

Script

脚本之家

www.jb51.net

首页

网页制作

脚本专栏

脚本下载

网络编程

数据库

CMS教程

电子书籍

平面设计

媒体动画

操作系统

网站运营

网络安全

基础知识

javascript类库

表单特效

广告代码

网页特效

黑客性质

javascript技巧

DOM

node.js

特色栏目:

vbscript

正则表达式

javascript

批处理

服务器软件

素材下载

搜索

页面导航: [首页](#) → [网络编程](#) → [JavaScript](#) → [javascript技巧](#) → 正文内容 [js 随机 颜色](#)

js随机颜色代码的多种实现方式

作者: 字体: [增加 减小] 类型: 转载

本篇文章小编为大家介绍，js随机颜色代码的多种实现方式。需要的朋友参考下

JS随机颜色有很多地方要用到：比如大家看到很多标签连接都是五颜六色。那就需要到这个了。下面开始：

方法思路总共有二。一是准备一组漂亮的候选颜色，二是随机生成颜色。

实现1

代码如下:

```
var getRandomColor = function(){

return '#' +
(function(color){
return (color += '0123456789abcdef'[Math.floor(Math.random()*16)])
&& (color.length == 6) ? color : arguments.callee(color);
})("");
}


```

复制代码

随机生成6个字符然后再串到一起，闭包调用自身与三元运算符让程序变得内敛，初心者应该好好学习这种写法。

实现2

代码如下:

```
var getRandomColor = function(){

return (function(m,s,c){
return (c ? arguments.callee(m,s,c-1) : '#') +
s[Math.floor(m.random() * 16)]
})(Math,'0123456789abcdef',5)
}


```

复制代码

把Math对象，用于生成hex颜色值的字符串提取出来，并利用第三个参数来判断是否还继续调用自身。

实现3

代码如下:

```
Array.prototype.map = function(fn, thisObj) {

var scope = thisObj || window;


```

复制代码

以下为引用的内容:

```
Array.prototype.map = function(fn, thisObj) {

var scope = thisObj || window;


```

最近更新

js去除重复字符串两种实现方法

在Windows上安装Node.js模块的方法

一行代码实现纯数据json对象的深度克隆实

一个不错的js html页面倒计时可精确到秒

js父窗口关闭时子窗口随之关闭完美解决方

通过身份证号得到出生日期和性别的js代码

js数值计算时使用parseInt进行数据类型转

JavaScript的IE和Firefox兼容性汇编

javascript用DIV模拟弹出窗口_窗体滚动跟

从父页面读取和操作iframe中内容方法

热点排行

js刷新页面方法大全

清除网页历史记录，屏蔽后退按钮

js中cookie的使用详细分析

javascript深入理解js闭包

JS鼠标事件大全 推荐收藏

Javascript 的addEventListener(

js验证表单大全

js 获取浏览器高度和宽度值(多浏

js关闭当前页面(窗口)的几种方式

js中eval详解

http://www.jb51.net/article/35931.htm

1/5

```
var a = [];  
for ( var i=0, j=this.length; i < j; ++i ) {  
    a.push(fn.call(scope, this[i], i, this));  
}  
return a;  
};  
var getRandomColor = function(){  
    return '#'+'0123456789abcdef'.split('').map(function(v,i,a){  
        return i>5 ? null : a[Math.floor(Math.random()*16)] }).join('');  
}
```

这个要求我们对数组做些扩展，**map**将返回一个数组，然后我们再用**join**把它的元素串成字符。

实现4

代码如下:

[复制代码](#)

以下为引用的内容:

```
var getRandomColor = function(){  
    return '#' + Math.floor(Math.random()*16777215).toString(16);  
}
```

这个实现非常逆天，虽然有点小bug。我们知道hex颜色值是从#000000到#ffffff，后面那六位数是16进制数，相当于"0x000000"到"0xffffffff"。这实现的思路是将hex的最大值ffffff先转换为10进制，进行random后再转换回16进制。我们看一下，如何得到16777215 这个数值的。

代码如下:

[复制代码](#)

以下为引用的内容:

```
<!doctype html>  
<meta charset="utf-8"/>  
<meta http-equiv="X-UA-Compatible" content="IE=8"/>  
<title>hex的最大值</title>  
<script type="text/javascript" charset="utf-8">  
    window.onload = function () {  
        alert(parseInt("0xffffffff",16).toString(10));  
    };  
</script>  
<div id="text"></div>
```

实现5

代码如下:

[复制代码](#)

以下为引用的内容:

```
var getRandomColor = function(){  
    return '#' + (Math.random()*0xffffffff<0).toString(16);  
}
```

基本实现4的改进，利用左移运算符把0xffffffff转化为整型。这样就不用记16777215了。由于左移运算符的优先级比不上乘号，因此随机后再左移，连Math.floor也不用了。

实现6

代码如下:

[复制代码](#)

以下为引用的内容:

```
var getRandomColor = function(){
    return '#' + (function(h){
        return new Array(7 - h.length).join("0") + h
    })(Math.random() * 0x1000000 < 0).toString(16))
}
```

修正上面版本的bug（无法生成纯白色与hex位数不足问题）。0x1000000相当0xffffffff+1，确保会抽选到0xffff。在闭包里我们处理hex值不足5位的问题，直接在未位补零。

实现7

代码如下:

[复制代码](#)

以下为引用的内容:

```
var getRandomColor = function(){
    return '#' + ('00000' + (Math.random() * 0x1000000 < 0).toString(16)).substr(-6);
}
```

这次在前面补零，连递归检测也省了。

实战一下:

代码如下:

[复制代码](#)

以下为引用的内容:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html lang="zh-cn">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>初级饼图</title>
<script src="http://bloghighlighter.googlecode.com/files/raphael-min.js" type="text/javascript" ></s
cript>
<script type="text/javascript" charset="utf-8">
var getRandomColor = function(){
    return '#' + ('00000' + (Math.random() * 0x1000000 < 0).toString(16)).substr(-6);
}

window.onload = function () {
    var paper = Raphael("canvas", 700, 700);
    paper.rect(0, 0, 640, 480, 10).attr({fill: "#F2F1D7", stroke: "none"}); //设置画板

    function drawSector(cx, cy, r, paper, oc, startAngle){
        var angleplus = 360 * oc / 100, //360度乘以40%
            startAngle = startAngle || 0,
            endAngle = startAngle + angleplus,
            rad = Math.PI / 180,
            x1 = cx + r * Math.cos(-startAngle * rad),
            x2 = cx + r * Math.cos(-endAngle * rad),
            y1 = cy + r * Math.sin(-startAngle * rad),
            y2 = cy + r * Math.sin(-endAngle * rad);
        var path = ["M", cx, cy, "L", x1, y1, "A", r, r, 0, +(endAngle - startAngle > 180), 0, x2, y2, "z"],

            path = path.join(" ");
        paper.path({fill: getRandomColor(), path});
        return endAngle
    }

    var ocs = [40, 25, 17, 10, 8];
    for(var i=0, l=ocs.length, startAngle; i<l; i++){
```

```
startAngle = drawSector(300,300,100,paper,ocs[i],startAngle);
    }
};
</script>
<style type="text/css" media="screen">
    #canvas {
        width: 700px;
        height: 700px;
    }
</style>
<title>初级2324234饼图</title>
</head>
<body>
    <p>初级23232饼图</p>
    <div id="canvas"></div>
</body>
</html>
```

- 您可能感兴趣的文章：
- js 随机数代码大全
 - js随机密码产生函数
 - Js 随机数产生6位数字
 - js 生成随机汉字的问题
 - JS随机生成不重复数据的实例方法

QQ空间 新浪微博 腾讯微博 搜狐微博 人人网 开心网 百度搜藏 更多 0

Tags: js 随机 颜色

复制链接 | 收藏本文 | 打印本文 | 关闭本文 | 返回首页

上一篇: js对象与打印对象分析比较 下一篇: JavaScript中的style.cssText使用教程

相关文章	
js Function类型	2011-12-12
js中根据字数截取字符串，不能截断url	2012-01-01
js COL能很好的控制表格的列	2008-04-04
解决function函数内的循环变量	2008-10-10
js+ajax实现的A*游戏路径算法整理	2007-05-05
控制input输入框中提示信息的显示和隐藏的方法	2014-02-02
用javascript实现gb2312转utf-8的脚本	2007-05-05
javascript 隐藏/显示指定的区域附HTML元素【legend】用法	2010-03-03
JavaScript中发布/订阅模式的简单实例	2014-11-11
cnblogs csdn 代码运行框实现代码	2009-11-11

文章评论

社交帐号登录: 微博 QQ 人人 豆瓣 更多»



说点什么吧...

发布

评论

最新 最早 最热

还没有评论，沙发等你来抢

脚本之家正在使用多说

