①添加事件方法

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | addHandler：function(element,type,handler){      if(element.addEventListener){ //检测是否为DOM2级方法          element.addEventListener(type, handler, false);      }else if (element.attachEvent){ //检测是否为IE级方法          element.attachEvent("on" + type, handler);      } else { //检测是否为DOM0级方法          element["on" + type] = handler;      }  } |

**②移除之前添加的事件方法**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | removeHandler：function(element, type, handler){         if (element.removeEventListener){             element.removeEventListener(type, handler, false);         } else if (element.detachEvent){             element.detachEvent("on" + type, handler);         } else {             element["on" + type] = null;         }     } |

**③获取事件及事件对象目标**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | //获取事件对象的兼容性写法   getEvent: function(event){       return event ? event : window.event;   },   //获取事件对象目标的兼容性写法   getTarget: function(event){       return event.target || event.srcElement;   } |

**④阻止浏览器默认事件的兼容性写法**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | preventDefault: function(event){          if (event.preventDefault){              event.preventDefault();          } else {              event.returnValue = false;          }      } |

**⑤阻止事件冒泡的兼容性写法**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | stopPropagation: function(event){          if (event.stopPropagation){              event.stopPropagation();          } else {              event.cancelBubble = true;          }      } |

**⑥mouseover和mouseout 事件才包含的获取相关元素的方法**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | //mouseover和mouseout 事件才包含的获取相关元素的方法  getRelatedTarget: function(event){      if (event.relatedTarget){          return event.relatedTarget;      } else if (event.toElement){          return event.toElement;      } else if (event.fromElement){          return event.fromElement;      } else {          return null;      }  } |

**⑦鼠标滚轮判断**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32 | /\*对于mousedown 和mouseup 事件来说，则在其event 对象存在一个button 属性，  表示按下或释放的按钮。DOM的button 属性可能有如下3 个值：0 表示主鼠标按钮，1 表示中间的鼠  标按钮（鼠标滚轮按钮），2 表示次鼠标按钮。在常规的设置中，主鼠标按钮就是鼠标左键，而次鼠标  按钮就是鼠标右键。  IE8 及之前版本也提供了button 属性，但这个属性的值与DOM 的button 属性有很大差异。  θ 0：表示没有按下按钮。  θ 1：表示按下了主鼠标按钮。  θ 2：表示按下了次鼠标按钮。  θ 3：表示同时按下了主、次鼠标按钮。  θ 4：表示按下了中间的鼠标按钮。  θ 5：表示同时按下了主鼠标按钮和中间的鼠标按钮。  θ 6：表示同时按下了次鼠标按钮和中间的鼠标按钮。  θ 7：表示同时按下了三个鼠标按钮。\*/  getButton: function(event){      if (document.implementation.hasFeature("MouseEvents", "2.0")){          return event.button;      } else {          switch(event.button){              case 0:              case 1:              case 3:              case 5:              case 7:              return 0;              case 2:              case 6:              return 2;              case 4:              return 1;          }      }  } |

**⑧能够取得鼠标滚轮增量值（delta）的方法**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | getWheelDelta: function(event){      if (event.wheelDelta){          return (client.engine.opera && client.engine.opera < 9.5 ?              -event.wheelDelta : event.wheelDelta);      } else {          return -event.detail \* 40;//firefox中的值为+3表示向上滚，-3表示向下滚      }  } |

**⑨跨浏览器的方式取得字符编码**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | getCharCode: function(event){      if (typeof event.charCode == "number"){          return event.charCode;      } else {          return event.keyCode;      }  } |

**⑩访问剪贴板中的数据**

|  |  |
| --- | --- |
| 1  2  3  4 | getClipboardText: function(event){          var clipboardData = (event.clipboardData || window.clipboardData);          return clipboardData.getData("text");      } |

**⑪设置剪贴板中的数据**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | setClipboardText: function(event, value){          if (event.clipboardData){              return event.clipboardData.setData("text/plain", value);          } else if (window.clipboardData){              return window.clipboardData.setData("text", value);          }      } |