JavaScript基础练习

1. 输入框输入一个数num，计算出从1~num的累加结果，展示在屏幕上。
2. 创建有返回值的函数，已知将两个输入框的值value作为参数返回值，创建一个return按钮将两个值相加后输出。
3. 在练习1中添加如下功能

（1）输入框输入字母“A”,显示图片A；输入字母“B”,显示图片B；输入字母“M”,演示音乐。

（2）在练习1中输入验证，用户需要在页面上输入用户名和电话号码，格式如下:用户名（3到6位字母数字的组合）电话号码(6位数字)。输入格式正确，则弹出录入正确，格式错误则弹出录入错误！

（3）添加图片的翻滚功能。

**Jason练习**

**第1部分**

在JS中解析JSON数据,JSON文件data.json如下：

文件有如下结构:

{

}

"pets": [

{

"name": "Scooby",

"breed": "Labrador"

},

{

"name": "Lassie",

"breed": "Collie"

},

{

"name": "Toto",

"breed": "Terrier"

},

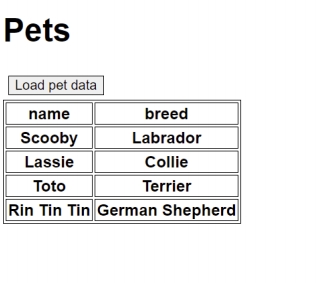
{

"name": "Rin Tin Tin",

"breed": "German Shepherd"

}

]



在工作空间文件夹中创建另一个名为“javascript.js”的文件——将在这个文件中保存所有的JS代码和JSON文件。另外，在工作空间文件夹中创建一个CSS文件“stylesheet.css”——它将包含CSS代码。

你需要创建一个html文件例如index.html链接到CSS和JS文件。将该链接添加到头部的CSS文件中。

将链接添加到body底部的JS文件中采用如下代码：

**<script src="javascript.js" defer></script>**

接下来现在要在“javascript.js”中编写JS代码，实现解析JSON文件，并在html中动态地创建一个包含有宠物信息的表格。要做到这一点，请在html中的body中添加以下代码:

**<body>**

**<h1>Pets</h1>**

**<button onclick="loadPetData()">Load pet data</button>**

**<table id="pets"></table>**

**<script src="javascript.js" defer></script>**

**</body>**

（注：）在完成解析和显示这个JSON文件之后，后面将给你一个更复杂的JSON文件，完成解析和显示这些数据。

在javascript.js代码最前面，声明一个constant “pet\_data”并赋值JSON数据:

**const pet\_data =**

**{**

**"pets": [**

**{**

**"name": "Scooby",**

**"breed": "Labrador"**

**},**

**{**

**"name": "Lassie",**

**"breed": "Collie"**

**},**

**{**

**"name": "Toto",**

**"breed": "Terrier"**

**},**

**{**

**"name": "Rin Tin Tin",**

**"breed": "German Shepherd"**

**}**

**]**

**};**

接下来，创建表头，它是表的第一行，包含单词“name”和“breed”:

const tableHeaders =

'<tr><th>name</th><th>breed</th></tr>';

然后需要解析JSON数据，并创建包含宠物数据的表格。这将作为一个字符串存储在一个名为petsHTML的变量中，声明该变量，并将tableHeaders添加到字符串petsHTML中:

**let petsHTML = '';**

**petsHTML += tableHeaders;**

接下来，需要考虑表头下的每一行都具有什么结构。这将作为模板，我们将在每个表格单元格中使用大写字母来表示将放入的数据

const petRow =

'<tr><td>PET\_NAME</td><td>PET\_BREED</td></tr>';

现在，编写一个JS函数来解析JSON文件，并使用模板结构一次创建一行

**function loadPetData(){**

**pet\_data.pets.forEach(pet=>{**

**let thisPetHTML = petRow;**

**thisPetHTML = thisPetHTML.replace('PET\_NAME', pet.name);**

**thisPetHTML = thisPetHTML.replace('PET\_BREED', pet.breed);**

**petsHTML += thisPetHTML;**

**});**

**document.getElementById //this line needs to be completed**

}

这个函数使用forEach遍历JSON宠物数组中的每个对象，并将对象赋值给变量pet 。每次迭代，它都会将变量petRow中的模板赋值给变量thisPetHTML。然后调用模板上的replace函数，用实际保存在pet.name和pet.breed中的JSON数据替换PET\_NAME和 PET\_BREED 占位符。然后更新petsHTML。

当所有的JSON都被解析后，该函数更新HTML，在index.html中有table ID处完成代码，实现当按钮被按下时，表格就会出现在按钮下面。

可以对表格进行一些样式化处理，例如将在表格中的每个单元格周围添加边框:

table, th, td {

border: 1px solid black;

}

还应该在按钮和表格之间添加一些空间，并在页面中使用一致的字体。

测试代码是否工作，在浏览器窗口中打开index.html，单击按钮，查看表是否正确地显示宠物数据。

**第2部分**

练习1演示了如何解析一个JSON文件，并动态创建一个HTML表来显示来自JSON文件的数据。

在练习2中，你将解析一个更复杂的JSON文件(也包含不同宠物的数据)，使用与我们在练习1中使用的相同的技术。此练习的JSON文件如下所示。

第一步是查看JSON数据，并考虑它在表中会是什么样子。然后创建一个模板，这样就可以使用与练习1中相同的替换技术。

提示1:如果你将光标停留在一个元素上，那么它将指示该元素包含什么JSON。这可以帮助开发你的算法。

提示#2:由于JSON文件的复杂性增加，你将需要遍历JSON数据的不同部分。

JSON数据如下

{

"pets": [

{

"type": "fish",

"pets": [

{

"name": {

"first": "Nemo",

"last": "Finding"

},

"colours": ["orange", "white"]

},

{

"name": {

"first": "Michael",

"last": "Thomas"

},

"colours": ["yellow"]

},

{

"name": {

"first": "Dory",

"last": "Grubber"

},

"colours": ["blue", "yellow", "black"]

}

]

},

{

"type": "cat",

"pets": [

{

"name": {

"first": "Squid",

"last": "Johnson"

},

"colours": ["black"]

},

{

"name": {

"first": "Peanut",

"last": "Day"

},

"colours": ["black", "brown"]

},

{

"name": {

"first": "Marley",

"last": "Day"

},

"colours": ["black", "brown"]

},

{

"name": {

"first": "Bear",

"last": "Blue"

}

]

},

{

}

},

"colours": ["gray"]

}

]

"type": "dog",

"pets": [

{

"name": {

"first": "Rin",

"last": "Tin Tin"

},

"colours": ["black"]

},

{

"name": {

"first": "Sharik",

"last": "Gajos"

},

"colours": ["black", "brown"]

}

]