基于Gulp的工程化流程说明

1. 安装说明：

1、gulp是基于node.js环境，要先安装node.js，

2、全局安装 gulp：（作为本地电脑cmd使用gulp命令）

$ npm install --global gulp

3、作为项目的开发依赖（devDependencies）安装（安装的插件名字会出现在gulpfile.js devDependencies内，便于其他人引入项目）：

$ npm install --save-dev gulp

例如：项目中可以引入var gulp = require('gulp');

4、项目中需要用到的开发依赖（devDependencies）（gulp需要用到的其他插件）安装：

$ npm install --save-dev 插件名

5、插件安装完后项目中会出现自动生成

node\_modules文件夹，里面全是devDependencies，本地依赖插件

6、如果直接引入别人项目：可以直接依赖package.json内的插件安装；

只需要，打开cmd->进入到当前工程目录->输入npm install 命令即可，会自动安装所有项目依赖包

1. 文件夹说明：
   1. build :
      1. gulpfile.config:放置gulp的基本路径配置，把路径提取为变量，方便不同项目路径修改
      2. gulpfile.dev : 主要配置项目工程在开发环境下的任务，后期如要加入功能，可直接在此文件夹内定义新的gulp.task( )任务即可，并加入到服务器监听，让浏览器可实时更新。具体插件使用可到npm官网查询。
      3. gulpfile.prod : 主要配置项目在上线前，把项目压缩打包，css自动添加前缀兼容等作用！配置可参照！
   2. data: 主要放置一些mock的数据，用于前后端分离后，后台只要定义好接口及数据格式，具体字段，前端可自行模拟数据测试。推荐mock.js来模拟数据。
   3. dist: 运行开发命令、或者生产命令后，放置生成后的目标文件。
   4. docs：主要放置一些开发文档，接口文档之类，便于以后查找。
   5. node\_moudles：放置项目的插件依赖包
   6. src：放置源码文件----保存项目所有的源码文件
      1. assets:放置一些外部资源等，作为项目公用依赖
         1. fonts:字体图标
         2. js:例如：jquery.js、Tween.js等
         3. scss:放置一些，项目组长期积累下来的scss文件或者第三方文件：例如：rest.scss文件等
      2. html：放置html文件---模板用的是nunjucks插件（用来继承模板，引用模板等）
         1. pages内放置要写的页面，足以可以继承layout.html基础模板
         2. templates放置公用模板
            1. partials放置公用的html模块，可新增加
            2. layout.html基础模板
      3. images：图片文件
         1. favicon:放置网站标题favicon
         2. icon：放置一些小图标
         3. icon-2x：移动端2x高清图
         4. 其他图片资源可直接放在images根目录内，也可新建文件存放。
      4. Js:存放js文件
      5. Sass:存放scss文件----模块化思想。
         1. var 存放变量文件，工程项目的变量文件
         2. mixin 存放混合文件
         3. 其他 .scss文件可直接见与根目录下
   7. .jshintrc：js语法检查配置，主要是同意项目组内各成员的书写格式，已经写好配置，不过项目组也可以自定义。具体用法去官网查。
   8. gulpfile.js : gulp启动任务的入口文件，整个gulp任务都是基于这个文件的，不过这个文件又依赖了build内的配置（其实是把gulpfile.js的配置放到了build文件夹内去配置，便于管理）
   9. packages.json：项目依赖包，有了这个文件，其他人引入后，运行命令npm install，即可全部安装项目依赖插件
2. gulp启用说明：
   1. 开发环境启用：
      1. 运行命令：npm run dev；项目即可启动，gulp会帮助用户编译，实时刷新浏览器等一系列事情。
   2. 生产打包启用：
      1. 运行命令：npm run build；项目即可启动，gulp会帮助用户编译，打包，自动加前缀等一系列事情。
3. 添加或更改gulp任务：
   1. dev环境下所有gulp任务都是在下面函数内新增加

function dev() {

//例如：新增加一个”html:dve”任务

gulp.**task**(**'html:dev'**, **function** () {

return …你的任务详情

});

}

* 1. prod环境下所有gulp任务都是在下面函数内新增加

function prod() {

//例如：新增加一个”html”任务

gulp.**task**(**'html'**, **function** () {

return …你的任务详情

});

}