

# 第十三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。

# 决赛（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 分一分（5 分）
- 02 新鲜的蔬菜（5 分）
- 03 水果消消乐（10 分）
- 04 用什么来做计算 A（10 分）
- 05 开学礼物大放送（15 分）
- 06 权限管理（15 分）
- 07 一起会议吧（20 分）
- 08 天气趋势 A（20 分）
- 09 JSON 生成器（25 分）
- 10 商城管理系统（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 或 rar 格式后上传提交，其他格式一律不得分。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，不要嵌套压缩包，不要使用其他压缩格式。
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。

# 分一分

## 介绍

如果给你一个数组，你能很快将它分割成指定长度的若干份吗？

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，达到分割数组的要求。

## 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：

```
|— effect.gif
|— index.html
|— js
  |— index.js
```

其中：

- effect.gif 是最终实现的效果图。
- index.html 是主页面。
- js/index.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：



初始效果

## 目标

请在 js/index.js 文件中补全函数 splitArray 中的代码，最终返回按指定长度分割的数组。

具体要求如下：

1. 将待分割的（一维）数组**升序**排序。
2. 将排序后的数组从**下标为 0** 的元素开始，按照从 `id=sliceNum` 的输入框中获取到的数值去分割，并将分割好的数据存入一个新数组中。如：输入框中值为 `n`，将原数组按每 `n` 个一组分割，不足 `n` 个的数据为一组。
3. 将得到的新数组返回（即 `return` 一个二维数组）。

例如：

```
var arr = [3, 1, 4, 2, 5, 6, 7];
// 分割成每 1 个一组
var newA = splitArray(arr, 1);
console.log(newA); // => [[1],[2],[3],[4],[5],[6],[7]]

// 分割成每 2 个一组
newA = splitArray(arr, 2);
console.log(newA); // => [[1,2],[3,4],[5,6],[7]]

// 分割成每 4 个一组
newA = splitArray(arr, 4);
console.log(newA); // => [[1,2,3,4],[5,6,7]]

// 分割成每 7 个一组
newA = splitArray(arr, 7);
console.log(newA); // => [[1,2,3,4,5,6,7]]
```

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。

完成后的效果见文件夹下面的 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，以免造成无法判题通过。

## 判分标准

- 完全实现题目目标得满分，否则得 0 分。

# 新鲜的蔬菜

## 介绍

厨房里新到一批蔬菜，被凌乱地堆放在一起，现在我们给蔬菜分下类，把相同的蔬菜放到同一个菜板上，拿给厨师烹饪美味佳肴吧。

## 准备

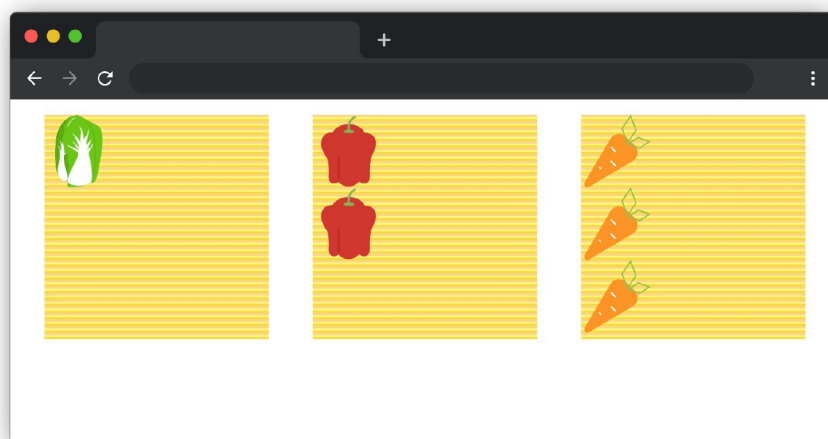
开始答题前，需要先打开本题的项目代码文件夹，目录结构如下：

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

其中：

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

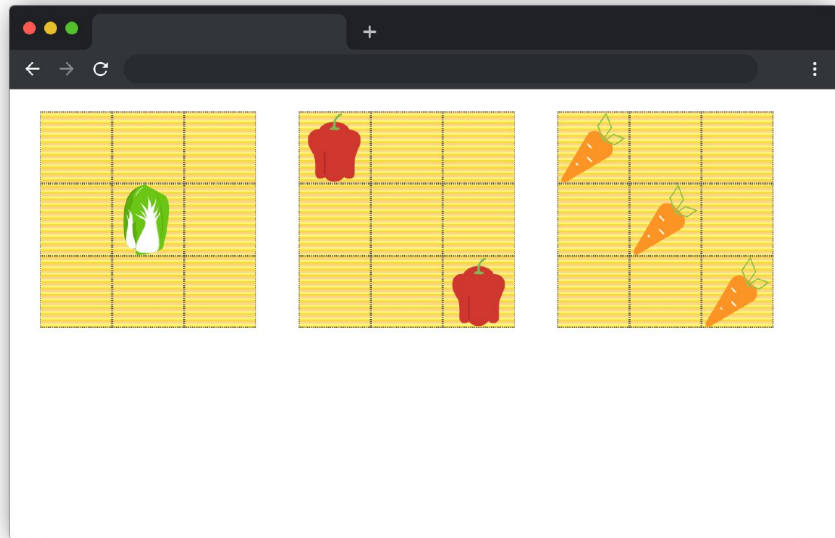
在浏览器打开后，效果如下：



初始效果

## 目标

完成 `css/style.css` 中的 TODO 部分。所有元素的大小都已给出，无需修改，完成后效果如下（图中灰色线条为布局参考线无需实现）：



布局位置示意图

## 规定

- 请勿修改页面中所有给出元素的大小，以免造成无法判题通过。
- 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，以免造成无法判题通过。

## 判分标准

- 本题完全实现题目目标得满分，否则得 0 分。

# 水果消消乐

## 介绍

消消乐是一款益智类休闲游戏，在排队等待做核酸的时候可以打开手机玩一会，陪你度过这漫长且无聊的等待期。

本题需要实现一款水果消消乐的游戏，在 4 X 4 的格子上，消除相同的水果得两分，否则扣两分，考验一下大家的记忆力，看看最后谁的分数最多。

## 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：

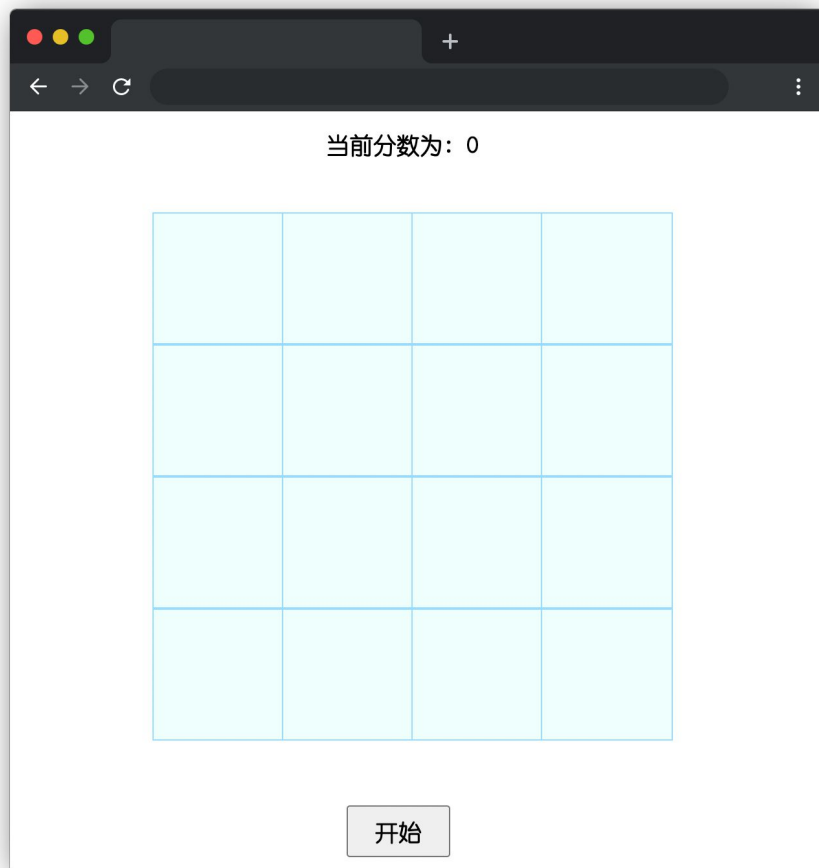
```
├── css
│   └── style.css
├── images
│   ├── apple.png
│   ├── cherry.png
│   ├── grape.png
│   ├── peach.png
│   ├── pear.png
│   ├── strawberry.png
│   ├── tangerine.png
│   └── watermelon.png
├── js
│   ├── index.js
│   └── jquery-3.6.0.min.js
├── effect.gif
└── index.html
```

其中：

- `css/style.css` 是样式文件。
- `images` 是图片文件夹。
- `js/index.js` 是实现游戏功能的 js 文件。
- `js/jquery-3.6.0.min.js` 是 jQuery 文件。
- `effect.gif` 是最终实现的效果动图。
- `index.html` 是主页面。

在浏览器中预览 `index.html` 页面，点击开始按钮不能正常进行游戏，效果如下：





未实现效果的游戏界面

## 目标

请完善 `js/index.js` 文件。

完成后的效果见文件夹下面的 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）。

具体说明如下：

- 点击**开始**按钮后，该按钮被隐藏，方格上的图片显示后又隐藏。
- 点击方格，方格中的图片显示，页面显示两张图片后，比较图片是否相同。
- 如果图片上的水果相同，方格即可消除，并得 **2 分**；如果图片上的水果不相同，图片隐藏，并扣 **2 分**（分数可以为负数）。
- 在文本 “当前分数为：” 的冒号后面会**实时**统计当前的得分情况。

## 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，以免造成无法判题通过。

## 判分标准

- 本题完全实现题目目标得满分，否则得 0 分。

# 用什么来做计算 A

## 介绍

古以算盘作为计算工具。算盘常为木制矩框，内嵌珠子数串，定位拨珠，可做加减乘除等运算。站在前人的肩膀上，后人研究出计算器，便利了大家的生活，我们不用带着笨重的计算工具出门，打开手机上的计算器就可以了。

本题需要实现一个可以进行加减乘除等运算的计算器。

## 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：

```
|— css
|   |— style.css
|— js
|   |— index.js
|— effect.gif
|— index.html
```

其中：

- `css/style.css` 是样式文件。
- `js/index.js` 是实现计算功能的 js 文件。
- `effect.gif` 是最终实现的效果动图。
- `index.html` 是主页面。

在浏览器中预览 `index.html` 页面，点击计算器上的按钮，不能进行计算（图上的红色文字只是对计算器的功能区域进行说明，无需实现），效果如下：



未实现计算功能的计算器界面

## 目标

请完善 `js/index.js` 文件，当鼠标点击计算器上的按钮时，能够正常进行计算。

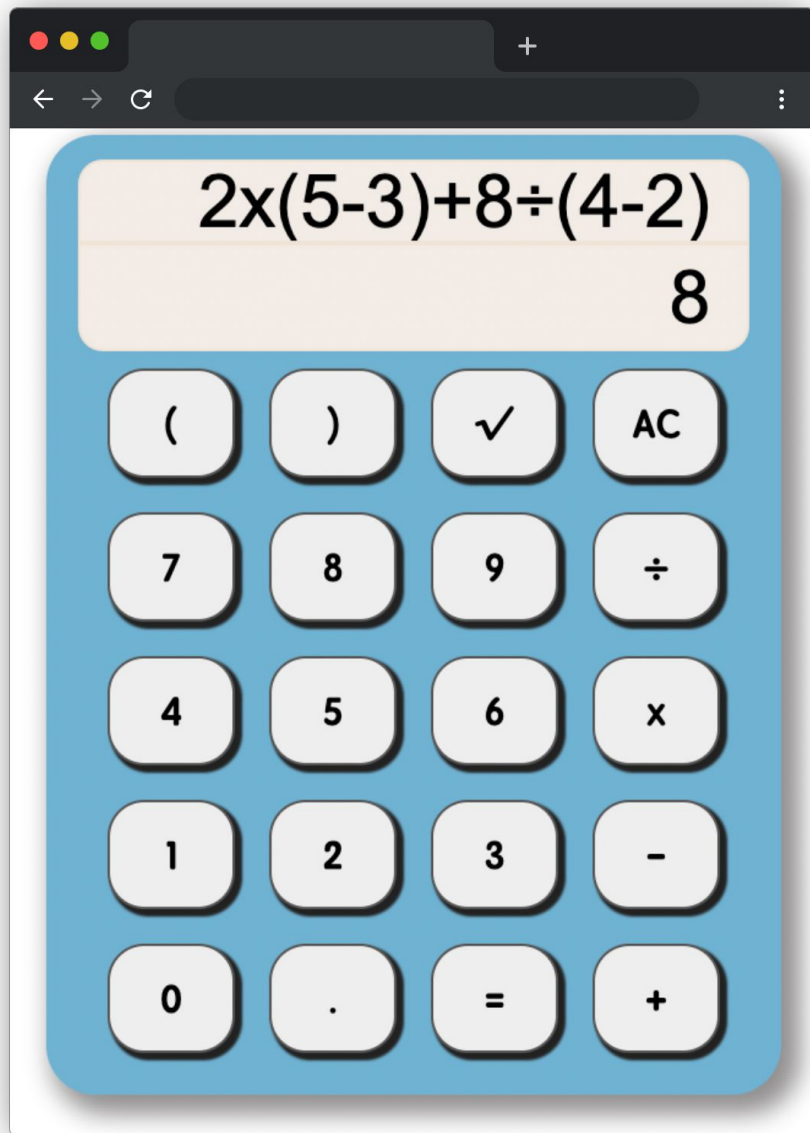
完成后的效果见文件夹下面的 gif 图，图片名称为 `effect.gif`（提示：可以通过 VS Code 或者浏览器预览 gif 图片）。

具体说明如下：

- 点击按钮会在计算式子区域显示当前输入的计算式子，当点击**等号 (=)** 后，在结果显示区域应该显示出正确的结果。
- 计算器需要具有**加 (+)**、**减 (-)**、**乘 (x)**、**除 (÷)**、**开二次方 (√)**、**重置 (AC)**、**小数点 (.)**、**括号运算**这八个功能。
- 计算器的计算遵循四则混合运算的法则，括号的优先级最高，其次是乘除，加减的优先级最低。

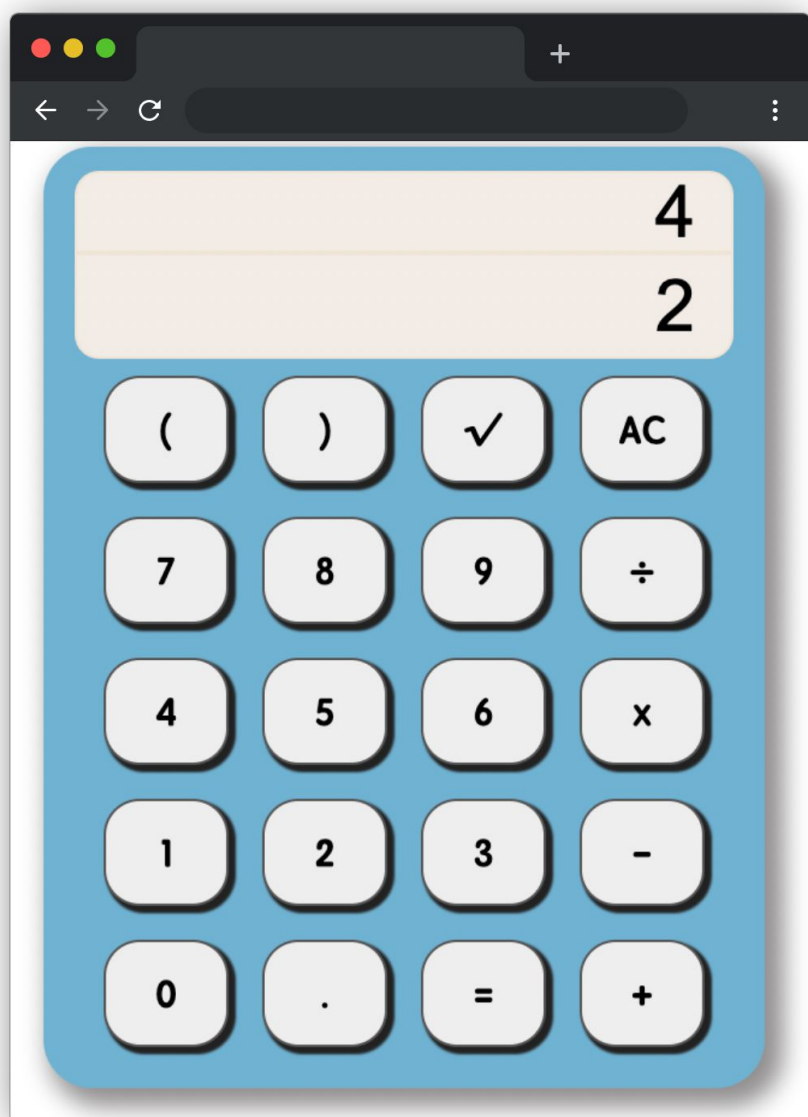
效果说明如下：

- 能够正确进行加、减、乘、除的混合运算，样例如下所示：

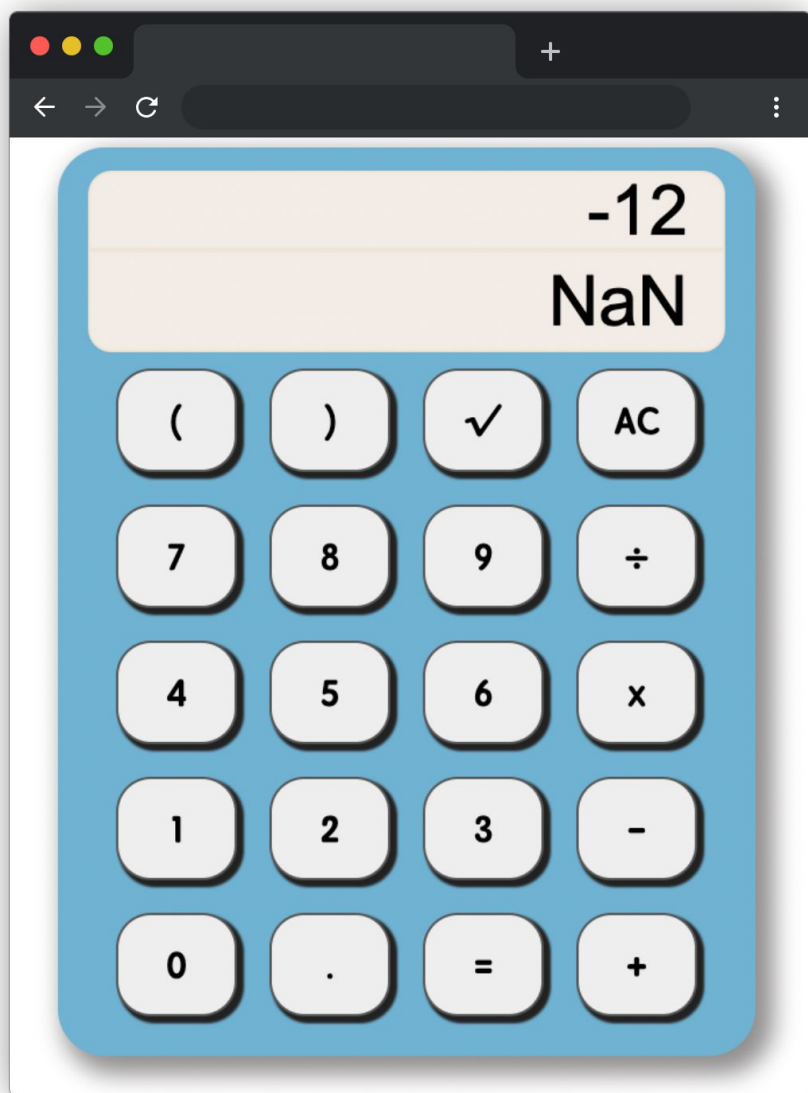


加减乘除混合运算样例图

- 能够正确进行开二次方，输入一个可以平方的数，点击开方（ $\sqrt{\quad}$ ）按钮即可在结果显示区域显示结果。如果输入的值是不能平方，结果显示 NaN，样例如下所示：

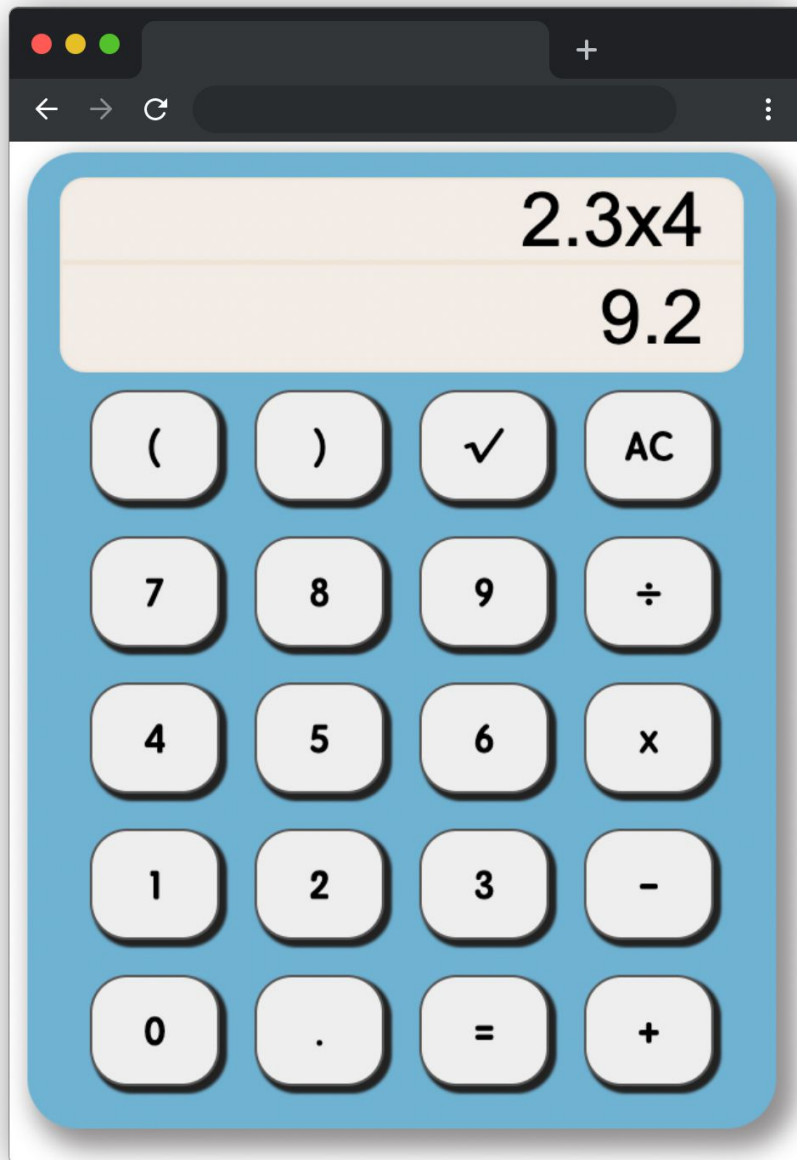


开方运算样例图



开方运算样例图

- 点击重置按钮（AC），可以清空计算式子显示区域和结果显示区域的值。
- 能够进行小数的加减乘除运算，样例如下所示：



小数运算样例图

## 规定

- 请勿修改除 `js/index.js` 文件以外任何文件的内容。
- 请注意，在 `js/index.js` 文件中编写代码时，必须使用文件已经给出的 class 名、id 名、标签来进行 DOM 操作。
- 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、文件夹路径等，以免造成无法判题通过。

## 判分标准

- 实现重置（AC）功能，得 1 分。



- 实现计算式子和结果显示功能，得 3 分。
- 实现计算功能，得 6 分。

# 开学礼物大放送

## 介绍

又是一年开学季，蓝桥为大家准备了开学礼物，想制作一个页面来宣传一下该活动。

本题需要按照要求完成一个以“开学季”为主题的页面布局。

## 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
├── images
├── mark
│   ├── preview
│   └── index.html
└── index.html
```

其中：

- `css/style.css` 是样式文件。
- `images` 是页面布局需要用到的图片素材。
- `mark/preview` 是页面效果图。
- `mark/index.html` 是布局参数标记页面。
- `index.html` 是主页面。

## 目标

请完善 `css/style.css` 和 `index.html` 文件。

请根据 `mark/preview` 最终效果图和 `mark/index.html` 上的参数标注来完成页面布局。

在浏览器打开 `mark/index.html` 页面，鼠标点击页面可以在右侧看到相应的参数标注。

## 规定

- 本题只能使用项目文件夹中提供的素材。

- 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、文件夹路径等，以免造成无法判题通过。

## 判分标准

- 本题根据页面布局的相似度给分，低于 50% 得 0 分，高于 50% 则按比例给分。

# 权限管理

## 介绍

你有没有想过，在我们日常浏览的网页中，那些新闻或者商品内容是如何被输入到数据库中的呢？大家虽然没有用过，但是肯定听过“后台管理系统”，运营人员就是通过它将批量的商品上传到数据库并呈现在网页中的。那么随便一个人就可以通过该管理系统操作这些商品数据吗？当然不行，这就涉及到了权限管理问题。

本题需要在已提供的基础项目中使用 JS/jQuery 知识完善代码，最终实现权限管理的功能。

## 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
├─ index.html
├─ css
├─ js
│   ├── index.js
│   ├── jquery-3.6.0.min.js
│   └─ userList.json
```

其中：

- effect.gif 是最终实现的效果图。
- index.html 是主页面。
- css 是样式文件夹。
- js/index.js 是需要补充代码的 js 文件。
- js/jquery-3.6.0.min.js 是 jQuery 文件。
- js/userList.json 是需要请求的数据文件。

在浏览器中预览 index.html 页面，显示如下所示：



初始效果

## 目标

请在 `js/index.js` 文件中补全代码，最终实现**管理用户权限**的功能。

具体需求如下：

### 1. 实现异步数据读取和渲染功能。

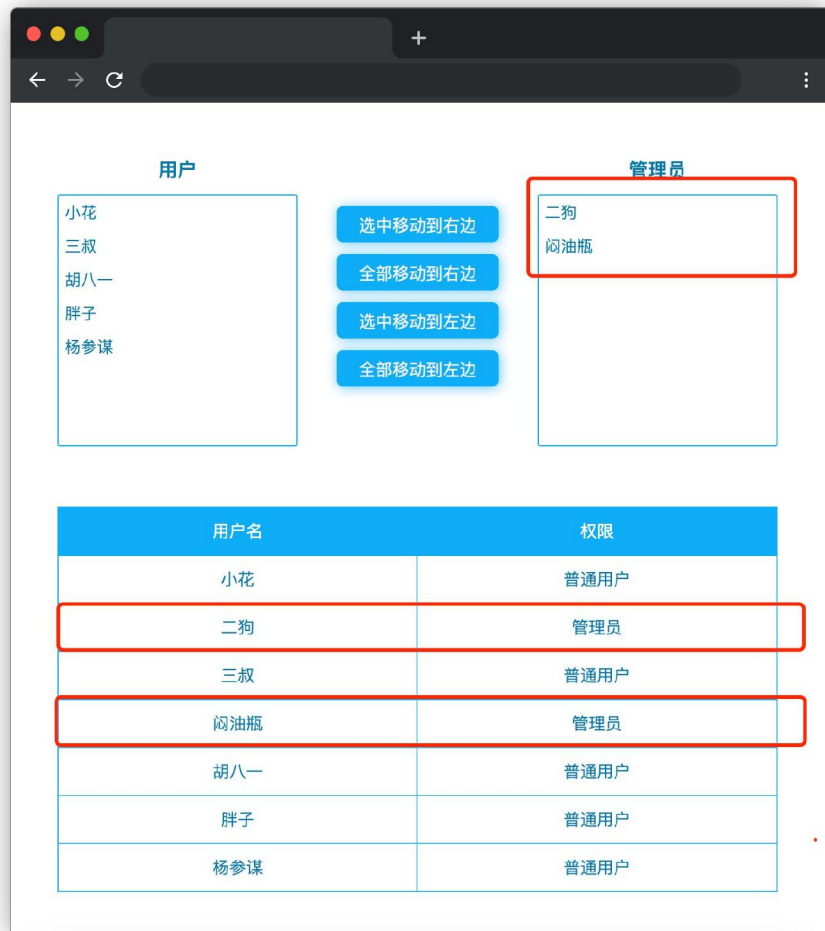
- 使用 ajax 异步获取 `./js/userList.json` 中的用户数据，并以正确的方式显示在页面下的用户权限表格中（注意：调试完成后请将请求地址写死为 `./js/userList.json`）。
- 页面初始化时，左边用户列表显示 7 个用户（**页面数据和结构不能随意篡改**），右边的管理员列表无选项。效果如下：



ajax 获取数据后的初始效果

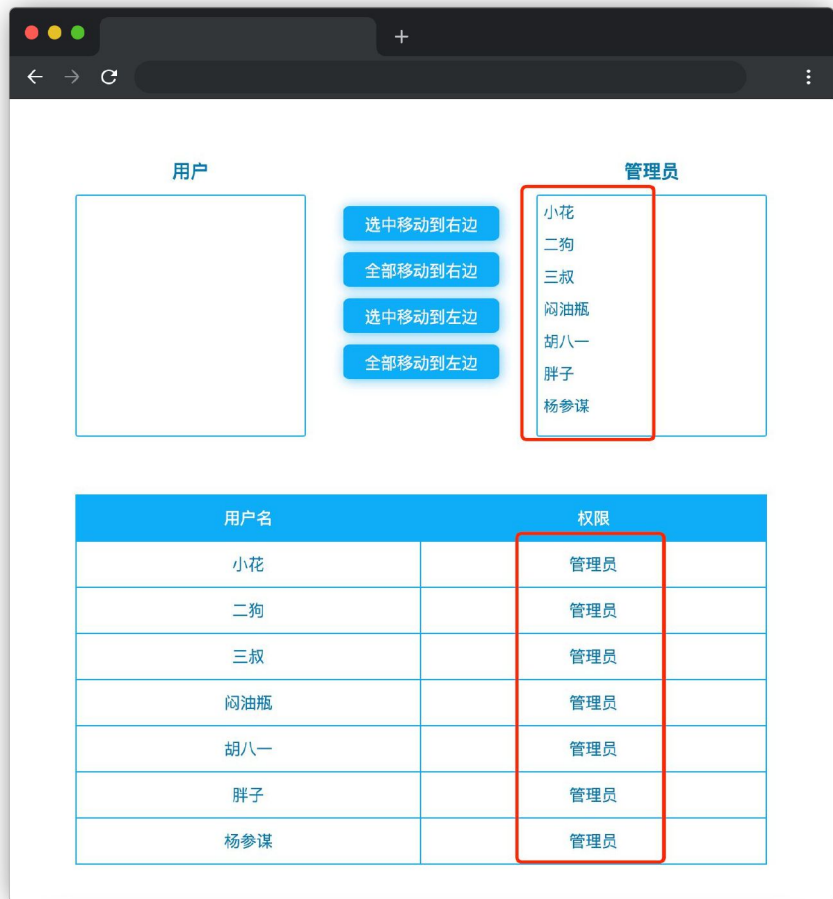
## 2. 实现左/右单个、多个或全部互移功能。

- 选中任意一个或多个（按 Ctrl 键可多选）左/右边**用户/管理员**列表中的用户，并点击**选中移动到右/左边**按钮，即可将选中的用户/管理员移动至另一边的**管理员/用户**列表中。效果如下：



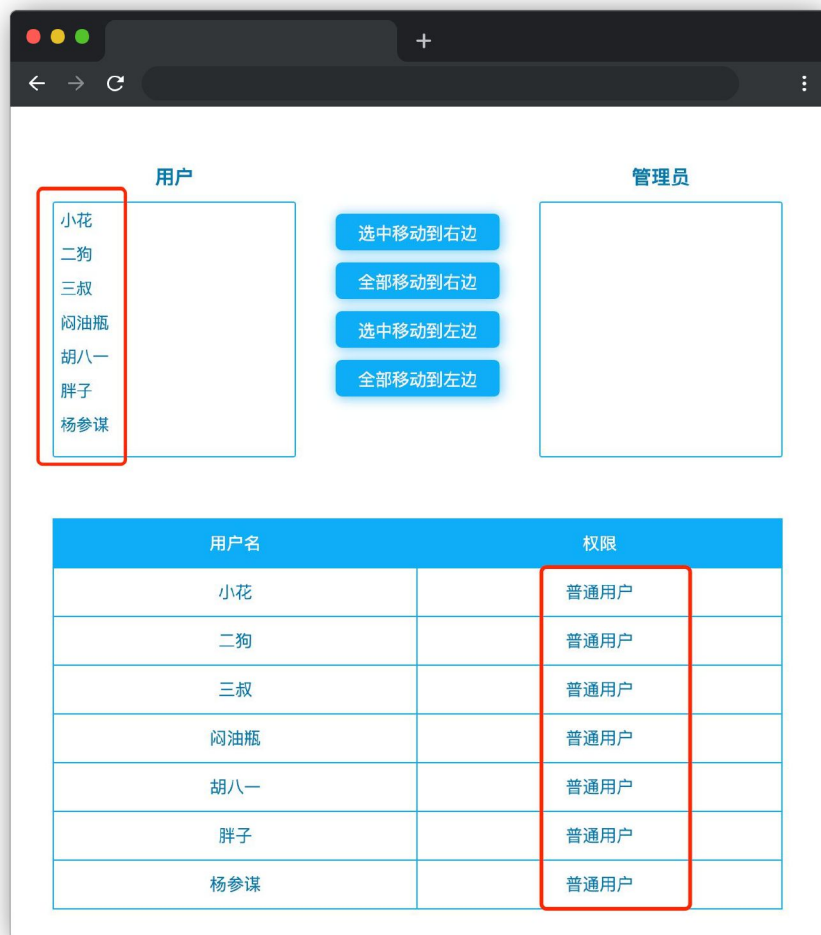
选中用户被移至右侧

- 点击**全部移动到右/左边**按钮，则将左/右边**用户/管理员**列表中的用户**全部移动**到右/左边的**管理员/用户**列表中（即：左/右边列表被清空）。效果如下：



左侧选择框内选项被全部移至右侧，所有用户权限均变为管理员





右侧选择框内选项被全部移至左侧，所有用户权限均变为普通用户

### 3. 实现用户权限表格中的权限，随用户移动发生变化功能。

- 页面下的**用户权限表格**中所有用户的权限，会根据页面上方所处列表不同发生相应改变，即**管理员**列表中的用户权限为**管理员**，用户列表中的用户权限为**普通用户**。

完成后的效果见文件夹下面的 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。
- 请严格按照上述要求来完成代码，页面中需要显示的文本内容，与具体说明里的内容保持一致，不要自定义，以免造成无法判题通过。
- 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，以免造成无法判题通过。

## 判分标准

- 实现异步数据读取和渲染功能，得 5 分。
- 实现左/右单个、多个或全部互移功能，得 6 分。
- 实现用户权限表格中权限变化功能，得 4 分。

# 一起会议吧

## 介绍

网络会议已经成为当下最流行的会议模式，为网络会议提供支持的当然是一些优秀的会议软件。

本题需要在已提供的基础项目中使用 Vue 2.x 知识完善代码，最终实现**网络会议软件中，参会人员列表**的几个展示效果。

## 准备

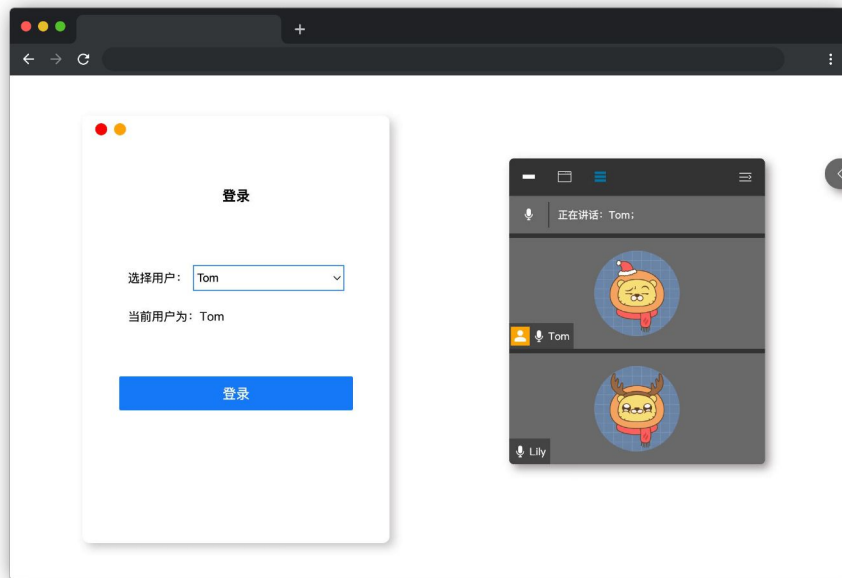
开始答题前，需要先打开本题的项目代码文件夹，目录结构如下：

```
|— effect.gif
|— index.html
|— css
|— images
└— js
    |— axios.min.js
    |— userList.json
    └— vue.js
```

其中：

- effect.gif 是最终实现的效果图。
- index.html 是主页面。
- css 是样式文件夹。
- images 是图片文件夹。
- js/axios.min.js 是 axios 文件。
- js/userList.json 是需要请求的数据文件。
- js/vue.js 是 Vue 2.x 文件。

在浏览器中预览 index.html 页面，显示如下所示：



初始效果

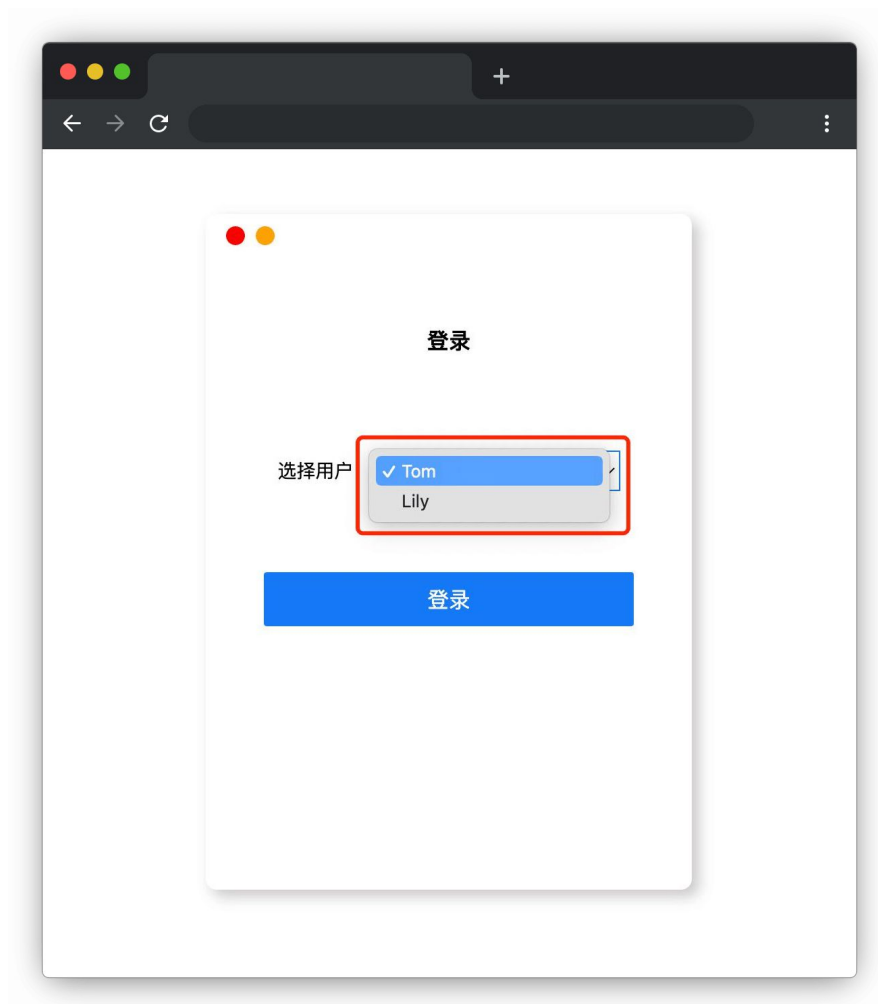
## 目标

请在 index.html 文件中补全代码，最终实现网络会议中参会人员列表的几个展示效果。

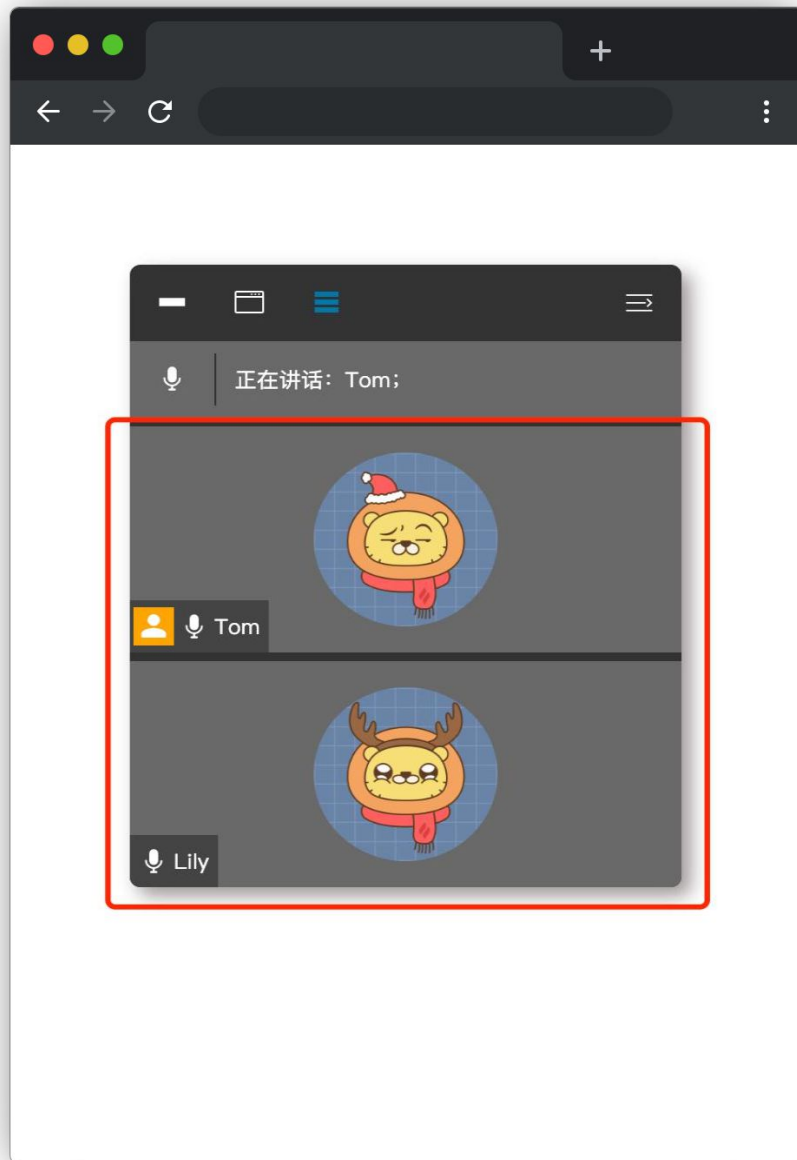
具体需求如下：

1. 实现异步数据读取和渲染功能。

- 使用 axios 异步获取 ./js/userList.json 中的用户数据（注意：调试完成后请将请求地址写死为 ./js/userList.json），并显示在登录窗口及参会人员窗口中。效果如下：



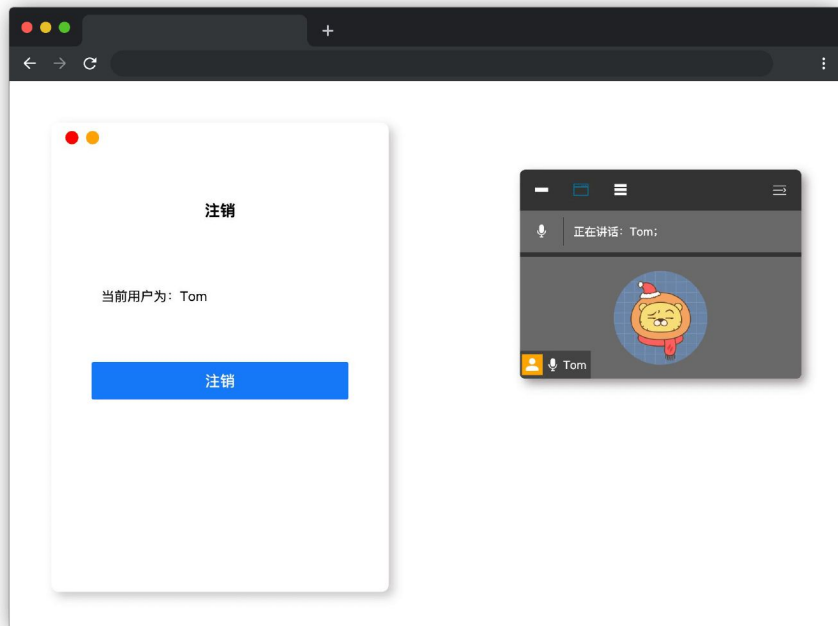
登录窗口异步数据渲染效果



参会人员窗口异步数据渲染效果

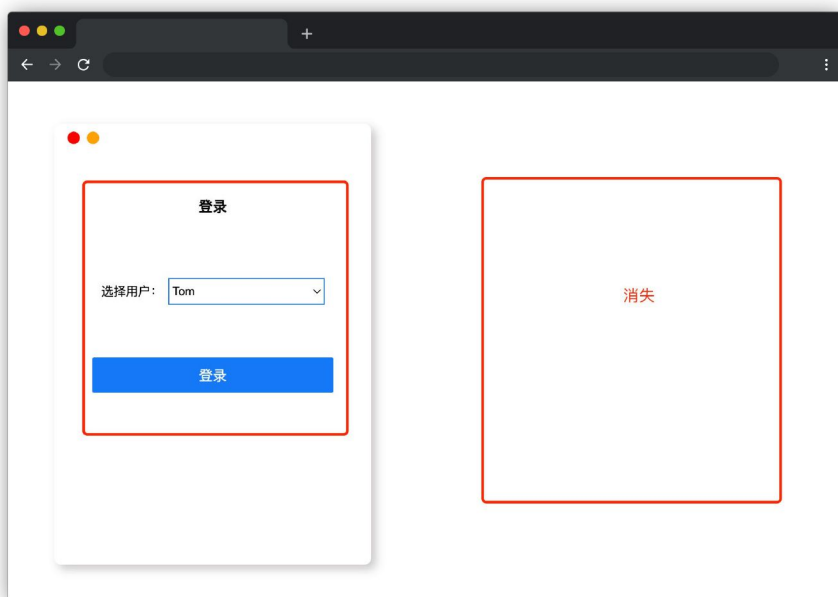
## 2. 实现登录、注销切换功能。

- 在登录窗口选取用户登录后，登录窗口切换为注销窗口，具体变化为：**登录标题变为注销字样**；**选择用户下拉框变为显示当前登录用户名**；**登录按钮变为注销按钮**。**参会人员窗口显示，并默认只显示当前登录用户信息**。效果如下：



登录后的效果

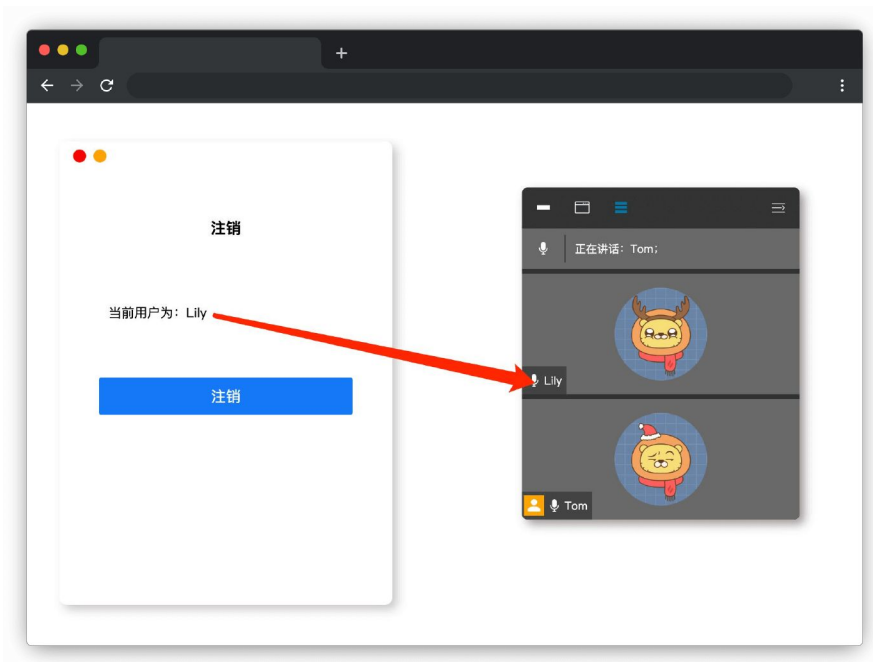
- 在注销窗口点击注销按钮后，注销窗口切换为登录窗口，参会人员窗口消失。效果如下：



注销后的效果图

### 3. 实现参会列表中的首位用户始终为登录用户功能。

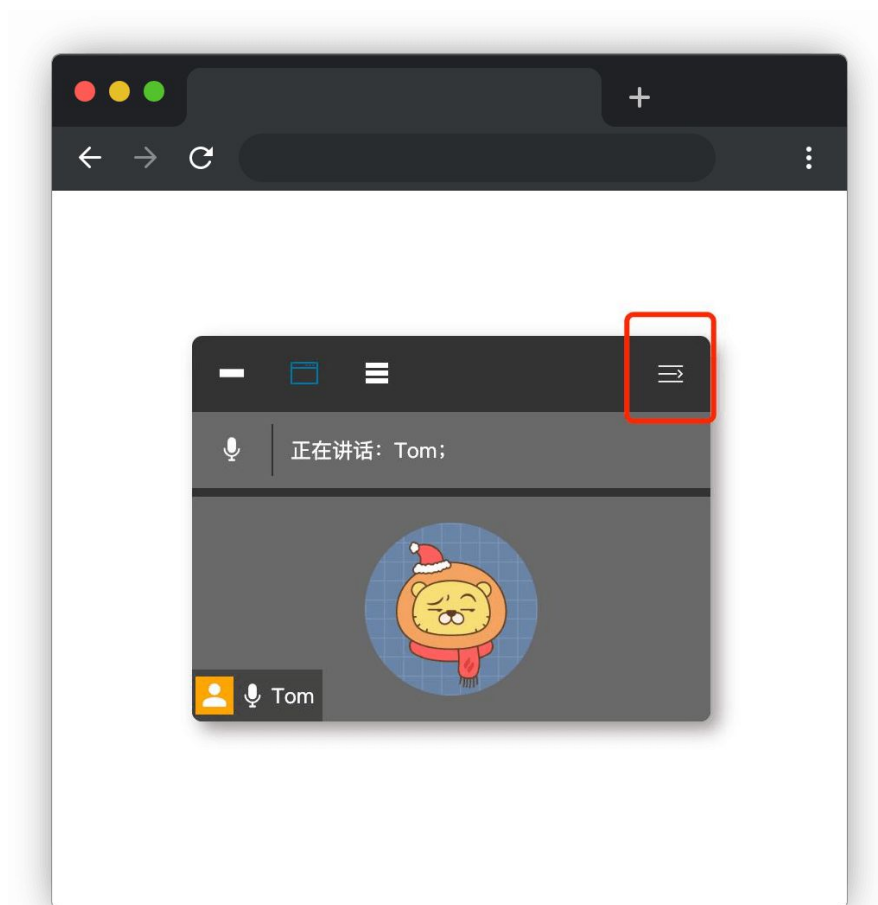
- 参会列表中的首位用户始终为登录用户。效果如下：



参会列表中的首位用户始终为登录用户

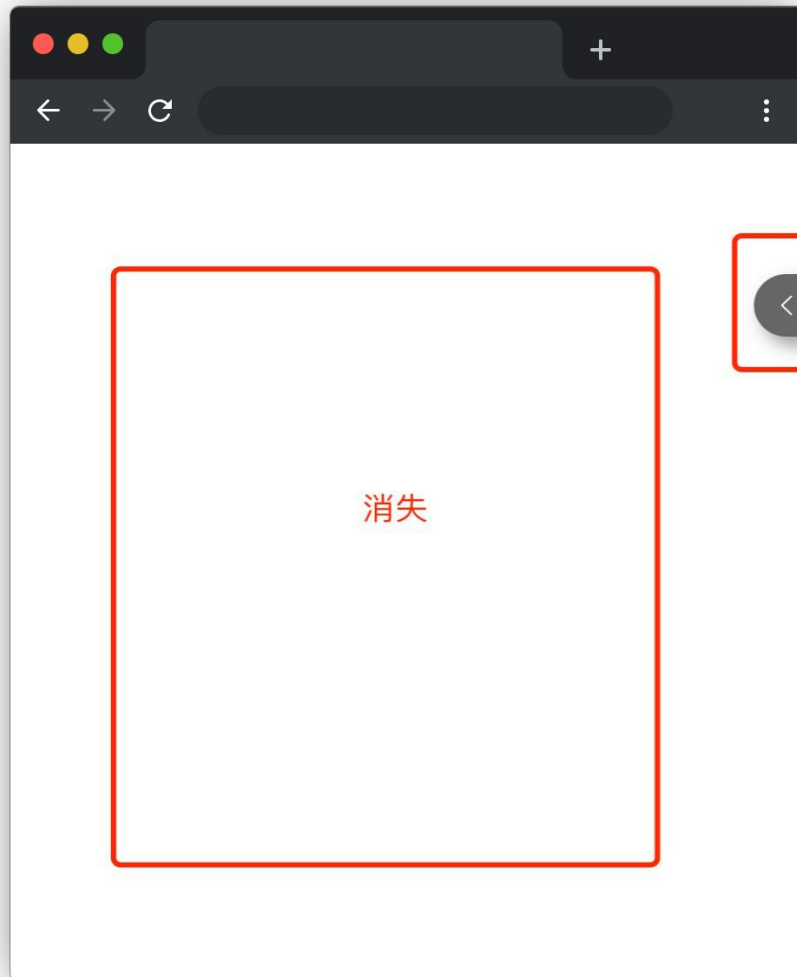
#### 4. 实现用户窗口显隐切换功能。

- 通过点击隐藏参会用户窗口或显示参会用户窗口按钮图标，可切换参会人员窗口的显隐。效果如下：



隐藏参会用户窗口按钮图标位置

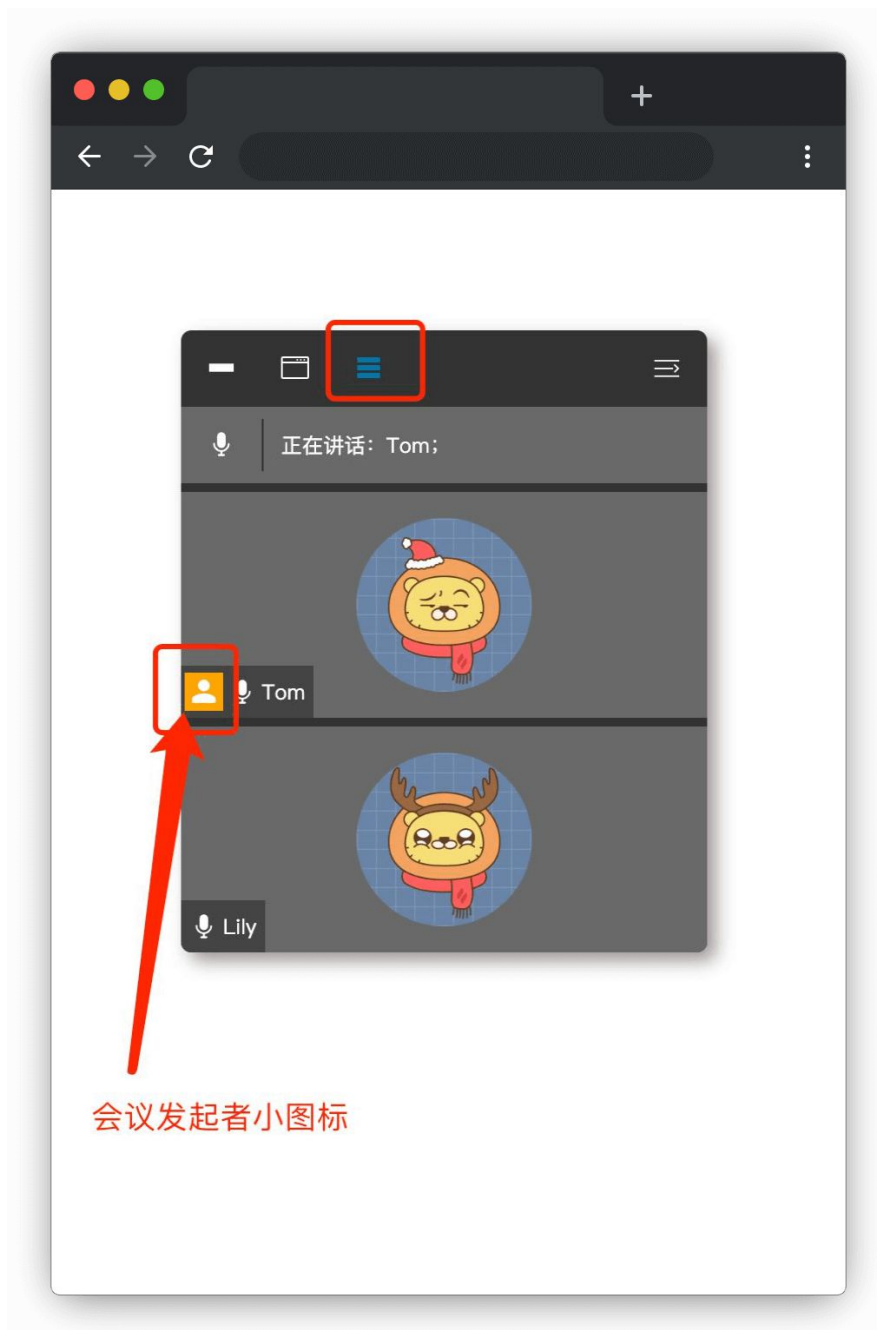




显示参会用户窗口按钮图标位置

##### 5. 实现用户列表显示效果切换功能。

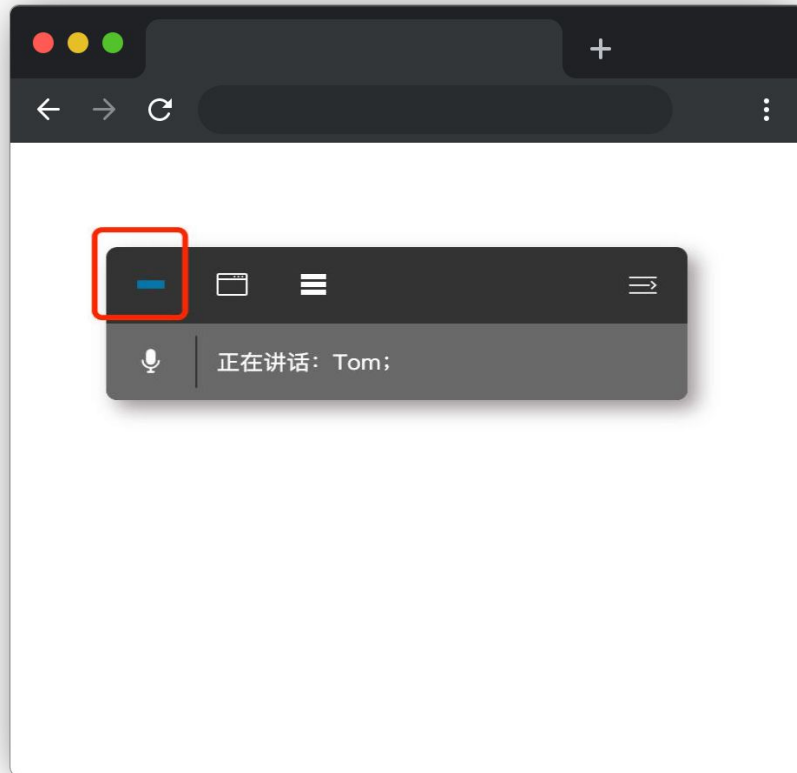
- 点击显示所有参会人员按钮图标，图标变色（即 `class=active`），参会人员窗口内的人员列表显示所有参会人员信息。效果如下：



显示所有参会人员效果图

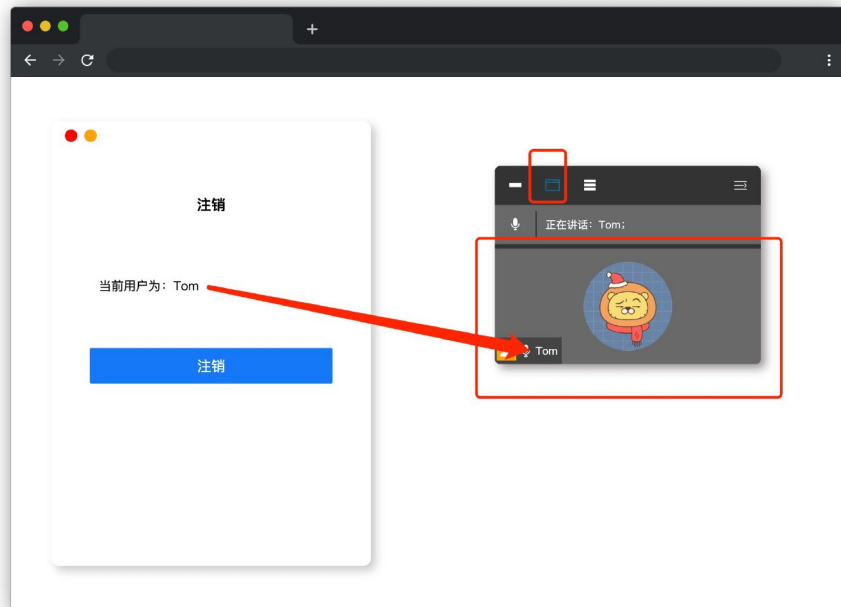
注意，参会人员列表中用户名称前面的黄色小图标，代表的是该成员为会议发起者，与当前登录用户无关。与之对应的 `./js/userList.json` 中的字段为 `isHost`，其中 `"isHost": true` 为会议发起人，`"isHost": false` 为普通参会成员。

- 点击最左侧的不显示参会人员列表的按钮图标，图标变色（即 `class=active`），参会人员窗口内的人员列表隐藏。效果如下：



不显示参会人员列表效果图

- 点击只显示当前登录用户的按钮图标，图标变色（即 `class=active`），参会人员窗口内的人员列表中只显示当前登录用户信息。效果如下：



只显示当前登录用户效果图

完成后的效果见文件夹下面的 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）。

注意：本题涉及到的登录/注销流程仅为当前题目需要而设计，非真实场景模拟。

## 规定

- 请勿修改 `index.html` 文件外的任何内容。
- 请严格按照上述要求来完成代码，页面中需要显示的文本内容，与具体说明里的内容保持一致，不要自定义，以免造成无法判题通过。
- 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，以免造成无法判题通过。

## 判分标准

- 实现异步数据读取功能，得 5 分。
- 实现登录、注销切换功能，得 4 分。
- 实现参会列表首位用户始终为登录用户功能，得 3 分。
- 实现用户窗口显隐切换功能，得 2 分。
- 实现用户列表显示效果切换功能，得 6 分。

# 天气趋势 A

## 介绍

日常生活中，气象数据对于人们的生活具有非常重要的意义，数据的表现形式多种多样，使用图表进行展示使数据在呈现上更加直观。

本题请实现一个 Y 城 2022 年的天气趋势图。

## 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-1.gif
├─ effect-2.gif
├─ index.html
└─ js
    ├── axios.js
    ├── echarts.min.js
    ├── vue.min.js
    └─ weather.json
```

其中：

- `css/style.css` 是样式文件。
- `index.html` 是主页面。
- `js/axios.js` 是 axios 文件。
- `js/vue.min.js` 是 vue2.x 文件。
- `js/echarts.min.js` 是 echarts 文件。
- `js/weather.json` 是请求需要用到的天气数据。
- `effect-1.gif` 是当月和未来七天切换的效果图。
- `effect-2.gif` 是最终完成的效果图。

在浏览器中预览 `index.html` 页面效果如下：

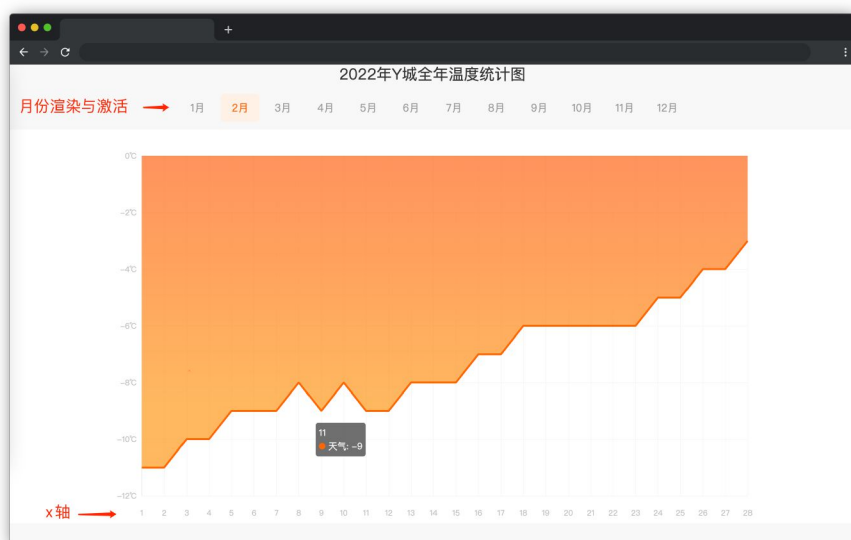


初始效果

## 目标

请在 index.html 文件中补全代码，具体需求如下：

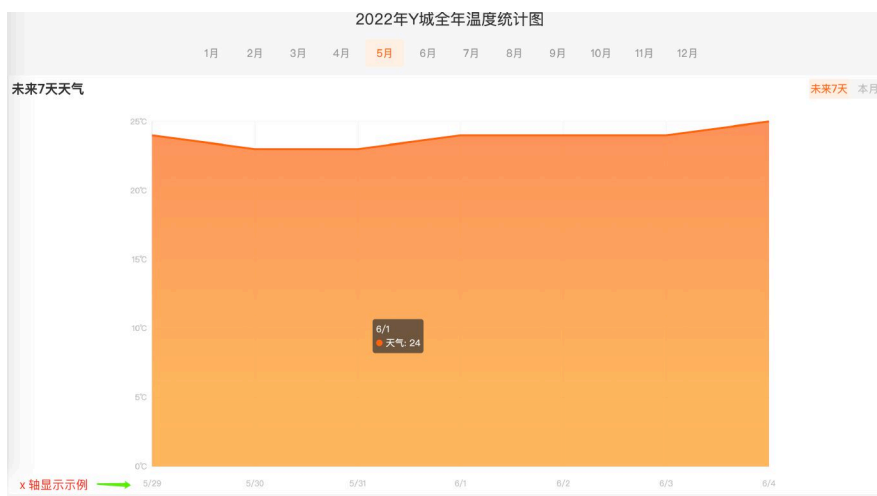
- 1.完成数据请求（数据来源 ./js/weather.json），weather.json 中存放的数据为 12 个月对应的温度数据。在项目目录下已经提供了 axios，考生可自行选择是否使用。注意：调试完成后请将请求地址写死为 ./js/weather.json。
- 2.把 data 中的月份数据 monthList，在 class=month 标签下面的 li 上完成渲染，点击月份则切换对应月份的温度数据同时被点击的月份会变成激活状态(.active 类)，x 轴为日期，y 轴为温度，默认显示 1 月份数据。



目标2示例图

3. 如果点击的月份是当天（通过时间函数动态获取的时间）所在月份，本月和未来七天切换的 tab（即 id=currentMonth 元素）显示，其他月份 currentMonth 元素不显示。
- 默认显示本月数据。
  - 点击**本月**显示当月数据，点击**未来七天**显示从当天(包含当天)开始未来七天的数据，当显示未来七天数据时 x 轴需要显示为月/日格式。
  - 点击 tab 上**本月**和**未来七天**会切换激活状态（.active）。

以当天为 5 月 29 号为例，未来七天 x 轴显示示例（即 x 轴显示成：5/29，5/30，5/31，6/1，6/2，6/3，6/4）：



目标3示例图

本月和未来七天 切换效果见文件夹下 effect-1.gif。

最终效果见文件夹下面的 gif 图，图片名称为 effect-2.gif（提示：可以通过 VS Code 或者浏览器预览 gif 图片）

## 规定

- 请求地址为 ./js/weather.json，切勿写成其他地址格式，以免造成无法判题通过。
- 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，以免造成无法判题通过。

## 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。



# JSON 生成器

## 介绍

JSON 已经是大家必须掌握的知识点，JSON 数据格式为前后端通信带来了很大的便利。在开发中，前端开发工程师可以借助于 JSON 生成器快速构建一个 JSON 用来模拟数据。

本题请你开发一个简易的 JSON 生成器工具，使它能根据模板生成对应格式的 JSON。

## 准备

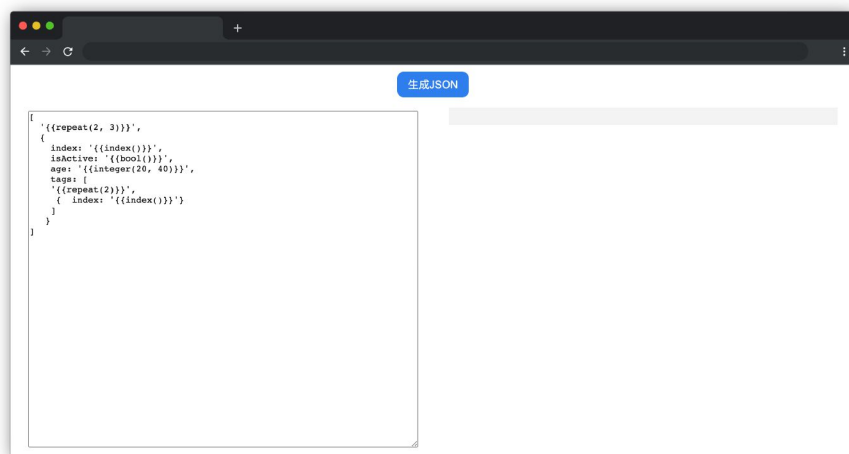
开始答题前，需要先打开本题的项目代码文件夹，目录结构如下：

```
├─ css
├─ index.html
├─ js
  │├─ highlight.min.js
  └─ index.js
```

其中：

- css 是样式文件夹。
- index.html 是主页面。
- js/highlight.min.js 是 json 格式化文件。
- js/index.js 是需要补充代码的 js 文件。

在浏览器中预览 index.html 页面效果如下：



初始效果

## 目标

请在 index.js 文件中补全 generateData 函数代码，并最终返回一个 js 对象(说明：

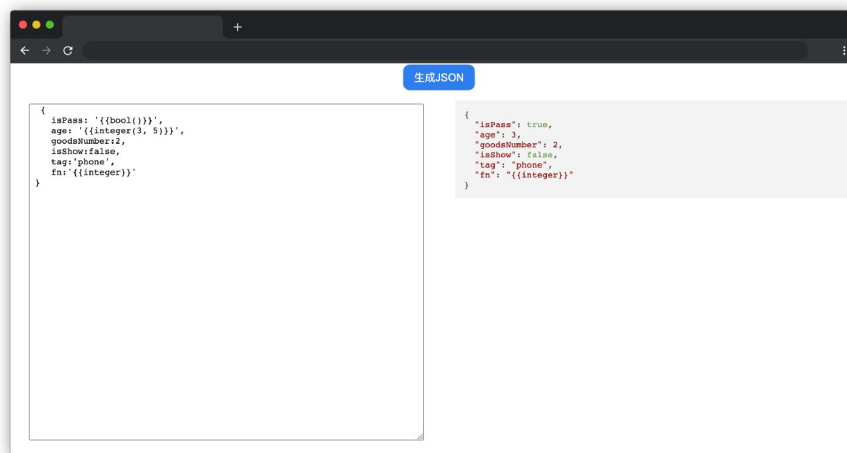
generateData 生成的数据会由插件自动转化成 JSON)。

在左侧的输入框中输入指定格式的数据模板，点击生成JSON 按钮，右侧会自动生成对应格式的 JSON 数据。

1. 数据模板中对象的 key 对应的 value 如果是 `{{}}` 并且符合下述规则，则根据下述规则解析，否

则一律返回原始 value 值。具体规则如下：

- `{{bool()}}` 表示随机生成布尔值。
- `{{integer(n, m)}}` 表示生成 n-m 之间（包含 n、m）的随机整数（注意：n<m）。



目标1测试用例

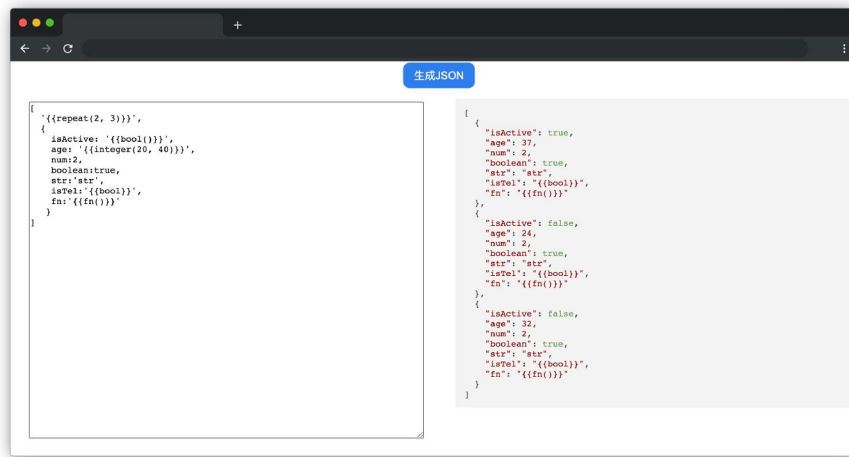
附目标 1 测试用例：

```
{
  isPass: '{{bool()}}',
  age: '{{integer(3, 5)}}',
  goodsNumber: 2,
  isShow: false,
  tag: 'phone',
  fn: '{{integer()}}'
}
```

2. 数据模板中 `{{repeat()}}`（此项只会出现在数组首位）表示重复生成数组中的数据，

如：`"repeat(5, 7)"` 则表示随机生成 5-7 条数组数据，repeat 中值只包含一个数字，

如`"repeat(5)"` 表示生成 5 条数组数据。



目标2测试用例

附目标 2 测试用例(3 组):

```
// (1) 随机生成 `2-5` 条数组数据
[
  "{{repeat(2, 5)}}",
  {
    isActive: "{{bool()}}",
    age: "{{integer(20, 40)}}",
    num: 2,
    boolean: true,
    str: "str",
    isTel: "{{bool}}",
    fn: "{{fn()}}",
  },
]

// (2) 固定生成 `7` 条数组数据
[
  ("{{repeat(7)}}",
  {
    isTrue: "{{bool()}}",
    score: "{{integer(3, 7)}}",
    tag: "android",
    isSamll: true,
    fn: "{{fn()}}",
  })
]

// (3) 无 repeat 的情况
[
  { maxNum: 10 }
];
```

**注意：**本题不考虑用户输入和传参不合法的情况，只处理合法的数据格式即可，实际测试中 **key** 和 **value** 为非固定值。提供的测试用例仅为方便测试代码使用，实际使用中需要对所有符合要求的数组/对象结构的模板生效。

## 规定

- 请勿修改 `js/index.js` 文件外的任何内容。
- 请不要使用固定值，如对象的 **key** 只对测试用例的 **key** 生效，以免造成无法判题通过。
- 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，以免造成无法判题通过。

## 判分标准

- 实现目标 1，得 10 分。
- 实现目标 2，得 15 分。

# 商城管理系统

## 介绍

在商城管理系统中，超级管理员和普通管理员因为权限不同，登录进入后看到的菜单也会是不同的。

本题需要你完成商城管理系统中权限数据的处理。

## 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
|— images
|— js
|   |— auth.js
|   |— menu.js
|— effect.gif
|— login.html
|— index.html
```

其中：

- `css` 是样式文件夹。
- `images` 是图片文件夹。
- `login.html` 是登录页。
- `index.html` 是商城首页。
- `js/menu.js` 是渲染左侧菜单的 `js` 文件。
- `js/auth.js` 是需要补充代码的 `js` 文件。
- `effect.gif` 是最终完成的效果图。

在浏览器中预览 `login.html` 页面效果如下：



登录页面

## 目标

请在 `js/auth.js` 文件中补全 `getMenuListAndAuth` 函数代码。

在登录页 `login.html` 点击**管理员登录**和**超级管理员登录**时会根据管理员的权限不同，在商城首页 `index.html` 的左侧显示不同的菜单列表。但是后端同学给的数据是不符合前端展示要求的，所以我们需要做些处理，假如后端提供的数据是这样的：

```
[
  { parentId: -1, name: "商品管理", id: 1, auth: "cart" },
  { parentId: 1, name: "商品列表", id: 4, auth: "cart-list" },
  { parentId: -1, name: "添加管理员", id: 10, auth: "admin" },
];
```

其中数据中对象字段的含义说明：

- `id` 表示当前节点
- `parentId` 表示父级节点，如果为 `-1` 则表示顶级数据
- `auth` 表示权限

具体需求如下：

1. 将待处理数据（一维数组）根据 `parentId` 字段值处理成树形结构存入 `menus` 变量中。

把数据处理成如下格式（每一项都必须有 `children` 字段，没有子级时，`children` 为空数组）：

```
[
  {
    parentId: -1,
```

```

    name: "商品管理",
    id: 1,
    auth: "cart",
    children: [
      {
        parentId: 1,
        name: "商品列表",
        id: 4,
        auth: "cart-list",
        children: [],
      },
    ],
  },
  {
    parentId: -1,
    name: "添加管理员",
    id: 10,
    auth: "admin",
    children: [],
  },
];

```

注意: js/auth.js 中的 menuList 仅为后端返回的数据结构示例, 非固定数据。实际使用中, 后端返回的数据转化成树形结构时层级可能更多, 封装方法时务必考虑通用性。

2. 将待处理数据中的 auth 字段提取出来并存入 auths 变量中, 左侧菜单会根据传入的权限列表数组和 auths 对比后进行菜单渲染。

经过 getMenuListAndAuth 处理后 auths 的结果为:

```
["cart", "cart-list", "admin"];
```

完成后的效果见文件夹下面的 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

- getMenuListAndAuth 需要对所有符合规则的数组有效。
- 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, 以免造成无法判题通过。

## 判分标准

- `auths` 数组返回值正确，得 5 分。
- `menus` 树形结构返回值正确，得 20 分。