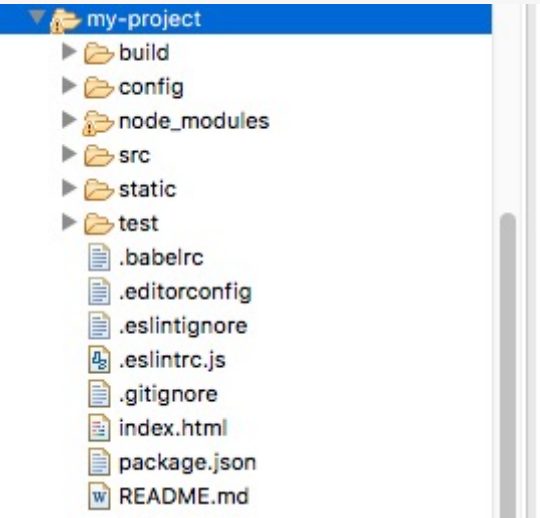


# Vue.js 目录结构

上一章节中我们使用了 npm 安装项目，我们在 IDE（Eclipse、Atom等）中打开该目录，结构如下所示：



## 目录解析

目录/文件	说明
build	项目构建(webpack)相关代码
config	配置目录，包括端口号等。我们初学可以使用默认的。
node_modules	npm 加载的项目依赖模块
src	这里是我们要开发的目录，基本上要做的事情都在这个目录里。里面包含了几个目录及文件： <ul style="list-style-type: none"><li>assets: 放置一些图片，如logo等。</li><li>components: 目录里面放了一个组件文件，可以不用。</li><li>App.vue: 项目入口文件，我们也可以直接将组件写这里，而不使用 components 目录。</li><li>main.js: 项目的核心文件。</li></ul>
static	静态资源目录，如图片、字体等。
test	初始测试目录，可删除
.xxxx文件	这些是一些配置文件，包括语法配置，git配置等。
index.html	首页入口文件，你可以添加一些 meta 信息或统计代码啥的。
package.json	项目配置文件。

目录/文件	说明
README.md	项目的说明文档，markdown 格式

在前面我们打开 APP.vue 文件，代码如下（解释在注释中）：

#### src/APP.vue

```
<!-- 展示模板 -->
<template>
<div id="app">

<hello></hello>
</div>
</template>
<script>
// 导入组件
import Hello from './components/Hello'
export default {
  name: 'app',
  components: {
    Hello
  }
}
</script>
<!-- 样式代码 -->
<style>
#app {
  font-family: 'Avenir', Helvetica, Arial, sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
  text-align: center;
  color: #2c3e50;
  margin-top: 60px;
}
</style>
```

接下来我们可以尝试修改下初始化的项目，将 Hello.vue 修改为以下代码：

#### src/components/Hello.vue

```
<template>
<div class="hello">
<h1>{{ msg }}</h1>
</div>
</template>
<script>
export default {
  name: 'hello',
  data () {
    return {
      msg: '欢迎来到菜鸟教程！'
    }
  }
}
```

```
}  
</script>
```

重新打开页面 <http://localhost:8080/>，一般修改后会自动刷新，显示效果如下所示：



← Vue.js 条件语句

Vue.js 计算属性 →

✍ 点我分享笔记