**二维码的生成与识别**

**一、实验目的**

1、通过编程实现二维码的生成与识别；

2、熟悉所使用的编程语言；

3、理解二维码编码与解码的过程；

**二、实验准备**

1、环境器材：Sublime Text3（编程软件）、Google Chrome（浏览器）；

2、实验前了解：

1. 二维码又称QR Code，QR全称Quick Response，是一个近几年来移动设备上超流行的一种编码方式，它比传统的Bar Code条形码能存更多的信息，也能表示更多的数据类型。它是用某种特定的几何图形按一定规律在平面（二维方向上）分布的黑白相间的图形记录数据符号信息的。
2. 常用的码制有：Data Matrix, Maxi Code, Aztec, QR Code, Vericode, PDF417, Ultracode, Code 49, Code 16K等。

（3）二维条码/二维码可以分为堆叠式/行排式二维条码和矩阵式二维条码：

堆叠式/行排式二维条码形态上是由多行短截的一维条码堆叠而成；

矩阵式二维条码以矩阵的形式组成，在矩阵相应元素位置上用“点”表示二进制“1”， 用“空”表示二进制“0”，“点”和“空”的排列组成代码。

（4）特点：

1)存储大容量信息,还可以支持所有类型的数据

2)在小空间内打印

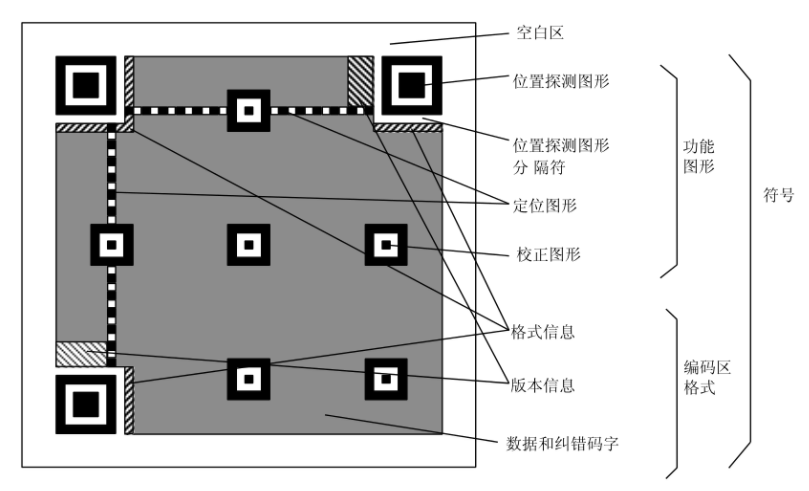
3)有效表现各种字母

4)具备“纠错功能”，即使部分编码变脏或破损，也可以恢复数据

5)可以从任意方向读取

6)支持数据合并功能

（5）基本结构图



位置探测图形、位置探测图形分隔符、定位图形：用于对二维码的定位，对每个QR码来说，位置都是固定存在的，只是大小规格会有所差异；

校正图形：规格确定，校正图形的数量和位置也就确定了；

格式信息：表示该二维码的纠错级别，分为L、M、Q、H；

版本信息：即二维码的规格，QR码符号共有40种规格的矩阵（一般为黑白色），从21x21（版本1），到177x177（版本40），每一版本符号比前一版本 每边增加4个模块。

数据和纠错码字：实际保存的二维码信息，和纠错码字（用于修正二维码损坏带来的错误）。

**三、实验原理**

编码过程：

（1）数据分析：确定编码的字符类型，按相应的字符集转换成符号字符； 选择纠错等级，在规格一定的条件下，纠错等级越高其真实数据的容量越小。

（2）数据编码：将数据字符转换为位流，每8位一个码字，整体构成一个数据的码字序列。其实知道这个数据码字序列就知道了二维码的数据内容。

|  |  |
| --- | --- |
| **格式** | **容量** |
| 数字 | 最多7089字符 |
| 字母 | 最多4296字符 |
| 二进制数（8 bit） | 最多2953字节 |
| 日文汉字/片假名 | 最多1817字符（采用Shift JIS） |
| 中文汉字 | 最多984字符（采用UTF-8） |
| 中文汉字 | 最多1800字符（采用BIG5） |

表一 格式容量表

|  |  |
| --- | --- |
| **模式** | **指示符** |
| ECI | 0111 |
| 数字 | 0001 |
| 字母数字 | 0010 |
| 8位字节 | 0100 |
| 日本汉字 | 1000 |
| 中文汉字 | 1101 |
| 结构链接 | 0011 |
| FNCI | 0101（第一位置）  1001（第二位置） |
| 终止符（信息结尾） | 0000 |

表二 模式编码表

数据可以按照一种模式进行编码，以便进行更高效的解码，例如：对数据：01234567编码（版本1-H），

1）分组：012 345 67

2）转成二进制：

012→0000001100

345→0101011001

67 →1000011

3）转成序列：0000001100 0101011001 1000011

4）字符数 转成二进制：8→0000001000

5）加入模式指示符（上图数字）0001：0001 0000001000 0000001100 0101011001 1000011

对于字母、中文、日文等只是分组的方式、模式等内容有所区别。基本方法是一致的。

1. 纠错编码：按需要将上面的码字序列分块，并根据纠错等级和分块的码字，产生纠错码字，并把纠错码字加入到数据码字序列后面，成为一个新的序列。

|  |  |
| --- | --- |
| **纠错等级** | **纠错水平** |
| L | 7%字码修正 |
| M | 15%字码修正 |
| Q | 25%字码修正 |
| H | 30%字码修正 |

表三 纠错等级与水平对照表

（在二维码规格和纠错等级确定的情况下，其实它所能容纳的码字总数和纠错码字数也就确定了，比如：版本10，纠错等级时H时，总共能容纳346个码字，其中224个纠错码字。）

就是说二维码区域中大约1/3的码字时冗余的。对于这224个纠错码字，它能够纠正112个替代错误（如黑白颠倒）或者224个据读错误（无法读到或者无法译码），这样纠错容量为：112/346=32.4%。

（4）构造最终数据信息：在规格确定的条件下，将上面产生的序列按次序放如分块中。

按规定把数据分块，然后对每一块进行计算，得出相应的纠错码字区块，把纠错码字区块按顺序构成一个序列，添加到原先的数据码字序列后面。

如：D1, D12, D23, D35, D2, D13, D24, D36, … D11, D22, D33, D45, D34, D46, E1, E23,E45, E67, E2, E24, E46, E68，…

（5）构造矩阵：将探测图形、分隔符、定位图形、校正图形和码字模块放入矩阵中。

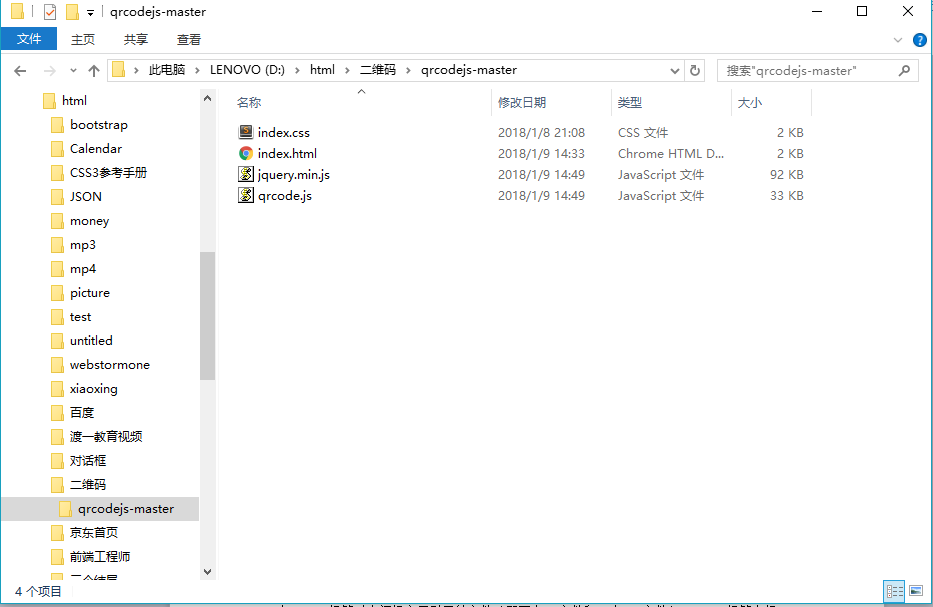
（6）掩摸：将掩摸图形用于符号的编码区域，使得二维码图形中的深色和浅色（黑色和白色）区域能够比率最优的分布。

1. 格式和版本信息：生成格式和版本信息放入相应区域内。（版本7-40都包含了版本信息，没有版本信息的全为0。二维码上有两个位置包含了版本信息，它们是冗余的。版本信息共18位，6X3的矩阵，其中6位是数据位，如版本号8，数据位的信息是 001000，后面的12位是纠错位。）

**四、实验内容**

1、本次实验使用的是前端（HTML、CSS、JavaScript）技术实现的；

这是所有的文件：



2、首先来看HTML的关键代码：

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ko" lang="ko">

<head>

<title>二维码的生成</title>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

<meta name="viewport" content="width=device-width,initial-scale=1,user-scalable=no" />

<link rel="stylesheet" type="text/css" href="index.css">

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

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

</head>

<body>

<div class="div1">

<p>网址输入</p>

<input id="text" type="text" value="http://jindo.dev.naver.com/collie" placeholder="输入网址" />

<input type="button" name="button1" value="生成二维码" class="btn">

<div id="qrcode"></div>

</div>

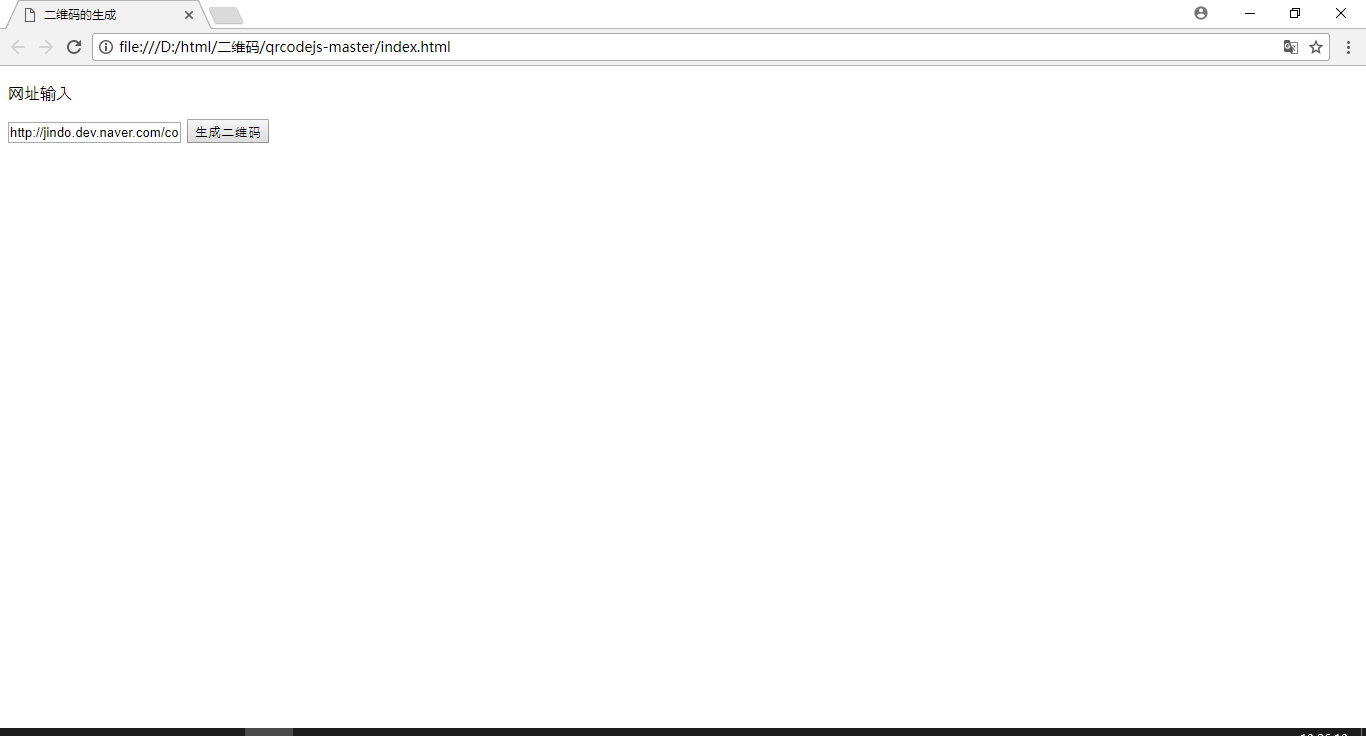
</body>

</html>

*<!-- HTML指超文本标记语言，用来展现页面结构，由一对一对的标签以及标签的属性组成；*

*在<head>标签对中还规定了引用的文件（即两个.js文件和一个.css文件），<meta>标签内规定元数据信息，是页面上看不到的；<div>标签就类似于一个盒子，包裹起我要的东西，分好区块，保证不同的元素之间不互相影响；<p>标签就是一个段落标签；<input>标签为表单标签，在这里，我们添加一个文本框text以及一个按钮button；*

*每个标签还有自己所能够添加的属性，通过设置属性值，可以给元素带来很多的改变，实现不同的功能；-->*

在只有HTML部分代码时，页面的展现效果是： 

图一 html页面展示图

可见，只是有元素，并没有任何其他的东西，就像是我们在网络不好的时候看到只有“文本”的页面一样，一点都不美观；所以我们就需要用另一个语言来将它美化了；

3、CSS（层叠样式表）关键代码（CSS的作用通俗来讲就是美化，规定元素的样式，展示方式，就是<link>标签内src所引入的文件内容，这是外部引用）

\*{

margin: 0;

padding: 0;

}

.div1{

position: absolute;

left: 50%;

top: 10%;

margin-left:-176px;

border:1px solid #aaa;

width: 350px;

text-align: center;

border-radius: 5px;

}

p{

text-align: left;

margin-left: 25px;

color:/\* #3385ff \*/#999;

font-size: 18px;

margin-top: 10px;

font-family: 楷体;

}

#text{

margin-top: 5px;

width: 300px;

height: 30px;

line-height: 30px;

border:1px solid #aaa;

border-radius: 5px;

text-align: center;

}

#text::placeholder{

font-size: 16px;

opacity: 0.6;

font-family: 楷体;

}

.btn{

border:1px solid #c90000;

background-color: #c90a16;

width: 300px;

height: 30px;

letter-spacing: 20px;

margin-top: 30px;

border-radius: 5px;

color: white;

font-size: 18px;

cursor: pointer;

}

.btn:hover{

background-color: #a51b24;

}

#qrcode{

width: 300px;

height: 300px;

margin-top: 15px;

margin-left: 25px;

margin-bottom: 20px;

}

*/\* 前面说过，每个元素都有可以添加的属性，*

*元素的class属性和id属性值就可以作为我们的属性选择器，*

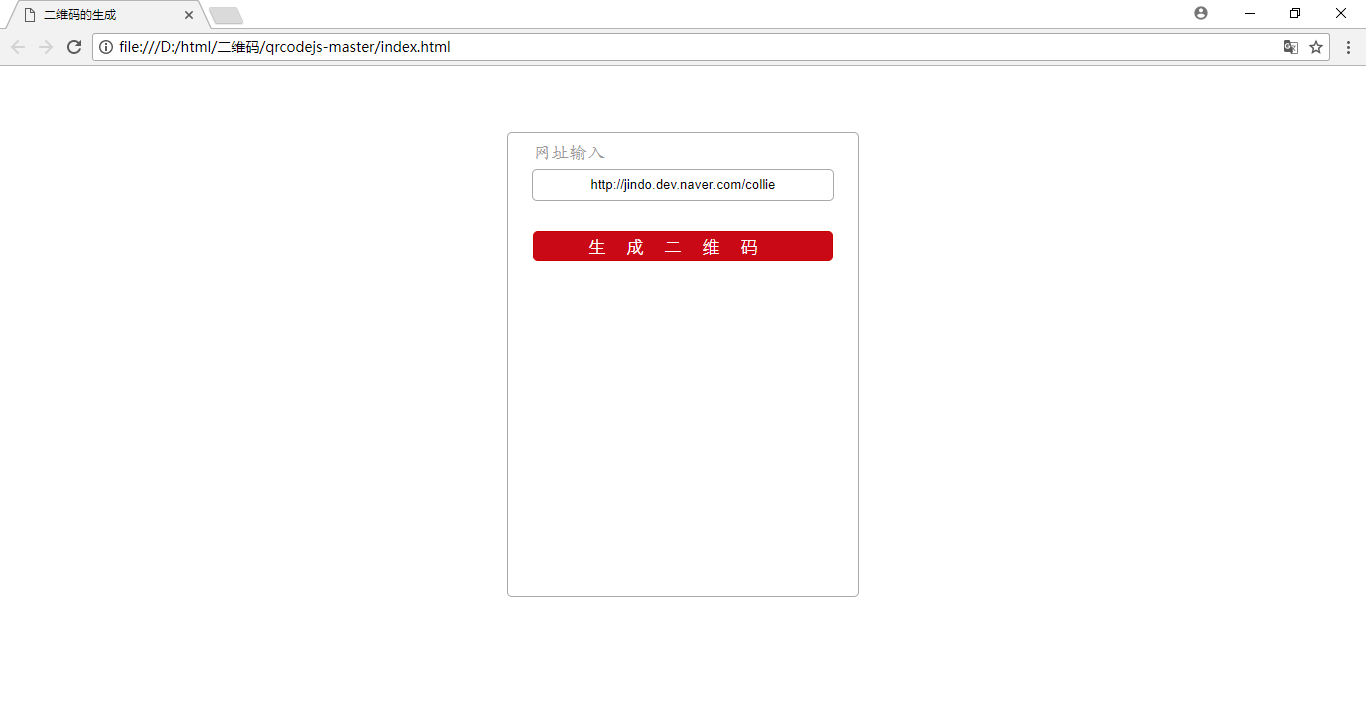
*通过属性选择器来给我们的元素添加样式；*

*例如，我们HTML中的第一个<div>标签有一个class="div1"，*

*我们就可以通过选中它来为我们的<div>添加样式，看代码的第六行用.div来选中<div>*

*并且为其添加了样式；同理，其他的元素也是通过这种方式添加； \*/*

添加了css之后，页面展现如下：



图二 html+css页面展示图

下方的空白区域是用来展现二维码的，因为此时还没有添加动态生成的效果，所以二维码还无法显示；那么如果页面需要有事件发生或者是响应动态效果的话，就需要添加JavaScript代码了；

4、JavaScript是一种脚本语言，用于页面交互；在HTML文件中通过<script>标签来将html和JavaScript语言结合起来，此处引用了两个js的文件，其中一个jquery.min.js是一个JQuery的包，他是JavaScript的一个框架，可以对元素进行一些操作；还有一个是二维码qrcode.js是一个二维码编码的插件，可以直接应用；

然后我们在页面中添加了一些内联的js代码，来对引入的文件进行一个应用，核心代码如下：

<script type="text/javascript">

var qrcode = new QRCode(document.getElementById("qrcode"), {

width : 300,

height : 300

});

*/\*因为直接应用的是插件，应用包里面的对象，创建一个QRCode对象（跟java有一点类似）；*

*通过DOM模型获得文档id为qrcode的元素，设置宽高都为300像素；*

*这里崭新的就是二维码生成的大小了；\*/*

function makeCode () {

var elText = document.getElementById("text");

if (!elText.value) {

elText.focus();

return;

}

qrcode.makeCode(elText.value);

}

*/\*设置一个生成编码的函数，利用if语句条件判断，使鼠标定位选中；*

*再根据输入的二维码的值，生成相应的编码；\*/*

makeCode();

*/\*引用函数\*/*

$(".btn").on("click",function(){

makeCode();

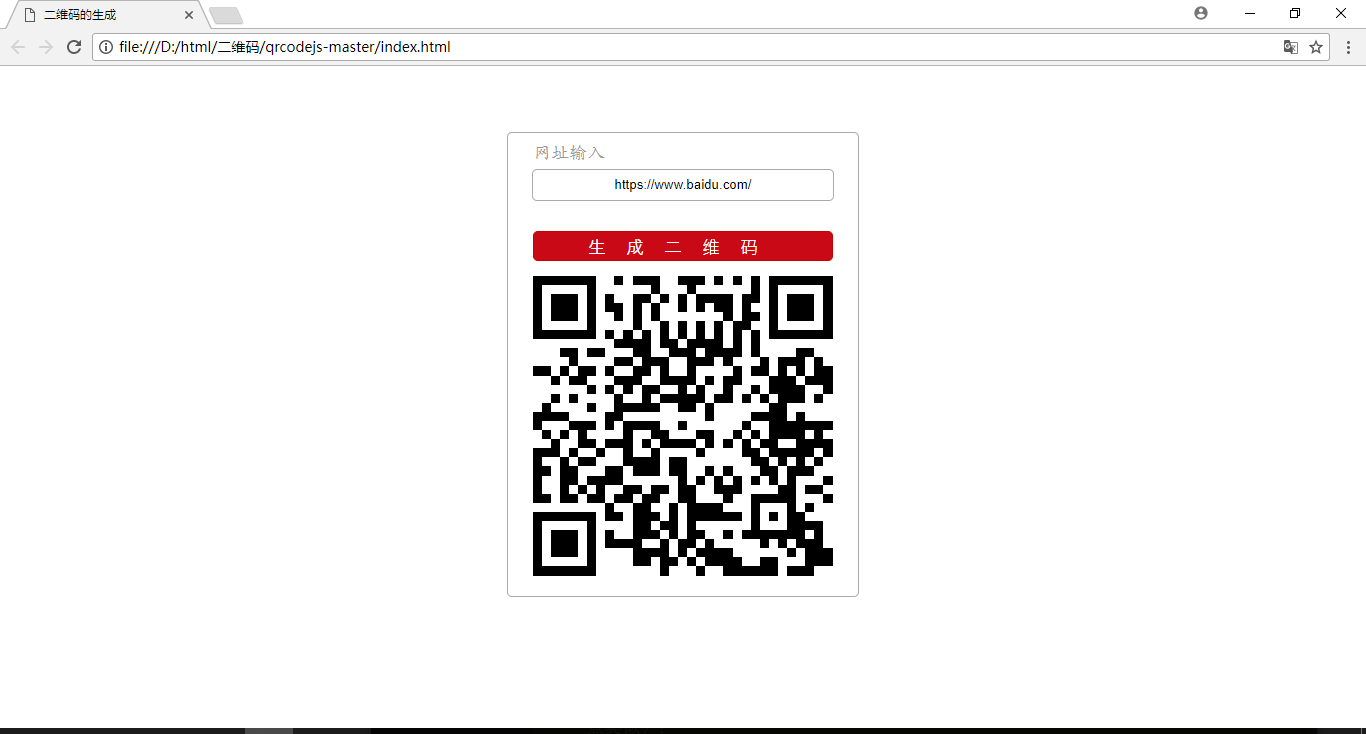
})

*/\*这个地方就是绑定一个事件了，当按钮（生成二维码）被点击的时候，就会执行上面定义的函数，生成相应的二维码；*

*美元符号是jquery的别名，就代表jquery对象\*/*

</script>

添加了内联的js之后，页面展示情况就是（默认用了百度的网址）：

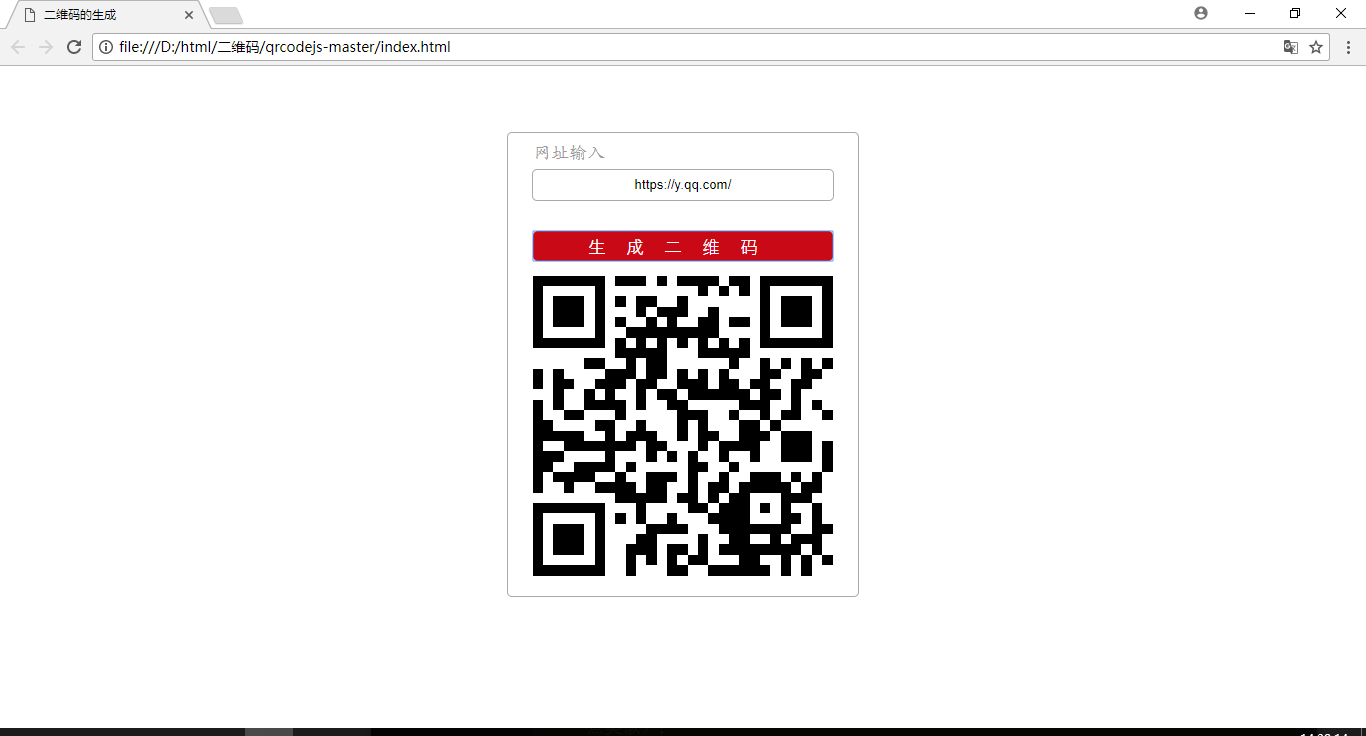


图三 html+css+JavaScript页面展示图

以百度网址为例

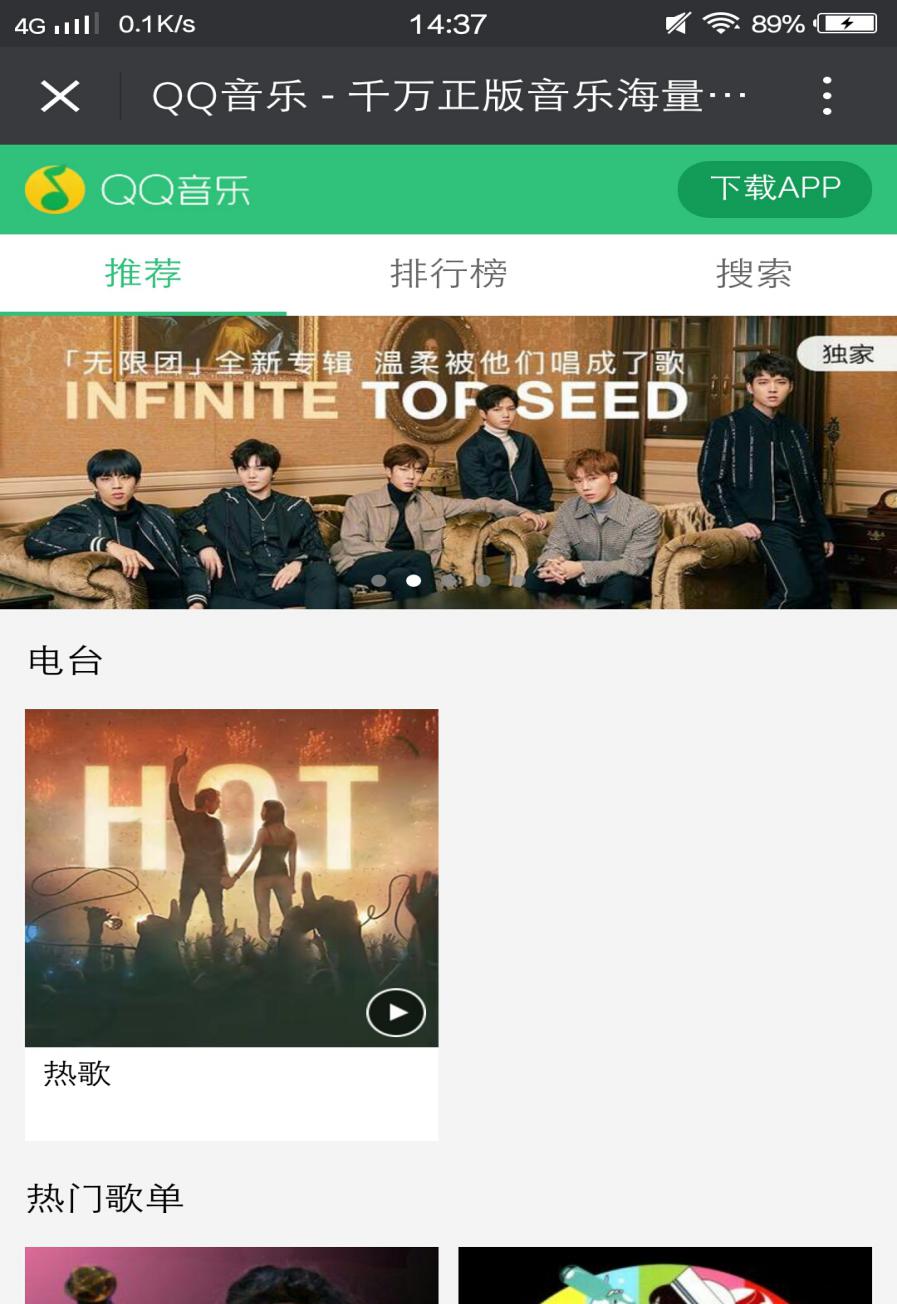
5、然后可以通过在上方的网址输入框修改其他的网址，然后点击下面红色的按钮就可以生成相应的二维码，然后利用手机的扫一扫，可以扫到相应的网页（例如试一下QQ音乐官网）：

二维码页面：



图四 QQ音乐二维码生成页面

手机扫描识别页面：



图五 QQ音乐二维码手机识别页面

综上就完成了二维码的生成与识别；并进过验证可行；

1. **实验小结**
2. 本次实验基本都是按照自己的想法来做的，实验前，先有了一定的想法，查阅了一下关于二维码生成的编码以及识别的一些规则，为后面写代码和看懂插件打下了一定的基础；
3. 然后实验过程中，也没有遇到比较难以理解的问题，在引用插件以及前端框架的时候，要先熟悉一下它的结构，才能真正更好的运用到自己的代码中，得到自己想要的效果；
4. 总的来说，收获还是很有的，虽然我们现在还没有能力自己写一些插件代码，但是去了解和知道怎么去运用也是很有意义的；