

# 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（职业院校组）

## 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

试题序号、试题名称及基础源代码文件夹名称对应如下：

- 01 传送门（5 分）
- 02 新鲜的蔬菜（5 分）
- 03 小兔子找胡萝卜（10 分）
- 04 猜硬币（10 分）
- 05 用什么来做计算 B（15 分）
- 06 开学礼物大放送（15 分）
- 07 阅读吧（20 分）
- 08 新增地址（20 分）
- 09 天气趋势 B（25 分）
- 10 JSON 生成器（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 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。

# 传送门

## 介绍

日常浏览网页的时候，我们会发现一个问题，当页面太长、内容太多的时候我们很难快速浏览到心仪的内容。为了解决这个烦恼，优秀的产品研发团队发明了一种类似传送门的功能，以使用户能通过它快速定位到感兴趣的内容。这个功能也被通俗地比喻为“电梯”。

本题需要在已提供的基础项目中使用 JS/jQuery 等知识完善代码，最终实现该功能。

## 准备

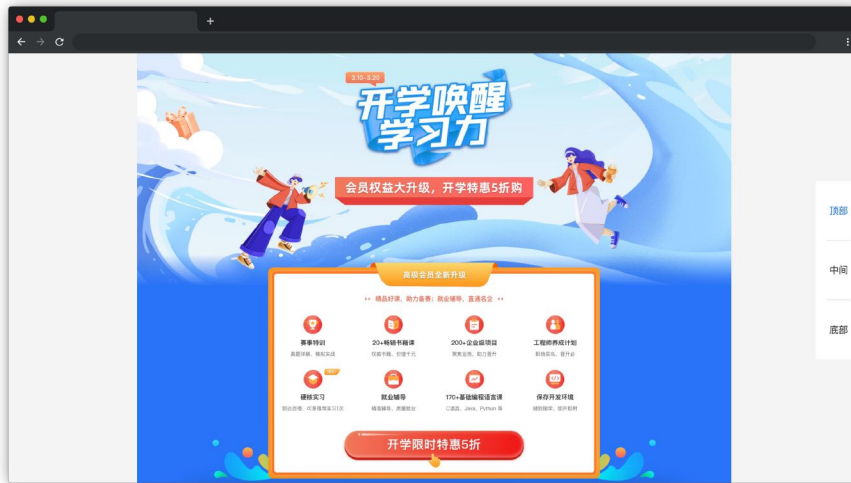
开始答题前，需要先打开本题的项目文件夹，目录结构如下：

```
|— effect.gif
|— index.html
|— css
|— images
└— js
    |— index.js
    └— jquery-3.6.0.min.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 是素材图片文件夹。
- 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。

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，最终实现传送门的功能。

具体需求如下：

1. 点击页面侧边的**顶部**、**中间**或**底部**按钮，页面滚动到**顶部**、**中间**或**底部**对应的范围内。这三个范围的设定如下：
  - 顶部：滚动条距离页面顶部  $0 \sim 960\text{px}$ （不包含 960）的范围。
  - 中间：滚动条距离页面顶部  $960 \sim 1920\text{px}$ （包含 960，不包含 1920）的范围。
  - 底部：滚动条距离页面顶部大于等于  $1920\text{px}$  的范围。
2. 当页面滚动到**顶部**、**中间**或**底部**位置时，对应的侧边按钮（即，**顶部**、**中间**或**底部**按钮）添加激活样式（即 `.active-color`），其余按钮样式变为默认（即 `.default-color`）。

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

- 实现点击侧边按钮快速定位内容功能，得 2 分。

- 实现侧边按钮文字颜色随页面内容滚动而变化功能，得 3 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

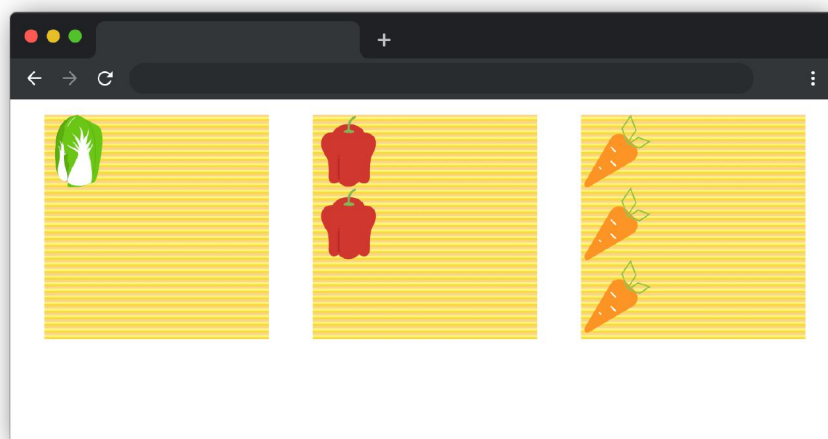
开始答题前，需要先打开本题的项目代码文件夹，目录结构如下：

```
├── 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。

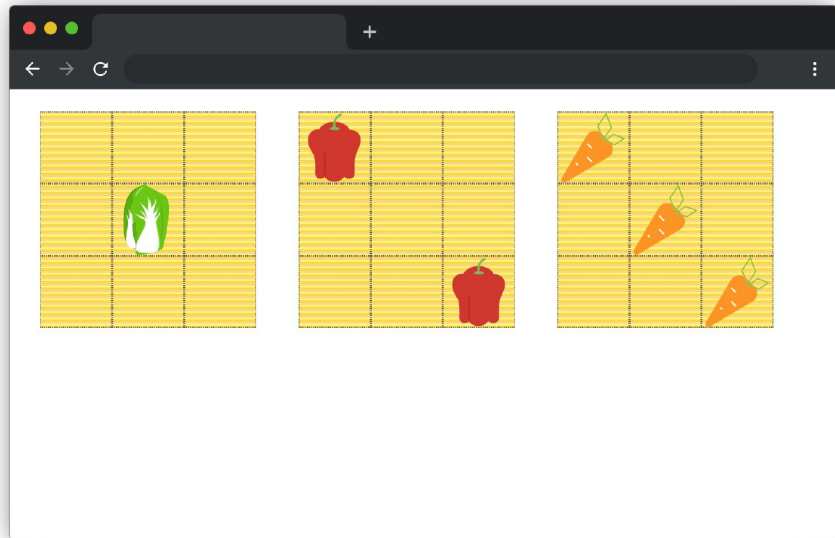
在浏览器打开后，效果如下：



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。



# 小兔子找胡萝卜

## 介绍

本题需要实现一款小兔子找胡萝卜的小游戏。在遥远的兔子王国，有一只小灰兔已经长大了，需要独自去寻找食物。点击“开始游戏”后，在输入框中输入移动步数，点击移动按钮，小灰兔可以在指定形状的草坪上跳动一步或者两步，草坪里藏着炸弹和胡萝卜。如果小灰兔踩到炸弹，本次觅食行动失败；如果小灰兔踩到胡萝卜，本次觅食行动获胜。

## 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
│   ├── bomb.png
│   ├── carrot.png
│   └─ rabbit.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，点击开始游戏按钮没有任何反应，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**开始游戏**的按钮后，**移动按钮**显示，**开始游戏**按钮隐藏，在右侧输入框中输入步数后，点击**移动按钮**，小兔子可以在草坪上向前移动。
- 输入框中只能输入 **1** 或者 **2**，输入其他数字，点击**移动按钮**后，小兔子不会移动，并且在按钮下方（`.result` 元素中）会显示一句话：“输入的步数不正确，请重新输入。”。
- 点击**移动按钮**后，输入框中的值不会被清空。
- 如果小兔子踩到第 13 块草坪上的炸弹，**移动按钮**隐藏，**重置按钮**显示，并且在**重置按钮**的下方（`.result` 元素中）会显示一句话：“哎呀！兔子踩到炸弹了，游戏结束！”。
- 如果小兔子踩到第 24 块草坪上的胡萝卜，**移动按钮**隐藏，**重置按钮**显示，并且在**重置按钮**的下方（`.result` 元素中）会显示一句话：“小兔子吃到胡萝卜啦，游戏获胜！”。
- 点击**重置按钮**小兔子回到游戏的初始化状态，并且输入框中的值和按钮下方的提示文字都会被清空。

## 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，按钮下方显示的文本内容，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

- 本题完全实现题目目标得满分，否则得 0 分。

# 猜硬币

## 介绍

为了打发无聊的时间，小蓝开发了一款人机对战的猜硬币游戏，页面中一共有 9 个杯子，系统会随机挑选 3 个杯子在里面放入硬币，玩家通过输入含有杯子序号的字符串进行猜选，但是遇到了一些问题。

本题需要你帮助小蓝完成猜硬币游戏。

## 准备

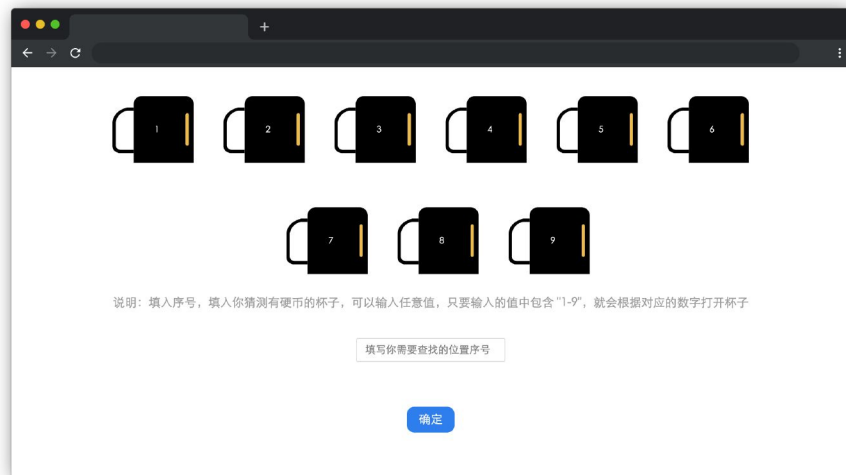
开始答题前，需要先打开本题的项目代码文件夹，目录结构如下：

```
├─ css
├─ effect.gif
├─ images
├─ index.html
└─ js
    ├── findCoin.js
    └─ index.js
```

其中：

- `css` 是样式文件夹。
- `images` 是图片文件夹。
- `index.html` 是主页面。
- `js/index.js` 是硬币渲染和动画逻辑。
- `js/findCoin.js` 是需要补充代码的 `js` 文件。
- `effect.gif` 是最终实现的效果图。

在浏览器中预览 `index.html` 页面效果如下：



初始效果

## 目标

请在 `js/findCoin.js` 文件中补全 `findNum` 和 `randomCoin` 函数代码，最终实现猜硬币游戏。

具体需求如下：

1. 页面加载后，系统会在 9 个杯子中随机挑选 3 个杯子生成硬币。
2. 用户在文本框中输入任意字符串，点击**确定**按钮后，系统会找到字符串中含有 1-9 的数字，并根据数字打开对应的杯子。

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 `findNum` 函数，得 5 分。
- 完成 `randomCoin` 函数，得 5 分。

# 用什么来做计算 B

## 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。

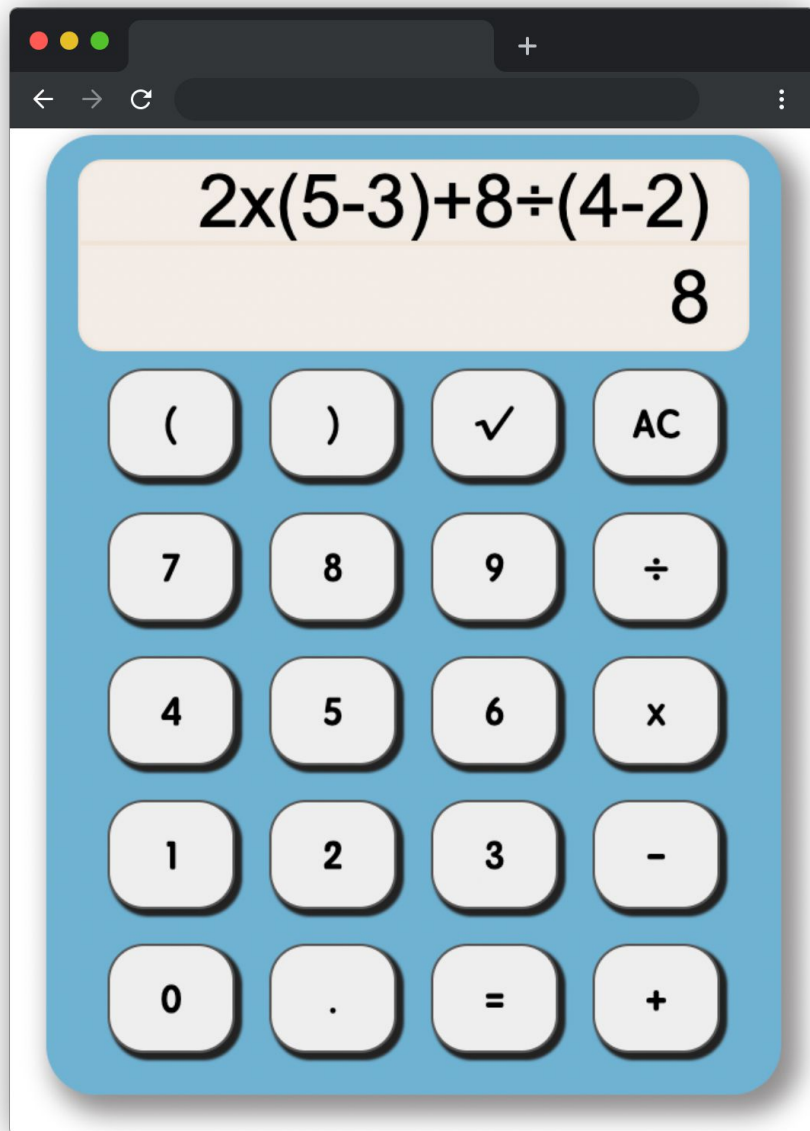
完成后的效果见文件夹下面的 gif 图，图片名称为 `effect.gif`（提示：可以通过 VS Code 或者浏览器预览 gif 图片）。

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：

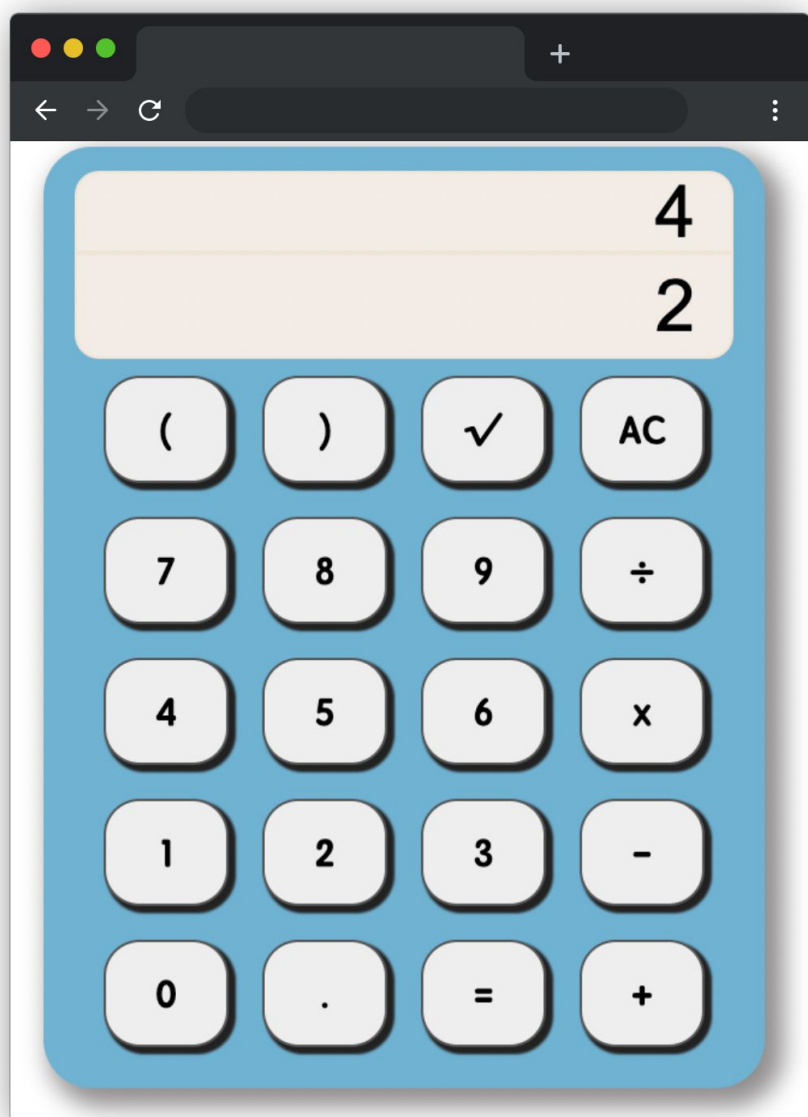
- 能够正确进行加、减、乘、除的混合运算，样例如下所示：



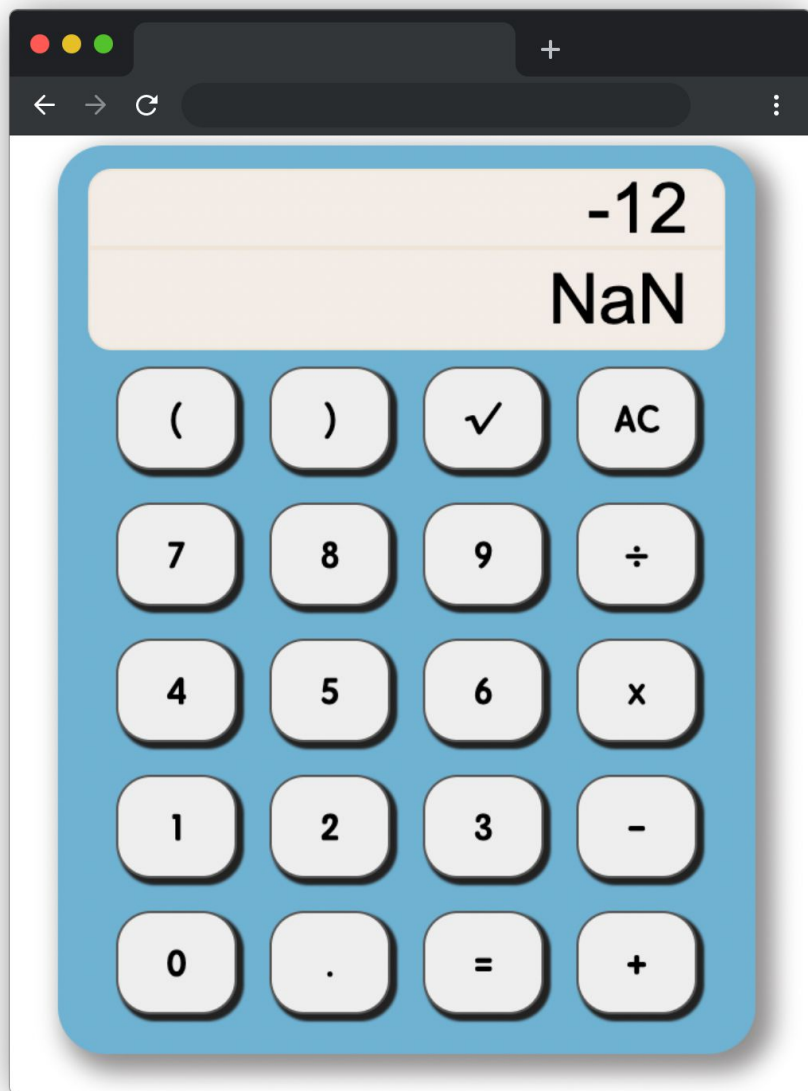
加减乘除混合运算样例图

- 能够正确进行开二次方，输入一个可以平方的数，点击开方 ( $\sqrt{\phantom{x}}$ ) 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：



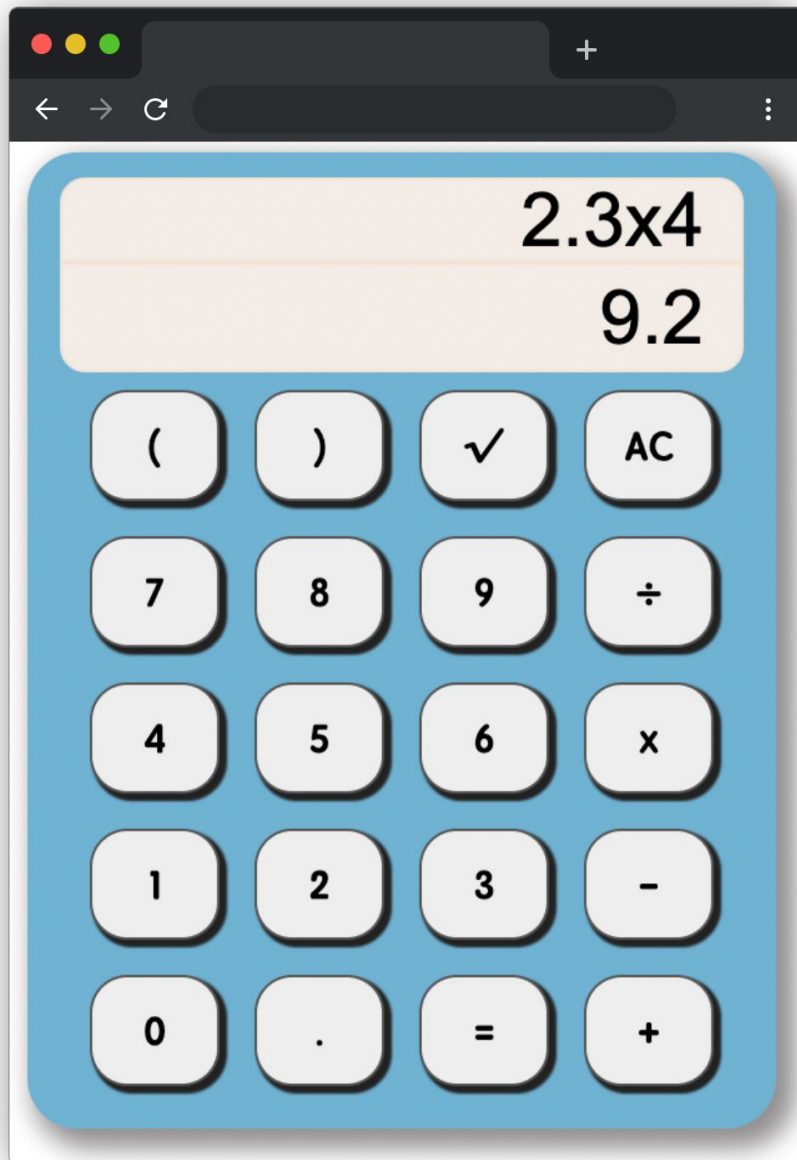


开方运算样例图



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 2 分。

- 实现计算式子和结果显示功能，得 5 分。
- 实现计算功能，得 8 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。

# 阅读吧

## 介绍

“读万卷书，行万里路”，无论你现在贫穷或富有，身和心一定要有一个在路上。那么，在快节奏的今天，人们是如何利用碎片化的时间去阅读的呢？没错，那就是电子书了。有了它，我们就可以随时随地，想读就读～

本题需要在已提供的基础项目中使用 Vue 2.x 完善代码，最终实现一个简易的电子书阅读器。

## 准备

开始答题前，需要先打开本题的项目文件夹，目录结构如下：

```
├─ effect.gif
├─ index.html
├─ css
├─ js
└─ 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。
- `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，最终实现以下功能：

## 1. 实现头部工具栏显隐功能。

- 头部工具栏中的关闭图标和页面右上侧的设置图标，可以进行头部工具栏的显隐切换。效果如下：



关闭按钮



设置按钮

## 2. 实现设置阅读主题功能。

- 点击头部工具栏阅读主题后面任意一个圆形色块，该色块显示勾选状态（即 .icon-selected），同时阅读区的背景色修改为该色块对应颜色。效果如下：



选中阅读主题色效果

- 当阅读区的背景色为黑色时，阅读区的字体色为白色（即 #ffffff），其它背景色对应的字体颜色为黑色（即 #333333）。





背景色为黑色时，阅读区的字体色为白色

### 3. 实现设置字体大小功能。

- 点击工具栏中“A-”或“A+”按钮，阅读区的字体大小呈 2px 递减或递增，并将字号同步更新到“A-”和“A+”之间的元素（即 .lang）中，需要注意的是：字体最小为 12px，最大为 48px，且阅读区的行高（即 line-height CSS 属性）始终比其字体大小多 10px。



阅读区字体大小和行高效果图

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）。

## 规定

- 请勿修改 index.html 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

- 实现头部工具栏显隐功能，得 4 分。
- 实现设置阅读主题功能，得 8 分。
- 实现设置字体大小功能，得 8 分。

# 新增地址

## 介绍

网购大家应该再熟悉不过，网购中有个很难让人忽略的功能就是地址管理，接下来就让我们通过完善代码来探索下地址管理中常用功能的实现吧～

本题需要在已提供的基础项目中使用 JS 知识完善代码，最终实现需求中的具体功能。

## 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
|— images
└— js
    |— data.js
    └— index.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/data.js 是省市列表数据。
- 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` 文件中补全代码，最终实现以下需求：

1. 实现省市二级联动功能。
  - 页面加载后省份下拉列表数据正常渲染。
  - 在省份列表中选取某个数据后，对应的市区数据会渲染在其右侧的市区列表中。效果如下：

新建收货地址

省市 上海市 黄浦区

\* 地址 填写详细地址，例：XX 小区 1 幢 101

标签 家 公司 学校

\* 联系人 收货人姓名

\* 手机号 收货人手机号码

保存

省市联动效果

## 2. 实现对提交的表单信息的验证功能。

- 填写表单信息，其中**地址**、**联系人**和**手机号**为必填项，有一项为空则点击**保存**按钮会弹出提示窗，点击提示窗周围的半透明蒙层后，该窗口会关闭。效果如下：



弹窗提示效果

3. 实现选择标签显示激活样式功能。

- 点击标签列表中的某个标签（例如：公司），该标签显示激活样式（即 `.active`），其他未选择标签显示默认样式。效果如下：

新建收货地址

省市 上海市 闵行区

\* 地址 vvvvv

标签 家 公司 学校

\* 联系人 tom

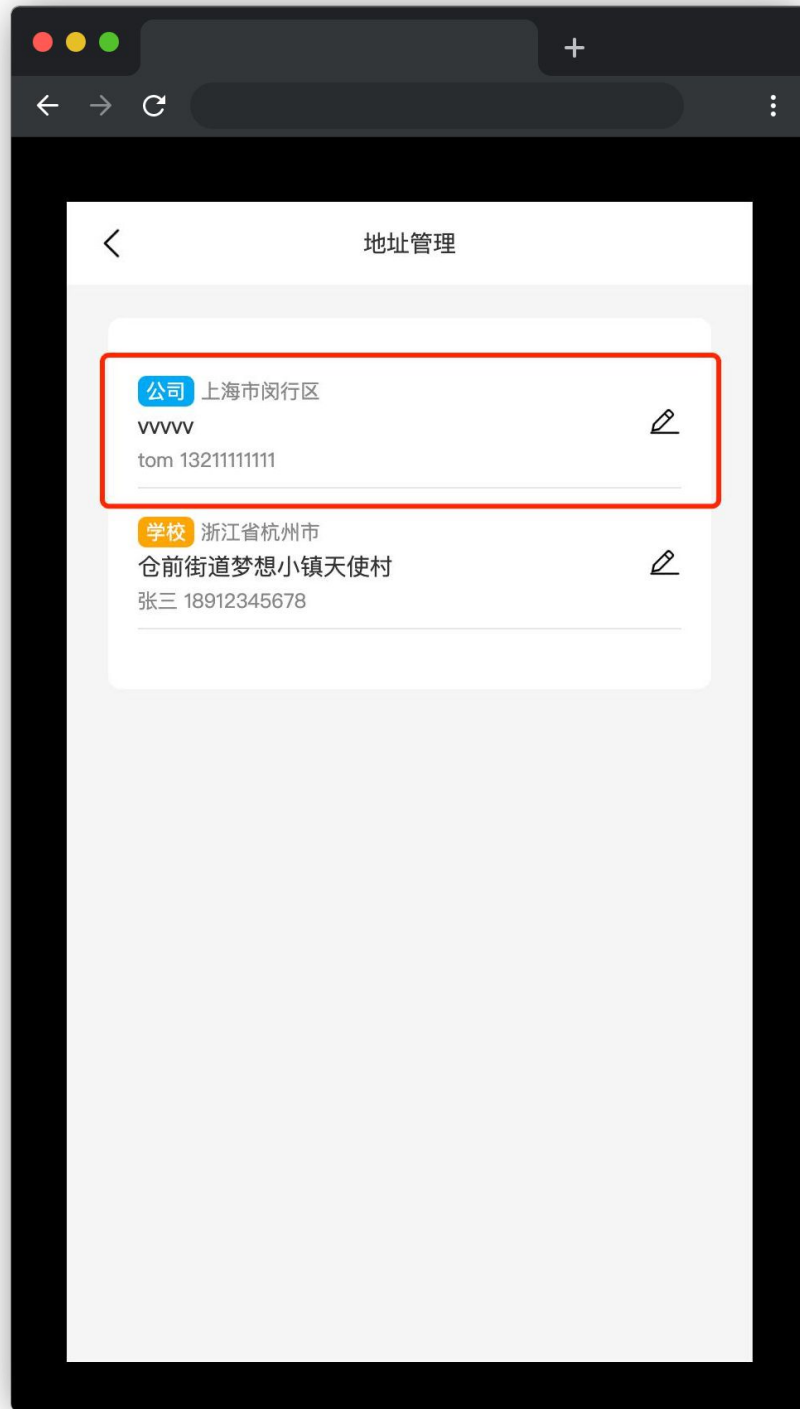
\* 手机号 13211111111

保存

标签显示激活样式

4. 实现新增地址被正确渲染功能。

- 点击保存按钮，表单数据将以列表的形式显示在原有数据的上方，即整个地址列表的首位。效果如下：



表单提交后在地址列表中的显示效果

- 表单数据在地址列表中显示时，对应的标签有其不同的样式，具体如下：
- 家：.home。
- 公司：.company。
- 学校：.school。

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

- 实现省市二级联动功能，得 7 分。
- 实现表单信息验证功能，得 2 分。
- 实现标签动态选取功能，得 3 分。
- 实现新增地址渲染功能，得 8 分。



# 天气趋势 B

## 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：

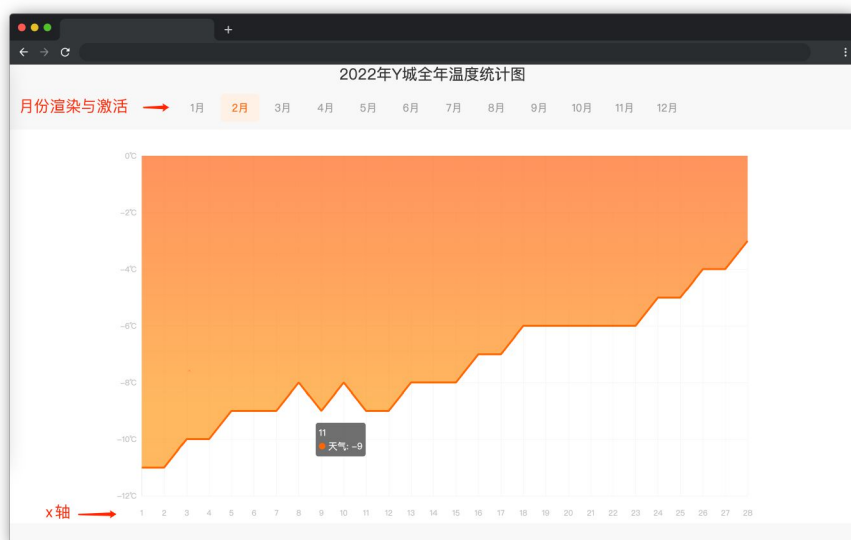


初始效果

## 目标

请在 `index.html` 文件中补全代码，具体需求如下：

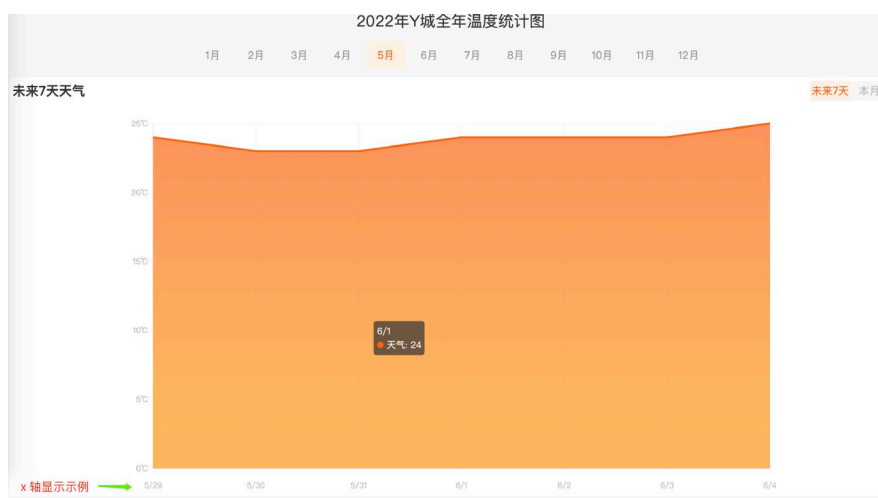
- 1.完成数据请求（数据来源 `./js/weather.json`），`weather.json` 中存放的数据为 12 个月对应的温度数据。在项目目录下已经提供了 `axios`，考生可自行选择是否使用。注意：调试完成后请将请求地址写死为 `./js/weather.json`。
- 2.把 `data` 中的月份数据 `monthList`，在 `class=month` 标签下面的 `li` 上完成渲染，点击月份则切换对应月份的温度数据同时被点击的月份会变成激活状态(`.active` 类)，x 轴为日期，y 轴为温度，默认显示 1 月份数据。



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 10 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

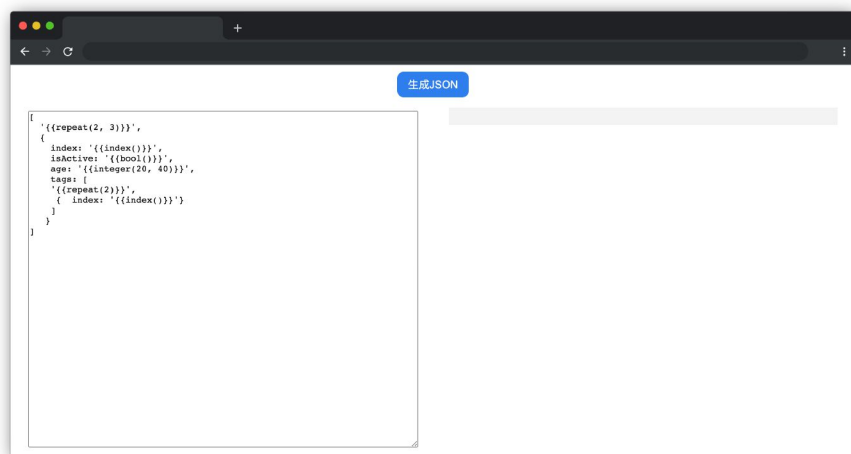
开始答题前，需要先打开本题的项目代码文件夹，目录结构如下：

```
├─ 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：

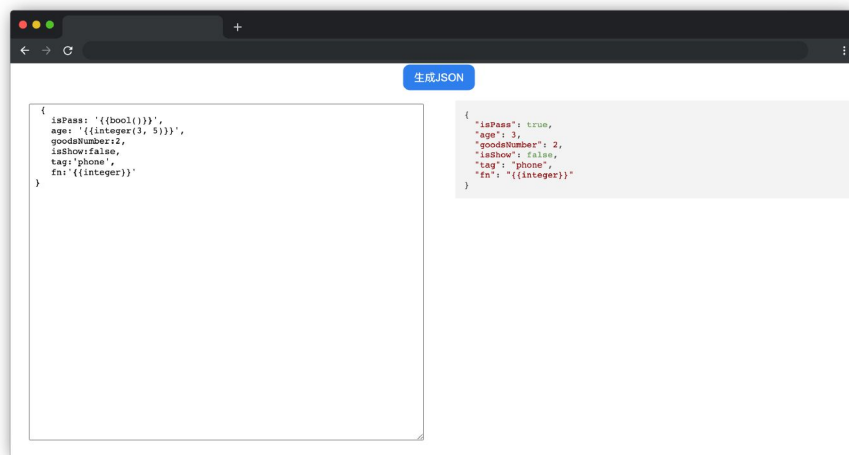
generateData 生成的数据会由插件自动转化成 JSON)。

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

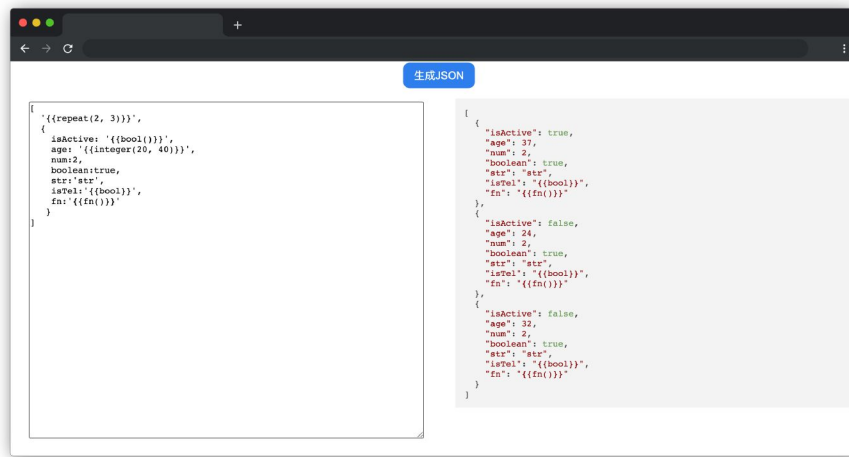
附目标 1 测试用例：

```
{
  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。