# 前面的事件虚构标记,参考: [学习笔记4-事件处理1](coderwhy-jsdom学习笔记4-事件处理1.docx)

# 9.事件委托-案例1 p164

## 1>什么是事件委托

|  |
| --- |
|  |

## 小案例:新建一个html文件,然后添加一个ul元素,在里面添加10个li元素,然后我们有一个需求,就是被点击的li我们需要给他一个样式,比如修改字体颜色和背景颜色,此时我们可以先把这些li都获取到放到一个lis列表里面.然后我们利用循环来添加点击事件处理函数,在里面把每一个li的样式的classList列表添加一个active样式class,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>事件委托</title>      <style>          .active{              background-color: pink;              color: white;              font-size: 20px;              width: 200px;          }      </style>  </head>  <body>      <ul>          <li>1</li>          <li>2</li>          <li>3</li>          <li>4</li>          <li>5</li>          <li>6</li>          <li>7</li>          <li>8</li>          <li>9</li>          <li>10</li>      </ul>      <script>        let lis = document.querySelectorAll("li")        for(let li of lis){          li.addEventListener("click",function(){              this.classList.add("active") //这里只能用this,不能用li否则只有最后一个有效果          })        }      </script>  </body>  </html> |

### 效果

|  |
| --- |
|  |
|  |

## 虽然能够实现功能,但是有点啰嗦,我们可以使用事件委托的发生把事件交给ul处理

|  |
| --- |
|  |

### 这样子也能够实现效果,利用的就是事件冒泡的原理

|  |
| --- |
|  |

## 我们再提一个需求,就是如果不是当前点击的项目,需要把样式还原,一个方法可以实现,不过有点麻烦,就是先把所有的li是样式移除,然后再给当前选中的li添加样式

|  |
| --- |
|  |

### 效果

|  |
| --- |
|  |
|  |
|  |

### 我们还有一种方法,就是利用.active来查找元素,找到就把他的样式移除,因为我们在点击第二个li的时候就需要做了,所以他前面应该只有一个元素有这个样式

|  |
| --- |
|  |

### 这样子也是可行的,而且性能更高,另外还有一种方法,就是用变量记录上一次的选择,然后把它的样式移除

|  |
| --- |
|  |

#### 这种方法很好,效率也很高,也很常用.

## 不过这个案例有点问题,就是当我们点击ul的时候,所以的li都会被添加样式,这个问题的解决办法如下,只需要做一个判断,保证它是li

|  |
| --- |
|  |

## 案例完整代码如下

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>      <meta charset="UTF-8">      <meta name="viewport" content="width=device-width, initial-scale=1.0">      <title>事件委托</title>      <style>          .active{              background-color: pink;              color: white;              font-size: 20px;              width: 200px;          }      </style>  </head>  <body>      <ul>          <li>1</li>          <li>2</li>          <li>3</li>          <li>4</li>          <li>5</li>          <li>6</li>          <li>7</li>          <li>8</li>          <li>9</li>          <li>10</li>      </ul>      <script>        //1.传统的方式      //   let lis = document.querySelectorAll("li")      //   for(let li of lis){      //     li.addEventListener("click",function(){      //         this.classList.add("active") //这里只能用this,不能用li否则只有最后一个有效果      //     })      //   }      //2.使用事件委托,统一在ul中监听      // let ul = document.querySelector("ul")      // ul.onclick = function(e){      //     // console.log("某个li被点击了...");     //    if(e.target !== ul){      //     e.target.classList.add("active")      //    }      // }      //3.新需求,从前选中的对象才添加样式,其他的样式把添加这个样式      // let lis = document.querySelectorAll("li")      // let ul = document.querySelector("ul")      // ul.onclick = function(e){      //     // console.log("某个li被点击了...");      //     //先吧所有的li是active样式移除,      //     for(let li of lis){      //         if(li.classList.contains("active"))      //         {      //             li.classList.remove("active") //这里不能用this      //         }        //     }      //     //然后再给当前选中的li添加样式      //   if(e.target !== ul){      //     e.target.classList.add("active")      //    }      // }      //4.新需求的第二种实现方法      // let ul = document.querySelector("ul")      // ul.onclick = function(e){      //     // console.log("某个li被点击了...");      //     //先找到有active样式的样式      //     let liAct = ul.querySelector(".active")      //     if(liAct){      //         liAct.classList.remove("active")      //     }      //     //然后再给当前选中的li添加样式      //   if(e.target !== ul){      //     e.target.classList.add("active")      //    }      // }       //5.新需求的第三种实现方法,用变量记录选中的元素,在下一次点击的时候把这个元素的样式移除      let ul = document.querySelector("ul")      let lastLi = null      ul.onclick = function(e){          // console.log("某个li被点击了...");          //存在才移除样式          if(lastLi && e.target.tagName ==="LI"){              lastLi.classList.remove("active")          }          //然后再给当前选中的li添加样式,ul是不添加这个样式的.          if(e.target !== ul){              e.target.classList.add("active")               //保存本次选中的项,然后在下一次点击的时候移除他的样式              lastLi = e.target          }          }      </script>  </body>  </html> |

# 10.事件委托-案例2 p165

## 比较简单,就是工具点击按钮的属性来判断点击了哪个按钮,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>事件委托2</title>  </head>  <body>      <div class="box">          <button data-action="remove">移除</button>          <button data-action="new">新建</button>          <button data-action="search">搜索</button>      </div>      <script>         //在div的点击事件中(通过按钮的事件冒泡触发)根据点击的按钮的文本来输出不同的信息      //    方式1      //    let div = document.querySelector(".box")      //    div.onclick = function(e){      //        switch(e.target.textContent){      //           case "移除":      //             console.log("点击了移除按钮...");      //             break;      //           case "新建":      //             console.log("点击了新建按钮...");      //             break;      //           case "搜索":      //              console.log("点击了搜索按钮...");      //              break;      //        }      //    }      //    方式2      let div = document.querySelector(".box")      div.onclick = function(e){             switch(e.target.dataset.action){                case "remove":                  console.log("点击了移除按钮...");                  break;                case "new":                  console.log("点击了新建按钮...");                  break;                case "search":                   console.log("点击了搜索按钮...");                   break;             }         }      </script>  </body>  </html> |

### 效果

|  |
| --- |
|  |

# 11.鼠标常见事件p167

|  |
| --- |
|  |

# 12,

# 13.

# 14.常见键盘事件和案例

|  |
| --- |
|  |