

# 自动轮播图

2019年4月6日 21:27

```
<body>
<div class="box" id="box">
  <div class="inner">
    <!--轮播图-->
    <ul>
      <li><a href="#"></a></li>
      <li><a href="#"></a></li>
      <li><a href="#"></a></li>
      <li><a href="#"></a></li>
      <li><a href="#"></a></li>
    </ul>
    <ol class="bar">
      小按钮数量无法确定，由js动态生成
    </ol>
    <!--左右焦点-->
    <div id="arr">
      <span id="left"><img alt="left arrow" /></span>
      <span id="right"><img alt="right arrow" /></span>
    </div>
  </div>
</div>
</body>
```

```
<style>
* {
  margin: 0;
  padding: 0;
}
.box {
  width: 500px;
  height: 300px;
  border: 1px solid #ccc;
  margin: 100px auto;
  padding: 5px;
}
.inner {
  width: 500px;
  height: 300px;
  position: relative;
  overflow: hidden;
}
.inner img {
  width: 500px;
  height: 300px;
  vertical-align: top;
}
ul {
  width: 1000%;
  position: absolute;
  list-style: none;
  left: 0;
  top: 0;
}
.inner li {
  float: left;
}

ol {
  position: absolute;
  height: 20px;
  right: 20px;
  bottom: 20px;
  text-align: center;
  padding: 5px;
}
ol li {
  display: inline-block;
  width: 20px;
  height: 20px;
  line-height: 20px;
  background-color: #fff;
  margin: 5px;
  cursor: pointer;
}
ol .current {
  background-color: #ccc;
}
```

```

<script>
/**
 *
 * @param id 传入元素的id
 * @returns {HTMLElement | null} 返回标签对象，方便获取元素
 */
function my$(id) {
    return document.getElementById(id);
}

//获取各元素，方便操作
var box=my$("box");
var inner=box.children[0];
var ulObj=inner.children[0];
var list=ulObj.children;
var olObj=inner.children[1];
var arr=my$("arr");
var imgWidth=inner.offsetWidth;
var right=my$("right");
var pic=0;
//根据li个数，创建小按钮
for(var i=0;i<list.length;i++){
    var liObj=document.createElement("li");

    olObj.appendChild(liObj);
    liObj.innerText=(i+1);
    liObj.setAttribute("index",i);

    //为按钮注册mouseover事件
    liObj.onmouseover=function () {
        //先清除所有按钮的样式

        for (var j=0;j<olObj.children.length;j++){
            olObj.children[j].removeAttribute("class");
        }
        this.className="current";
        pic=this.getAttribute("index");
        animate(ulObj,-pic*imgWidth);
    }
}

//设置ol中第一个li有背景颜色
olObj.children[0].className = "current";
//克隆一个ul中第一个li,加入到ul中的最后=====克隆
ulObj.appendChild(ulObj.children[0].cloneNode(true));

var timeld=setInterval(onmouseclickHandle,1000);
//左右焦点实现点击切换图片功能
box.onmouseover=function () {
    arr.style.display="block";
    clearInterval(timeld);
};
box.onmouseout=function () {

```

```
}
ol .current{
    background-color: red;
}
#arr{
    display: none;
}
#arr span{
    width: 40px;
    height: 40px;
    position: absolute;
    left: 5px;
    top: 50%;
    margin-top: -20px;
    background: #fff;
    cursor: pointer;
    line-height: 40px;
    text-align: center;
    font-weight: bold;
    font-family: '黑体';
    font-size: 30px;
    color: #000;
    opacity: 0.5;
    border: 1px solid #fff;
}
#arr #right {
    right: 5px;
    left: auto;
}
```

```

clearInterval(timerId),
};
box.onmouseout=function () {
    arr.style.display="none";
    timerId=setInterval(onmouseclickHandle,1000);
};

right.onclick=onmouseclickHandle;
function onmouseclickHandle() {
    //如果pic的值是5,恰巧是ul中li的个数-1的值,此时页面显示第六
    个图片,而用户会认为这是第一个图,
    //所以,如果用户再次点击按钮,用户应该看到第二个图片
    if (pic == list.length - 1) {
        //如何从第6个图,跳转到第一个图
        pic = 0;//先设置pic=0
        ulObj.style.left = 0 + "px";//把ul的位置还原成开始的默认位置
    }
    pic++;//立刻设置pic加1,那么此时用户就会看到第二个图片了
    animate(ulObj, -pic * imgWidth);//pic从0的值加1之后,pic的值是
    1,然后ul移动出去一个图片
    //如果pic==5说明,此时显示第6个图(内容是第一张图片),第一个
    小按钮有颜色,
    if (pic == list.length - 1) {
        //第五个按钮颜色干掉
        olObj.children[olObj.children.length - 1].className = "";
        //第一个按钮颜色设置上
        olObj.children[0].className = "current";
    } else {
        //干掉所有的小按钮的背景颜色
        for (var i = 0; i < olObj.children.length; i++) {
            olObj.children[i].removeAttribute("class");
        }
        olObj.children[pic].className = "current";
    }
}
left.onclick=function () {
    if (pic==0){
        pic=list.length-1;
        ulObj.style.left=-pic*imgWidth+"px";
    }
    pic--;
    animate(ulObj,-pic*imgWidth);
    for (var i = 0; i < olObj.children.length; i++) {
        olObj.children[i].removeAttribute("class");
    }
    //当前的pic索引对应的按钮设置颜色
    olObj.children[pic].className = "current";
};

//设置任意的一个元素,移动到指定的目标位置
function animate(element, target) {
    clearInterval(element.timerId);
    //定时器的id值存储到对象的一个属性中
    element.timerId = setInterval(function () {
        //获取元素的当前的位置,数字类型
        var current = element.offsetLeft;

```



```
//每次移动的距离
var step = 10;
step = current < target ? step : -step;
//当前移动到位置
current += step;
if (Math.abs(current - target) > Math.abs(step)) {
    element.style.left = current + "px";
} else {
    //清理定时器
    clearInterval(element.timerId);
    //直接到达目标
    element.style.left = target + "px";
}
}, 10);
}
</script>
```