

第 6 章 JavaScript 与 CSS 交互

主讲师：黄玲





学习目标



· 掌握 `style` 属性和 `className` 属性的应用



· 掌握 `visibility` 属性和 `display` 属性的应用



· 掌握 `scrollTop` 属性的应用

本章任务



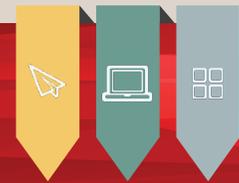
 任务 1 : 使用 JavaScript 动态改变导航菜单样式

任务 2 : 制作图片轮显效果

任务 3 : 制作随鼠标滚动的广告图片



任务 1：使用 JavaScript 动态改变导航菜单样式



任务描述：



1. 制作贵美网站的导航部分，使用 style 属性或 className 属性动态改变菜单样式，要求如下。
 2. 默认情况下，“首页”、“家用电器”等文本的背景图片为 bg1.jpg，字体大小为 14px，加粗显示，字体颜色为白色。
 3. 当鼠标指针移到“首页”等文本区域上时，背景图片为 bg2.jpg，字体颜色为黑色，字体大小为 15px，加粗显示。
 4. 当鼠标指针移出文本区域时，文本显示样式与默认情况下相同。

JavaScript 访问样式的常用方法



如何动态改变页面元素的样式？

1. 使用 getElement 系列方法访问元素
2. 改变样式属性：
 - ① Style 属性
 - ② className 属性



Style 属性



 HTML 元素 `.style`. 样式属性 = " 值 "

```
document.getElementById("titles").style.color="#ff0000";  
document.getElementById("titles").style.font-size="25p";
```



style 对象的属性



· style 对象的常用属性

类别	属性	描述
Padding (边距)	padding	设置元素的填充
	paddingTop	设置元素的上、下、左、右填充
	paddingBottom	
	paddingLeft	
	paddingRight	
Border (边框)	border	设置四个边框所有的属性
	borderTop	设置上、下、左、右边框的属性
	borderBottom	
	borderLeft	
	borderRight	



使用 style 属性制作菜单



 如何制作当鼠标移动菜单上时，菜单背景变化？



- 1、鼠标移到菜单上时改变菜单样式
- 2、鼠标移出菜单时恢复为原来的样式



常用事件



· JavaScript 中常用的事件列表

名称	描述
onclick	当用户单击某个对象时调用事件
onmouseover	鼠标移到某元素之上
onmouseout	鼠标从某元素移开
onmousedown	鼠标按钮被按下



使用 style 属性改变样式



使用 style 属性改变样式

this , 通常指向正在执行函数本身或指向函数所属的对象
，对于 onmouseover 事件，它指当前的 HTML 元素

```
.....  
<ul>  
  <li onmouseover="this.style.backgroundImage='url(images/bg2.gif)'"  
  onmouseout="this.style.backgroundImage='url(images/bg1.gif)'"> 资讯动态  
</li>  
  <li onmouseover="this.style.backgroundImage='url(images/bg2.gif)'"  
  onmouseout="this.style.backgroundImage='url(images/bg1.gif)'"> 产品世界  
</li>  
  <li onmouseover="this.style.backgroundImage='url(images/bg2.gif)'"  
  onmouseout="this.style.backgroundImage='url(images/bg1.gif)'"> 市场营销  
</li>  
</ul>
```

是否可以对这些相似的重复代码进行简化？





```
<script type="text/javascript">
function over(num) {
    var overB=document.getElementsByTagName("li");
    overB[num].style.backgroundImage="url(images/bg2.gif)";
}
function out(num) {
    var overB=document.getElementsByTagName("li");
    overB[num].style.backgroundImage="url(images/bg1.gif)";
}
</script>
</head>

<body>
<ul>
<li onmouseover="over(0)" onmouseout="out(0)">资讯动态</li>
<li onmouseover="over(1)" onmouseout="out(1)">产品世界</li>
<li onmouseover="over(2)" onmouseout="out(2)">市场营销</li>
</ul>
```



匿名函数



- 匿名函数定义：没有函数名的函数

```
function(){  
    this.style.backgroundImage="url(images/bg2.gif)";  
}
```

- 匿名函数调用：匿名函数定义的整个语句可以像值一样赋给一个变量进行保存，也可以将它赋给一个事件触发。

```
XXX.onmouseover=function(){  
this.style.backgroundImage="url(images/bg2.gif)";  
}
```



完善菜单特效



· 思路分析

- 设置项目列表的初始状态。
- 使用 `document.getElementsByTagName("li")` 获取所有的 `` 标签。
- 在每一个 `` 标签设置事件，使用 “事件名 = function(){...}” 的方式

```
var objli=document.getElementsByTagName("li");
for(var i=0;i< objli.length;i++){
    objli [i].onmouseover=function(){
        this.style.backgroundImage="url(images/bg2.gif)";
    }
    objli [i].onmouseout=function(){
        this.style.backgroundImage="url(images/bg1.gif)";
    }
}
```



className 属性



 HTML 元素 .className=" 类样式名 "

使用 className 属性修改菜单特效的例子

实现思路：

1. 标签的初始状态不改变，设置两个类样式 over 和 out ，

分别表示鼠标移至菜单上和移出菜单的效果

2. 在 标签中添加上 onmouseover 事件和 onmouseout



className 属性



 over 和 out 分别为鼠标移到菜单上和移出菜单时的菜单样式

```
<ul>  
<li onmouseover="this.className='over'"  
onmouseout="this.className='out'"> 资讯动态 </li>  
<li onmouseover="this.className='over'"  
onmouseout="this.className='out'"> 产品世界 </li>  
<li onmouseover="this.className='over'"  
onmouseout="this.className='out'"> 市场营销 </li>  
</ul>
```



使用函数改变菜单特效



· 实现思路

- 设置每一个 li 标签的初始状态
- 设置两个样式 over 和 out ，表示鼠标移至菜单和移出菜单的效果
- 统一为每一个 li 标签设置 onmouseover 事件和 onmouseout 事件效果

```
var  
len=document.getElementsByTagName("li");  
for (var i=0;i<len.length;i++) {  
    len[i].onmouseover=function() {  
        this.className="over";  
    }  
    len[i].onmouseout=function() {  
        this.className="out";  
    }  
}
```



任务实现



```
var len=document.getElementsByTagName("li");
for(var i=0;i<len.length;i++){
    len[i].onmouseover=function(){
this.style.backgroundImage="url(images/bg2.jpg)";
    this.style.color="#000000";
    this.style.fontSize="15px";  }
    len[i].onmouseout=function(){
this.style.backgroundImage="url(images/bg1.jpg)";
    this.style.color="#ffffff";
    this.style.fontSize="14px";
}}
```



练一练



- 利用 style 属性和 className 属性实现
- 动态改变导航菜单样式



任务 2：制作图片轮显效果



任务描述：

带按钮的轮换显示的横幅广告，五张广告图片循环显示，并且下面的标题随图片显示不同的内容，单击数字按钮时可以即时显示对应的图片和相应的标题。





实现思路



1. 页面加载时，使用定时器将图片自动播放，即当前片显示时，其他图片使用 `display` 属性设置隐藏
2. 当图片播放到相应图片时，相应数字添加 `numberOver` 样式，其他图片相应数字添加 `number` 样式。
3. 用户单击数字播放相应图片。



相关知识



如何实现如图所示的页面效果？



图书周排行榜 TOP 100

近7天销量，每日更新

小说 非小说 少儿

1. 时间旅行者的妻子
2. 女心理师（下）
3. 鬼吹灯之精绝古城
4. 女心理师（上）
5. 小时候
6. 追风筝的人
7. 盗墓笔记2
8. 输赢



设置图片、层等元素的显示和隐藏



visibility



· visibility 属性的值

值	描述
visible	表示元素是可见的
hidden	表示元素是不可见的



`object.style.visibility=" 值 "`



display



· display 属性的值

值	描述
none	表示此元素不会被显示
block	表示此元素将显示为块级元素， 此元素前后会带有换行符
inline	表示此元素将显示为行级元素



`object.style.display`
= " 值 "



任务实现



1. 实现加载时图片自动播放

```
.....  
var title=new Array();  
title[0]="1. 店庆第一波 限时抢购 一日三疯!";  
title[1]="2. 十年店庆均价场 39/99/169专场!";  
title[2]="3. 全场69折封顶 享当当的超值低价!";  
var n=0;title[3]="4. 店庆钜献 海量图书69折封顶";  
var m=0;title[4]="5. 十年店庆 名家跨界祝贺 珍藏签名本专区";  
function showPic(){  
var now=1; //当前显示的图片  
var maxp=5;  
function showPic(){  
for(var i=1;i<maxp+1;i++){  
if(i==now){  
document.getElementById("Rotator"+i).style.display="block";  
document.getElementById("fig_"+i).innerHTML=title[i-1];  
document.getElementById("fig_"+i).className="numberOver";  
}  
else{  
document.getElementById("Rotator"+i).style.display="none";  
document.getElementById("fig_"+i).innerHTML=i;  
document.getElementById("fig_"+i).className="number";  
}  
if(now<5){  
now++;  
}  
else{  
now=1;  
}  
var timer=setTimeout("showPic()",2000);  
}  
window.onload=showPic; //页面加载时运行函数show()*  
.....
```

le.di

HTML=

me=



练一练：



根据任务 2 提供的素材实现页面加载时每隔 4

秒自动播放图片。





2. 如何实现单击数字时播放相应图片？

```
.....  
    function show(num)  
    {  
        if(Number(num))  
        {  
            nowFrame=num; // 通过参数改变当前播放的图片  
        }  
    }  
.....
```



任务实现



3. 当多次单击数字时，为什么播放速度会加快，如何解决？

```
.....  
function show(num)  
{  
  if(Number(num))  
  {  
    nowFrame=num; // 通过参数改变当前播放的图片  
    clearTimeout(timer); // 清除原来创建的定时器  
  }  
}  
.....
```



练一练：



实现任务 2 图片轮显效果，页面加载时图片自动播放，单击数字时可以播放相对应的图片。



任务 3：制作随鼠标滚动的广告图片



任务描述：

1. 当滚动条向下或向右移动时，图片和关闭按钮随滚动条移动，相对于浏览器的位置固定。
2. 单击关闭按钮，页面中的图片和关闭按钮不显示。



实现思路



1. 把广告图片放在一个 div 中，并且 div 总是显示在页面的上方
2. 使用 `getElementById()` 方法获取层对象，并且获取层在页面上的初始位置
3. 根据鼠标滚动事件，获取滚动条滚动的距离
4. 随着滚动条的移动改变层在页面上的位置





如何实现广告图片总是显示在页面上方，并且随滚动条同步移动？

1. 使用 JavaScript 获取层的 top 和 left 样式值
2. 使用 JavaScript 获取滚动条滚动的距离 scrollTop 值
3. 同步移动：设置层的 top 值 = 初始值 + 滚动距离



获取样式属性值



- 获取行内样式的方法

`document.getElementById(elementId). 样式属性值`

```
var divObj=document.getElementById("test");  
var objTop=divObj.style.top;
```

获取类样式的方法

`currentStyle` (IE、Opera)

`getComputedStyle()` (Firefox、chrome、safari)

```
var divObj=document.getElementById("test");  
var objTop= divObj.currentStyle.top;  
var objTop =document.defaultView.getComputedStyle(divObj,null).top;
```



制作随鼠标滚动的广告图片



使用 `currentStyle` 或 `getComputedStyle()` 可以获得层在网页中的位置，但是如何获取滚动条滚动的距离呢？

`scrollTop`、`scrollLeft` 属性

属性	描述
<code>scrollTop</code>	设置或获取位于对象最顶端和窗口中可见内容的最顶端之间的距离
<code>scrollLeft</code>	设置或获取位于对象左边界和窗口中目前可见内容的最左端之间的距离





获取滚动条在窗口中滚动的距离



scrollTop 属性有浏览器兼容性问题，在不同浏览器使用不同方法获取：

IE、FireFox 浏览器

`document.documentElement.scrollTop`

谷歌浏览器

`document.body.scrollTop`



任务实现



```
function inix(){
    adverObject=document.getElementById("adver");// 获得层对象
    if(adverObject.currentStyle){
        adverTop=parseInt(adverObject.currentStyle.top);
        adverLeft=parseInt(adverObject.currentStyle.left);
    }
    else{
        adverTop=parseInt(document.defaultView.getComputedStyle(adverObject,null).top);
        adverLeft=parseInt(document.defaultView.getComputedStyle(adverObject,null).left);
    }
}
```

任务实现



实现广告层关闭功能

```
function closeAdv()  
{  
  
document.getElementById("adver").style.display="none";  
  
}
```



总结



使用 JavaScript 改变样式的两种方法

使用 style 属性

使用 className 属性

获取对象样式属性值

使用 style 属性对象获取内联样式属性值

使用 currentStyle 对象和

getComputedStyle 方法获取样式中的属性值



谢谢观看

