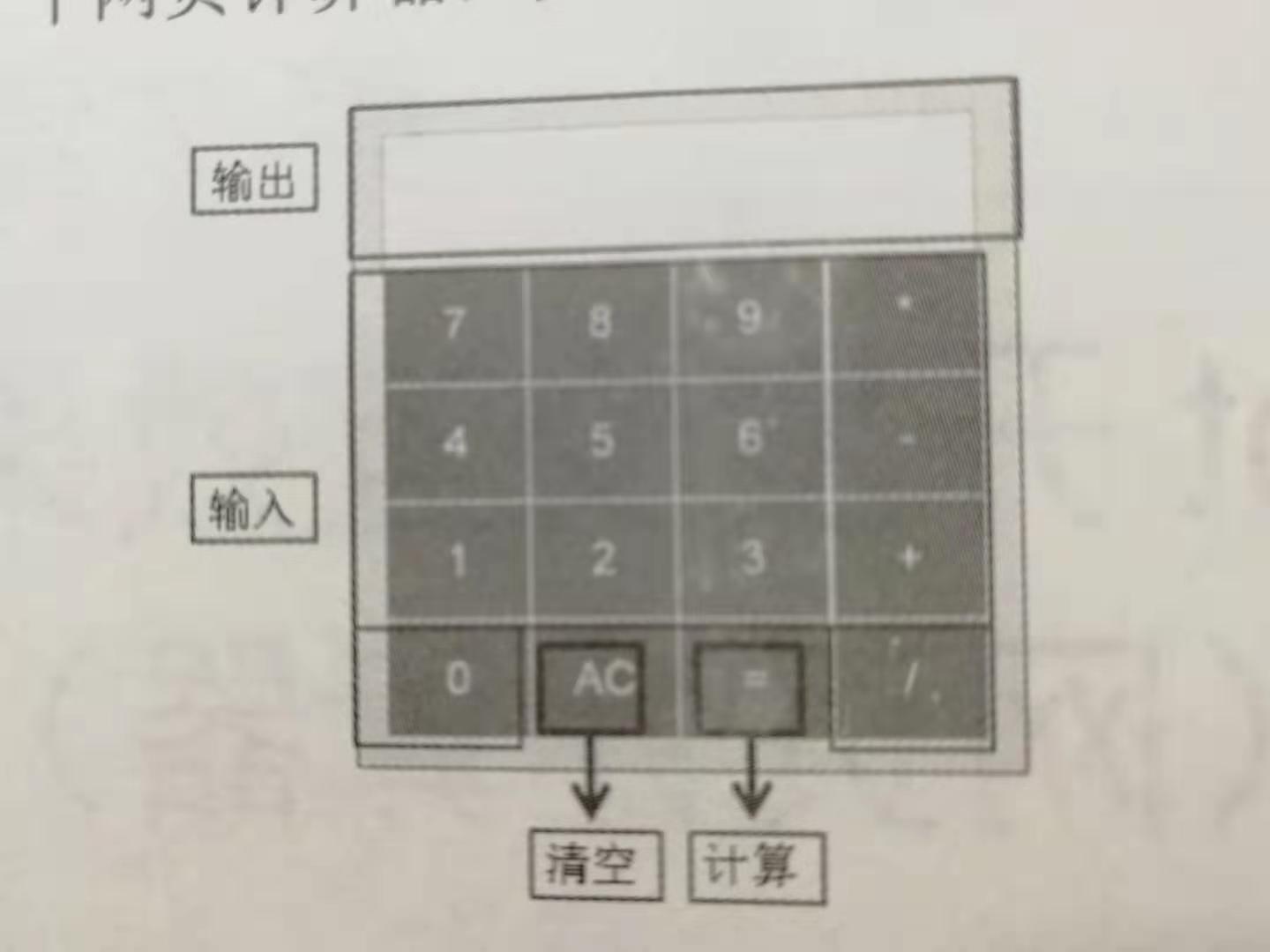
**JavaScript开发交互效果页面\_网页计算器**

**一、实验任务**

使用js完成一个网页计算器，页面效果如下图所示：



（1）计算器页面包括数字按键、运算符按键、计算按键和清空按键，以及计算区域输出文本框，并设置计算区域输出文本框为不可编辑。

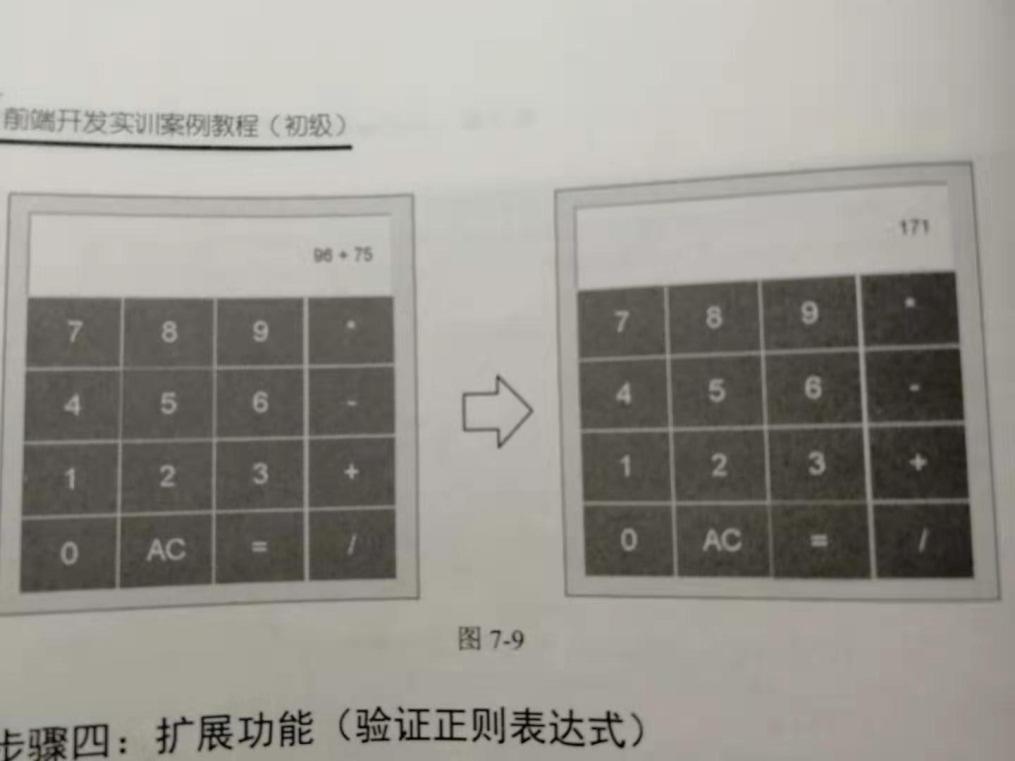
（2）界面输入规则：输入一次数字，再输入一次运算符，然后输入一次数字，接着输入一次运算符，如此往复，形成类似“3+5\*6”的形式。

（3）单击“=”按键出发js函数，进行计算。

（4）计算结果显示至页面计算区域输出文本框。

（5）单击“AC”按键清空计算区域输出文本框值。

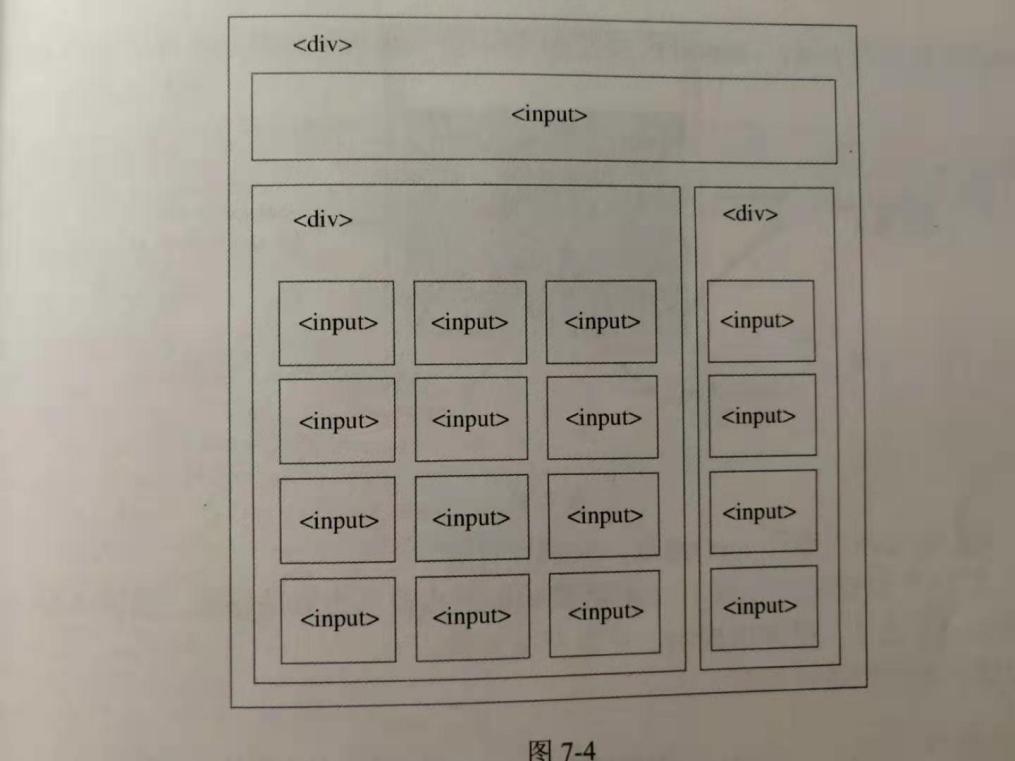
（6）扩展功能：将计算区域输出文本框改为可编辑文本框，可直接输入算式，如下图所示，单击“=”按键使用正则表达式验证文本框输入内容并进行计算。



**二、设计思路**

（1）用HTML和CSS布局网页计算器的界面。

页面结构如图所示：



（2）创建js文件：calculator.js。

在文件中创建一个calculator的对象，并定义1个“number数组”和4个计算函数。

* number数组：存储输入的数字和运算符。
* numberClick：输入数字函数，通过数字按键得到输入的数字，并显示到计算区域输出文本框。
* operatorClick：输入运算符函数，通过运算符按键得到输入的运算符，并显示到计算区域输出文本框。
* equalClick：获取计算区域输出文本框数字和运算符，解析之后保存到“number数组”，计算并显示结果。
* cleanClick：清空数据。

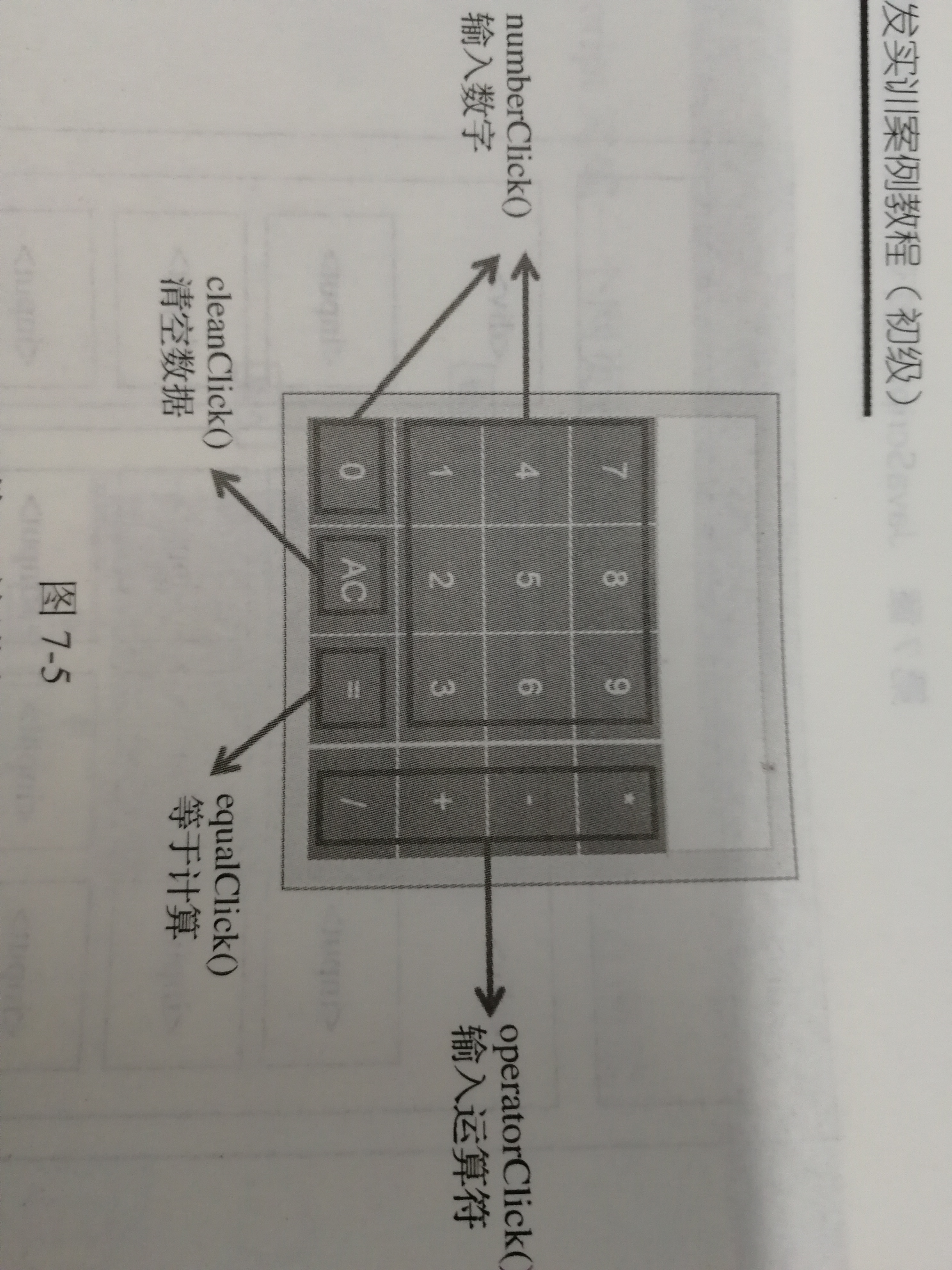
（3）为每个按键的<input>标签绑定click事件。

（4）输入数字和运算符，每次click事件，都更新计算区域输出文本框，将输入的内容添加到计算区域输出文本框。

（5）编写equalClick()计算函数，响应“=”click事件，计算步骤如下：

* 获取显示框字符串。
* 将字符串分割并赋给数组，然后对数组进行计算。
* 将计算结果显示在input显示框上。

各个按键对应的函数如下图所示：



（6）针对扩展功能，设计reg()函数，并进行计算。

* 当单击“=”按键时，先调用reg()函数验证键盘输入内容的格式。
* 如果验证不通过，则弹出提示，并置空文本框。
* 如果验证通过则进行计算。

三、实验步骤（跟我做）

1、步骤一：HTML布局

（1）创建计算器页面，命名为calculator.html。

（2）按照计算器样式添加<input>进行按键布局。

* 添加数字输入区域和文本框，文本框id属性为output，class属性为output，设置文本框不可编辑。

<div class="calculator">

<input type="text" class="output" id="output" value="" disabled />

</div>

* 添加运算符输入区域。
* 加入数字按键，归零（AC）按键和“=”按键。

<div class="numbers">

<input type="button" value="7">

<input type="button" value="8">

<input type="button" value="9">

<input type="button" value="4">

<input type="button" value="5">

<input type="button" value="6">

<input type="button" value="1">

<input type="button" value="2">

<input type="button" value="3">

<input type="button" value="0">

<input type="button" value="AC">

<input type="button" value="=">

</div>

* 加入运算符按键。

<div class="operators">

<input type="button" value="\*">

<input type="button" value="-">

<input type="button" value="+">

<input type="button" value="/">

</div>

2、步骤二：CSS添加样式

（1）创建calculator.css文件。

（2）在页面中引入该文件。

（3）编辑该文件，为页面添加样式。

* 设置计算区域样式。

/\* 设置计算区域的宽度、边框、背景色、边距 \*/

.calculator{

width: 405px;

border: solid 1px white;

background: #ffefd5;

margin: 50px;

padding: 20px;

}

* 设置输出区域样式。

/\* 设置计算区域文本框宽度、高度、边距、字体大小、文本右对齐、背景颜色 \*/

.output{

width: 356px;

height: 50px;

padding: 20px;

font-size: 20px;

text-align: right;

background: white;

}

* 设置输出按键样式。

/\* 设置按键样式 \*/

input[type=button]{

border: solid 1px white;

width: 100px;

height: 80px;

background: grey;

cursor: pointer;

color: white;

font-size: 30px;

}

* 设置输出数字样式。

/\* 数字样式 \*/

.numbers{

width: 300px;

display: inline-flex;

flex-wrap: wrap;

}

* 设置输出符号样式。

/\* 设置输出符号样式 \*/

.operators{

width: 100px;

position: relative;

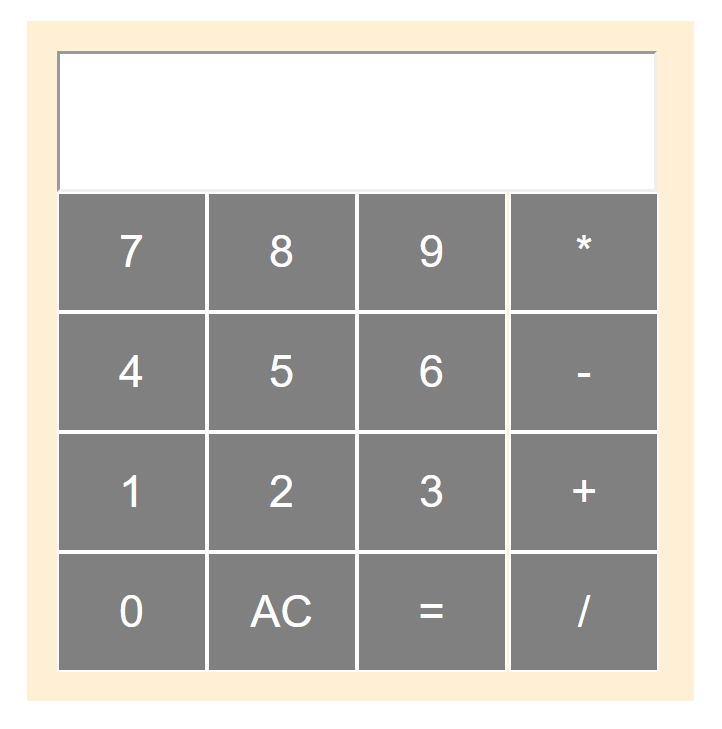
left: -3px;

display: inline-flex;

flex-wrap: wrap;

}

（4）页面效果如图所示：



3、步骤三：js计算

（1）创建calculator.js文件。

（2）导入该文件。

<script type="text/javascript" src="calculator.js"></script>

（3）编辑该文件，实现计算功能。

* 定义一个变量和一个对象，对象中包含一个计算数组和4个计算方法。

//计算对象

var calculator={

//用于保存输入的数字和符号数据

number:[],

//计算方法

numberClick:numberClick,

operatorClick:operatorClick,

equalClick:equalClick,

cleanClick:cleanClick

}

* 数字按键click事件调用方法。

//输入数字方法

var numberClick=function(value){

var val=document.getElementById("output").value;

//显示框为0时，输入0无效

if(value=="0" && val=="0"){

return;

}

//如果显示框为0，则去掉0，只显示输入值

if(val=="0"){

document.getElementById("output").value=value;

}else{

//在显示框显示对应字符

document.getElementById("output").value=val+value;

}

}

* 运算符按键click事件调用方法。

//输入运算符方法

var operatorClick=function(value){

var val=document.getElementById("output").value;

//判断是否连续输入了两次运算符，运算符后面输入数字，不能练习输入多个运算符（运算符两端都有空格）

if(val[val.length-1]==" "){

return;

}

//在显示框显示对应运算符

document.getElementById("output").value=val+" "+value+" ";

}

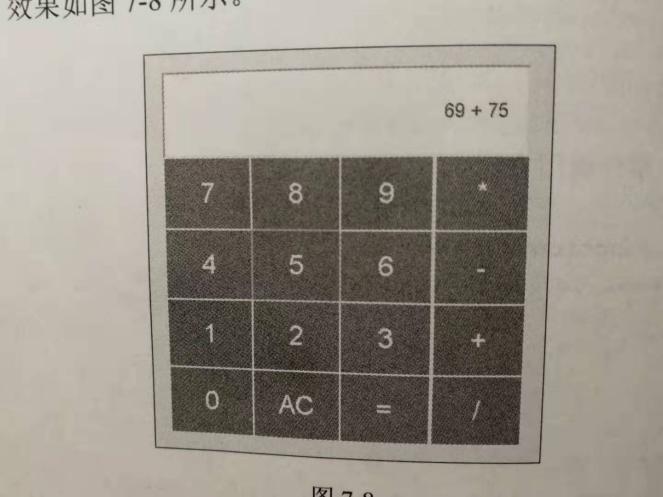
* 调用数字和运算符按键click事件。

在每个input标签内部注册对应的对象中的计算方法。

<input type="button" value="7" onclick="calculator.numberClick(7)" />

<input type="button" value="\*" onclick="calculator.operatorClick('\*')" />

* 输入算式，效果如下图所示：



* 计算并显示结果。