

考核第三周

进度报告

完成登陆框，轮播图，服务项目三个部分

- 要求
 1. 轮播图（实现点击左右prev，next按钮会切换前一张和后一张图；点击下方圆点切换相应图片）
- 待改进：轮播图的大小没有自适应屏幕宽度，未将transition和js结合，视觉效果较差。
 2. 登陆框（实现点击弹出登陆框并屏蔽主页按钮）

学习

1. 接触js轮播图动画

```
<script type="text/javascript">
    /* 知识点: */
    /*      this用法 */
    /*      DOM事件 */
    /*      定时器 */

    window.onload = function() {
        var container = document.getElementById('container');
        var list = document.getElementById('list');
        var buttons = document.getElementById('buttons').getElementsByTagName('span');
        var prev = document.getElementById('prev');
        var next = document.getElementById('next');
        var index = 1;
        var timer;

        function animate(offset) {
            //获取的是style.left, 是相对左边获取距离, 所以第一张图后style.left都为负值,
            //且style.left获取的是字符串, 需要用parseInt()取整转化为数字。
            var newLeft = parseInt(list.style.left) + offset;
            list.style.left = newLeft + 'px';
            //无限滚动判断
            if (newLeft > -600) {
                list.style.left = -3000 + 'px';
            }
            if (newLeft < -3000) {
                list.style.left = -600 + 'px';
            }
        }

        function play() {
            //重复执行的定时器
            timer = setInterval(function() {
                next.onclick();
            }, 2000)
        }
    }
</script>
```

```

}

function stop() {
    clearInterval(timer);
}

function buttonsShow() {
    //将之前的小圆点的样式清除
    for (var i = 0; i < buttons.length; i++) {
        if (buttons[i].className == "on") {
            buttons[i].className = "";
        }
    }
    //数组从0开始, 故index需要-1
    buttons[index - 1].className = "on";
}

prev.onclick = function() {
    index -= 1;
    if (index < 1) {
        index = 5
    }
    buttonsShow();
    animate(600);
};

next.onclick = function() {
    //由于上边定时器的作用, index会一直递增下去, 我们只有5个小圆点, 所以需要做出判断
    index += 1;
    if (index > 5) {
        index = 1
    }
    animate(-600);
    buttonsShow();
};

for (var i = 0; i < buttons.length; i++) {
    (function(i) {
        buttons[i].onclick = function() {

            /* 这里获得鼠标移动到的小圆点的位置, 用this把index绑定到对象buttons[i]上, 去谷歌this的用法 */

            /* 由于这里的index是自定义属性, 需要用到getAttribute()这个DOM2级方法, 去获取自定义index的属性*/

            var clickIndex = parseInt(this.getAttribute('index'));
            var offset = 600 * (index - clickIndex); //这个index是当前图片停留时的index
            animate(offset);
            index = clickIndex; //存放鼠标点击后的位置, 用于小圆点的正常显示
            buttonsShow();
        }
    })(i)
}

container.onmouseover = stop;
container.onmouseout = play;
play();

```

```

}
</script>

```

2. 用js改变动画和过渡效果的触发事件，实现点击时开始播放动画

```
//animation绑定用该语句
elemen.classList.add('属性');
//transistion绑定只需先设置transition, 再添加js即可
```

3. 用html5 的canvas标签结合js绘图

```
//画一个3/4的圆环
var bulb1=document.getElementsByClassName("bulb1")[0];
var ctx=bulb1.getContext("2d");
ctx.beginPath();
ctx.arc(150,80,40,0.75*Math.PI,0.25*Math.PI);
ctx.lineWidth=8;
ctx.strokeStyle='rgb(244,152,0)';
ctx.stroke();
```

遇到的问题

1. 函数封装失败（不知道哪出问题了）

```
for(var i=0;i<labelBoxes.length;i++){
    labelBoxes[i].onmouseover=change(i);
    labelBoxes[i].onmouseout=reset(i);
}
function change(num){
    if(num<3){
        verticalLines[num].style.marginTop="0px";
    }
}
function reset(num){
    if(num<3){
        verticalLines[num].style.marginTop="-50px"
    }
}

---
function change(){
    verticalLines[0].style.marginTop="0px";
}
function reset(){
    verticalLines[0].style.marginTop="-50px";
}
labelBoxes[0].onmouseover=change;
labelBoxes[0].onmouseout=reset;
reset();
function change1(){
    verticalLines[1].style.marginTop="0px";
}
function reset1(){
    verticalLines[1].style.marginTop="-50px";
}
labelBoxes[1].onmouseover=change1;
```

```
        labelBoxes[1].onmouseout=reset1;
        reset1();
    function change2(){
        verticalLines[2].style.marginTop="0px";
    }
    function reset2(){
        verticalLines[2].style.marginTop="-50px";
    }
    labelBoxes[2].onmouseover=change2;
    labelBoxes[2].onmouseout=reset2;
    reset2();
}
```

2. 表单标签赋予margin属性没有效果

3. 无法做出能移动的登陆框

心得

- 布局
- 每次相对定位的开始位置是同一级的上次相对定位的底端
- 绝对定位是相对其最近的父级元素的顶部开始的
- 绝对定位的使用不会影响到原先布局
- 实现w3c盒子content部分居中，可以将width设为x%，将padding或者margin设为(100-x)%/2（此时盒子在水平方向上占满屏幕）
- 有时给一个类命赋予hover属性后，页面没有显示，可能是hover前的类命的权值小于之前的权值，故之后的属性被覆盖