

# JavaScript奇淫技巧：隐写术

原创

w2sfot 于 2022-03-09 08:52:55 发布 478 收藏

文章标签：[javascript](#) [隐写术](#) [内容加密](#) [信息隐藏](#)

版权声明：本文为博主原创文章，遵循[CC 4.0 BY-SA](#) 版权协议，转载请附上原文出处链接和本声明。

本文链接：<https://blog.csdn.net/w2sf/article/details/123368234>

版权

本文将用JavaScript实现图片隐写术。

## 什么是隐写术？

简单的来说，比如下面这张图中，隐藏着秘密信息。通过特殊的技术手段可以读取到其信息。



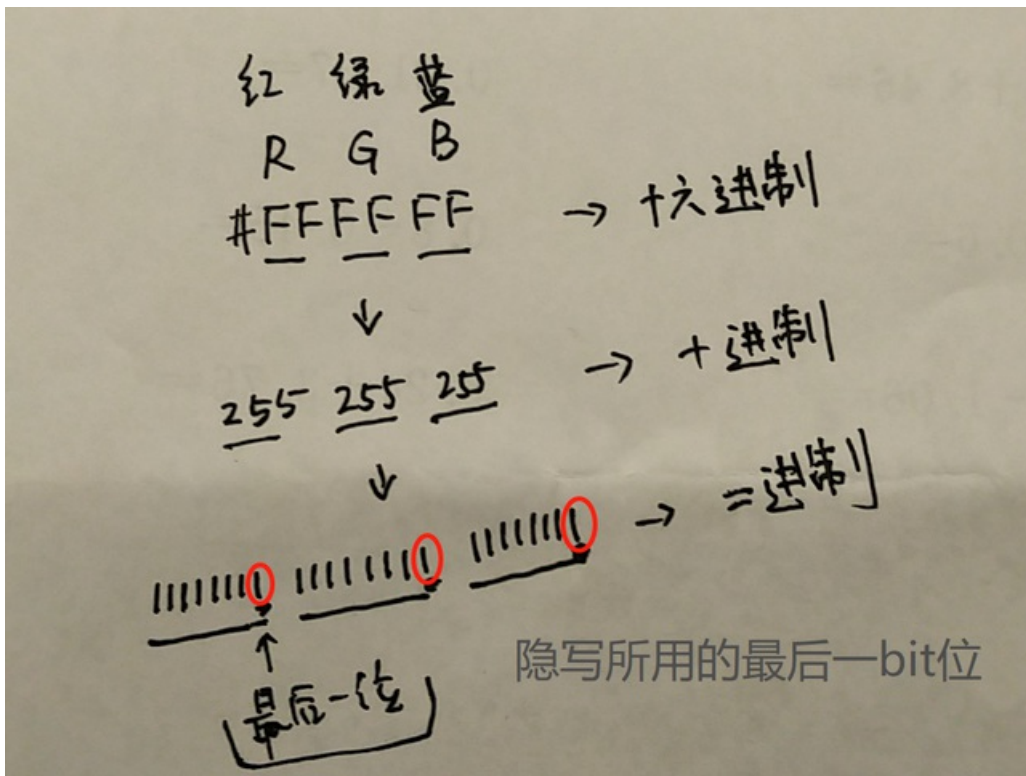
## 效果与用途：

图片隐写术的效果是：可将信息写入图片，并可读取。

可用于：在图片中存放隐秘内容、给图片做不可见的版权标识，等等。

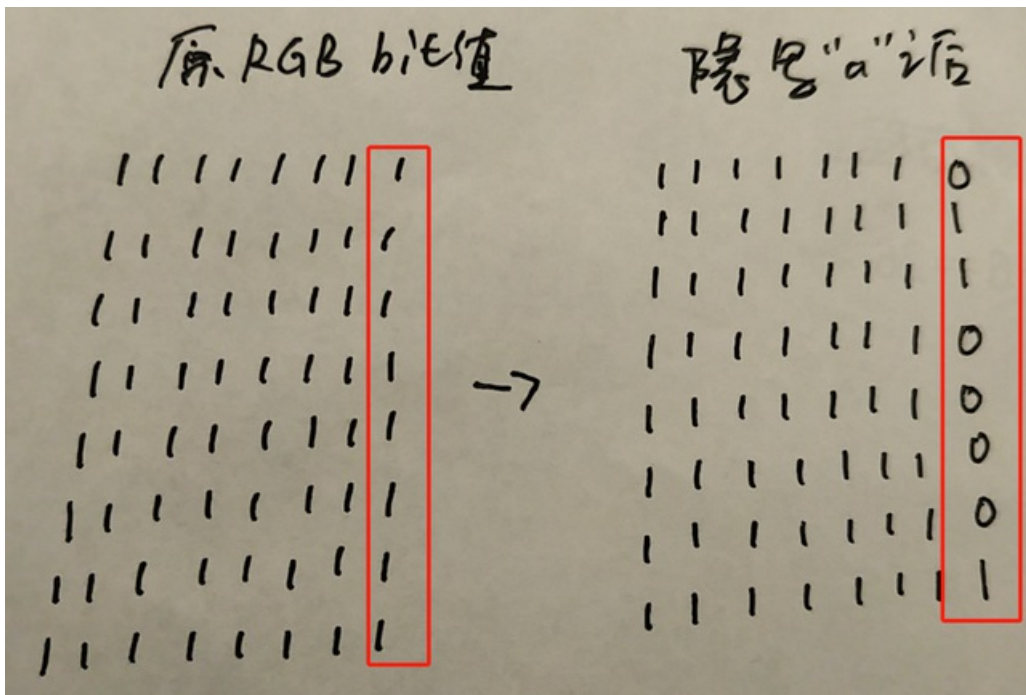
## 技术原理：

图片，是由像素构成；像素都是由RGB三原色组成，比如css中用十六进制定义颜色：`#FFFFFF`。即红绿蓝各为FF。一个FF是转为二进制，是8bit:11111111，如果修改最后一位，是不影响颜色以及图片显示的。正因如此，“隐写术”会将要隐藏的信息，写入到这个8bit的最后一位。



例如：要隐写字符“a”，先转化a为2进制：`"a".charAt(0).toString(2)`

得到值：`"01100001"`，隐写会进行如下操作：



用程序实现时，流程为：

- 1、读取原图，得到图的像素级bit信息；
- 2、将要隐写的信息，转为2进制；
- 3、将二进制的隐写信息，写入到每个像素最后一位；
- 4、保存，生成新图片。

要读取信息时，获取图片像素最后一位，并转二进制为字符信息。

## 源码解析:

以上介绍了技术原理, 接下来, 编程实现。先对源码功能做介绍, 最后将给出完整的功能源码。

本例程源码, 将在一个html文件中实现, 有html代码和JavaScript代码两部分组成。

## Html部分:

```
> demo.html X
C: > Users > WangLiwen > Desktop > <> demo.html > html > body > img#output
1 <html>
2 <body>
3 <!--选择图片文件-->
4 选择文件: <input type='file' id='file' /><br>
5 <!--帆布, 获取图片像素用-->
6 图片预览: <canvas id='canvas' style="width: 600px;"></canvas><br>
7 <!--要隐写的信息-->
8 隐写信息: <textarea id='message'></textarea><br>
9 <button id='encode' class='submit'>隐写</button><br>
10
11 <!--隐写后的图片-->
12 隐写图片: <img id='output' style="width: 600px;" /><br>
13 <button id='decode'>从隐写图片读取信息</button><br>
14 <!--解密出的信息-->
15 读出的隐写内容: <div id='messageDecoded'></div><br>
16
17 <script>
```

html代码有两个组成部分: 读取图片并隐写内容、从隐写图片读取信息。

## JavaScript部分:

初始化时, 给图片文件读取、隐写信息按钮、读取信息按钮等页面元素绑定操作函数:

```
<script>
window.onload = function() {
  //文件读取
  var input = document.getElementById('file');
  input.addEventListener('change', importImage);
  //隐写
  var encodeButton = document.getElementById('encode');
  encodeButton.addEventListener('click', encode);
  //从隐写图片读取
  var decodeButton = document.getElementById('decode');
  decodeButton.addEventListener('click', decode);
};

//选择文件后, 将图片显示在canvas中
var importImage = function(e) {
  var reader = new FileReader();
  reader.onload = function(event) {
    var img = new Image();
```

选择图片后, 显示在画布中:

```
26     var decodeButton = document.getElementById('decode');
27     decodeButton.addEventListener('click', decode);
28 };
29
30 //选择文件后, 将图片显示在canvas中
31 var importImage = function(e) {
32     var reader = new FileReader();
33     reader.onload = function(event) {
34         var img = new Image();
35         img.onload = function() {
36             var ctx = document.getElementById('canvas').getContext('2d');
37             ctx.canvas.width = img.width;
38             ctx.canvas.height = img.height;
39             ctx.drawImage(img, 0, 0);
40         };
41         img.src = event.target.result;
42     };
43     reader.readAsDataURL(e.target.files[0]);
44 };
45
46 //隐写并保存图片
47 var encode = function() {
```

将信息隐写到图片中:

核心代码: 隐写操作:

```
143
144     var encodeMessage = function(colors, message) {
145         var messageBits = getBitsFromNumber(message.length);
146         messageBits = messageBits.concat(getMessageBits(message));
147
148         var history = [];
149         var pos = 0;
150         while (pos < messageBits.length) {
151             var loc = getNextLocation(history, colors.length);
152             colors[loc] = setBit(colors[loc], 0, messageBits[pos]);
153             while ((loc + 1) % 4 !== 0) {
154                 loc++;
155             }
156             colors[loc] = 255;
157
158             pos++;
159         }
160     };
161
```

隐写时, 将信息转为二进制并保存到图片各像素最低位。

前面是隐写部分, 接下来, 是从隐写图片中读取信息:

```
//读出隐写的信息
var decode = function() {
    var ctx = document.getElementById('canvas').getContext('2d');
    var imgData = ctx.getImageData(0, 0, ctx.canvas.width, ctx.canvas.height);
    //核心功能: 从图片数据中读取隐写信息
    var message = decodeMessage(imgData.data);
    alert(message);
    document.getElementById('messageDecoded').innerHTML = message;
};
```

```
var decodeMessage = function(colors) {
    var history = [];

    var messageSize = getNumberFromBits(colors, history);
    if ((messageSize + 1) * 16 > colors.length * 0.75) {
        return '';
    }
    var message = [];
    for (var i = 0; i < messageSize; i++) {
        var code = getNumberFromBits(colors, history);
        message.push(String.fromCharCode(code));
    }
    return message.join('');
};
```

隐写时，是将信息转为二进制，此时是逆操作：

```
88 //将二进制编码信息转为字符串
89 //-----//
90 var getNumberFromBits = function(bytes, history) {
91     var number = 0, pos = 0;
92     while (pos < 16) {
93         var loc = getNextLocation(history, bytes.length);
94         var bit = getBit(bytes[loc], 0);
95         number = setBit(number, pos, bit);
96         pos++;
97     }
98     return number;
99 };
100 var getNextLocation = function(history, total) {
101     var pos = history.length;
102     var loc = Math.abs(pos + 1) % total;
103     while (true) {
104         if (loc >= total) {
105             loc = 0;
106         } else if (history.indexOf(loc) >= 0) {
107             loc++;
108         } else if ((loc + 1) % 4 === 0) {
109             loc++;
110         } else {
111             history.push(loc);
112             return loc;
113         }
114     }
115 };
116 var setBit = function(number, location, bit) {
117     return (number & ~(1 << location)) | (bit << location);
118 };
```

以上即是全部功能代码。总计不到200行，实现了完整的隐写和读取。

实际应用时，隐写部分和读取信息部分，应该是独立、分离使用的。

为了防止读取逻辑被他人分析识破，可以对读取隐写的功能代码做保护。

比如对下面这部分JS代码，使用JShaman进行混淆加密：

```
161
162 var decodeMessage = function(colors) {
163     var history = [];
164
165     var messageSize = getNumberFromBits(colors, history);
166     if ((messageSize + 1) * 16 > colors.length * 0.75) {
167         return '';
168     }
169     var message = [];
170     for (var i = 0; i < messageSize; i++) {
171         var code = getNumberFromBits(colors, history);
172         message.push(String.fromCharCode(code));
173     }
174     return message.join('');
175 };
176
177 </script>
178 </body>
```

在JShaman平台，对代码保护：



生成加密的代码:



再复制回原位置:



这样JS代码将不能被分析，而运行不受影响。

运行效果:

/C:/Users/WangLiwen/Desktop/ +

← → ↻ 🏠 file:///C:/Users/WangLiwen/Desktop/demo.html

选择文件: 浏览... 微信截图\_20220309080834.png



图片预览:

隐写信息: JShaman.com是专业的JavaScript代码保护提供商。

隐写



隐写图片:

从隐写图片读取信息

读出的隐写内容:

JShaman.com是专业的JavaScript代码保护提供商。

### 完整源码:

最后, 附上完整源码, 保存为html文件即可使用。

```
<html>

<body>

<!--选择图片文件-->

选择文件: <input type='file' id='file' /><br>

<!--帆布, 获取图片像素用-->

图片预览: <canvas id='canvas' style="width: 600px;"></canvas><br>

<!--要隐写的信息-->

隐写信息: <textarea id='message'></textarea><br>

<button id='encode' class='submit'>隐写</button><br>
```

<!--隐写后的图片-->

隐写图片: <img id='output' style='width: 600px;' /><br>

<button id='decode'>从隐写图片读取信息</button><br>

<!--解密出的信息-->

读出的隐写内容: <div id='messageDecoded'></div><br>

<script>

window.onload = function() {

//文件读取

var input = document.getElementById('file');

input.addEventListener('change', importImage);

//隐写

var encodeButton = document.getElementById('encode');

encodeButton.addEventListener('click', encode);

//从隐写图片读取

var decodeButton = document.getElementById('decode');

decodeButton.addEventListener('click', decode);

};

//选择文件后, 将图片显示在canvas中

var importImage = function(e) {

var reader = new FileReader();

reader.onload = function(event) {

var img = new Image();

img.onload = function() {

var ctx = document.getElementById('canvas').getContext('2d');

ctx.canvas.width = img.width;

ctx.canvas.height = img.height;

ctx.drawImage(img, 0, 0);

};

img.src = event.target.result;

};

reader.readAsDataURL(e.target.files[0]);



```
};

//隐写并保存图片

var encode = function() {

//信息

var message = document.getElementById('message').value;

//隐写后的图片

var output = document.getElementById('output');

//画布

var canvas = document.getElementById('canvas');

var ctx = canvas.getContext('2d');

console.log(message)

//是否超过能隐写的最大量

var pixelCount = ctx.canvas.width * ctx.canvas.height;

if ((message.length + 1) * 16 > pixelCount * 4 * 0.75) {

alert('内容太多了，超过了可写入的最大量');

return;

}

//核心函数：隐写

var imgData = ctx.getImageData(0, 0, ctx.canvas.width, ctx.canvas.height);

encodeMessage(imgData.data, message);

ctx.putImageData(imgData, 0, 0);

alert('隐写成功，信息已隐藏到图片中');

//显示出隐写后的图片

output.src = canvas.toDataURL();

};

//读出隐写的信息

var decode = function() {

var ctx = document.getElementById('canvas').getContext('2d');

var imgData = ctx.getImageData(0, 0, ctx.canvas.width, ctx.canvas.height);

//核心功能：从图片数据中读取隐写信息

var message = decodeMessage(imgData.data);
```

```
alert(message)

document.getElementById('messageDecoded').innerHTML = message;

};

//将二进制编码信息转为字符串
//-----//

var getNumberFromBits = function(bytes, history) {

var number = 0, pos = 0;

while (pos < 16) {

var loc = getNextLocation(history, bytes.length);

var bit = getBit(bytes[loc], 0);

number = setBit(number, pos, bit);

pos++;

}

return number;

};

var getNextLocation = function(history, total) {

var pos = history.length;

var loc = Math.abs(pos + 1) % total;

while (true) {

if (loc >= total) {

loc = 0;

} else if (history.indexOf(loc) >= 0) {

loc++;

} else if ((loc + 1) % 4 === 0) {

loc++;

} else {

history.push(loc);

return loc;

}

}

};
```

```

var setBit = function(number, location, bit) {

return (number & ~(1 << location)) | (bit << location);

};

//-----//

//将信息字符串转为二进制编码

//-----//

var getMessageBits = function(message) {

var messageBits = [];

for (var i = 0; i < message.length; i++) {

var code = message.charCodeAt(i);

messageBits = messageBits.concat(getBitsFromNumber(code));

}

return messageBits;

};

var getBitsFromNumber = function(number) {

var bits = [];

for (var i = 0; i < 16; i++) {

bits.push(getBit(number, i));

}

return bits;

};

var getBit = function(number, location) {

return ((number >> location) & 1);

};

//-----//

var encodeMessage = function(colors, message) {

var messageBits = getBitsFromNumber(message.length);

messageBits = messageBits.concat(getMessageBits(message));

var history = [];

var pos = 0;

while (pos < messageBits.length) {

```

```

var loc = getNextLocation(history, colors.length);

colors[loc] = setBit(colors[loc], 0, messageBits[pos]);

while ((loc + 1) % 4 !== 0) {

loc++;

}

colors[loc] = 255;

pos++;

}

};

var decodeMessage = function(colors) {

/*Obfuscated by JShaman.com*/var _0x265a=['length','push','fromCharCode','join'];var
_0x1c66=function(_0x265a55,_0x1c6643){_0x265a55=_0x265a55-0x0;var _0x2081ad=_0x265a[_0x265a55];return
_0x2081ad;};var _0x2ad986=function(_0x5d3dbb,_0x36e20f,_0x4c778b,_0x1e11f6,_0x377eb9){return
_0x1c66(_0x1e11f6- -0x169,_0x4c778b);};var
_0xeb9032=function(_0x1ff9d9,_0x7ca6ec,_0x5d43c2,_0xaf192e,_0x146982){return _0x1c66(_0xaf192e- -
0x169,_0x5d43c2);};var _0x4e4429=function(_0x1099e4,_0x3d47d8,_0x9623bb,_0x8f809f,_0x2632e4){return
_0x1c66(_0x8f809f- -0x169,_0x9623bb);};var
_0x3ab227=function(_0xe7a97,_0x3e4f1d,_0x3dc243,_0x7d20c1,_0x541a1b){return _0x1c66(_0x7d20c1- -
0x169,_0x3dc243);};var history=[];var
messageSize=getNumberFromBits(colors,history);if((messageSize+0x1)*0x10>colors[_0x2ad986(-0x167,-0x169,-
0x168,-0x169,-0x16a)]*0.75){return"";var message=[];for(var i=0x0;i<messageSize;i++){var
code=getNumberFromBits(colors,history);message[_0x2ad986(-0x168,-0x169,-0x16a,-0x168,-0x169)]
(String[_0x4e4429(-0x168,-0x168,-0x167,-0x167,-0x167)](code));return message[_0x4e4429(-0x166,-0x165,-0x166,-
0x166,-0x166)]("");

};

</script>

</body>

</html>

```



[创作打卡挑战赛](#) >

[赢取流量/现金/CSDN周边激励大奖](#)