day17——获取元素节点的方式

2022年6月23日

14:47

## //DOM介绍

    // 1.DOM document object model  文档对象模型

    // 2.一个HTML文件就是一个网页，一个网页就是一个文档，一个文档在JS眼中就是一个document对象

    // 3.document对象就包括页面中的所有元素

    // 4.我们学习DOM，其实就是学习通过document对象去操作页面中的各种元素，使页面动起来

## // 节点

    // 1.节点英文单词：node

### // 2.节点的类型有：

    // 文档节点  document对象

    // 元素节点  页面中的各种元素

    // 属性节点   元素的属性

    // 文本节点   元素的文本内容

    // 注释节点    注释

    // 文档片段节点   页面中的一段标签内容

    // 文档声明节点    页面中的文档声明

### // 3.元素节点之间的关系

    // 父子关系

    // 兄弟关系

    // 节点属性

    // 1.页面中的内容，JS都认为是一个个的节点

    // 2.节点都有以下3个属性

    // nodeType 节点类型

    // nodeName 节点名称

    // nodeValue 节点的值

    // 注意：不同的节点类型，nodeType属性值不同

    // 文档节点  9

    // 元素节点    1

    // 属性节点   2

    // 文本节点   3

## // 获取元素节点的方式

    // getElementById(): 根据元素的id属性去获取元素

    // getElementsByClassName(): 根据元素的class属性获取元素

    // getElementsByTagName(): 根据标签获取元素

    // getElementsByName(): 根据元素的name属性获取元素

    // querySelector()、querySelectorAll(): 根据选择器去获取元素，选择器就是css中写的选择器

    // getElementById()方法只能由document对象去调用，不能用别的元素对象去调用

    // getElementsByTagName('标签')，该方法不仅document对象可以使用，别的元素节点也可以调用

### // 创建元素节点

    // document.createElement('标签名')

    // 该方法会返回创建好的元素对象

### // 创建文本节点

    // document.createTextNode('文本内容');

    // 该方法会返回创建好的文本对象，文本节点是元素节点的子节点

### // 创建属性节点

    // document.createAttribute('属性名');

    // 该方法会返回创建好的属性节点对象，属性节点是没有属性值的  属性节点.value = '属性值'

//一般使用的是标准属性，不需要自己创建

### // 给元素节点添加属性节点

    // 元素节点.setAttributeNode(属性节点)

day18——操作元素节点之间的内容及属性节点

2022年6月24日

9:20

### // 操作元素节点之间内容

// 1.目前我们会创建元素节点，创建属性节点、创建文本节点

// 2.创建元素节点需要掌握！创建属性节点、创建文本节点了解即可，之前方式比较繁琐，后面有简单方法

// 3.操作元素节点之间的内容，指的就是开始标签和结束标签之间的东西！

// 4.操作元素节点之间的内容有三种方式

// 1)innerHTML 属性

// 1)innerText 属性

// 1)textContent 属性

### // innerHTML属性

// 1.通过元素节点的innerHTML属性可以获取元素之间的内容，也可以设置元素之间的内容

// 2.语法格式：元素节点.innerHTML = 值；

// 3.innerHTML不仅可以获取|设置元素之间的文本内容，还可以获取|设置元素之间标签

// 4.通过innerHTML属性赋值，会将之前的内容给覆盖掉，不想覆盖就需要在原来的基础上拼接

### // innerText属性

// 1.通过元素节点的innerText属性可以获取文本之间的内容，也可以设置文本之间的内容

// 2.innerText属性不识别标签

### // innerText属性和textContent属性的区别？

// innerText属性不识别隐藏的内容 (隐藏内容指什么）

// textContent属性识别隐藏的内容

### // 表单内容回顾：

// action:指定表单提交给服务器的地址

// method:指定表单提交的方式，get提交数据会在地址栏显示，post提交数据不会在地址栏显示

// 表单项的内容要想能提交给服务器，表单项元素需要有name属性

// 表单项元素中可以写value属性，value的值会显示在表单中，也就是一个默认值

// 用户在表单项元素中输入内容后，也是赋值给value属性

// 获取|设置表单元素之间的文本内容，要用value属性

### // 属性节点的操作

// 通过元素节点的attributes获取属性节点

// 1.语法格式：元素节点.attributes

// 2.返回的是一个类数组对象，可以通过遍历该结果查看效果

// 通过元素节点的getAttributeNode()方法获取属性节点

// 1.语法格式：元素节点.getAttributeNode('属性名')

// 操作元素的class属性

// 元素节点.className;

// 元素节点['className'];

// 元素节点.className = 值;

// 元素节点['className'] = 值;

晨测：

  // 获取元素的方式有哪些，写出具体语法

    // document.getElementById

    // document.getElementsByName

    // document.getElementsByTagName

    // document.getElementsByClassName

    // document.querySlector

    // document.querySlectorAll

    //怎样创建元素节点、属性节点、文本节点

    // document.createElement

    // document.createAttribute

    // document.createTextNode

day19——操作自定义属性的方法、元素节点之间的关系

2022年6月27日

20:25

### checked属性

对于复选框，check属性表示让复选框默认选中

chenked属性是标准属性，使用元素节点.checked;获取到的值不是checked,而是true,或者是false

### selected属性

1.下拉列表中，通过selected属性可以指定某个选项默认选中

2.option元素节点.selected;返回的是true或者false

### 操作自定义属性的方法

获取属性：  元素节点.getAttribute('属性名')

设置属性:   元素节点.setAttribute('属性名','属性值')

判断元素是否有某个属性:   元素节点.hasAttribute('属性名')

判断元素是否有属性:  元素节点.hasAttributes()

删除属性:  元素节点.removeAttribute('属性名')

### 元素节点的关系属性

1.parentElement属性:  元素节点的该属性，可以返回该元素的父元素

2.children属性: 获取元素的所有子元素，返回一个类数组对象

3.firstElementChild属性:  获取元素的第一个子元素

4.lastElementChild属性:  获取元素的最后一个子元素

5.nextElementSibling属性:  获取元素的下一个兄弟元素

6.previousElementSibling属性:  获取元素的上一个兄弟元素

### 节点之间的关系（了解）

1.parentNode属性:  获取到元素的父元素

2.childNodes属性: 获取元素的所有子节点,注意，空白也是一个子节点

3.firstChild属性: 获取元素的第一个子节点

4.lastChild属性: 获取元素的最后一个子节点

5.previousSibling属性: 获取元素的上一个兄弟节点

6.nextSibling属性: 获取元素的下一个兄弟节点

### 两类获取页面元素方式的对比

getElementxxx

querySelector

后者IE低版本不支持，前者中大部分IE低版本支持

后者不能动态的获取最新的元素，前者可以获取到最新的结果，但不是绝对的，得看代码写的

位置

day20——localtion对象、定时器

2022年6月28日

8:32

### document对象属性的分类

1)快捷方式属性

2)节点集合属性

3)网页静态信息属性

快捷方式属性

1.就是通过这些属性可以很方便、快捷的获取一些内容

2.doctype属性:可以获取网页的文档声明，如果没有文档声明返回null

3.documentElement属性:可以获取到网页的根元素，也就是获取到html元素

4.body属性:可以获取到body元素

5.head属性:可以获取head元素

6.title属性:可以获取到title元素的内容，也可以给title元素设置内容

### 节点集合属性

    1.节点集合属性也是一种快捷获取一些元素的方式，只不过是它获取到的是多个元素，返回的是一个类数组对象

    2.links属性:获取页面中所有设置了href属性的超连接，返回一个类数组对象 document.links

    3.forms属性:获取页面中所有的表单标签，返回一个类数组对象 document.forms

    4.images属性:获取页面中所有的img标签，返回一个类数组对象 document.images

    5.script属性:获取页面中所有的script标签,返回一个类数组对象 document.script

### 文档静态信息属性

    1.documentURI属性:获取网页的地址

    2.domain属性:获取网页地址中的域名

    域名对应的就是一个IP,因为IP地址不好记,所以可以花钱买一个域名，去绑定对应的IP地址

    3.location属性:返回一个location对象,这个location对象中存储了网页地址栏相关的信息

    而且location对象是js内置的，可以直接使用

### location对象

    1.location对象是JS内置的对象,里面存储了地址相关的信息

    2.href属性:获取当前页面的地址，也可以设置地址栏的地址实现页面的跳转

    3.potocol属性:获取到地址栏地址的协议 协议:就是双方或者多方公共遵守的规则

    4.host属性:获取地址栏地址的主机和端口号  主机:指的是就是服务器的地址，也就是服务器的ip地址

    5.hostname属性:获取地址栏地址的主机名，也就是ip名

    6.port属性:获取地址栏地址的主机和端口号

     每个软件在电脑上运行时，会占用电脑一个端口号，互联网中，两个应用程序在不同电脑上要进行通信，必要条件:ip地址、端口号

     7.pathname属性:获取地址栏中路径名 路径名就是域名后面的路径信息

     8.search属性:获取地址栏中地址上的搜索内容  其实就是地址中问号后面的内容

### 定时器介绍

     1.定时器，英文单词 timer

     2.定时器就是可以实现每隔指定时间就做一次某个事情

     3.JS中定时器，就是使用setInterval()函数

     4.setInterval()函数是window对象的函数，window对象是一个全局对象，可以省略，所以我们在使用定时器时，直接调用setInterval()函数即可

     5.语法格式  let 定时器编号 = setInterval(回调函数,时间间隔);时间间隔后面可以写更多的参数，这些参数会传递给回调函数

### 定时器的停止

     1.定时器，一旦启动后，只要不停止，会无限次的执行下去

     2.定时器用完后记得停止，语法 clearInterval();

， ： 逗 回 们 · 么 ？ 
． 根 斟 一 个 索 。 如 何 获 取 它 的 父 元 素 ． 下 一 个 兄 弟 ， 
一 个 旯 弟 、 子 元 素 

day21——增加元素删除元素的方法

2022年6月29日

8:35

### 通过元素节点去调用的DOM操作相关方法

### appendChild()

1)作用:在元素A里面的最后添加一个元素B

2)语法:   元素A.appendChild(元素B)

3)注意:如果添加的元素是已经存在的元素,实际上是相当于元素的移动

### insertBefore()

1)作用:在父元素中插入一个子元素，而且可以指定在哪个子元素之前插入

2)语法:   父元素.insertBefore(新元素,指定的元素)

3)注意:如果添加的元素是已经存在的元素,实际上是相当于元素的移动

### cloneNode()

1)作用:复制一个元素

2)语法:  元素.cloneNode(true|false);

3)说明: true表示不仅复制元素,还复制其内容

        false表示仅复制元素,不复制内容

        不论是true还是false,都不复制事件

### contains()

1)作用:判断父元素中是否有某个子元素

2)语法:父元素.contains(子元素)

### hasChildNodes()

1)作用:判断元素中是否有子节点,注意:空白也是一个节点

2)元素.hasChildNodes()

isEqualNode()

1)作用:判断两个节点是否一样,指的是：节点类型、内容、属性

2)语法   元素.isEqualNode(另一个元素)

### remove()

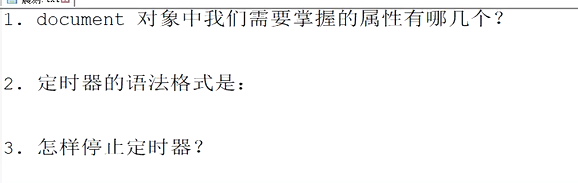
1)作用:删除元素

2)语法: 元素.remove()

### removeChild()

1)作用:删除父元素中的一个子元素

2)语法: 父元素.removeChild(子元素)



day22——\*\*\*js操作行内样式

2022年6月30日

8:37

### js操作行内样式

元素节点.style = 'width:200px;height:200px;'

元素节点.setAttribute('style','font-size:20px;background-color:red;');

元素节点.className = 'box';//在页面中写css样式 给class='box'的元素设置样式

元素节点.style.样式名 = 样式值；//设置样式

元素节点.style.样式名;//获取样式

元素节点.style.样式名 = '';//清除样式

样式名中含有-的话，需要写成驼峰命名，比如font-size写成 fontSize

float是js中的关键字，所以float写成cssFloat

元素节点.style.cssText = '样式:样式值;' //可以用于减少回流

元素节点.style.setProperty('样式名:样式值',优先级)//优先级是可选的，要写的话只能写！important

元素节点.style.getPropertyValue('样式名');

元素节点.style.removeProperty('样式名');

### 获取元素的最终样式

使用window.getComputedStyle()方法可以获取到元素最终生效的所有样式

语法格式：

let 变量 = window.getComputedStyle(元素)；

变量.样式名;//获取到某个生效的样式的值

语法说明：

返回的是一个对象,该对象中就含有该元素上所有生效的样式

样式名有-的话,写成驼峰命名

样式名不要写简写，比如font、margin

这种写法，IE浏览器低版本不支持！IE低版本指的是IE8及其以下的版本

如果是IE低版本的话，使用如下方式

let 变量 = 元素节点.currentStyle;

变量.样式名

与 
div 丿 匚 
给 div 元 素 中 设 置 内 容 ： 你 好 
2 ． 
使 用 ] s 给 div 设 置 宽 度 是 200px ， 
3 ． 
． 删 除 元 素 的 语 法 是 ？ 
body 
高 度 是 200px ， 背 景 颜 色 是 红 色 

day23——获取窗口高度、宽度的属性

2022年7月1日

8:38

### DOM常用属性

1.DOM元素节点,有一些常用的属性,分为3个系列

1)client系列

2)scroll系列

3)offset系列

### client系列的属性

1.client系列的属性有4个，都是clientXXX

2.client系列的属性调用格式:元素节点.clientxxx;

3.client系列的属性返回值都是整数,有小数的话也会四舍五入为整数,而且没有单位

4.client系列的属性:

1)clientWidth:指的是元素的width和左右padding

2)clientHeight:指的是元素的height和上下padding

3)clientLeft:指的是左边框的宽度

4)clientTop:指的是上边框的宽度

### scroll系列的属性

1.scroll系列的属性有4个,调用方式  元素节点.scrollXXX

2.scroll系列的属性返回值也是整数,有小数的话也会四舍五入,没有单位

3.scroll系列的属性

1)scrollWidth:指的是元素的width和左右padding和溢出部分

2)scrollHeight:指的是元素的height和上下padding和溢出部分

3)scrollLeft:返回元素水平滚动条与左边的距离

4)scrollTop:返回元素垂直滚动条与上边的距离

垂直方向的滚动条有的浏览器算在body上，有的算在html上

let scrollTop = document.body.scrollTop || document.documentElement.scrollTop;

利用的是非布尔值参与计算的短路机制,哪个表达式能决定最终的结果，结果就是这个表达式的值

### offset系列的属性（了解）

1.offset系列的属性有5个，使用有点类似于绝对定位，是相对与最近的有定位的祖先元素而言的

2.offset系列的属性使用格式,元素节点.offsetXXX;返回的值也是整数，没有单位。（除了第一个属性外）

3.offset系列的属性

1)offsetParent:返回最近的有定位的祖先元素,如果祖先都没有定位,返回body元素

2)offsetWidth:返回元素的宽度，包括width、padding和border

3)offsetHeight:返回元素的高度,包括:height、padding和border

4)offsetLeft:返回元素与最近的有定位的祖先元素左边的距离

4)offsetTop:返回元素与最近的有定位的祖先元素上边的距离

2 ． 定 义 动 0 的 语 法 格 式 ， 调 用 动 0 的 语 法 格 式 ， 举 例 说 明 
3 ． 一 200p 又 —SOOpx; 表 刁 孓 1 上 么 意 思 
如 回 获 取 表 项 元 肃 中 的 们 

day24——\*\*\*事件绑定的方式

2022年7月4日

8:36

### 事件的介绍

事件，单词 event

事件就是指网页的浏览者、浏览器本身、文档元素本身的行为

### 事件的绑定

1.事件的绑定是指，给某个元素绑定事件，然后当该事件被触发后，就会执行相应的函数

2.事件绑定的方式有

1)直接在元素的开始标签上绑定  (不推荐)

2)使用DOM0级方式绑定事件

3)使用DOM2级方式绑定事件

直接在元素的开始标签上绑定

1.可以在html元素的开始标签上绑定事件,调用相关的js函数执行

2.这种方式不推荐,因为html代码和js代码耦合性过高！不利于后期的运行维护

### DOM0级事件绑定

1.这种方式是我们之前一直用的方式

2.该方法没有浏览器兼容问题

3.语法格式

元素节点.on+事件名称 = function(){}

或者    元素节点.on+事件名称 = 有名函数的函数名 (函数名就代表一个函数声明)

4.语法说明

1)利用这种方式可以给一个元素绑定多个不同的事件

2)利用这种方式给一个元素绑定多个相同的事件的话，后写的生效！！

### DOM2级事件绑定

1.这种方式有浏览器兼容性问题

2.这种方式可以给一个元素绑定多个相同的事件，这些事件都生效！

3.语法格式：

元素节点.addEventListener('事件名称',匿名函数或有名函数函数名,事件流);

4.语法说明

1)参数1是事件名称，不要写on,直接写事件名

2)参数3是事件流,后面讲,它是可选的,默认是false,表示事件冒泡

//通用写法,解决兼容性问题

function binEvent(ele,type,fun,stream){ //元素  事件名称(事件名称不写on)  函数  事件流

    if(ele.addEventlistener){

        ele.addEventListener(type,fun);

    }else{

        ele.attachEvent("on"+type,fun)

    }

}

### 事件的删除

1.我们可以给元素绑定事件，也可以删除元素上的事件

2.绑定事件的方式有3种，我们主要考虑DOM0和DOM2这两种方式的事件删除

### DOM0绑定的事件进行删除

1.利用DOM0绑定事件的特点:给一个元素绑定多个相同的事件,是最后面的生效！

2.语法格式: 元素节点."on"+事件名 =null;

### DOM2绑定的事件进行事件删除

1.DOM2绑定事件时,有浏览器兼容性问题,删除DOM2绑定的事件时,也有浏览器兼容性问题

2.DOM2可以给一个元素绑定多个相同的事件,都生效,删除DOM2绑定的事件时,就得指定要删除的事件名及对应的函数

3.语法格式:  元素节点.removeEventListener('事件名称',函数名,事件流);

4.语法说明:既然语法中要求得写要删除的事件对应的函数名,那就意味着如果事件的函数是一个匿名函数，该事件无法删除

### 事件流

1.事件流就是指页面中事件被响应的顺序

2.事件流有两种

1).事件冒泡,一般默认都是事件冒泡,false

2).事件捕获,true

### 事件冒泡

1.事件冒泡就是指事件是从内向外或者说是从子元素向父元素去执行的顺序!

2.DOM0绑定事件时不涉及事件流,DOM2绑定事件时涉及,主要演示DOM2绑定事件时事件流的情况

### 事件捕获

1.事件捕获就是指事件是从外向内或者说是从父元素向子元素去执行的顺序!

day25——事件对象的属性

2022年7月5日

8:48

### 事件对象的介绍

1.事件发生以后,会产生一个事件对象,并作为一个参数传递给事件处理函数

2.事件对象都是Event类型的实例

3.事件对象中记录了与事件相关的一些信息,比如:事件的类型、鼠标的位置、事件是否支持冒泡等等

### 事件对象的获取

1.IE低版本和非IE低版本的浏览器获取事件对象的代码不太一样,需要考虑写兼容性的代码

2.事件对象,是在事件被触发后,由浏览器创建的,会将其作为参数传递给事件处理函数。

所以,我们要用事件对象的话,在事件处理函数写形参即可。(非IE低版本)

元素节点.onclick = function(e){

    //使用事件对象

}

3.在IE低版本浏览器中不能直接通过在事件处理函数中写形参的方式获取事件对象。

IE低版本需要使用window.event;的方式来获取事件对象!

### 事件对象的属性

1.事件对象既然是对象,就有属性和方法

2.bubbles属性:判断事件是否支持冒泡,支持的话返回true,不支持返回false。

该属性只能读取,不能修改。bubble是冒泡的意思

3.eventPhase属性: 可以获取到事件所处的阶段。phase是阶段、时期的意思

0 - 事件未发生  1 - 事件处于捕获阶段  2 - 事件到达目标节点  3 - 事件处于冒泡阶段

4.keyCode属性:键码值,我们键盘上的每个按键,在计算中都有一个键码值,所以,通过键码值就可以判断用户按下的是哪一个键

该属性要配合键盘事件使用

5.key属性:可以根据你按下的键返回一个字符说明,比如 a - a, 上移 - Arrowup

6.clientX、clientY属性:

1)配合鼠标事件使用

2)返回鼠标位置与视口左上角的水平方向的距离和垂直方向的距离

7.screenX、screenY属性

1)配合鼠标事件使用

2)返回鼠标位置与屏幕左上角的水平方向的距离和垂直方向的距离

8.offsetX、offsetY属性

1)配合鼠标事件使用

2)返回鼠标位置与元素自己的左上角的水平方向和垂直方向的距离

3)元素的左上角,包含 内容区、内边距,不包含边框和外边距

9.pageX、pageY属性

1)配合鼠标事件使用

2)返回鼠标位置与文档(body)左上角的水平方向与垂直方向的距离

### 晨测

DOM0级事件绑定

    ele."on"+事件 = 函数表达式;

    DOM2级事件绑定

    ele.addEventListener(type,fun,stream);

    事件冒泡就是从子元素向父元素扩散，绑定的是同一事件，所有事件都会执行

    事件捕获就是从父元素向子元素扩散，绑定的是同一事件，所有事件都会执行

    ele.style.width;

    ele.style.height;

    document.body.clientWidth;

    document.body.clientHeight;

    兼容性问题指某些方法在低版本浏览器中不支持使用，解决方法是使用一个方法时，先对这个方法进行一个判断

    functioon bindEvent(){

        if(ele.addEventListener){

ele.addEventListener(type,fun,stream);

        }else{

            ele.addEventListener(type,fun,stream);

        }

    }

day26——事件对象的属性、键盘、鼠标、表单事件

2022年7月5日

11:10

10.cancelable属性：

    1)判断事件是否可以取消,可以的话返回true,不可以返回false

    2)它要配合preventDefault()方法使用,用来组止元素事件默认的行为,比如组止超连接点击后自动跳转

11.type属性:返回一个字符串,表示事件的类型,比如:click、mouseover

12.ctrlKey、altKey、shiftKey属性:

    1)配合键盘事件使用

    2)可以用来判断是否按下ctrl、alt、shift键,是的话返回true,不是返回false

13.button属性

    1)配合鼠标事件使用

    2)判断按下的是哪个鼠标按键

    0 - 左键

    1 - 滚轮

    2 - 右键

    3)该属性只能判断按下的一个键,如果同时按下两个键判断不了

14.buttons属性

    1)配合鼠标事件使用

    2)判断按下是哪个鼠标按键,可以判断同时按下的多个按键

    1 - 左键

    2 - 右键

    4 - 滚轮

    3)如果同时按下左右键的话,返回他们的和

15.currenTarget、target属性

    1)currentTarget表示绑定事件的元素节点,等价于this

    2)target表示真正触发事件执行的元素节点

    3)target属性有兼容性问题,在IE低版本中要写成srcElement

    var target = e.target ||e.srcElement;

### 事件对象的方法

1.preventDefault()

1)作用:阻止元素的默认行为,比如:点击超链接后要进行跳转,是超链接的默认行为

2)该方法有兼容性问题,如果是IE低版本的话,使用  事件对象.returnValue = false;

2.stopPropagation()

1)作用:阻止事件冒泡

2)该方法有兼容性问题,如果是IE低版本的话,使用: 事件对象.cancelBubble  = true;

有默认行为的:

1.超链接有默认行为,点击超链接后跳转

2.表单有默认行为,点击提交按钮后表单提交

3.鼠标右击有默认行为,右击后显示菜单

### 鼠标事件

1.鼠标事件有很多,都是通过鼠标去触发执行的

2.鼠标事件有:

click  鼠标单击

dblclick  鼠标双击

mousemove 鼠标移动

mousedown 鼠标按下

mouseup   鼠标松开

mouseover 鼠标滑入,会冒泡

mouseout  鼠标滑出,会冒泡

mouseenter 鼠标滑入,不会冒泡

mouseleave 鼠标滑出,不会冒泡

contextmenu 鼠标右击

3.说明:一个元素可以绑定多个事件,如果给一个元素绑定了mousedown、mouseup和click事件,

那触发时的执行顺序:先走mousedown,再走mouseup,最后click

### 键盘事件

    1.键盘事件就是通过键盘按键去触发的事件

    2.键盘事件有3个:

    1)keydown:键盘按下事件

    2)keyup:键盘松开事件

    3)keypress:键盘按下事件,要求你按下的键得是有值的键,比如:a、b、1、2等是有值的键

      ctrl、shift、alt等是无值的键

    3.说明:如果一个元素同时绑定了keydown和keypress事件,当我们按下一个有值的键时,先执行keydown然后执行keypress

### 表单事件

    1.表单事件就是与表单或者表单项有关的事件

    2.表单事件有:

    1)focus:聚焦事件,比如:光标在input中,就是input获取到了焦点,也就是聚焦

    2)blur:失焦事件

    3)input:内容改变事件,一旦表单项内容发生改变就会触发

    4)change:内容确定改变事件,一旦表单项内容确定改变就会触发

    5)submit:表单提交事件,当表单提交时会触发

    6)reset:表单重置事件,当表单重置时会触发

 // let lis = document.querySelectorAll('li');

        // let ul = document.querySelector("ul");

        // //第一种方式

        // for (let i = 0; i < lis.length; i++) {

        //     lis[i].onclick = function () {

        //         console.log(lis[i].innerHTML);

        //     }

        // };

        //第二种方式

        // for(var i=0;i<lis.length;i++){

        //     lis[i].onclick = function(){

        //         console.log(i);

        //         console.log(this.innerHTML);

        //     }

        // };

        //第三种方式

        // ul.onclick = function (e) {

        //     console.log(e.target.innerHTML);

        // }

        //第四种方式

        // for (var i = 0; i < lis.length; i++) {

        //     (function (i) {

        //         lis[i].onclick = function () {

        //             console.log(lis[i].innerHTML);

        //         }

        //     })(i);

        // };

day27——滚动条、文档加载事件(load、error)

2022年7月7日

19:01

### 滚动条事件

1.网页中有滚动条的话,一般也是网页右侧的垂直滚动条

2.我们可以给整个文档绑定一个滚动条事件

3.格式 document.onscroll = function(){}

4.document.body.scrollTop || document.documentElement.scrollTop;

可以获取到垂直滚动条与上边的距离

### 文档加载事件介绍

1.客户端(浏览器,前端)发送请求给服务器,服务器收到请求后,对请求进行处理,

然后将处理的结果响应给浏览器,浏览器收到服务器响应回来的数据后,对数据进行加载解析,然后展示给用户看

2.文档加载事件就是指在文档响应过程中的一些事件

3.文档加载相关的事件:

1)load：文档加载完成事件

2)error：文档或者资源加载出错事件

3)DOMContentLoaded:DOM结构加载完成事件

4)readystatechange:准备状态发生改变事件,指的是文档加载过程中状态发生改变后触发

### load事件

1.load事件指的是网页或者某个资源加载完毕后会触发的事件

2.load事件我们一般是给window对象去绑定,表示整个网页都加载完毕

### error事件

1.error事件是指页面或资源加载出错时触发

day28——DOMContentLoaded、readystatechange、鼠标滚轮事件

2022年7月8日

19:43

### DOMContentLoaded 事件

    1.DOMContentLoaded 事件是指:DOM结构加载完成

    2.浏览器发送请求给服务器,服务器收到请求后对请求进行处理,处理完成后,响应结果给浏览器,

    浏览器收到响应的数据后,对数据进行解析,当元素节点都解析完成后,就会触发DOMContentLoaded事件,也就是说

    DOMContentLoaded事件要比load事件发生的要早。

    3.要求:

    1)DOMContentLoaded事件必须给domcument对象绑定

    2)必须用DOM2的方式绑定事件

### readystatechange事件

    1.表示准备状态发生改变

    2.document对象readyState属性的值,就反映了文档加载过程中的阶段、状态:

    1)loading:正在加载中....

    2)interactive:表示网页DOM结构解析完成(引入的外部资源还没完事),可以触发

DOMContentLoaded事件

    3)complete:表示整个网页文档加载完毕,可以触发load事件

    3.上面三个网页加载过程中的状态,一旦状态发生改变就会触发readystatechange事

件！

    4.该事件要给document对象去绑定,用来监听文档加载状态的变化！

### 鼠标滚轮事件

    1.鼠标:mouse  滚动:wheel

    2.鼠标滚轮事件也是鼠标事件的一种,只不过是它比较特殊

    3.鼠标滚轮事件有浏览器兼容性问题,谷歌和IE浏览器写法一样,火狐浏览器是另一种写法

    4.给谷歌和IE浏览器绑定鼠标滚轮事件语法:

    documnet.onmousewheel = function(){}

    5.通过事件对象的wheelDelta属性可以判断滚轮滚动的方向,-120表示向下滚动,120表示向上滚动

### 