

我如何完成此项目？

查看电子简历[项目评估准则](#)。

1. 在此项目中，你将按照下面给出的模式 将简历数据存储在四个 `JavaScript` 对象中。像往常一样，利用 `API` 时，对象必须完全遵循该模式。所有 属性必须存在并具有真或假值。名称 必须与模式中的名称匹配（注意，对象名和属性名 区分大小写）。所有属性值必须是模式中 为属性提供的数据类型。例如，如果给定数据类型为数组，则不接受使用字符串作为该属性 的值。
2. 创建 `JavaScript` 对象后，你将编写 需要动态显示在简历中这些对象包含的 所有简历数据的代码。你的简历应该可以 显示所有根据给定的模式提供数据 的简历。注意，**forEach**或**for**（而不是**for in**）循环必须用于显示**技能、工作、项目、项目图片、学校和在线课程**数组。你的解决方案应作用于对应数组中的所有 数量的项目。
3. 建立简历所需的所有 `HTML` 代码存储在**js/helper.js** 变量中。变量名称表示它们的 函数。你将以 `JavaScript` 对象中的数据替换这些变量字符串值（如 `%data%` 和 `#`）中的子字符串，并将格式化的结果附加或预置到 相应位置的简历中。
4. 如果你需要复习 `JavaScript` 语法，请转至 [Javascript 基础](#) 课程；如果你想要了解该项目如何操控和遍历 `DOM`，请参阅 [jQuery 简介](#)。
5. 通过本课程的视频和作业了解建立简历所需的 `JavaScript` 语法。
6. 在 Github 上 Fork [此代码库](#)，开始建立你的电子简历。
7. 根据 [项目评估准则](#) 检查你的工作。
8. 当你对项目满意时，根据下一页的“提交说明”将其提交。



最后，你的简历应该看起来像这样。

你的代码库中应包括以下文件：

- **index.html**：主要 HTML 文档。包含呈现简历所需的所有 CSS 和 JS 资源的链接，其中包括 resumeBuilder.js。
- **js/helper.js**：包含格式化简历和构建地图所需的帮助程序代码。它还具有一些函数 shell 用于其他功能。更多有关 helper.js 的深入分析。
- **js/resumeBuilder.js**：此文件是空的。你应该在此编写代码。
- **js/jquery.js**：jQuery 库。
- **css/style.css**：包含设置页面样式所需的所有 CSS。
- **README.md**：GitHub README 文件。
- 图片目录中的一些图片。

你的起点...

js/helper.js

在 helper.js 内，你将发现包含 HTML 代码段的大量字符串。在许多代码段内，你将发现 **%data%** 或 **%contact%** 格式的占位符数据。

每个字符串具有一个描述应如何使用它的标题。例如，**HTMLworkStart** 应该是简历“工作”部分的第一个 **<div>**。**HTMLschoolLocation** 包含 **%data%** 占位符，应该替换为一处学校地点。

你的流程：

简历具有四个不同的部分：工作、教育、项目和具有个人信息的标题。你将需要：

1. 构建四个 javascript 对象，每个对象代表一个不同的简历部分。你创建的对象（包括属性名称及其值的数据类型）需完全遵循以下模式。所有属性均应包括在内并且包含指定类型的值，除非该属性标记为 'optional'。属性值可能包含真或假数据。属性名称区分大小写。确保使用 jshint.com 将你的 javascript 对象正确格式化。
2. **bio** 包含：

```
name: 字符串
role: 字符串
contacts: 对象
  mobile: 字符串
  email: 字符串
  github: 字符串
  twitter: 字符串 (可选)
  location: 字符串
welcomeMessage: 字符串
skills: 字符串数组
```

biopic: url
display: 函数

3. education 包含:

schools: 对象数组
 name: 字符串
 location: 字符串
 degree: 字符串
 majors: 字符串数组
 dates: 字符串（之间使用连字符）
 url: 字符串
onlineCourses: 对象数组
 title: 字符串
 school: 字符串
 dates: 字符串（之间使用连字符）
 url: 字符串
display: 函数

4. work 包含

jobs: 对象数组
 employer: 字符串
 title: 字符串
 location: 字符串
 dates: 字符串（可以是 'in progress'）
 description: 字符串
display: 函数

5. projects 包含:

projects: 对象数组
 title: 字符串
 dates: 字符串（之间使用连字符）
 description: 字符串
 images: 字符串格式的 url 数组
display: 函数

1. 迭代每个 JavaScript 对象并将其信息附加到 index.html 正确的部分。

1. 首先，你将使用 jQuery 的 `selector.append()` 和 `selector.prepend()` 函数来修改 index.html。`selector.append()` 使元素显示在所选部分的末尾。`selector.prepend()` 使元素显示在所选部分的开始。
2. 密切关注 index.html 中 `<div>` 的 id 和 helper.js 中的 HTML 代码段。它们作为用于 `selector.append()` 和 `selector.prepend()` 的 jQuery 选择器，将十分有用。

2. 你还将使用 JavaScript 方法 `string.replace(old, new)` 将所有占位符文本（例如，`%data%`）替换为简历 JavaScript 对象中的数据。

3. 这是一些代码的示例，会把你公司的地址添加到页面中：

1. `var formattedLocation = HTMLworkLocation.replace("%data%", work.jobs[job].location);`
2. `$(".work-entry:last").append(formattedLocation);`
3. 使用文档底部的模型作为顺序指南将元素附加到页面。

4. 简历包括交互式地图。执行以下操作将其添加。

1. 在 resumeBuilder.js 中，将 googleMap 字符串添加到 `<div id="mapDiv">` 中。
2. 在 index.html 中，取消注释 Google 脚本元素：`<script type="text/javascript" src="http://maps.googleapis.cn/maps/api/js?libraries=places"></script>`
3. 在 helper.js 文件底部，取消注释代码以初始化地图并设置 fitBounds。

5. 用于将元素添加到简历的所有代码应 包含在函数内。

6. 如 JavaScript 对象模式中所描述，每个 `display` 函数应封装其在简历中显示的 JavaScript 对象内。例如，用于将 'work' 经历数据 附加到简历的 `display` 函数应封装在 `workJavaScript` 对象内。`display` 函数 可以定义其他属性的方式封装在 `workJavaScript` 对象内， 或者可稍后使用点标记封装在文件内。例如：`work.display =`

7. 当你单击地图上的大头针时，可以显示附加信息。请检查 helper.js 中的 174 行和 Google Maps API 予以了解