# dom学习笔记1,参考: [coderwayDOM学习笔记1](coderwhy-jsdom学习笔记1.docx)

# dom学习笔记2,参考: [coderwayDOM学习笔记2](coderwhy-jsdom学习笔记2.docx)

# 20.DOM操作元素的大小-位置-滚动p151

## 元素可以有很大信息,如图

|  |
| --- |
|  |

## 课堂案例

|  |
| --- |
|  |

# 21,window获取创建的大小位置滚动p152

|  |
| --- |
|  |

## 课堂案例1.按下按钮,窗口往下滚动

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>      <meta charset="UTF-8">      <meta name="viewport" content="width=device-width, initial-scale=1.0">      <title>window获取创建的大小-滚动-方法</title>      <style>          .box{              width: 100px;              height: 70px;              background-color: deeppink;              position: fixed;              text-align: center;              line-height: 70px;          }      </style>  </head>  <body>        <div class="box">           <button class="btn">滚动窗口</button>      </div>          <!-- 下面的br只是为了让浏览器窗口程序滚动条,方便我们的学习 -->      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <script>          //outerWidth 不常用          //outerHeight          console.log(window.outerWidth);          console.log(window.outerHeight);          //innerWidth 比较常用          //innerHeight          console.log(window.innerWidth);          console.log(window.innerHeight);          console.log("=======================");            //获取html元素的offsetWidth和offsetHeight          console.log(document.documentElement.offsetWidth);          console.log(document.documentElement.offsetHeight);          //获取window的滚动大小,这个还是比较常用的          //window.scrollX == window.pageXOffset          //window.scrollY == window.pageYOffset          window.onclick = function () {              console.log(window.scrollX);              console.log(window.scrollY);          }          //监听window的滚动事件          window.onscroll = function (){          //    console.log(window.scrollY);            }          //小案例,实现点击一下按钮,窗口就往下滚动100px的效果          let btn = document.querySelector("button")          let box = document.querySelector(".box")          let moveVal = 0          btn.onclick = function () {              window.scrollBy(0,100) //滚动是滚动了,但是如何让按钮同步下降呢,注意这里有一个单位问题              // moveVal +=200 //先在这里改变数值,然后在下面设置              // box.style.marginTop =moveVal+ "px" //需要加上px,否则不起作用              // console.log(box.style.marginTop);              //当然,也可以用css把按钮固定,参考上面的css代码          }          </script>  </body>  </html> |

## 课堂案例2.添加一个返回按钮,点击按钮,窗口会自动滚动到最上面

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>      <meta charset="UTF-8">      <meta name="viewport" content="width=device-width, initial-scale=1.0">      <title>window获取创建的大小-滚动-方法</title>      <style>          .box{              width: 100px;              height: 70px;              background-color: deeppink;              position: fixed;              text-align: center;              line-height: 70px;          }          .back{              position: fixed;              right:20px;              bottom:20px;          }      </style>  </head>  <body>        <div class="box">           <button class="btn">滚动窗口</button>      </div>        <button class="back">回到顶部</button>      <!-- 下面的br只是为了让浏览器窗口程序滚动条,方便我们的学习 -->      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <script>          //outerWidth 不常用          //outerHeight          console.log(window.outerWidth);          console.log(window.outerHeight);          //innerWidth 比较常用          //innerHeight          console.log(window.innerWidth);          console.log(window.innerHeight);          console.log("=======================");            //获取html元素的offsetWidth和offsetHeight          console.log(document.documentElement.offsetWidth);          console.log(document.documentElement.offsetHeight);          //获取window的滚动大小,这个还是比较常用的          //window.scrollX == window.pageXOffset          //window.scrollY == window.pageYOffset          window.onclick = function () {              console.log(window.scrollX);              console.log(window.scrollY);          }          //监听window的滚动事件          window.onscroll = function (){          //    console.log(window.scrollY);            }          //小案例,实现点击一下按钮,窗口就往下滚动100px的效果          let btn = document.querySelector("button")          let box = document.querySelector(".box")          let moveVal = 0          btn.onclick = function () {              window.scrollBy(0,100) //滚动是滚动了,但是如何让按钮同步下降呢,注意这里有一个单位问题              // moveVal +=200 //先在这里改变数值,然后在下面设置              // box.style.marginTop =moveVal+ "px" //需要加上px,否则不起作用              // console.log(box.style.marginTop);              //当然,也可以用css把按钮固定,参考上面的css代码          }            let btnBack = document.querySelector(".back")          btnBack.onclick = function(){              window.scrollTo(0,0)          }      </script>  </body>  </html> |

## 课堂案例3,在课堂案例2.的基础上面改进,就说如果窗口没有滚动,下面的按钮不显示,只要一滚动,返回按钮立马出现,这个需要通过css的display属性来实现,我们可以先把它的display设置为none,然后在滚动按钮的点击事件里面把它设置为display:inline; 然后在返回按钮的点击事件里面把它又设置为none

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>      <meta charset="UTF-8">      <meta name="viewport" content="width=device-width, initial-scale=1.0">      <title>window获取创建的大小-滚动-方法</title>      <style>          .box{              width: 100px;              height: 70px;              background-color: deeppink;              position: fixed;              text-align: center;              line-height: 70px;          }          .back{              position: fixed;              right:20px;              bottom:20px;              display: none;          }      </style>  </head>  <body>        <div class="box">           <button class="btn">滚动窗口</button>      </div>        <button class="back">回到顶部</button>      <!-- 下面的br只是为了让浏览器窗口程序滚动条,方便我们的学习 -->      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <script>          //outerWidth 不常用          //outerHeight          console.log(window.outerWidth);          console.log(window.outerHeight);          //innerWidth 比较常用          //innerHeight          console.log(window.innerWidth);          console.log(window.innerHeight);          console.log("=======================");            //获取html元素的offsetWidth和offsetHeight          console.log(document.documentElement.offsetWidth);          console.log(document.documentElement.offsetHeight);          //获取window的滚动大小,这个还是比较常用的          //window.scrollX == window.pageXOffset          //window.scrollY == window.pageYOffset          window.onclick = function () {              console.log(window.scrollX);              console.log(window.scrollY);          }          //监听window的滚动事件          window.onscroll = function (){          //    console.log(window.scrollY);            }          //小案例,实现点击一下按钮,窗口就往下滚动100px的效果          let btn = document.querySelector("button")          let box = document.querySelector(".box")          let btnBack = document.querySelector(".back") //返回按钮          let moveVal = 0          btn.onclick = function () {              window.scrollBy(0,100) //滚动是滚动了,但是如何让按钮同步下降呢,注意这里有一个单位问题              // moveVal +=200 //先在这里改变数值,然后在下面设置              // box.style.marginTop =moveVal+ "px" //需要加上px,否则不起作用              // console.log(box.style.marginTop);              //当然,也可以用css把按钮固定,参考上面的css代码              //设置返回按钮看见              btnBack.style.display = "inline"          }              btnBack.onclick = function(){              window.scrollTo(0,0)              //窗口滚动回去后,可以把按钮隐藏               btnBack.style.display = "none"          }      </script>  </body>  </html> |

## 老师的案例写法1,只有当窗口的滚动距离大于等于600px返回按钮财显示,否则返回按钮隐藏,这个需要监听window的滚动事件-使用css的display属性

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>      <meta charset="UTF-8">      <meta name="viewport" content="width=device-width, initial-scale=1.0">      <title>window获取创建的大小-滚动-方法</title>      <style>          .box{              width: 100px;              height: 70px;              background-color: deeppink;              position: fixed;              text-align: center;              line-height: 70px;          }          .back{              position: fixed;              right:20px;              bottom:20px;              display : none          }      </style>  </head>  <body>        <div class="box">           <button class="btn">滚动窗口</button>      </div>        <button class="back">回到顶部</button>      <!-- 下面的br只是为了让浏览器窗口程序滚动条,方便我们的学习 -->      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <script>          //outerWidth 不常用          //outerHeight          console.log(window.outerWidth);          console.log(window.outerHeight);          //innerWidth 比较常用          //innerHeight          console.log(window.innerWidth);          console.log(window.innerHeight);          console.log("=======================");            //获取html元素的offsetWidth和offsetHeight          console.log(document.documentElement.offsetWidth);          console.log(document.documentElement.offsetHeight);          //获取window的滚动大小,这个还是比较常用的          //window.scrollX == window.pageXOffset          //window.scrollY == window.pageYOffset          window.onclick = function () {              console.log(window.scrollX);              console.log(window.scrollY);          }          //监听window的滚动事件,如果滚动距离大于600像素,显示返回按钮,否则不显示          window.onscroll = function (){          //    console.log(window.scrollY);            let scrollY = window.scrollY             if(scrollY>=600){                btnBack.style.display = "inline"               } else {              btnBack.style.display = "none"               }          }          //小案例,实现点击一下按钮,窗口就往下滚动100px的效果          let btn = document.querySelector("button")          let box = document.querySelector(".box")          let btnBack = document.querySelector(".back") //返回按钮          let moveVal = 0          btn.onclick = function () {             window.scrollBy(0,100)          }              btnBack.onclick = function(){              window.scrollTo(0,0)              //窗口滚动回去后,可以把按钮隐藏               btnBack.style.display = "none"          }      </script>  </body>  </html> |

## 老师的案例写法2,只有当窗口的滚动距离大于等于600px返回按钮财显示,否则返回按钮隐藏,这个需要监听window的滚动事件---使用dom的hidden属性

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>      <meta charset="UTF-8">      <meta name="viewport" content="width=device-width, initial-scale=1.0">      <title>window获取创建的大小-滚动-方法</title>      <style>          .box{              width: 100px;              height: 70px;              background-color: deeppink;              position: fixed;              text-align: center;              line-height: 70px;          }          .back{              position: fixed;              right:20px;              bottom:20px;          }      </style>  </head>  <body>        <div class="box">           <button class="btn">滚动窗口</button>      </div>        <button class="back">回到顶部</button>      <!-- 下面的br只是为了让浏览器窗口程序滚动条,方便我们的学习 -->      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <br><br><br><br><br><br><br>      <script>          //outerWidth 不常用          //outerHeight          console.log(window.outerWidth);          console.log(window.outerHeight);          //innerWidth 比较常用          //innerHeight          console.log(window.innerWidth);          console.log(window.innerHeight);          console.log("=======================");            //获取html元素的offsetWidth和offsetHeight          console.log(document.documentElement.offsetWidth);          console.log(document.documentElement.offsetHeight);          //获取window的滚动大小,这个还是比较常用的          //window.scrollX == window.pageXOffset          //window.scrollY == window.pageYOffset          window.onclick = function () {              console.log(window.scrollX);              console.log(window.scrollY);          }           //小案例,实现点击一下按钮,窗口就往下滚动100px的效果          let btn = document.querySelector("button")          let box = document.querySelector(".box")          let btnBack = document.querySelector(".back") //返回按钮          btnBack.hidden = true //默认把按钮隐藏          //监听window的滚动事件,如果滚动距离大于600像素,显示返回按钮,否则不显示          window.onscroll = function (){          //    console.log(window.scrollY);            let scrollY = window.scrollY             if(scrollY>=600){                btnBack.hidden = false             } else {              btnBack.hidden = true             }          }            let moveVal = 0          btn.onclick = function () {             window.scrollBy(0,100)          }              btnBack.onclick = function(){              window.scrollTo(0,0)              //窗口滚动回去后,可以把按钮隐藏              btnBack.hidden = true //如果上面用display,这里也用display,如果上面用hidden,这里也用hidden          }      </script>  </body>  </html> |

# 22.DOM练习,动态创建列表p153

## 老师的案例

|  |
| --- |
|  |

### 也可以这么写

|  |
| --- |
|  |

## 我的案例

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>      <meta charset="UTF-8">      <meta name="viewport" content="width=device-width, initial-scale=1.0">      <title>动态创建列表</title>  </head>  <body>      <input type="text" id="new"> <button>add</button>      <ul></ul>      <script>         let btnAdd = document.querySelector("button")         let newInput = document.querySelector("#new")         let fruitList = document.querySelector("ul")         let fruits = ["apple","cherry","banana"]         fruits.forEach(item=>{           let li = document.createElement("li")           li.textContent = item           fruitList.append(li)         })         //动态添加项         btnAdd.onclick = function (){           let li = document.createElement("li")           li.textContent = newInput.value           fruitList.append(li)           newInput.value = " "         }        </script>  </body>  </html> |

# 23.Dom练习,创建一个屏幕时钟p154

## 我的案例

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>      <meta charset="UTF-8">      <meta name="viewport" content="width=device-width, initial-scale=1.0">      <title>屏幕时钟案例</title>  </head>  <body>      <div class="box">          <h2>显示当前时间</h2>          <h3 class="info"></h3>      </div>      <script>          let box = document.querySelector(".box")          let info = document.querySelector(".info")            //这个函数比较麻烦但是效果更好 格式:2025-09-10 18:28:30          function showDateTimeString() {              let d = new Date() //每一次都需要new              let yr =d.getFullYear()              let mn =new String( d.getMonth()+1).padStart(2,"0")              let dy =new String( d.getDate()).padStart(2,"0")              let h = d.getHours()              if(h<10){                  h = "0" + h              }              let m = d.getMinutes()              if(m<10){                  m = "0" + m              }                let s = d.getSeconds()              if(s<10){                  s = "0"+ s              }              info.innerHTML = `${yr}-${mn}-${dy} ${h}:${m}:${s}`          }            //显示18:28:30          function showTimeString2(params) {           let d = new Date() //每一次都需要new           info.innerHTML = d.toLocaleTimeString()         }         //显示2025/9/10 18:28:30         function showDatetimeStr(params) {           let d = new Date() //每一次都需要new           info.innerHTML = d.toLocaleString()         }          setInterval(()=>{              showDateTimeString()              // showTimeString2()              // showDatetimeStr()          },1000)          // console.log(dateStr);        </script>  </body>  </html> |

## 老师的案例

|  |
| --- |
|  |
|  |

# 24.Dom练习,制作一个倒计时案例,截至时间是监听晚上24:00:00

## 难点: 需要找到现在的时间戳和结束时间的时间戳,然后需要使用整除和取余数算法来得到相差的时分秒,然后分别设置到不同的span标签中,注意:需要使用setInterval函数否则得到的只是一次的数据,违法动态减小,算法原理如图,只不过时间的进制是60而不是10:

|  |
| --- |
|  |

## 案例的完整代码如下

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>      <meta charset="UTF-8">      <meta name="viewport" content="width=device-width, initial-scale=1.0">      <title>倒计时案例</title>      <style>          .countdown{              color:deeppink;              font-size: 20px;          }          .countdown .time{              /\* border: 1px solid #999; \*/              display: inline-block;              padding: 5px;              background-color: #cff7ec;              border-radius: 5px;          }      </style>  </head>  <body>      <div class="countdown">        <span class="time hour">03</span>        <span class="divider">:</span>        <span class="time minute">10</span>        <span class="divider">:</span>        <span class="time second">20</span>      </div>      <script>          //格式化函数,返回xx如01,09,11,12...,其实开源把这个函数作为一个工具函数写到另外一个js文件中.          function padLeft(content,count,padstr)          {              count = count ||2              padstr = padstr || '0'              content = String(content)              return content.padStart(count,padstr)          }         //获取时分秒对应的span         let hour = document.querySelector(".hour")         let minute = document.querySelector(".minute")         let second = document.querySelector(".second")         let dateEnd = new Date() //这个是我们的目标,只需要获取一次.         //获取结束的时分秒         dateEnd.setHours(24)         dateEnd.setMinutes(0)         dateEnd.setSeconds(0)         dateEnd.setMilliseconds(0) //这些代码写在外面可以提高运行效率.         setInterval(()=>{          //获取倒计时的hour,minute,second的值          let dateNow = new Date() //这个需要放置setInterval里面因为每一次都需要获取.          //计算这两个时间的间隔,然后对它取整          let timeInterval = Math.floor((dateEnd.getTime()-dateNow.getTime())/1000)          console.log(timeInterval);          //然后需要获取这个时间间隔的时分秒,使用整除和取余数的方法          let hrs = Math.floor(timeInterval/3600)          let mns = Math.floor((timeInterval/60)%60)          let secs = Math.floor(timeInterval%60)          //分别给他设置内容          hour.textContent=padLeft(hrs)          minute.textContent=padLeft(mns)          second.textContent=padLeft(secs)         },1000)        </script>  </body>  </html> |