

全部课程 (/courses/) / 基于 HTML5 实现刮刮乐效果 (/courses/133) / HTML5实现刮刮乐效果

在线实验，请到PC端体验

基于 HTML5 实现刮刮乐效果

一、实验介绍

1.1 实验内容

欢迎来到实验楼之基于 **HTML5** 实现刮刮乐的课程。

本课程较为简单，适合刚学完前端基础(HTML+JavaScript)的同学。只需要花费很短的时间，你也可以实现刮刮乐的效果。

实验完成后的效果图：

Feeling Lucky? Try![Try Again](#)



1.2 实验知识点

本次实验涉及到的知识点如下：

- HTML5
- JavaScript

1.3 实验环境

1. Terminal：Linux 命令行终端，打开后会进入 **Bash** 环境，可以使用 **Linux** 命令
2. Firefox：浏览器，可以在需要前端界面的课程里，只需要打开环境里写的 **HTML/JS** 页面即可
3. GVim：非常好用的编辑器，最简单的用法可以参考课程 **Vim编辑器** (<http://www.shiyanlou.com/courses/2>)（或者使用 **Sublime Text** 编辑器）

实验楼的 **Sublime Text** 编辑器位于：应用程序菜单->开发 下。

1.4 适合人群

本课程较为简单，适合刚学完前端基础(HTML+JavaScript)的同学。

1.5 代码获取

```
$ wget http://labfile.oss.aliyuncs.com/courses/133/guaguale.zip
```

三、开发准备

先建立下面的目录结构：

动手实践是学习 **IT** 技术最有效的方式！

开始实验

```
guaguale
|__index.html
|__main.js
```

然后我们需要在项目目录下下载俩张实验要用到的图片：

```
$ wget http://labfile.oss.aliyuncs.com/courses/133/p_0.jpg
$ wget http://labfile.oss.aliyuncs.com/courses/133/p_1.jpg
```

四、实验步骤

4.1 编写 HTML

代码后的注释部分只是方便你理解代码，不是必须敲进去的。

由于代码十分简单，所以对于代码的讲解都写在其中的注释了：

```
<!--代码注释：可理解为声明文档类型为HTML-->
<!DOCTYPE HTML>
<html>
<head>

<!--代码注释：设置编码为utf-8-->
<meta charset="utf-8">

<!--代码注释：此段代码的效果你可以理解为实现自动适应屏幕的效果-->
<meta name="viewport" content="width=device-width; initial-scale=1.0">

<title>使用HTML5实现刮刮乐效果</title>
</head>
<body>
  <div id="main">
    <div class="msg">Feeling Lucky? Try!<a href="javascript:void(0)" onClick="window.location.reload()">Try Again</a>
  </div>
  <!--代码注释： window.location.reload()可理解为刷新页面的意思-->
  <div>
    <!--代码注释：下面就是引入<canvas>-->
    <canvas></canvas>
  </div>
</div>
<!--代码注释：引入 JavaScript 代码 -->
<script src="main.js"></script>
</body>
</html>
```

恭喜你，第一步完成，可以看到 HTML 的代码很简单，只是提供了 canvas 区域和引入了 JavaScript 文件。

4.2 编写 JavaScript

下面的就来编写我们的 JavaScript 代码，代码中已经给出了详细的注释：

//代码注释:首先,我们要禁用页面的鼠标选中拖动的事件,就是不运行执行选中操作。

```
var bodyStyle = document.body.style;
bodyStyle.mozUserSelect = 'none';
bodyStyle.webkitUserSelect = 'none';
```

//代码注释:接着我们定义图片类,获取canvas元素,并设置背景和位置属性。

```
var img = new Image();
var canvas = document.querySelector('canvas');
canvas.style.backgroundColor='transparent';
canvas.style.position = 'absolute';
```

//代码注释:我们在本例中用到两张随机照片,每次刷新随机一张图片作为背景。

```
var imgs = ['p_0.jpg','p_1.jpg'];
var num = Math.floor(Math.random()*2);
img.src = imgs[num];
```

//然后进入主体,当检测到图片加载完的时候,首先定义一些属性和函数,函数layer()用来绘制一个灰色的正方形,eventDown()定义了按下事件eventUp()定义了松开事件,eventMove()定义了移动事件,其中当按下时,获取坐标位移,并通过arc(x, y, 10, 0, Math.PI * 2)来绘制小圆点。

```
img.addEventListener('load', function(e) {
    var ctx;
    var w = img.width,
        h = img.height;
    var offsetX = canvas.offsetLeft,
        offsetY = canvas.offsetTop;
    var mousedown = false;

    function layer(ctx) {
        ctx.fillStyle = 'gray';
        ctx.fillRect(0, 0, w, h);
    }

    function eventDown(e){
        e.preventDefault();
        mousedown=true;
    }

    function eventUp(e){
        e.preventDefault();
        mousedown=false;
    }

    function eventMove(e){
        e.preventDefault();
        if(mousedown) {
            if(e.changedTouches){
                e=e.changedTouches[e.changedTouches.length-1];
            }
            var x = (e.clientX + document.body.scrollLeft || e.pageX) - offsetX || 0,
                y = (e.clientY + document.body.scrollTop || e.pageY) - offsetY || 0;
            with(ctx) {
                beginPath()

                //代码注释: 绘制圆点。
                arc(x, y, 10, 0, Math.PI * 2);
                fill();
            }
        }
    }
}
```

//最后,通过canvas调用以上函数,绘制图形,并且侦听触控及鼠标事件,调用相应的函数。

```
canvas.width=w;
canvas.height=h;
canvas.style.backgroundImage='url('+img.src+')';
ctx=canvas.getContext('2d');
ctx.fillStyle='transparent';
ctx.fillRect(0, 0, w, h);//代码注释: 绘制矩形。
layer(ctx);
```

```
ctx.globalCompositeOperation = 'destination-out';
```

```
canvas.addEventListener('touchstart', eventDown);
canvas.addEventListener('touchend', eventUp);
canvas.addEventListener('touchmove', eventMove);
canvas.addEventListener('mousedown', eventDown);
canvas.addEventListener('mouseup', eventUp);
canvas.addEventListener('mousemove', eventMove);
```

```
});
```

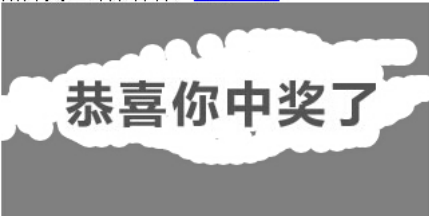
动手实践是学习新技术最有效的方式!

开始实验

完成上述文件后保存，然后使用 **Firefox**浏览器 打开 index.html 文件。

文件打开的效果如下图所示：

刮开灰色部分看看，[再来一次](#)



五、实验总结

通过本次实验，我们通过 **HTML5** 和 **JavaScript** 完成了一个简易的刮刮卡项目，考察了自己对于前端的掌握程度。

大家可以考虑下是否可以使用类似的 **HTML5** 的代码完成一个猜成语游戏，比如显示成语中的两个字，而隐藏另外两个字。

欢迎将你的作品写到实验楼的实验报告并发布，与其他同学交流讨论。

课程教师



Silence_Zhang
共发布过2门课程

[查看老师的所有课程 > \(/teacher/11944\)](#)

前置课程

[HTML基础入门 \(/courses/19\)](#)

[Javascript基础（新版） \(/courses/21\)](#)

进阶课程

[网页版扫雷 \(/courses/144\)](#)

[网页版别踩白块游戏 \(/courses/306\)](#)



动手做实验，轻松学IT



公司 <http://weibo.com/shiyanlou2013>

关于我们 (/aboutus)
联系我们 (/contact)
加入我们 (<http://www.simplecloud.cn/jobs.html>)
技术博客 (<https://blog.shiyanlou.com>)

服务

企业版 (/saas)
实战训练营 (/bootcamp/)
会员服务 (/vip)
实验报告 (/courses/reports)
常见问题 (/questions/?tag=%E5%B8%B8%E8%A7%81%E9%97%AE%E9%A2%98)
隐私条款 (/privacy)

合作

我要投稿 (/contribute)
教师合作 (/labs)
高校合作 (/edu/)
友情链接 (/friends)
开发者 (/developer)
学习路径
Python学习路径 (/paths/python)
Linux学习路径 (/paths/linuxdev)
大数据学习路径 (/paths/bigdata)
Java学习路径 (/paths/java)
PHP学习路径 (/paths/php)
全部 (/paths/)

动手实践是学习 IT 技术最有效的方式！ [开始实验](#)