

第3章 案例（原生态）

课程提要

- 原生JS返回顶部效果
- 原生JS折叠菜单效果

3.1 原生JS返回顶部效果

3.1.1 需求分析

- 创建一个有滚动条的界面；
- 当界面滚动高度大于可视区域高度时，显示“回到顶部”按钮，否则隐藏；
- 点击“回到顶部”按钮，页面滚动至顶部，回到顶部按钮隐藏。

3.1.2 效果图



3.1.3 制作过程

3.1.3.1 HTML结构

重点：

- 创建多个img标签，使页面出现滚动条；
- 创建a链接标签，id名为“btn”，作为“回到顶部”按钮。

代码：

```

<a href="javascript:;" id="btn"></a>


















```

3.1.3.2 CSS样式

重点：

- 设置#btn按钮长宽样式，设置背景图片为images/huojian1.jpg；
- 设置按钮绝对定位于页面右下角，不随页面滚动而滚动。

代码：

```

#btn{
    position: fixed;
    right: 30px;
    bottom: 10px;
    width: 99px;
    height: 175px;
    background: url(images/huojian1.jpg);
    display: none;
}

```

3.1.3.3 JS脚本分析

重点：

Window绑定scroll滚动事件：

- 当滚动高度大于可视区域高度时，则“回到顶部”按钮显示；
- 当滚动高度小于可视区域高度时，“回到顶部”按钮隐藏；
- 判断当点击回到顶部按钮后滚动条在回滚过程中，若手动滚动滚动条，则清除定时器。

“回到顶部”按钮绑定点击事件：

- 创建定时器用于设置滚动速度差；
- “回到顶部”按钮背景切换，页面滚动；
- 当页面滚动到顶部时，“回到顶部”按钮隐藏。

代码：

1、获取#btn回到顶部按钮元素

```
var btn = document.getElementById("btn");
```

2、获取可视区域高度

```
var clientHeight = document.documentElement.clientHeight;
```

3、为window窗口绑定滚动事件

- 获取滚动条的滚动高度；
- 如果滚动高度大于可视区域高度，则显示回到顶部按钮否则隐藏。

```
window.onscroll = function(){  
    //获取滚动条的滚动高度  
    var scrollTop = document.documentElement.scrollTop || document.body.scrollTop;  
    //如果滚动高度大于可视区域高度，则显示回到顶部按钮否则隐藏  
    if(scrollTop >= clientHeight){  
        btn.style.display = "block";  
    }else{  
        btn.style.display = "none";  
    }  
}
```

4、为btn绑定点击事件

- 定义全局变量timer用于定时器，在btn的点击事件中，创建setInterval定时器，用于设置滚动速度差，产生缓动的效果；
- 获取滚动条的滚动高度；
- 切换btn按钮的背景图片为“images/huojian2.jpg”。

```
var timer = null;    //定义一个定时器  
btn.onclick = function(){  
    timer = setInterval( function(){  
        //获取滚动条的滚动高度  
        var scrollTop = document.documentElement.scrollTop || document.body.scrollTop;  
        btn.style.backgroundImage = "url(images/huojian2.jpg)";  
        //用于设置速度差，产生缓动的效果  
        var speed = Math.floor(-scrollTop / 6);  
        document.documentElement.scrollTop = document.body.scrollTop = scrollTop +  
speed;  
    }, 50 );  
}
```

5、定义全局变量isTop=true用于判断是否到达顶部：为true则到达顶部，为false则未到达顶部。

在window的scroll滚动事件中判断当点击回到顶部按钮后滚动条在回滚过程中，若手动滚动滚动条，则清除定时器，且btn的背景图片设置为“images/huojian1.jpg”。且设置isTop = false;用于清除定时器。

```
var isTop = true;    //定义一个布尔值，用于判断是否到达顶部
//判断当点击回到顶部按钮后滚动条在回滚过程中，若手动滚动滚动条，则清除定时器
if(!isTop){
    btn.style.backgroundImage = "url(images/huojian1.jpg)";
    clearInterval(timer);
}
isTop = false;
```

6、在btn的点击事件中，设置isTop =true用于阻止滚动事件清除定时器。

```
isTop =true;    //用于阻止滚动事件清除定时器
```

7、在btn的点击事件中，判断滚动高度是否为0（即是否到达顶部），是则btn的背景图片设置为“images/huojian1.jpg”，清除定时器。

```
if(scrollTop == 0){
    btn.style.backgroundImage = "url(images/huojian1.jpg)";
    clearInterval(timer);
}
```

3.1.3.4 JS脚本整合

```

window.onload = function(){
    var btn = document.getElementById("btn");
    //获取可视区域的高度
    var clientHeight = document.documentElement.clientHeight;
    var timer = null;    //定义一个定时器
    var isTop = true;    //定义一个布尔值，用于判断是否到达顶部

    window.onscroll = function(){
        //获取滚动条的滚动高度
        var scrollTop = document.documentElement.scrollTop || document.body.scrollTop;

        //如果滚动高度大于可视区域高度，则显示回到顶部按钮否则隐藏
        if(scrollTop >= clientHeight){
            btn.style.display = "block";
        }else{
            btn.style.display = "none";
        }
        //判断当点击回到顶部按钮后滚动条在回滚过程中，若手动滚动滚动条，则清除定时器
        if(!isTop){
            btn.style.backgroundImage = "url(images/huojian1.jpg)";
            clearInterval(timer);
        }
        isTop = false;
    }
    //回到顶部按钮点击事件
    btn.onclick = function(){
        timer = setInterval( function(){

            //获取滚动条的滚动高度
            var scrollTop = document.documentElement.scrollTop || document.body.scrollTop;

            //用于设置速度差，产生缓动的效果
            var speed = Math.floor(-scrollTop / 6);
            document.documentElement.scrollTop = document.body.scrollTop = scrollTop + speed;
            btn.style.backgroundImage = "url(images/huojian2.jpg)";
            isTop = true;    //用于阻止滚动事件清除定时器
            if(scrollTop == 0){
                btn.style.backgroundImage = "url(images/huojian1.jpg)";
                clearInterval(timer);
            }
        },50 );
    }
}

```

3.2 原生JS折叠菜单效果

3.2.1 需求分析

- 创建一个垂直菜单，默认显示第一个子菜单下的内容；
- 当点击子菜单时，该子菜单下的内容下拉显示，其他子菜单下的内容隐藏。

3.2.2 效果图

HTML
超文本标记语言，标准通用标记语言下的一个应用。“超文本”就是指页面内可以包含图片、链接，甚至音乐、程序等非文字元素。超文本标记语言的结构包括“头”部分、和“主体”部分，其中“头”部提供关于网页的信息，“主体”部分提供网页的具体内容。
CSS
JS
JQ

3.2.3 制作过程

3.2.3.1 HTML结构

重点：

- 创建一个ul无序列表，在ul列表下创建四个li子列表；
- 在每个li子列表下嵌套一个h3标签元素，用于子菜单的标题；
- 在每个li子列表下嵌套一个div标签元素，class类名为text；
- 在div.text标签中嵌套一个p标签元素，用于存储子菜单中的内容。

代码：

```

<ul>
  <li>
    <h3>HTML</h3>
    <div class="text">
      <p>
        超文本标记语言，标准通用标记语言下的一个应用。“超文本”就是指页面内
        可以包含图片、链接，甚至音乐、程序等非文字元素。超文本标记语言的结构
        包括“头”部分、和“主体”部分，其中“头”部提供关于网页的信息，“主体”
        部分提供网页的具体内容。
      </p>
    </div>
  </li>
  <li>
  </li>
  <li>
  </li>
  <li>
  </li>
  <li>
  </li>
</ul>

```

3.2.3.2 CSS样式

重点：

- 设置所有元素box-sizing为border-box；
- UI无序列表设置width为300px，height为400px，box-shadow为2px 2px 5px #333333；
- 子菜单标题h3设置height为50px，字体垂直水平居中，设置cursor为pointer；
- 设置所有的div.text标签元素height为0px，溢出隐藏；
- 设置第一个li子菜单下的div.text的height为200px；
- 设置div.text盒子中的p段落内边距为5px，首行缩进。

代码：


```

*{
    padding: 0;
    margin: 0;
    list-style: none;
    box-sizing: border-box;
}
ul{
    width: 300px;
    height: 400px;
    border: 1px solid #ccc;
    margin: 50px auto;
    -moz-box-shadow: 2px 2px 5px #333333;
    -webkit-box-shadow: 2px 2px 5px #333333;
    box-shadow: 2px 2px 5px #333333;
}
ul li>h3{
    height: 50px;
    text-align: center;
    line-height: 50px;
    border-bottom: 1px solid #ccc;
    cursor: pointer;
}
.text{
    height: 0px;
    background-color: #ccc;
    overflow: hidden; /*溢出隐藏*/
}
/*设置第一个li子菜单下的div.text的height为200px*/
ul li:first-of-type>.text{
    height: 200px;
}
/*p段落内边距为5px，首行缩进*/
.text p{
    padding: 5px;
    text-indent: 2em;
    line-height: 25px;
}

```

3.2.3.3 JS脚本分析

重点：

遍历所有的h3：

- 当 $i \neq 0$ 时，设置 `div.text` 高度为 `0px`；
- 把 `i` 值赋值给 `h3` 的 `index`；
- 给每一个 `h3` 绑定鼠标点击事件，遍历所有的 `div.text`，被点击的 `div.text` 运用封装的运动函数高度变为 `200px`，其他 `.text` 高度设置为 `0px`。

封装一个运动动画：

- 定义一个定时器，每 `20ms` 调用一次计算表达式；
- 当传入的对象的高小于 `200px`，让当前高加上 `speed` 值，每执行一次定时器，`speed+10`；
- 当传入对象的高大于等于 `200px` 时，清除定时器。

代码：

获取 `document` 中的所有 `h3`，获取 `document` 中的所有 `.text`；

```
var $h3=document.getElementsByTagName('h3'); // 获取document中的所有h3
var $text=document.getElementsByClassName('text'); // 获取document中的所有.text
```

遍历 `h3` 数组：

当 $i \neq 0$ ，设置 `div.text` 高度为 `0px`；（即默认第一个 `div.text` 的高度为 `200px`，其他 `div.text` 高度为 `0px`）

```
//遍历h3数组
for(var i=0;i<$h3.length;i++){
    if(i!=0){
        $text[i].style.height='0px';
    }
}
```

循环把 `i` 的值赋值给 `h3` 的 `index`；

```
//循环把i的值赋值给h3的index
$h3[i].index=i;
```

给每一个 `h3` 绑定鼠标点击事件，遍历所有的 `div.text`，被点击的 `div.text` 运用封装的运动函数高度变为 `200px`，其他 `.text` 高度设置为 `0px`。

```
//给每一个h3注册鼠标点击事件
$h3[i].addEventListener("click",function(){
    for(var j=0;j<$text.length;j++){
        //除了被点击的h3下的div.text，其他.text高度为0
        if(this.index != j){
            $text[j].style.height='0px';
        }
    }
    ani($text[this.index]); //使用ani方法，把鼠标点击的h3下的div.
    text高度变为200px
},false);
```

封装一个运动动画函数，传入一个节点元素作为参数：

定义一个定时器。

```
//定义一个定时器  
var timer= setInterval(function(){  
  
},20);
```

如果传入的对象的高小于200px，让当前高加speed，每执行一次定时器，speed+10；当传入对象的高大于等于200px时，清除定时器。

```
var speed = 10;  
//如果传入的对象的高小于200px  
if(parseInt(obj.style.height) < 200){  
    //让当前高加speed  
    obj.style.height = parseInt(obj.style.height) + speed + "  
        px";  
    speed += 10; //每执行一次定时器，speed+10  
}else{  
    clearInterval(timer); //清除定时器  
}
```

3.2.3.4 JS脚本整合

```

window.onload=function(){
    var $h3=document.getElementsByTagName('h3');    //获取document中的所有h3
    var $text=document.getElementsByClassName('text');    //获取document中的所有.text
    /*animate动画*/
    function ani(obj){
        //定义一个定时器
        var timer= setInterval(function(){
            var speed = 10;
            //如果传入的对象的高小于200px
            if(parseInt(obj.style.height) < 200){
                //让当前高加speed
                obj.style.height = parseInt(obj.style.height) + speed + "px";
                speed += 10; //每执行一次定时器，speed+10
            }else{
                clearInterval(timer);    //清除定时器
            }
        },20);
    }
    //遍历h3数组
    for(var i=0;i<$h3.length;i++){
        if(i!=0){
            $text[i].style.height='0px';
        }
        //循环把i的值赋值给h3的index
        $h3[i].index=i;
        //给每一个h3注册鼠标点击事件
        $h3[i].addEventListener("click",function(){
            for(var j=0;j<$text.length;j++){
                //除了被点击的h3下的div.text，其他.text高度为0
                if(this.index != j){
                    $text[j].style.height='0px';
                }
            }
            ani($text[this.index]);    //使用ani方法，把鼠标点击的h3下的div.text高度变为200px
        },false);
    }
}

```

3.3 课程总结

- 原生JS返回顶部效果
- 原生JS折叠菜单效果

3.4 实训

- 1、完成上面两个案例。
- 2、实现选择水果效果。需求如下：

- 创建左右两个select多选列表框和四个按钮；
- 点击第一个按钮“>”，左侧被选中的select框中的选项移动到右侧的select框中；
- 点击第二个按钮“>>”，左侧select框中的所有选项移动到右侧的select框中；
- 点击第三个按钮“<”，右侧被选中的select框中的选项移动到左侧的select框中；
- 点击第四个按钮“<<”，右侧select框中的所有选项移动到左侧的select框中。

效果图如下：

