# 什么是构建？为什么要自动化构建？

软件的构建：将源文件转化为可执行软件的过程

编译过程的简化，提高前端开发的效率。虽然IDE也具有编译的功能，但是打包和发布都还是要交由不同人独自处理，效率不足。

# 前端构建工具介绍

第三方类库太多，需要前端构建工具提高开发效率

Gulp

-基于流的自动化构建工具，将一个步骤的结果输送给下一个步骤，减少IO，提高构建工具的性能。

Javascript的构建

Js不需要编译的脚本语言，不需要编译，但是浏览器对js的支持不是那么好，html对js的支持必须要在其中引入js，多个js文件的支持就要引入多个，那么不可以只写一个script标签从而引入多个js文件吗？

# 什么是Browserify?小例子（demo1）

实现js的模块化加载。使用nodejs的Commonjs来加载模块。

1. 定义模块

module.exports = function() {

return ‘hello’;

}

1. 引入模块

var hello = require(‘./a.js’);

1. Browserify的安装

npm install –g browserify

1. 使用browserify

browserify main.js > bundle.js

1. Html中<script src=’./bundle.js’></script>

上面例子上传在demo1

# Gulp, 通过gulp工具生成js文件

1. 安装Gulp

npm install gulp

1. 新建gulpfile.js

var gulp = require(‘gulp’);

gulp.task (‘default’, function(){console.log(‘this is default task’)})

1. 执行 gulp，出现bundle.js，证明安装成功

那么如何在gulp中使用browserify

1. 安装shelljs，通过shelljs执行shell脚本

npm install shelljs

1. 在gulpfile.js中引用和使用browserify

var gulp = require(‘gulp’);

var shelljs = require(‘shelljs’);

gulp.task (‘default’, function(){

shelljs.exec(‘browserify main.js > bundle.js’)

})

1. 执行 gulp

但是，方法是有问题的，是借助shell做到的，browserify有自己的method api，在gulp中直接调用API更加正规

1. npm install browserify
2. 修改gulpfile文件

var gulp = require(‘gulp’);

var browserify = require(‘browserify); /加入browserify的引用

var fs = require(‘fs’);//node的核心模块之一，所以不需要通过npm安装

gulp.task (‘default’, function(){

browserify ().add(‘main.js’).bundle().pipe(fs.createWriteStream()); //初始化.将main.js作为输入文件，bundle之后调用pipe将上一步生成的stream传递给fs方法

})

1. 执行gulp

# 自动化（demo2）

但是，又但是。。。。自动化呢，每次修改js文件，都要gulp，根本没有自动化呀，别急，使用gulp watch监测文件变化。

1. 修改gulpfile文件：

加上下面代码

gulp.task(‘watch’, function(){

});

1. 但是源文件和目标文件都放在一起不好。执行3新建源文件，并将文件都移动到其中
2. 新建文件夹

assets > js

将源文件移到下面

1. 最终的gulpfile文件内容如下：

var gulp = require(‘gulp’);

var browserify = require(‘browserify);

var fs = require(‘fs’);//node的核心内容，所以不需要通过npm安装

gulp.task (‘default’, function(){//原本将源文件的处理放在了default task下面，但是这样就使得需要经常修改default task，按照5安装run-sequence，这个模块的作用是根据任务参数的名称调用不同的任务

sequence(‘mainjs’, ‘watch’);

})

gulp.task(‘mainjs, function(){

browserify ().add(‘main.js’).bundle().pipe(fs.createWriteStream());

})

gulp.task(‘watch’, function(){//使用gulp watch监测入口文件

gulp.watch([‘assets/js/\*.js’], function(){//监控的文件，监控源文件目录下的全部文件，在任何文件发生变动时重新执行sequence

sequence('mainjs');

})

})

1. npm install run-sequence
2. 执行gulp