webAPI知识点

第一天笔记

**一.获取页面中元素方法（一个网页就是一个文档，标签=元素=对象）。**

1.根据**id**获取元素，返回值是一个元素的（dom）对象。

1.1语法：document.getElementId（"id的属性名"）。

Param（参数）：id的属性名（字符串）；

Return（返回值）：一个对象；

1.2特点：

a.id具有唯一性，只能用document来调用。

b.如果能找到对应的id，则返回值是元素（对象）；如果没有找到，则返回的是null。

2.根据**标签名**获取元素，返回的是一个（伪）数组，每个标签都是数组的一个元素.

2.1语法：document.getElementsByTagName("标签名字")。

Param（参数）：标签名字符串；

Return（返回值）：一个伪数组（伪数组，拥有数组的三要素，没有数组的API）；

2.2小技巧：

a.如果获取元素的api中element后面有s，则获取的一定是数组，否则是一个对象；

b.使用 document.getElementsByTagName('标签名') 获取的整个页面所有的标签名元素数组；

c.使用 父元素.getElementsByTagName('标签名') 获取父元素中的所有标签名元素数组；

d.获取id只能使用 document.getElementById('id')；

3.根据**类名**获取元素，返回的是一个（伪）数组，每个带有该类名的标签都是数组的一个元素.

3.1语法：document.getElementsByClassName('类名')。

Param（参数）：类名字符串；

Return（返回值）：一个伪数组（伪数组，拥有数组的三要素，没有数组的API）/ 空数组；

3.2特点：

a.使用类名获取元素用的不多，IE8及以下不支持。

b.尽量使用其他方式来代替。

4.根据**name属性**获取元素，返回的是一个（伪）数组，每个带有该name的属性值的标签都是数组的一个元素。

4.1语法：document.getElementsByName（’name的属性值’）。

Param（参数）：name的属性值；

Return（返回值）：一个伪数组（伪数组，拥有数组的三要素，没有数组的API）/ 空数组；

4.2特点：name属性是表单元素(input)独有的默认属性。

5.根据**选择器**获取元素。

4.1语法：

a. document.querySelector（’选择器名’）。

特点：如果有多个相同的选择器，则只会找到满足选择器的第一个元素。

Param： 选择器名；

Return： 返回一个对象 / null；

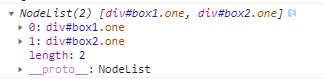


b. document.querySelectorAll（’选择器名’）。

特点：找到页面中满足选择器的所有元素。

Param： 选择器名；

Return： 如果找到对应的选择器，则返回一个数组：



如果找不到，则返回一个空数组：



❀总结

1.返回（伪）数组的元素获取方法包括：

a.根据标签名获取元素：document.getElementsByTagName（’标签名’）；

b.根据类名获取元素：document.getElementByClassName（’类名’）；

c.根据name属性获取元素：document.getElementByName（’name属性值’）；

d.根据选择器获取元素：document.querySelectorAll（’选择器名’）；

2.返回一个元素（对象）的元素获取方法包括：

a.根据id获取元素：document.getElementById（’id的属性值’）；

b.根据选择器获取元素：document.querySelector（’选择器名’）；

二.获取元素的属性特点：

1.获取的属性值都是字符串类型，并且带单位（行内写的是什么，拿的就是什么）；

2.只能获取行内属性，不能获取行外属性；

3.获取类名，使用className属性；

4.如果属性中有-，则获取时需要转成驼峰命名法；

1.获取类名属性需要使用className（因为class是js的一个关键字）。

**console.log ( box.**className **);**

2.

a.元素的style属性是一个对象，包含多个属性。

b. 获取的属性值都是字符串类型，并且带单位。

<**div** id=**"box"** class=**"div1"** style=**"***background-color*: **red;***width*:100**px;"** >**我是div**</**div**>

**console.log ( box.**style **);** //元素的样式style属性值是一个对象  
**console.log (** typeof **box.**style **);** //object;  
**console.log ( box.**style**.**width **);**//100px 获取宽度是一个带单位的字符串类型;  
**console.log (** typeof **box.**style**.**width **);** //string

如果css的属性带-，比如font- border- padding-,使用js获取需要转成驼峰命名法  
**a.去掉-，将-后面的单词首字母大写  
b.原因： - 不符合js中变量命名规范**

**console.log ( box.**style**.**backgroundColor **);**//red

4.点语法只能获取行内样式，无法获取行外（内外联）的属性。

**console.log ( box.**style**.**height **);**//空字符串

5.类名属性

a.修改类名，则会修改对应的样式。

b.注意点：

如果我想替换类样式： 元素.calssName = '类名'；

如果我想添加类样式： 元素.calssName += ' 类名'；

box.className = 'div2';（替换样式）；

box.className += ' div2';（添加样式，注意div2前面的空格）；

\*容易忘记的知识点：

在表单元素（input）中有一个特殊的属性，叫做disabled，这个属性只要写上去就会生效。

原理：disabled属性只有两个值true和false，默认情况下不写就是false，写了就是true，如果想用js来改变这个属性的值，只需要赋值true和false即可。

想要禁止，则把disabled的值设置成true；

想要开启，则把disabled的值设置成false；

第一天重点知识点:

\*在事件处理函数中如何得知是哪一个按钮（元素）触发了本次事件？

为什么只能通过**this**来得知谁触发了本次事件，而不能用元素[i]呢？



解:为什么只能通过**this**来得知谁触发了本次事件,而不能用元素[ i ]，以上面图片中的代码为例，当程序执行到for循环语时，这个for循环的作用是为每个按钮（元素）注册一个事件，但并不会执行事件处理函数，事件处理函数是程序中最后执行的，执行条件是：整个程序走完并且人为的点击某个按钮才会执行，而当for循环执行结束后，全局变量i的值为数组的长度，以上面图片中的代码为例，当for循环执行结束后，全局变量i的值已经变成5，所以当用元素[ i ]来获取某个属性或设置某个属性时，是获取或设置不到某个元素的属性的，此时程序会报错，所以不能用元素[ i ]这个方式。而用this这个方式的就可以实现，当某个按钮（元素）调用这个方法（事件处理函数）时，this就指向这个按钮（元素），相当于this就是当前调用这个方法的按钮（元素），表面上可以这样理解，所以说，想要获取某个属性或设置某个属性时，只能用this来实现，简单的说，就是谁调用这个方法，this就指向谁。