

第十四届蓝桥杯全国软件和信息技术专业人才大赛（Web 应用开发）

竞赛须知

请选手逐条仔细阅读以下须知：

1. 考试开始后，选手首先下载题目，并使用考场现场公布的解压密码解压试题。
2. 考试时间为 4 个小时，时间截止后，提交答案无效。
3. 在考试强制结束前，选手可以主动结束考试（需要身份验证），结束考试后将无法继续提交或浏览答案。
4. 选手可浏览自己已经提交的答案，被浏览的答案允许拷贝。
5. 对同一题目，选手可多次提交答案，以最后一次提交的答案为准。
6. 选手切勿在提交的代码中书写“姓名”、“考号”、“院校名”等与身份有关的信息或其它与竞赛题目无关的内容，否则成绩无效。
7. 选手必须通过浏览器方式提交自己的答案。选手在其它位置的作答或其它方式提交的答案无效。
8. 试题类型均为场景实战题。

注意事项

请选手逐条仔细阅读以下须知：

1. 基础源代码在无明确说明的情况下，请勿随意修改文件名称、文件夹名称、文件存放结构等。严格遵守题目中的注意事项。
2. 全部题目将使用前端自动化测试技术完成机器自动评分，务必严格规范根据题意操作，否则可能会影响最终机器阅卷的准确性。
3. 在评卷时使用的输入数据与试卷中给出的示例数据可能是不同的。选手的程序必须是通用的，不能只对试卷中给定的数据有效。

省赛（大学组）

试题介绍

场景实战题目均包含 2 部分内容：题面 PDF 文档和基础源代码压缩包。

1. 题面 PDF 文档可以使用本地 PDF 阅读器或者 Google Chrome 浏览器打开阅读。题面文档中会详细说明题目的背景、需求、目标。
2. 基础源代码压缩包中会给出相应题目的基础代码。一般情况下，选手需认真读题，结合需求和给出的基础源代码，通过修改、新增代码来实现题目最终目标。

部分题目可能包含前序准备步骤。例如解压缩相应的资源文件，在浏览器中预览网页效果等。大部分情况下，我们默认选手已经掌握了前端开发过程中可能涉及的基础知识和方法，不会给予单独的提示。同时，题目不会给予 IDE 开发工具的使用方法提示。

试题列表和分值

试题序号、试题名称及基础源代码文件夹名称对应如下：

- 01 电影院排座位（5 分）
- 02 图片水印生成（5 分）
- 03 收集帛书碎片（10 分）
- 04 自适应页面（10 分）
- 05 外卖给好评（15 分）
- 06 视频弹幕（15 分）
- 07 年度明星项目（20 分）
- 08 全球新冠疫情数据统计（20 分）
- 09 Markdown 文档解析（25 分）
- 10 组课神器（25 分）

试题提交说明

1. 考试给出题目压缩包中包含 PDF 题面，和 10 道题目的基础代码子文件夹，和题目序号一一对应。
2. 考生需根据题意和目标，在考试给出的基础代码上新增、修改代码。请勿修改给出代码文件夹名称，层次结构。
3. 相应题目的代码完成之后，**考生需将题目对应代码文件夹压缩成 zip 格式后上传提交，其他格式为 0 分**。例如 01 代码文件夹，应该在此文件夹基础上直接压缩后，为 01.zip ，然后提交压缩包到对应题目。请注意不要给压缩包添加密码。
4. 请务必严格按照规范提交题目代码，否则会造成无法被系统正确判分。

电影院排座位

介绍

随着人们生活水平的日益提升，电影院成为了越来越多的人休闲娱乐，周末放松的好去处。各个城市的电影院数量也随着市场的需求逐年攀升。近日，又有一个电影院正在做着开张前期的准备，小蓝作为设计工程师，需要对电影院的座位进行布局设计。

本题需要在已提供的基础项目中，使用 CSS 完成页面中电影院座位布局。

准备

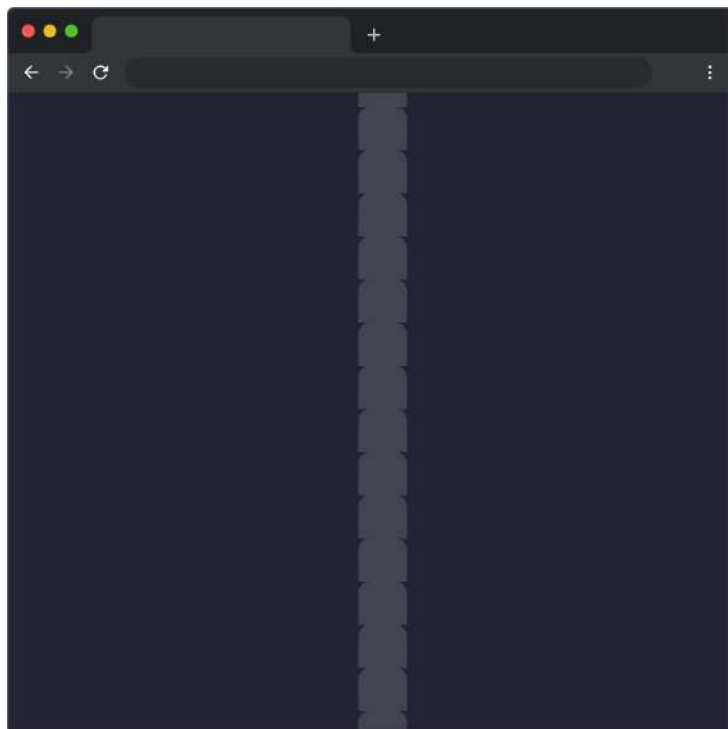
开始答题前，需要先打开本题的项目代码文件夹，目录结构如下：

```
├── css
│   └── style.css
└── index.html
```

其中：

- `index.html` 是主页面。
- `css/style.css` 是需要补充样式的文件。

在浏览器中预览 `index.html`，显示如下所示：



目标

请在 `css/style.css` 文件中的 TODO 下补全样式代码，最终达到预期布局效果，需要注意：

- 座位区域和荧幕间隔 50px。
- 座位区域每一行包含 8 个座位。
- 第 2 列和第 6 列旁边都是走廊，需要和下一列间隔 30px，其他列都只需要间隔 10px。

完成后的效果如下：



规定

- 请勿修改 `css/style.css` 文件外的任何内容。
- 请严格按照考试步骤操作，切勿修改考试默认提供项目中的文件名称、文件夹路径、class 名、id 名、图片名等，以免造成判题无法通过。
- 完成答题后，考生需将题目对应代码文件夹压缩成 zip 格式后上传提交，其他压缩包格式为 0 分。

判分标准

- 本题完全实现题目目标得满分，否则得 0 分。

图片水印生成

介绍

很多网站都会通过给图片添加水印的形式来标记图片来源，维护版权。前端生成水印通常是通过 `canvas` 实现，但实际上我们也可以直接利用 `CSS` 来实现图片水印，这样做会有更好的浏览器兼容性。

本题中你将封装一个创建文字水印的函数。

准备

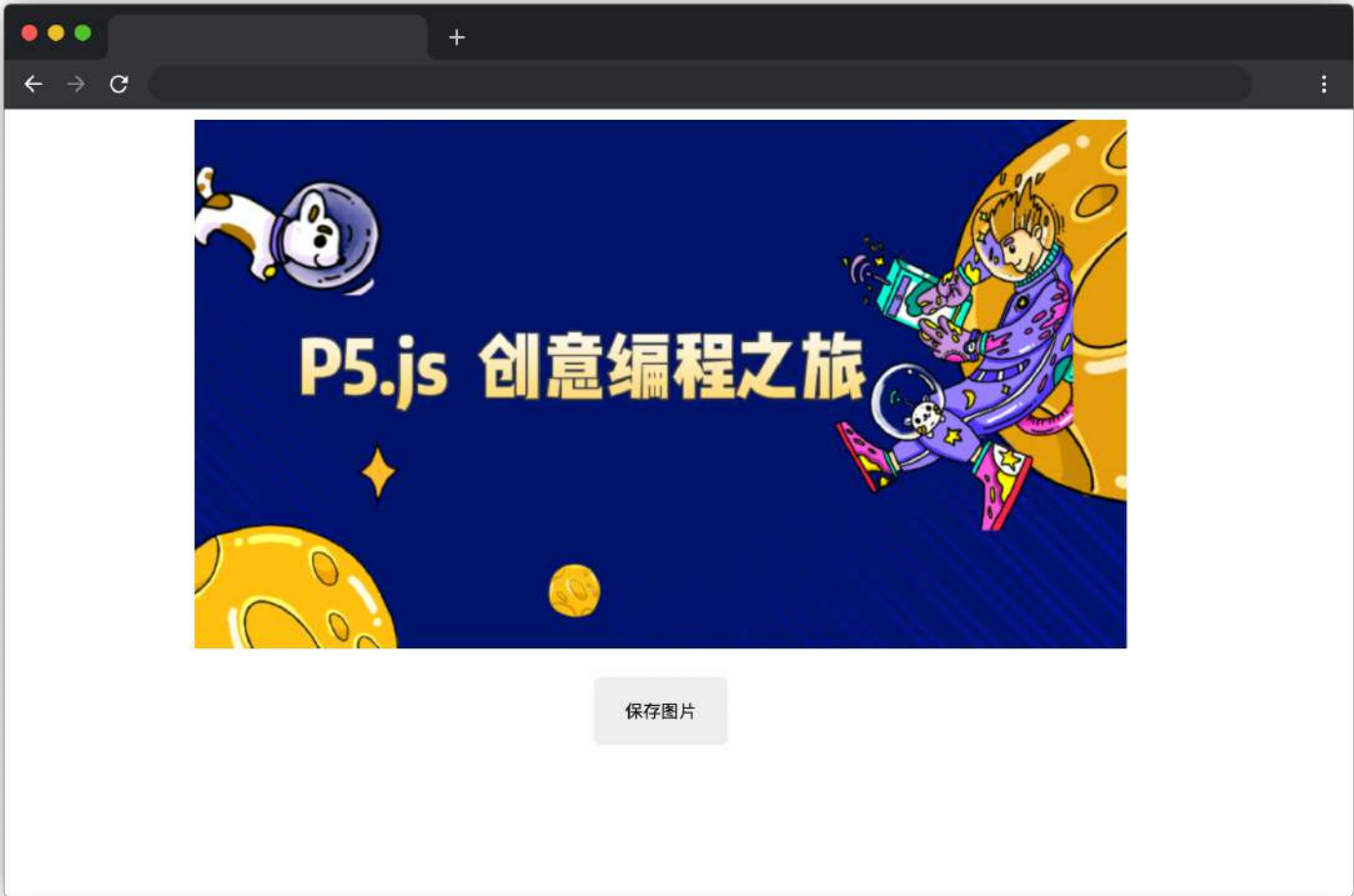
开始答题前，需要先打开本题的项目代码文件夹，目录结构如下：

```
├── css
│   └── style.css
├── images
│   └── origin.png
├── index.html
└── js
    ├── dom-to-image.min.js
    └── index.js
```

其中：

- `index.html` 是主页面。
- `css/style.css` 是样式文件。
- `js/index.js` 是需要补充代码的 `js` 文件。
- `js/dom-to-image.min.js` 是生成图片的第三方库，此文件无需修改。
- `images/origin.png` 是项目中的原始图片文件。

在浏览器中预览 `index.html` 页面效果如下所示：



目标

请完善 `js/index.js` 文件中的 `TODO` 部分，实现创建水印函数的功能，创建的水印需要使用 `` 标签展示。

`createWatermark` 函数参数说明：

参数	说明	类型
text	文字内容	string
color	颜色值	string
deg	旋转角度	number
opacity	透明度	number
count	水印数量	number

完成后的效果如下：



规定

- 请严格按照考试步骤操作，切勿修改考试默认提供项目中的文件名称、文件夹路径、class 名、id 名、图片名等，以免造成判题无法通过。
- 完成答题后，考生需将题目对应代码文件夹压缩成 zip 格式后上传提交，其他压缩包格式为 0 分。

判分标准

- 本题完全实现题目目标得满分，否则得 0 分。

收集帛书碎片

介绍

三叔在外出考古途中无意发现了一份战国帛书，帛书边缘有被明显裁剪过的痕迹，单从帛书片段，提到记录了神秘文物的地点，无奈帛书不完整，为了早日将文物带回博物馆，三叔号召当地村民闷油瓶、王胖子、潘子共同寻找剩余帛书碎片，相约一炷香后再聚。

本题需要在已提供的基础项目中使用 JS 知识封装一个函数，达到收集帛书碎片的要求。

准备

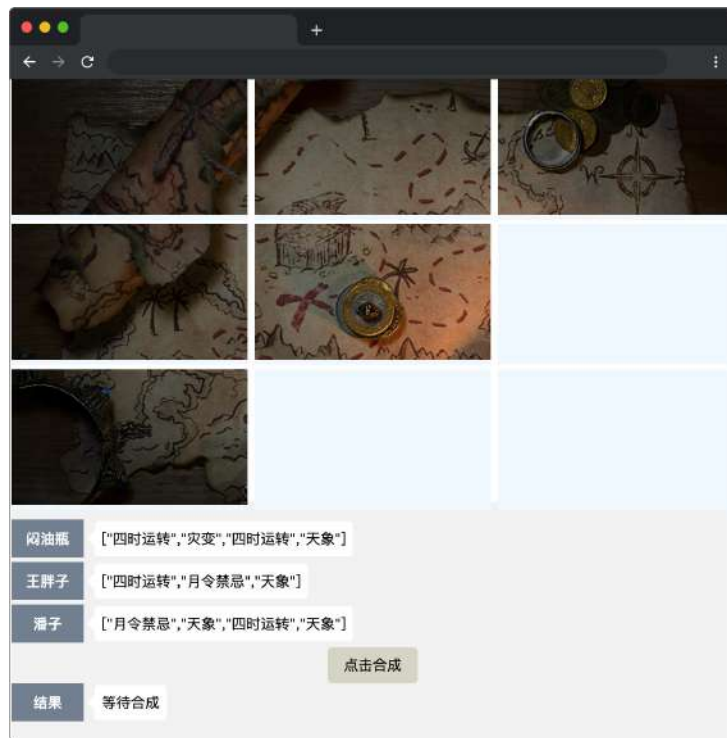
开始答题前，需要先打开本题的项目代码文件夹，目录结构如下：

```
|— js
|   |— collect-puzzle.js
|— css
|   |— style.css
|— images
|— index.html
```

其中：

- `index.html` 是主页面。
- `css/style.css` 是样式文件。
- `images` 是图片文件夹。
- `js/collect-puzzle.js` 是需要补充代码的 js 文件。

在浏览器中预览 `index.html` 页面，显示如下所示：



目标

请在 `collect-puzzle.js` 文件中补全函数 `collectPuzzle` 中的代码，返回包含不同帛书碎片的数组，最终拼出完整的战国帛书，需要注意：

- 同一个人收集的帛书碎片可能是有重复的。
- 同一组内不同的人收集的帛书碎片也可能是有重复的。
- 工具函数需要统计所有人获取的**不同帛书碎片**，而不是**帛书碎片数量**。

例如：

```
// 不同组内的帛书碎片不能重复计算
var arr1 = ["灾变", "四时运转", "天象"];
var arr2 = ["四时运转"];

var result1 = collectPuzzle(arr1, arr2);
console.log(result1); // 输出: ["灾变", "四时运转", "天象"]

// 同一组内的帛书碎片不能重复计算
var arr3 = ["灾变", "灾变", "天象"];
var arr4 = ["灾变", "月令禁忌", "天象"];

var result2 = collectPuzzle(arr3, arr4);
console.log(result2); // 输出: ["灾变", "天象", "月令禁忌"]
```

上述仅为示例代码，判题时会随机提供数组对该函数功能进行检测。

在题目所提供的数据的情况下，点击合成按钮，完成后的效果如下：



规定

- 请勿修改 `collect-puzzle.js` 文件外的任何内容。
- 判题时会随机提供数组对该函数功能进行检测，请保证函数的通用性，不能仅对测试数据有效。
- 请严格按照考试步骤操作，切勿修改考试默认提供项目中的文件名称、文件夹路径、class 名、id 名、图片名等，以免造成判题无法通过。
- 完成答题后，考生需将题目对应代码文件夹压缩成 zip 格式后上传提交，其他压缩包格式为 0 分。

判分标准

- 完全实现题目目标得满分，否则得 0 分。

自适应页面

介绍

响应式布局是在 2010 年 5 月份提出的一个概念，这个概念是为解决移动互联网浏览而诞生的。简而言之，就是一个网站能够兼容多个终端——而不是为每个终端做一个特定的版本。通过响应式布局可以为不同终端的用户提供更加舒适的界面和更好的用户体验，而且随着大屏幕移动设备的普及，用“大势所趋”来形容也不为过。因此越来越多的设计师采用这个技术。

本题需要在已提供的基础项目中，使用 CSS 或者 DOM 操作达到 Menu 和内容页自适应的效果。

准备

开始答题前，需要先打开本题的项目代码文件夹，目录结构如下：

```
├── css
│   └── style.css
├── images
│   ├── C++_course.png
│   ├── linux_course.png
│   └── python_course.png
├── index.html
├── js
│   └── jquery-3.6.0.min.js
├── default.gif
└── effect.gif
```

其中：

- index.html 是主页面。
- css/style.css 是需要补充样式的文件。
- js/jquery-3.6.0.min.js 是 jQuery 库文件。
- images 是图片文件夹。
- default.gif 是 PC 端默认效果图。
- effect.gif 是移动端自适应效果图。

在浏览器中预览 index.html 页面，默认 PC 端页面显示如下所示：



目标

初始页面已经具备基础的布局和交互效果，包含 hover 高亮，展示二级菜单等。效果见文件夹下面的 gif 图，图片名称为 default.gif（提示：可以通过 VS Code 或者浏览器预览 gif 图片）。

请通过补全 `css/style.css` 中代码或结合 jQuery 操作 DOM 的方式，达到根据屏幕大小显示不同布局的效果。

- 以 800px 为界限，800px 以上显示 PC 端布局，否则显示移动端布局，**需要实现移动端布局样式**如下：



- 移动端 **Menu** 由左上侧按钮（即 class 包含 icon-menu 的 label 标签）控制显隐，按钮大小已经默认提供，无需手动设置大小。且 Menu 按钮展示时，需要浮动在内容卡片上方，不能被遮挡，移动端和 PC 端顶部导航栏高度一致，均为 54px。
- 移动端导航栏的菜单项每一项独占一行。
- 显示移动端布局时，卡片描述和对应图片各占一行，且都撑满 #tutorials 容器。

移动端页面展开菜单栏效果如下所示：



最终效果可参考文件夹下面的 gif 图，图片名称为 effect.gif（提示：可以通过 VS Code 或者浏览器预览 gif 图片）。

考生可以根据需要使用 jQuery 实现脚本控制，也可以只使用纯 CSS 实现。

规定

- 请严格按照考试步骤操作，切勿修改考试默认提供项目中的文件名称、文件夹路径、class 名、id 名、图片名等，以免造成判题无法通过。

- 完成答题后，考生需将题目对应代码文件夹压缩成 zip 格式后上传提交，其他压缩包格式为 0 分。

判分标准

- 本题完全实现题目目标得满分，否则得 0 分。

外卖给好评

介绍

外卖是现代生活中必备的一环。收到外卖后，各大平台软件常常会邀请用户在口味，配送速度等多个方面给与评分。在 element-ui 组件中，已经有相应的 Rate 组件，但是已有组件只能对单一维度进行评分，在外卖评分这种场景中，样式基本上是固定的，功能也基本一样。若每写一个页面都要去复制一份类似代码，就会产生大量重复的代码，既不利于后期的维护，代码也不够简洁。为此需要前端工程师对 element-ui 的原 Rate 组件进行二次封装。

准备

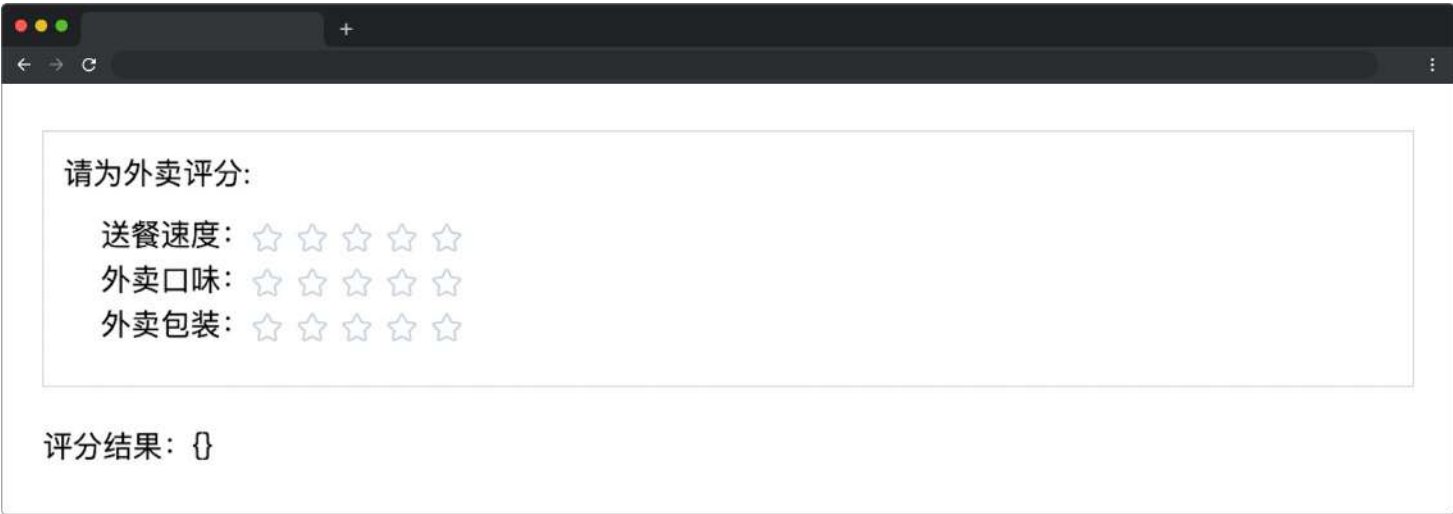
开始答题前，需要先打开本题的项目代码文件夹，目录结构如下：

```
├── element-ui-2.6.2
│   ├── element-icons.ttf
│   ├── element-icons.woff
│   ├── element-ui.min.js
│   └── element-ui.style.min.css
├── index.html
├── js
│   ├── http-vue-loader.min.js
│   └── vue.min.js
├── my-rate.vue
└── effect.gif
```

其中：

- index.html 是主页面。
- element-ui-2.6.2 文件夹中存放的是 element-ui 库相关的脚本文件、样式文件及字体。
- js 文件夹中存放的是 vue 及 http-vue-loader 库相关文件。
- my-rate.vue 是待封装的评分组件文件。
- effect.gif 是完成后的效果图。

在浏览器中预览 index.html 页面，显示如下所示：



目标

请在 `my-rate.vue` 文件中补充代码，具体要求如下：

- 1. `my-rate.vue` 组件能够对不同的维度进行评分。
- 2. `my-rate.vue` 组件对外抛出 `change` 事件，在三项评分均完成后，触发 `change` 事件，`change` 事件包含一个参数，用于传递改变后的分数值，其类型是对象，包含以下属性：

```
{
  speed: number;
  flavour: number;
  pack: number;
}
```

最终效果可参考文件夹下面的 gif 图，图片名称为 `effect.gif`（提示：可以通过 VS Code 或者浏览器预览 gif 图片）。

实现该功能所需的 `el-rate` 组件 api 如下：

参数	说明	类型	默认值
value/v-model	绑定值	number	0
show-score	是否显示当前分数， <code>show-score</code> 和 <code>show-text</code> 不能同时为真	boolean	false
change(event 事件)	分值改变时触发，参数是改变后的分值	(changed: object) => void	

规定

- 请勿修改 `my-rate.vue` 文件外的任何内容。
- 请严格按照考试步骤操作，切勿修改考试默认提供项目中的文件名称、文件夹路径、class 名、id 名、图片名等，以免造成判题无法通过。
- 完成答题后，考生需将题目对应代码文件夹压缩成 zip 格式后上传提交，其他压缩包格式为 0 分。

判分标准

- 完全实现题目目标得满分，否则得 0 分。

视频弹幕

介绍

弹幕指直接显现在视频上的评论，可以以滚动、停留甚至更多动作特效方式出现在视频上，是观看视频的人发送的简短评论。通过发送弹幕可以给观众一种“实时互动”的错觉，弹幕的出现让观看过程充满乐趣。本题需要在已提供的基础项目中，完成视频弹幕的功能。

准备

开始答题前，需要先打开本题的项目代码文件夹，目录结构如下：

```
|— effect.gif
|— css
|   |— style.css
|— video
|   |— video1.webm
|— index.html
|— js
|   |— index.js
```

其中：

- `index.html` 是主页面。
- `js/index.js` 是需要补充代码的 `js` 文件。
- `css/style.css` 是样式文件。
- `effect.gif` 是完成的效果图。
- `video` 是存放视频的文件夹。

在浏览器中预览 `index.html` 页面，显示如下所示：



目标

请在 `js/index.js` 文件中补全代码。具体要求如下：

1. 补全 `renderBullet` 函数中的代码，控制弹幕的显示颜色和移动。功能说明如下：

- 每个弹幕内容包裹在 `span` 标签中，作为子节点插入到 `#video` 元素节点内。
- 生成的 `span` 元素节点相对于 `#video` 元素绝对定位，初始位置的 `left` 是 `#video` 元素的宽，`top` 是 `#video` 元素的高内的随机数。

注意：需求中所需样式可直接通过已提供的 `getEleStyle` 方法获取。

- 弹幕每隔 `bulletConfig.time`（弹幕配置对象）时间，向左移动距离为 `bulletConfig.speed`（弹幕配置对象）。
- 当弹幕最右端完全移出 `#video` 元素时，移除 `span` 元素。

2. 补全 `#sendBulletBtn` 元素的绑定事件，点击发送按钮，输入框中的文字出现在弹幕中，样式不同于普通弹幕（样式红色字体红色框已设置，类名为 `create-bullet`）。通过调用 `renderBullet` 方法和正确的传参实现功能。

最终效果可参考文件夹下面的 gif 图，图片名称为 `effect.gif`（提示：可以通过 VS Code 或者浏览器预览 gif 图片）。

规定

- 请勿修改 `js/index.js` 文件外的任何内容。

- 请严格按照考试步骤操作，切勿修改考试默认提供项目中的文件名称、文件夹路径、class 名、id 名、图片名等，以免造成判题无法通过。
- 完成答题后，考生需将题目对应代码文件夹压缩成 zip 格式后上传提交，其他压缩包格式为 0 分。

判分标准

- 本题完全实现题目目标得满分，否则得 0 分。

年度明星项目

介绍

作为前端开发的主力语言，JavaScript 相关的开源项目是每一个前端开发者都应该多多关注的。我们可以通过这一年新增 star 的数量来判断一个开源项目的流行趋势。

本题请实现一个展示 2022 年 JavaScript 明星开源项目数据的网页。

准备

开始答题前，需要先打开本题的项目代码文件夹，目录结构如下：

```
├── css
│   └── style.css
├── effect.gif
├── images
├── index.html
└── js
    ├── all-data.json
    ├── index.js
    ├── jquery-3.6.0.min.js
    └── translation.json
```

其中：

- `css/style.css` 是样式文件。
- `index.html` 是主页面。
- `images` 是图片文件夹。
- `js/all-data.json` 是项目数据文件。
- `js/index.js` 是需要补充代码的 `js` 文件。
- `js/jquery-3.6.0.min.js` 是 jQuery 库文件。
- `js/translation.json` 是页面所用到的翻译数据。
- `effect.gif` 是页面最终的效果图。

在浏览器中预览 `index.html` 页面效果如下：



目标

请在 `js/index.js` 文件中补全代码，具体需求如下：

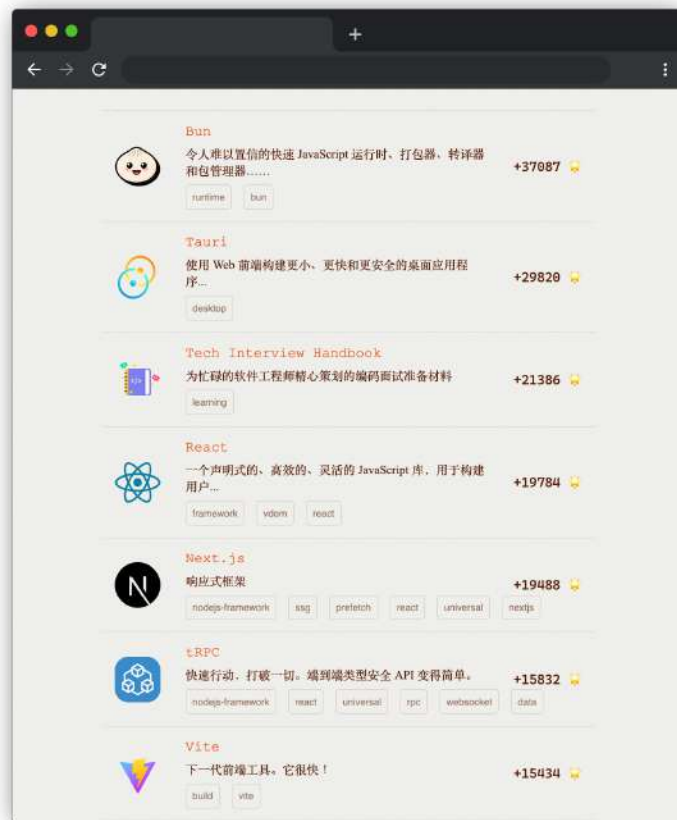
1. 在页面初始化时使用 AJAX 请求地址为 `./js/all-data.json` 以及 `./js/translation.json` 文件中的数据（必须使用给定的路径请求，否则可能会请求不到数据），并将后者中的数据保存至 `translation` 变量中。其中 `all-data.json` 文件中以数组的形式存储了明星项目的数据，`translation.json` 文件中包含了网站中英文转换所需的数据。

`all-data.json` 数据参数说明：

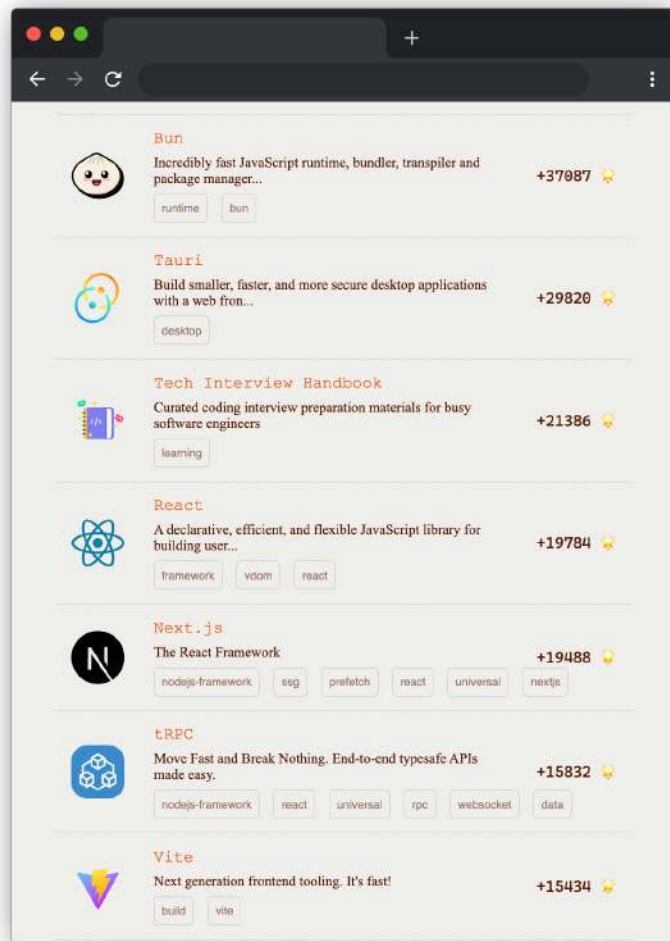
参数	说明	类型
<code>name</code>	项目名称	string
<code>icon</code>	项目 icon 路径	string
<code>stars</code>	项目新增 star 数量	number
<code>descriptionCN</code>	项目中文描述	string
<code>descriptionEN</code>	项目英文描述	string
<code>tags</code>	项目标签列表	array

2. 页面初始化时利用 `createProjectItem` 函数创建前 15 个项目数据（即 `all-data.json` 数组中的前 15 项）的列表元素并加载到页面中。当用户点击 **加载更多** 按钮时，则按顺序再显示 15 个项

目数据。直到所有项目数据都展示完毕（共 60 个）。所有项目展示完毕后需要隐藏 **加载更多** 按钮。项目展示效果如图所示：



3. 当用户点击页面右上方的中英文切换按钮时，根据用户的选择改变项目描述使用的语言（不改变原有项目展示数量）。当用户选择英语模式时的项目展示效果如图所示：



最终效果可参考文件夹下面的 gif 图，图片名称为 effect.gif（提示：可以通过 VS Code 或者浏览器预览 gif 图片）。

规定

- 请严格按照考试步骤操作，切勿修改考试默认提供项目中的文件名称、文件夹路径、class 名、id 名、图片名等，以免造成判题无法通过。
- 完成答题后，考生需将题目对应代码文件夹压缩成 zip 格式后上传提交，其他压缩包格式为 0 分。

判分标准

- 完成目标 1，得 5 分。
- 完成目标 2，得 10 分。
- 完成目标 3，得 5 分。

全球新冠疫情数据统计

介绍

新冠疫情席卷全球，在此期间有很多免费的 API 和网站为人们提供了各个国家疫情数据的查询功能，这些免费公开的数据体现出了互联网作为信息媒介的优越性，帮助全球人民更好的了解一线疫情信息。

本题请实现一个可以对各个国家的新冠疫情数据统计的页面。

准备

开始答题前，需要先打开本题的项目代码文件夹，目录结构如下：

```
|— css
|   |— style.css
|— effect.gif
|— index.html
|— js
|   |— axios.min.js
|   |— covid-data.json
|   |— echarts.min.js
|   |— vue.min.js
```

其中：

- `css/style.css` 是样式文件。
- `index.html` 是主页面。
- `js/axios.min.js` 是 axios 文件。
- `js/vue.min.js` 是 vue2.x 文件。
- `js/echarts.min.js` 是 echarts 文件。
- `js/covid-data.json` 是页面所用到的新冠数据。
- `effect.gif` 是页面最终的效果图。

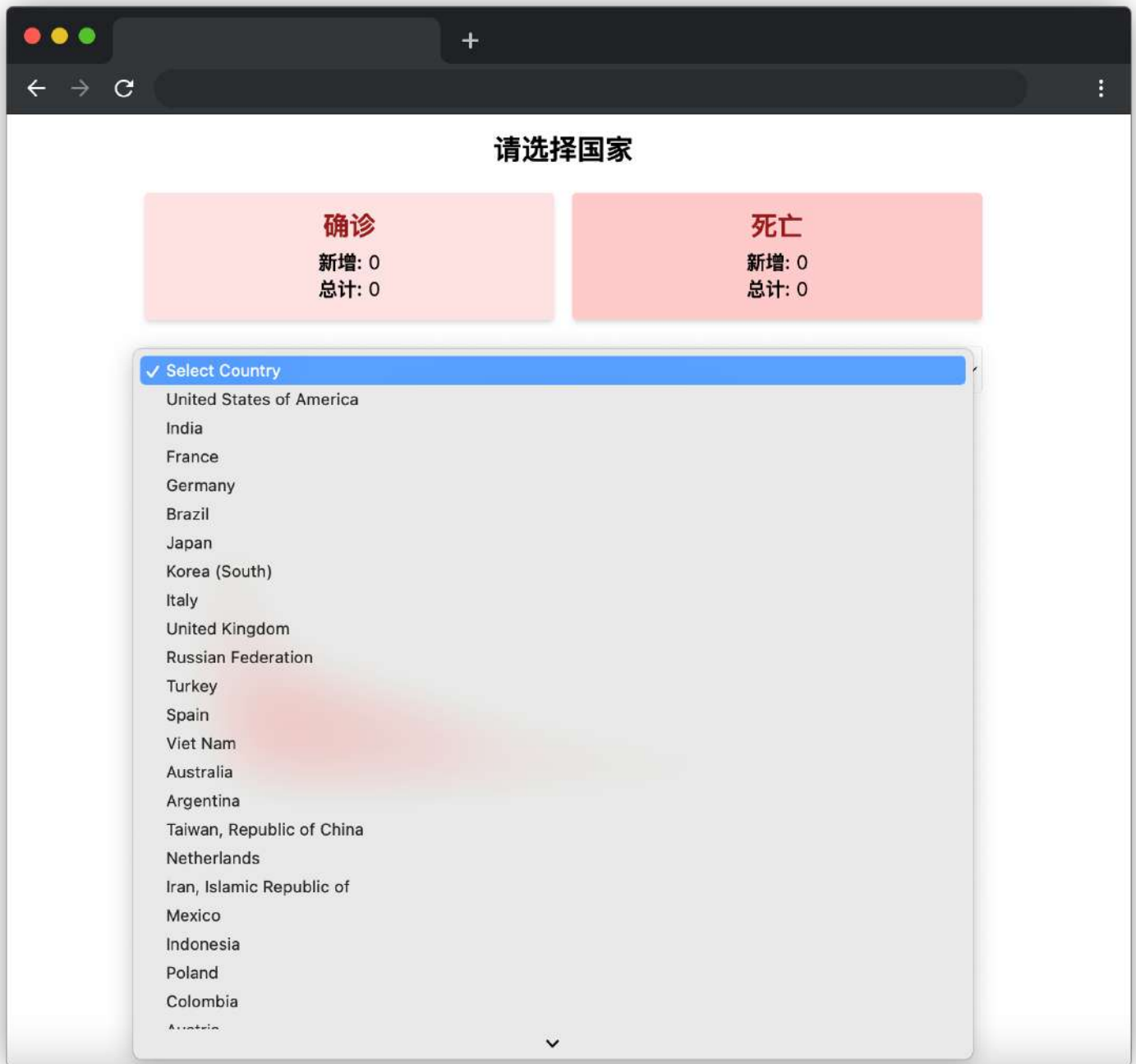
在浏览器中预览 `index.html` 页面效果如下：



目标

请在 `index.html` 文件中补全代码，具体需求如下：

1. 在组件加载时利用 `axios` 请求地址为 `./js/covid-data.json` （必须使用该路径请求，否则可能会请求不到数据）文件中的数据。并将所有国家名称在 `select` 标签下的 `option` 元素进行渲染（保留默认选项“Select Country”），效果如下：

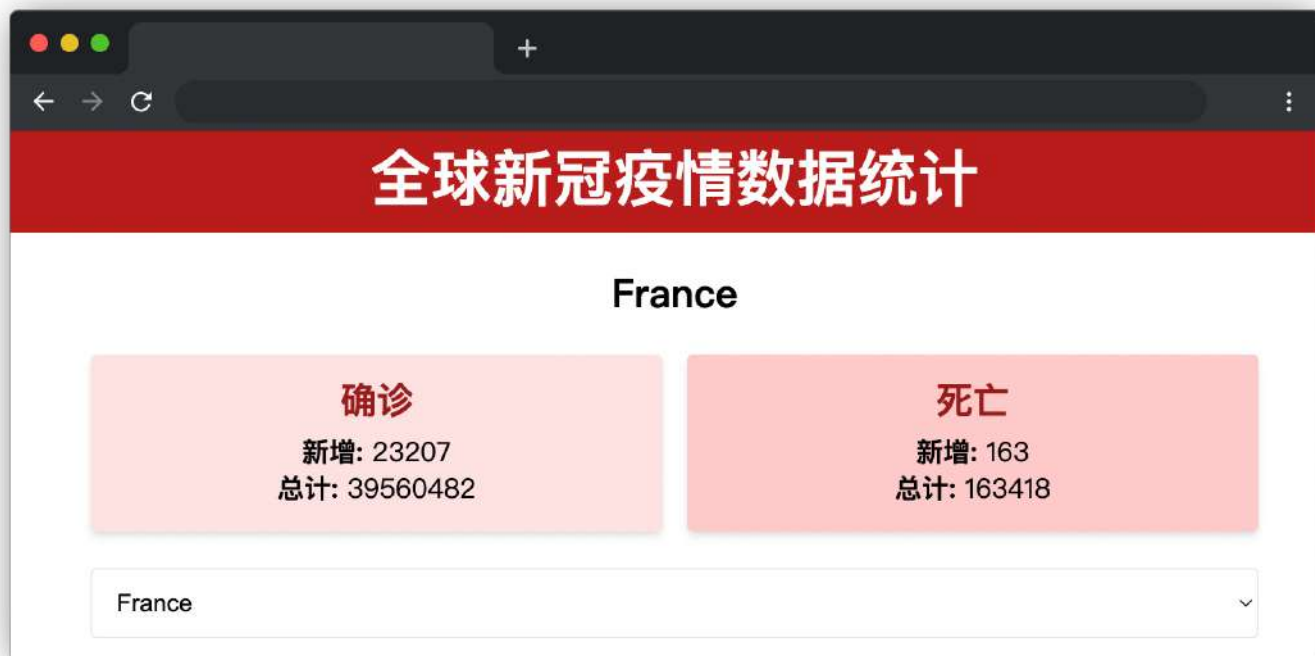


covid-data.json 数据参数说明:

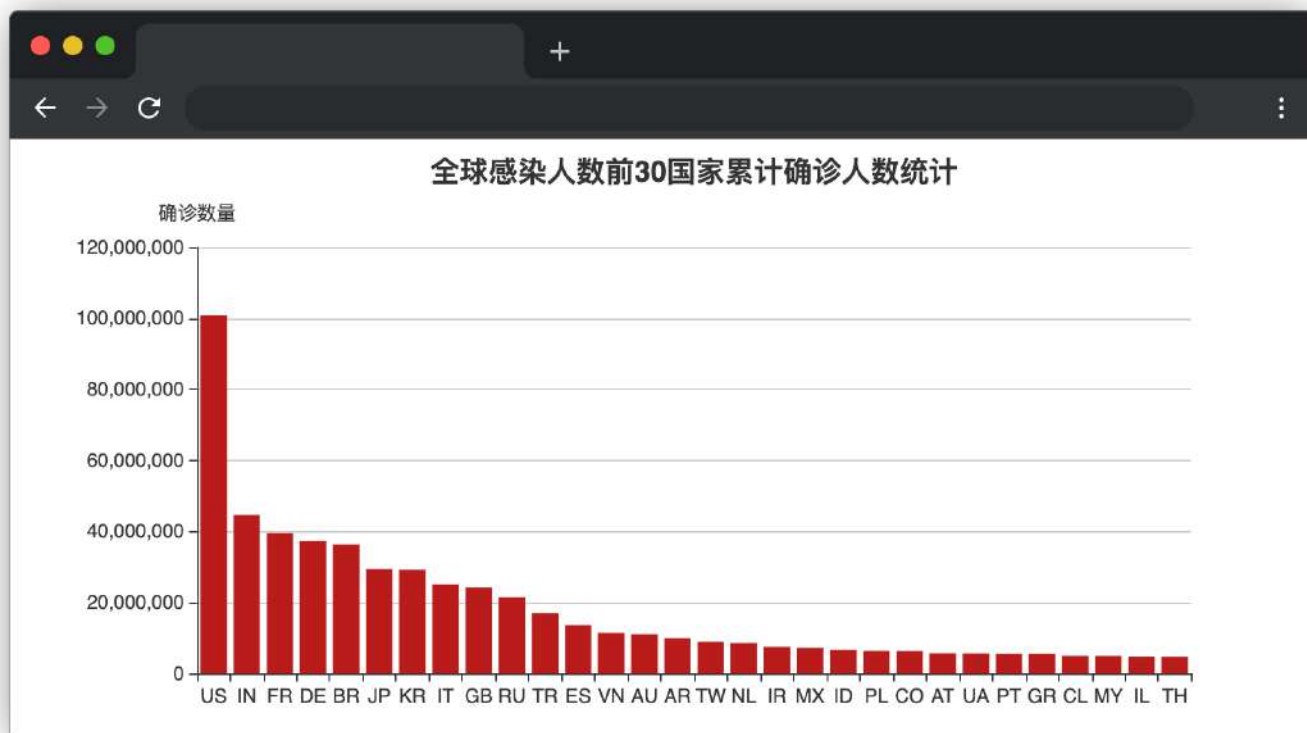
参数	说明	类型
Country	国家名称	string
CountryCode	国家简称	string
NewConfirmed	新增确诊	number
TotalConfirmed	累计确诊	number
NewDeaths	新增死亡	number

参数	说明	类型
TotalDeaths	累计死亡	number

2. 当用户改变 `select` 筛选器的选择时，根据用户的选择改变页面中展示的国家名以及确诊和死亡人数数据。如果用户没有选择任何国家，则展示默认值 0 和默认标题“**请选择国家**”。以选择法国为例：



3. 页面底部的 ECharts 图表希望显示各个国家的累计确诊人数，请修改 `initChart` 函数的内容，使得图表 x 轴数据为**国家简称**，y 轴数据为**累计确诊人数**，效果如下：



最终效果可参考文件夹下面的 gif 图，图片名称为 effect.gif（提示：可以通过 VS Code 或者浏览器预览 gif 图片）。

规定

- 请严格按照考试步骤操作，切勿修改考试默认提供项目中的文件名称、文件夹路径、class 名、id 名、图片名等，以免造成判题无法通过。
- 完成答题后，考生需将题目对应代码文件夹压缩成 zip 格式后上传提交，其他压缩包格式为 0 分。

判分标准

- 完成目标 1，得 5 分。
- 完成目标 2，得 10 分。
- 完成目标 3，得 5 分。

Markdown 文档解析

介绍

Markdown 因为其简洁的语法大受欢迎，已经成为大家写博客或文档时必备的技能点，众多博客平台都提倡用户使用 Markdown 语法进行文章书写，然后再发布后，实时的将其转化为常规的 HTML 页面渲染。

本题需要在已提供的基础项目中，使用 Nodejs 实现简易的 Markdown 文档解析器。

准备

开始答题前，需要先打开本题的项目代码文件夹，目录结构如下：

```
├── docs.md
├── images
│   └── md.jpg
├── index.html
└── js
    ├── index.js
    └── parse.js
```

其中：

- `index.html` 是主页面。
- `images` 是图片文件夹。
- `docs.md` 是需要解析的 Markdown 文件。
- `js/index.js` 是提供的工具脚本，用于快速验证代码结果。
- `js/parse.js` 是需要补充的脚本文件。

目标

在 `js/parse.js` 中实现几种特定的 Markdown 语法解析，目前初始文件中已实现标题解析（即从 `#` 前缀转换为 `<h1>` 标签），请你继续完善该文件 TODO 部分，完成剩余语法解析操作，具体需求如下：

1. 对分隔符进行解析，Markdown 中使用 --- （三条及以上的短横线） 作为分隔符，将其解析成为 <hr> 标签：

```
<!-- Markdown -->
-----
```

```
<!-- 对应 HTML -->
<hr />
```

2. 对引用区块进行解析，Markdown 中使用 > 作为前缀，将其解析成为 <blockquote> 标签：

```
<!-- Markdown 单个引用区块 -->
> 引用区块1
<!-- Markdown 引用区块 -->
> 多级引用区块2
> 多级引用区块2
```

```
<!-- 对应 HTML -->
<blockquote>
  <p>引用区块1</p>
</blockquote>
```

```
<blockquote>
  <p>多级引用区块2</p>
  <p>多级引用区块2</p>
</blockquote>
```

3. 对无序列表进行解析，Markdown 中使用 * 或者 - 作为前缀，将其解析成为 标签：

```
<!-- Markdown -->
* 无序列表
* 无序列表
* 无序列表
或者：
- 无序列表
- 无序列表
- 无序列表
```

```
<!-- 对应 HTML -->
<ul>
  <li>无序列表</li>
  <li>无序列表</li>
  <li>无序列表</li>
</ul>
```

4. 对图片进行解析，Markdown 中使用 ![alt](link) 表示，将其解析成为 标签：

```
<!-- Markdown -->
![图片](./images/md.jpg)

<!-- 对应 HTML -->

```

5. 对文字效果进行解析，比如粗体效果，和行内代码块，将其分别解析成 `` 和 `code` 标签：

```
<!-- Markdown -->
这是**粗体**的效果文字，这是内嵌的`代码行`

<!-- 对应 HTML -->
这是<b>粗体</b>的效果文字，这是内嵌的<code>代码行</code>
```

在验证代码效果时，你可以在终端运行：

```
node ./js/index.js
```

程序会将解析的结果输出到 `index.html` 文件中，然后通过浏览器查看输出的 `index.html` 是否符合解析要求（注意：程序不会实时的将结果更新到 `index.html` 文件中，在你的代码变更后，请重新执行上述命令）。

在题目所提供的数据的情况下，完成后的效果如下：



规定

- 请勿修改 `js/parse.js` 文件外的任何内容。
- 请严格按照考试步骤操作，切勿修改考试默认提供项目中的文件名称、文件夹路径、class 名、id 名、图片名等，以免造成判题无法通过。
- 完成答题后，考生需将题目对应代码文件夹压缩成 zip 格式后上传提交，其他压缩包格式为 0 分。

判分标准

- 完成目标 1，得 5 分。
- 完成目标 2，得 5 分。
- 完成目标 3，得 10 分。
- 完成目标 4 与 目标 5，得 5 分。

组课神器

介绍

在很多教育网站的平台上，课程的章节目录会使用树型组件呈现，为了方便调整菜单，前端工程师会为其赋予拖拽功能。本题需要在已提供的基础项目中，完成可拖拽树型组件的功能。

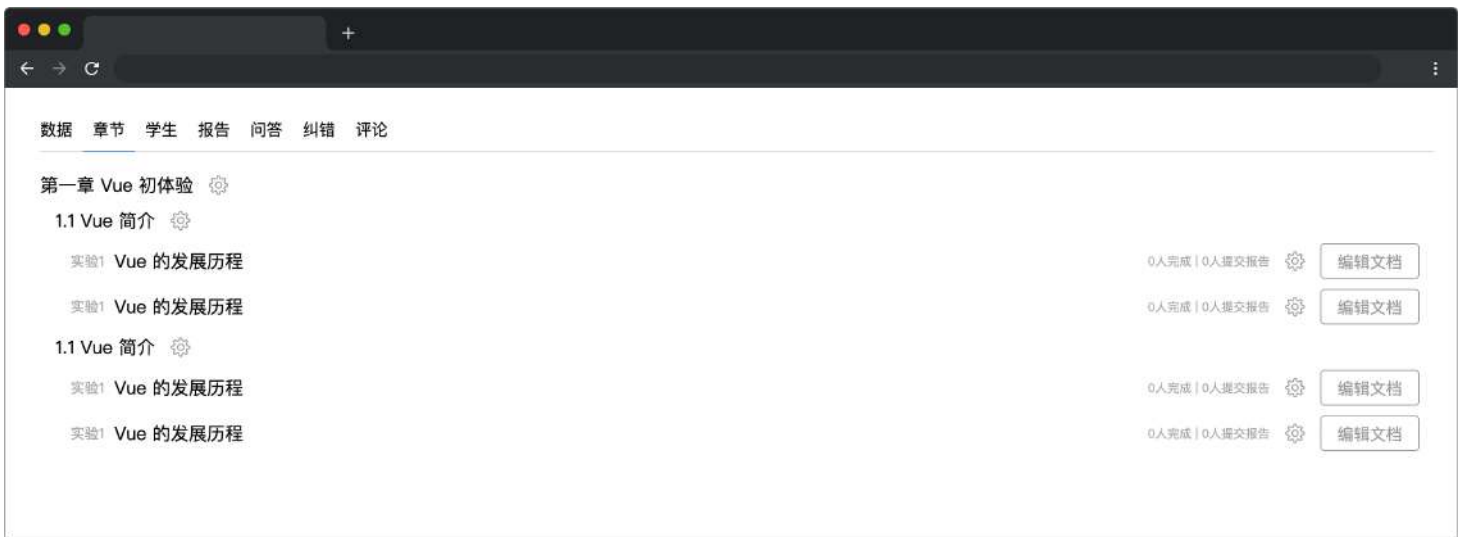
准备

```
|— effect.gif
|— css
|   |— style.css
|— index.html
|— images
|— js
|   |— data.json
|   |— axios.min.js
|   |— index.js
```

其中：

- index.html 是主页面。
- images 是图片文件夹。
- js/index.js 是需要补充代码的 js 文件。
- js/data.json 是存放数据的 json 文件。
- js/axios.min.js 是 axios 文件。
- css/style.css 是样式文件。
- effect.gif 是完成的效果图。

在浏览器中预览 index.html 页面，显示如下所示：



目标

请在 `js/index.js` 文件中补全代码。具体要求如下：

1. 补全 `js/index.js` 文件中 `ajax` 函数，功能为根据请求方式 `method` 不同，拿到树型组件的数据并返回。具体如下：
 - 当 `method === "get"` 时，判断 `localStorage` 中是否存在 `key` 为 `data` 的数据，若存在，则从 `localStorage` 中直接获取后处理为 `json` 格式并返回；若不存在则从 `./js/data.json`（必须使用该路径请求，否则可能会请求不到数据）中使用 `ajax` 获取并返回。
 - 当 `method === "post"` 时，将通过参数 `data` 传递过来的数据转化为 `json` 格式的字符串，并保存到 `localStorage` 中，`key` 命名为 `data`。

最终返回的数据格式如下：

```
[
  {
    "id": 1001,
    "label": "第一章 Vue 初体验",
    "children": [ ... ]
  },
  {
    "id": 1006,
    "label": "第二章 Vue 核心概念",
    "children": [
      {
        "id": 1007,
        "label": "2.1 概念理解",
        "children": [
          {
            "id": 1008,
            "label": "聊一聊虚拟 DOM",
            "tag": "文档 1"
          },
          ...
        ]
      },
      {
        "id": 1012,
        "label": "2.2 Vue 基础入门",
        "children": [
          {
            "id": 1013,
            "label": "Vue 的基本语法",
            "tag": "实验 6"
          },
          ...
        ]
      }
    ]
  }
]
}
```

2. 补全 `js/index.js` 文件中的 `treeMenusRender` 函数，使用所传参数 `data` 生成指定 DOM 结构的模板字符串（完整的模板字符串的 HTML 样例结构可以在 `index.html` 中查看），并在包含 `.tree-node` 的元素节点上加上指定属性如下：

属性名	属性值	描述
<code>data-grade</code>	<code>\${grade}</code>	表示菜单的层级，整数，由 <code>treeMenusRender</code> 函数的 <code>grade</code> 参数值计算获得，章节是 1，小节是 2，实验文档是 3。
<code>data-index</code>	<code>\${id}</code>	表示菜单的唯一 id，使用每层菜单数据的 <code>id</code> 字段值。

3. 补全 js/index.js 文件中的 treeDataRefresh 函数，功能为：根据参数列表

{ dragGrade, dragElementId }, { dropGrade, dropElementId } 重新生成拖拽后的树型组件数据 treeData （treeData 为全局变量，直接访问并根据参数处理后重新赋值即可）。

方便规则描述，现将 data.json 中的数据扁平化处理，得到的数据顺序如下：

```
[
  { grade: "1", label: "第一章 Vue 初体验", id: "1001" },
  { grade: "2", label: "1.1 Vue 简介", id: "1002" },
  { grade: "3", label: "Vue 的发展历程", id: "1003" },
  { grade: "3", label: "Vue 特点", id: "1004" },
  { grade: "3", label: "一分钟上手 Vue", id: "1005" },
  { grade: "1", label: "第二章 Vue 核心概念", id: "1006" },
  { grade: "2", label: "2.1 概念理解", id: "1007" },
  { grade: "3", label: "聊一聊虚拟 DOM", id: "1008" },
  { grade: "3", label: "感受一下虚拟 DOM", id: "1009" },
  { grade: "3", label: "聊一聊 MVVM 设计模式", id: "1010" },
  { grade: "3", label: "Vue 中的 MVVM 设计模式", id: "1011" }, // 即将被拖拽的元素节点
  { grade: "2", label: "2.2 Vue 基础入门", id: "1012" },
  { grade: "3", label: "Vue 的基本语法", id: "1013" },
  { grade: "3", label: "第一步，创建 Vue 应用实例", id: "1014" },
];
```

拖拽前后的规则说明如下：

- 情况一：若拖拽的节点和放置的节点为同级，即 treeDataRefresh 函数参数列表中 dragGrade == dropGrade，则将 id == dragElementId（例如：1011）的节点移动到 id==dropElementId（例如：1008）的节点后，作为其后第一个邻近的兄弟节点。最终生成的数据顺序如下：


```
[
  { grade: "1", label: "第一章 Vue 初体验", id: "1001" },
  { grade: "2", label: "1.1 Vue 简介", id: "1002" },
  { grade: "3", label: "Vue 的发展历程", id: "1003" },
  { grade: "3", label: "Vue 特点", id: "1004" },
  { grade: "3", label: "一分钟上手 Vue", id: "1005" },
  { grade: "1", label: "第二章 Vue 核心概念", id: "1006" },
  { grade: "2", label: "2.1 概念理解", id: "1007" },
  { grade: "3", label: "聊一聊虚拟 DOM", id: "1008" },
  // 在目标元素节点下方插入
  { grade: "3", label: "Vue 中的 MVVM 设计模式", id: "1011" },
  { grade: "3", label: "感受一下虚拟 DOM", id: "1009" },
  { grade: "3", label: "聊一聊 MVVM 设计模式", id: "1010" },
  // 移除被拖拽的元素节点
  { grade: "2", label: "2.2 Vue 基础入门", id: "1012" },
  { grade: "3", label: "Vue 的基本语法", id: "1013" },
  { grade: "3", label: "第一步, 创建 Vue 应用实例", id: "1014" },
];
```

- 情况二：若拖拽的节点和放置的节点为上下级，即 `treeDataRefresh` 函数参数列表中 `dragGrade - dropGrade == 1`，则将 `id == dragElementId`（例如：1011）的节点移动到 `id==dropElementId`（例如：1002）的节点下，并作为其第一个子节点。最终生成的数据顺序如下：

```
[
  { grade: "1", label: "第一章 Vue 初体验", id: "1001" },
  { grade: "2", label: "1.1 Vue 简介", id: "1002" },
  // 在目标元素节点下方插入
  { grade: "3", label: "Vue 中的 MVVM 设计模式", id: "1011" },
  { grade: "3", label: "Vue 的发展历程", id: "1003" },
  { grade: "3", label: "Vue 特点", id: "1004" },
  { grade: "3", label: "一分钟上手 Vue", id: "1005" },
  { grade: "1", label: "第二章 Vue 核心概念", id: "1006" },
  { grade: "2", label: "2.1 概念理解", id: "1007" },
  { grade: "3", label: "聊一聊虚拟 DOM", id: "1008" },
  { grade: "3", label: "感受一下虚拟 DOM", id: "1009" },
  { grade: "3", label: "聊一聊 MVVM 设计模式", id: "1010" },
  // 移除被拖拽的元素节点
  { grade: "2", label: "2.2 Vue 基础入门", id: "1012" },
  { grade: "3", label: "Vue 的基本语法", id: "1013" },
  { grade: "3", label: "第一步, 创建 Vue 应用实例", id: "1014" },
];
```

最终效果可参考文件夹下面的 gif 图，图片名称为 `effect.gif`（提示：可以通过 VS Code 或者浏览器预览 gif 图片）。

规定

- 请勿修改 `js/index.js` 文件外的任何内容。
- 请严格按照考试步骤操作，切勿修改考试默认提供项目中的文件名称、文件夹路径、class 名、id 名、图片名等，以免造成判题无法通过。
- 完成答题后，考生需将题目对应代码文件夹压缩成 zip 格式后上传提交，其他压缩包格式为 0 分。

判分标准

- 完成目标 1，得 5 分。
- 完成目标 2，得 10 分。
- 完成目标 3，得 10 分。