 gulp 与 browserify 这种小而美的工具完爆

作者：黄玄  
链接：https://www.zhihu.com/question/34460535/answer/73808112  
来源：知乎  
著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。

作者：黄玄  
链接：https://www.zhihu.com/question/34460535/answer/73808112  
来源：知乎  
著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。

Grunt gulp的区别

https://survivejs.com/webpack/appendices/comparison/

<http://m.blog.csdn.net/article/details?id=48834967>

文件级别的精确缓存控制

<http://blog.csdn.net/zhuchunyan_aijia/article/details/54963215>

# glup安装记录

**安装nodejs -> 全局安装gulp -> 项目安装gulp以及gulp插件 -> 配置gulpfile.js -> 运行任务** http://www.ydcss.com/archives/18

**全局安装node**

**全局安装gulp 便于执行任务**

**项目中npm init 生成package.json**

**项目中安装gulp及相关gulp插件**

**创建一个gulpfile.js文件，编写相关配置**

# Glup使用记录

**1.**[前端静态资源版本更新与缓存之——通过gulp 在原html文件上自动化添加js、css版本号](http://blog.csdn.net/zchcode/article/details/52421871)  http://blog.csdn.net/zchcode/article/details/52421871

**Require import**

require是commonjs规范  
import是es6规范, 但是主流实现都不支持(node, chrome), 一般还是要用babel转换  
基本上都是用babel-plugin-transform-es2015-modules-commonjs这个插件转成require形式

## gulp 是什么？

gulp是一个基于流的构建工具，相对其他构件工具来说，更简洁更高效。  
Tip：之前写过一篇[gulp 入门](http://www.jianshu.com/p/9723ca2a2afd)，可以参考下，如果对gulp已经有一定的了解请直接跳过。

## webpack 是什么？

webpack是模块化管理的工具，使用webpack可实现模块按需加载，模块预处理，模块打包等功能。  
Tip：之前写过一篇[webpack 入门](http://www.jianshu.com/p/f890cbfcb11b)，可以参考下，如果对webpack已经有一定的了解请直接跳过。

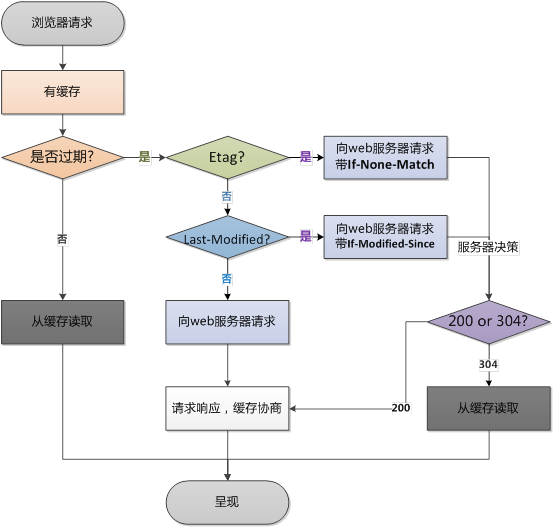
## 如何整合gulp & webpack

webpack是众多gulp子任务中比较复杂的部分，主要对JS/CSS进行相关处理。  
包括：模块分析、按需加载、JS代码压缩合并、抽离公共模块、SourceMap、PostCSS、CSS代码压缩等等...

gulp：处理html压缩/预处理/条件编译，图片压缩，精灵图自动合并等任务  
webpack：管理模块化，构建js/css。

至于为什么选择gulp & webpack，主要原因在于gulp相对来说更灵活，可以做更多的定制化任务，而webpack在模块化方案实在太优秀（情不自禁的赞美）。

**Webpack使用**



**动态页面静态页面**

<!--<meta http-equiv="pragma" content="no-cache">  
<meta http-equiv="cache-control" content="no-cache">  
<meta http-equiv="cache-control" content="no-store">  
<meta http-equiv="expires" content="0">-->  
<!-- <meta http-equiv="cache-control" content="max-age=0">-->  
<!-- <meta http-equiv="cache-control" content="no-store">  
 <meta http-equiv="expires" content="0">-->  
<!-- 禁止浏览器从本地缓存中调阅页面。-->  
<!--<meta http-equiv="pragram" content="no-cache">-->  
<!--网页不保存在缓存中，每次访问都刷新页面。-->  
<!--<meta http-equiv="cache-control" content="no-cache, must-revalidate">-->  
<!--同上面意思差不多，必须重新加载页面-->  
<meta http-equiv=**"expires"** content=**"-1"**>  
<!--网页在缓存中的过期时间为0，一旦网页过期，必须从服务器上重新订阅-->