Interactive greeding card design

Comment:

Play Music Function

Music Play

所有的气球，
做一个10秒的动画，
就是气球，分别各自
从下面往上面升起来。
顺序不一，一共10妙就成。

CAKE 要用Canvas API 来画出来。下面我有提供 Birthday cake的 canvas code

Make A Wish

Form Function
做一个form，是为了
Inputs are validated and display error messages.

Name:
Email:
Birthday Wish:

Sent

Re-fill the form

笑脸要用Canvas API 来画出来。下面我有提供笑脸的 canvas code

Decorate Your Cake

Candle

Purple Candy

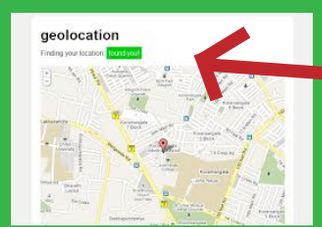
Blue Candy

Balloon

Chocolate

Chocolate

Drag & drop
做一个Drag Drop 的功能，点击选择每一个图案，就会出现在蛋糕上，然后可以随便的移动着图形来装饰蛋糕。
比如：点击蜡烛，蜡烛出现在粉色蛋糕上，然后可以随便移动蜡烛的位置，进行蛋糕装饰。



这里在来一个
geolocation html5 的功能。

Birthday cake canvas code ：

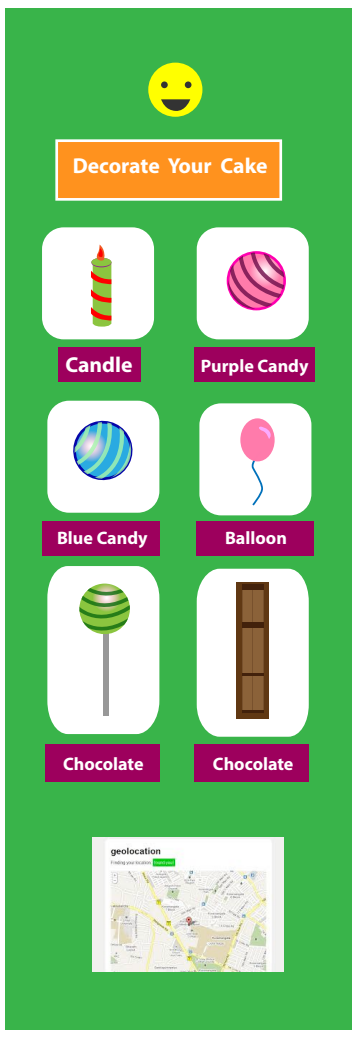
我是从这个网站上学的怎么生成的代码：

http://tomkrcha.com/?p=3686

```
UIColor* fillColor = [UIColor colorWithRed: 1 green: 0.482 blue: 0.674 alpha: 1];
UIBezierPath* path = [UIBezierPath bezierPath];
[path moveToPoint: CGPointMake(1513,957)];
[path addCurveToPoint: CGPointMake(1502,969) controlPoint1: CGPointMake(1513,962) controlPoint2: CGPointMake(1509,969)];
[path addLineToPoint: CGPointMake(1247,969)];
[path addCurveToPoint: CGPointMake(1234,957) controlPoint1: CGPointMake(1240,969) controlPoint2: CGPointMake(1234,962)];
[path addLineToPoint: CGPointMake(1234,897)];
[path addCurveToPoint: CGPointMake(1247,889) controlPoint1: CGPointMake(1234,892) controlPoint2: CGPointMake(1240,889)];
[path addLineToPoint: CGPointMake(1502,889)];
[path addCurveToPoint: CGPointMake(1513,897) controlPoint1: CGPointMake(1509,889) controlPoint2: CGPointMake(1513,892)];
[path addLineToPoint: CGPointMake(1513,957)];
[fillColor setFill];
[path fill];

UIColor* fillColor1 = [UIColor colorWithRed: 1 green: 0.482 blue: 0.674 alpha: 1];
UIBezierPath* path1 = [UIBezierPath bezierPath];
[path1 moveToPoint: CGPointMake(1477,877)];
[path1 addCurveToPoint: CGPointMake(1469,888) controlPoint1: CGPointMake(1477,882) controlPoint2: CGPointMake(1474,888)];
[path1 addLineToPoint: CGPointMake(1278,888)];
[path1 addCurveToPoint: CGPointMake(1268,877) controlPoint1: CGPointMake(1273,888) controlPoint2: CGPointMake(1268,882)];
[path1 addLineToPoint: CGPointMake(1268,829)];
[path1 addCurveToPoint: CGPointMake(1278,822) controlPoint1: CGPointMake(1268,824) controlPoint2: CGPointMake(1273,822)];
[path1 addLineToPoint: CGPointMake(1469,822)];
[path1 addCurveToPoint: CGPointMake(1477,829) controlPoint1: CGPointMake(1474,822) controlPoint2: CGPointMake(1477,824)];
[path1 addLineToPoint: CGPointMake(1477,877)];
[fillColor1 setFill];
[path1 fill];

UIColor* fillColor2 = [UIColor colorWithRed: 1 green: 0.482 blue: 0.674 alpha: 1];
UIBezierPath* path2 = [UIBezierPath bezierPath];
[path2 moveToPoint: CGPointMake(1438,812)];
[path2 addCurveToPoint: CGPointMake(1434,821) controlPoint1: CGPointMake(1438,816) controlPoint2: CGPointMake(1437,821)];
[path2 addLineToPoint: CGPointMake(1319,821)];
[path2 addCurveToPoint: CGPointMake(1313,812) controlPoint1: CGPointMake(1316,821) controlPoint2: CGPointMake(1313,816)];
[path2 addLineToPoint: CGPointMake(1313,775)];
[path2 addCurveToPoint: CGPointMake(1319,770) controlPoint1: CGPointMake(1313,771) controlPoint2: CGPointMake(1316,770)];
[path2 addLineToPoint: CGPointMake(1434,770)];
[path2 addCurveToPoint: CGPointMake(1438,775) controlPoint1: CGPointMake(1437,770) controlPoint2: CGPointMake(1438,771)];
[path2 addLineToPoint: CGPointMake(1438,812)];
[fillColor2 setFill];
[path2 fill];

UIColor* fillColor3 = [UIColor colorWithRed: 0.8 green: 0.8 blue: 0.8 alpha: 1];
UIBezierPath* path3 = [UIBezierPath bezierPathWithRect: CGRectMake(1312, 792, 127, 7)];
[fillColor3 setFill];
[path3 fill];

UIColor* fillColor4 = [UIColor colorWithRed: 0.8 green: 0.8 blue: 0.8 alpha: 1];
UIBezierPath* path4 = [UIBezierPath bezierPathWithRect: CGRectMake(1267, 853, 211, 8)];
[fillColor4 setFill];
[path4 fill];

UIColor* fillColor5 = [UIColor colorWithRed: 0.8 green: 0.8 blue: 0.8 alpha: 1];
UIBezierPath* path5 = [UIBezierPath bezierPathWithRect: CGRectMake(1232, 926, 282, 9)];
[fillColor5 setFill];
[path5 fill];

UIColor* fillColor6 = [UIColor colorWithRed: 0.8 green: 0.737 blue: 0.737 alpha: 1];
UIBezierPath* path6 = [UIBezierPath bezierPath];
[path6 moveToPoint: CGPointMake(1381,770)];
[path6 addLineToPoint: CGPointMake(1373,770)];
[path6 addLineToPoint: CGPointMake(1373,738)];
[path6 addLineToPoint: CGPointMake(1381,738)];
[path6 addLineToPoint: CGPointMake(1381,770)];
[fillColor6 setFill];
[path6 fill];

UIColor* fillColor7 = [UIColor colorWithRed: 0.945 green: 0.352 blue: 0.141 alpha: 1];
UIBezierPath* path7 = [UIBezierPath bezierPath];
[path7 moveToPoint: CGPointMake(1376,718)];
[path7 addCurveToPoint: CGPointMake(1374,729) controlPoint1: CGPointMake(1376,718) controlPoint2: CGPointMake(1376,725)];
[path7 addCurveToPoint: CGPointMake(1373,738) controlPoint1: CGPointMake(1372,733) controlPoint2: CGPointMake(1371,736)];
[path7 addCurveToPoint: CGPointMake(1381,739) controlPoint1: CGPointMake(1375,740) controlPoint2: CGPointMake(1379,740)];
[path7 addCurveToPoint: CGPointMake(1376,718) controlPoint1: CGPointMake(1383,737) controlPoint2: CGPointMake(1384,724)];
[fillColor7 setFill];
[path7 fill];

UIColor* fillColor8 = [UIColor colorWithRed: 1 green: 0 blue: 0 alpha: 1];
UIBezierPath* path8 = [UIBezierPath bezierPath];
[path8 moveToPoint: CGPointMake(1376,723)];
[path8 addCurveToPoint: CGPointMake(1375,729) controlPoint1: CGPointMake(1376,723) controlPoint2: CGPointMake(1376,727)];
[path8 addCurveToPoint: CGPointMake(1375,734) controlPoint1: CGPointMake(1374,731) controlPoint2: CGPointMake(1374,733)];
[path8 addCurveToPoint: CGPointMake(1379,735) controlPoint1: CGPointMake(1376,735) controlPoint2: CGPointMake(1378,735)];
[path8 addCurveToPoint: CGPointMake(1376,723) controlPoint1: CGPointMake(1381,733) controlPoint2: CGPointMake(1381,726)];
[fillColor8 setFill];
[path8 fill];
```

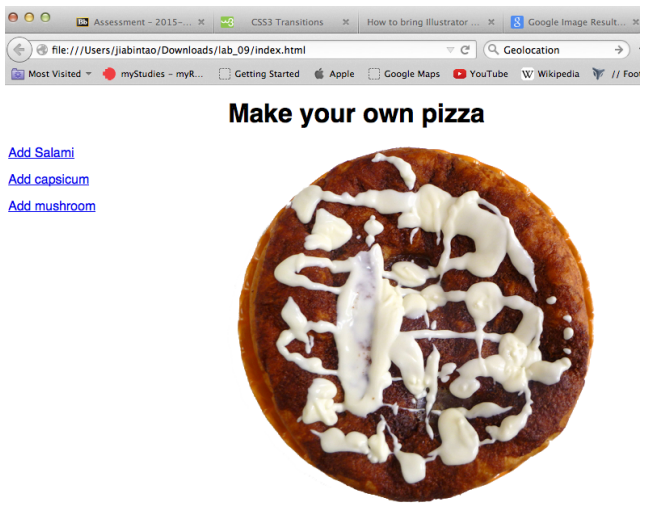
Smile face canvas code ：

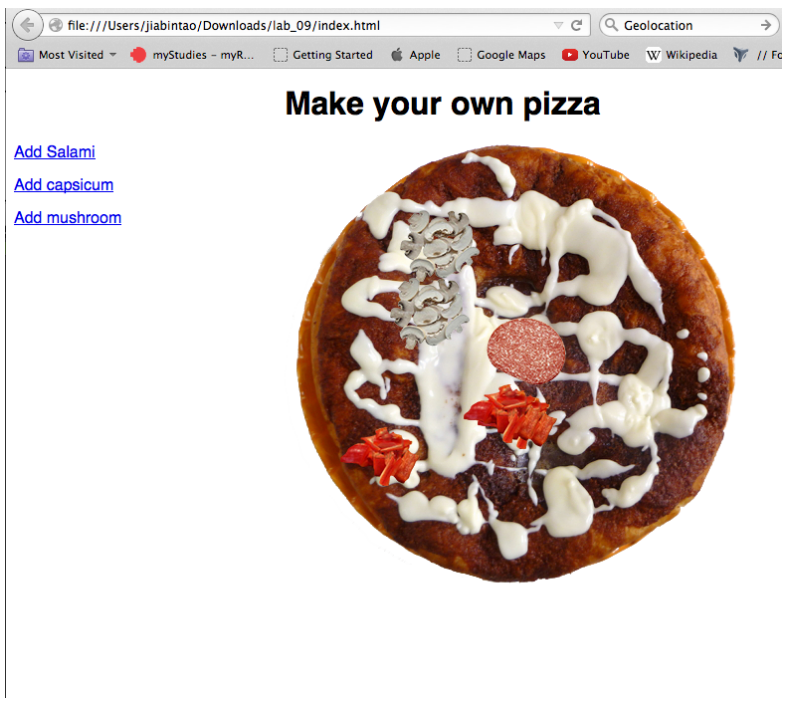
```
UIColor* fillColor = [UIColor colorWithRed: 1 green: 1 blue: 0 alpha: 1];
UIBezierPath* path = [UIBezierPath bezierPathWithOvalInRect: CGRectMake(2653, 654, 30, 30)];
[fillColor setFill];
[path fill];

UIColor* fillColor1 = [UIColor colorWithRed: 0.2 green: 0.2 blue: 0.2 alpha: 1];
UIBezierPath* path1 = [UIBezierPath bezierPath];
[path1 moveToPoint: CGPointMake(2664,666)];
[path1 addCurveToPoint: CGPointMake(2662,668) controlPoint1: CGPointMake(2664,667) controlPoint2: CGPointMake(2663,668)];
[path1 addCurveToPoint: CGPointMake(2660,666) controlPoint1: CGPointMake(2661,668) controlPoint2: CGPointMake(2660,667)];
[path1 addCurveToPoint: CGPointMake(2662,664) controlPoint1: CGPointMake(2660,665) controlPoint2: CGPointMake(2661,664)];
[path1 addCurveToPoint: CGPointMake(2664,666) controlPoint1: CGPointMake(2663,664) controlPoint2: CGPointMake(2664,665)];
[fillColor1 setFill];
[path1 fill];

UIColor* fillColor2 = [UIColor colorWithRed: 0.2 green: 0.2 blue: 0.2 alpha: 1];
UIBezierPath* path2 = [UIBezierPath bezierPath];
[path2 moveToPoint: CGPointMake(2677,666)];
[path2 addCurveToPoint: CGPointMake(2675,668) controlPoint1: CGPointMake(2677,667) controlPoint2: CGPointMake(2676,668)];
[path2 addCurveToPoint: CGPointMake(2673,666) controlPoint1: CGPointMake(2674,668) controlPoint2: CGPointMake(2673,667)];
[path2 addCurveToPoint: CGPointMake(2675,664) controlPoint1: CGPointMake(2673,665) controlPoint2: CGPointMake(2674,664)];
[path2 addCurveToPoint: CGPointMake(2677,666) controlPoint1: CGPointMake(2676,664) controlPoint2: CGPointMake(2677,665)];
[fillColor2 setFill];
[path2 fill];

UIColor* fillColor3 = [UIColor colorWithRed: 0.2 green: 0.2 blue: 0.2 alpha: 1];
UIBezierPath* path3 = [UIBezierPath bezierPath];
[path3 moveToPoint: CGPointMake(2660,675)];
[path3 addLineToPoint: CGPointMake(2676,675)];
[path3 addCurveToPoint: CGPointMake(2668,681) controlPoint1: CGPointMake(2676,675) controlPoint2: CGPointMake(2674,681)];
[path3 addCurveToPoint: CGPointMake(2660,675) controlPoint1: CGPointMake(2662,681) controlPoint2: CGPointMake(2661,677)];
[fillColor3 setFill];
[path3 fill];
```



Drag& Drop Code
Example from class making pizza ：

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js">
</script>

<title>Drag and Drop Pizzat</title>
<style>
#main{
        display:block;
        width:500px;
        margin:0 auto;
        position:relative;
        }
#pizza{
        display:block;
        width:454px;
        height:441px;
        background:url('images/pizza_base.png');
        position:absolute;
        }

#toppings{
                position:absolute;
                }

#toppings img{
                display:block;
                width:80px;
                height:80px;
        }
.draggable{
         cursor:pointer;
         position:absolute;
        }

</style>




</head>

<body>
        <div id="main">
           <h1> Make your own pizza</h1>
           <div id="pizza"></div>
           <div id="toppings"></div>
        </div>
        <p><a href="#" onclick="add('salami');return false">Add Salami</a></p>
         <p><a href="#" onclick="add('capsicum');return false">Add capsicum</a></p>
         <p><a href="#" onclick="add('mushroom');return false">Add mushroom</a></p>

<script type="text/javascript">
var startX, startY, dragTarget;
var image;

                function add(type){

                switch(type){
                case 'salami':
                image="images/salami.png";
   break;
                        case 'capsicum':
        image="images/capsicum.png";
   break;
                        case 'mushroom':
        image="images/mushroom.png";
   break;
}
                        //creat an object, to have a image inside, and content the topping image from div
                        var toppingitem=$('<img src="'+image+'" width="80" height="100" class="draggable">').css({
                                'left':Math.random()*300,
                                'top':Math.random()*300,
                                }).mousedown(function(e){
                                        startX=e.clientX;
                                        startY=e.clientY;
                                        dragTarget=$(this);
                                                //alert('TEST')
                                        document.onmousemove=startDrag;
                                        document.onmouseup=stopDrag;
                                        return false;
                                        });

                        $('#toppings').append(toppingitem);  //append means add some connents to
        }

        function startDrag(e){
                var diffX=e.clientX -startX;
                var diffY=e.clientY -startY;

                var left=parseInt(dragTarget.css('left'));
                var top=parseInt(dragTarget.css('top'));

                dragTarget.css({'left':diffX + left});
                dragTarget.css({'top':diffY + top});

                startX=e.clientX;
                startY=e.clientY;

                return false;

                }

        function stopDrag(){
                dragTarget=null;
                document.onmousemove=null;
                document.onmouseup=null;

                }

        //function startDrag(e){
                //var diffY=e.clientY -startY;
                //var up=parseInt(dragTarget.css('left'));
                //dragTarget.css({'up':diffY + up});
                //startY=e.clientY;
                //return false;
                //}



</script>

</body>
</html>
```





总体 作业 和判分要求：

DIG60011 Assignment 02: Interactive Greeting Card

Due: 5pm Friday Week 12
Weighting: 25% Final Grade

This assignment has two components:

1. Design and Construction of a Greeting Card mobile website that is as cross-device and cross- browser compatible as possible. It should demonstrate topics covered in the subject up to and including week 11.
2. A report listing the design choices made and providing a justification of those choices. In particular it should address responsive web design, progressive enhancement and usability.

The website will be graded according to two levels:

1. Meets all assignment criteria (max 85%)
2. Additional HTML5, CSS3, JavaScript or jQuery features not covered in the course (15%)

Carefully read the Marking Scheme at the end of this Assignment Specification.

1 Website (20% of final grade)

Design and construct an interactive greeting card in HTML5, CSS3 and JavaScript that is sufficiently cross-device and cross-browser compatible to be easily used on a range of platforms including smart phones, tablets, laptop / desktop computers. Your interactive card should:

- be for a birthday, Christmas, wedding…etc
- contain at least 3 different interactive elements (using the mouse or keyboard).
- Contain at least 1 animation, minimum 3 seconds long.
- Validate user input with JavaScript.

1.1 Evidence of individual research document

To get full marks you should provide evidence of individual research beyond content supplied in the textbook, study guides, lectures and labs. In addition, markers need to know where you have put features from supplied content. In a word document (see Submission Instructions, point 3) for each feature you have incorporated into the site:

a) that you have researched yourself , list :
   - The feature
   - A reference to the source of information
   - The file, page or location in your code where the feature can be found.

b) from the supplied content (in particular see Marking guide Sections 3 & 4) list:
   - The feature
   - The file, page or location in your code where the feature can be found.

总体 作业 和判分要求：

4 Website Marking Scheme

	Component			Marks
		Individual Criteria	**Mark**	**Total**
1	Structure			9
		File/folder naming	3	
		Folder structure	3	
		Submission structure	3	
2	Card Design			23
		HTML and CSS matches graphic design	3	
		Usability/Accessibility principles adhered to	3	
		Appropriate file type usage	3	
		At least 3 interactive elements	9	
		At least 1 animation >= 3 sec	3	
		Inputs are validated and display error messages	2	
2	Compatibility			12
		Most interactivity works across devices and browsers (Demonstrate progressive enhancement when feature not supported on device/browser).	6	
		Works on full-size, tablet and smartphone screens. Demonstrates responsive web design.	6	
3	HTML5			17-23
		Good use of semantic markup in structural elements	5	
		W3C validated	3	
		At least 3 of the following features: • The Canvas API • Session & Local storage • Drag and drop • Transitions • Dynamic HTML • Vector graphics • Geolocation • Social network API • Sound/Music playback Please list in the document of researched features. (Bonus 2 marks every extra feature, up to maximum of 6 marks)	9-15	
4	CSS3			10-13

总体 作业 和判分要求：

		Appropriate, working valid CSS	6	
		At least 2 of the following features: • Box shadow • Box-radius • Gradients • Transition • Pseudo selectors (nth-of-type, nth-child, :not…etc) • Media query Please list in the document of researched features. (Bonus 1 marks every extra feature, up to maximum of 3 marks)	4-7	
5	Forms			12
		At least 2 form elements	4	
		Validation via JavaScript	4	
		Demonstrate the onSubmit event (via JS or button)	4	
6	JavaScript / jQuery			11
		Demonstrates understanding of: • Event handling • DOM manipulation • Advanced variables, ie arrays • Functions with parameters/ return	7	
		Correct syntax	4	
7	Documentation			6
		25% Commenting in: • HTML • CSS • JavaScript	6	
	Total			**100**