# webApi第二天

1. 表单元素的属性操作
   1. value 用于大部分表单元素的内容获取(option除外)
   2. type 可以获取input标签的类型(输入框或复选框等)
   3. disabled 禁用属性
   4. checked 复选框选中属性
   5. selected 下拉菜单选中属性
2. 表单元素属性的案例
   1. 点击按钮禁用文本框 （练习disabled属性）
   2. 文本框模拟实现placeHolder

焦点： 页面中的鼠标的光标聚焦的位置

并不是所有的元素都有焦点 只有一些能够和用户参与交互的一些元素才有焦点

可以获取焦点 才会有焦点事件

比如 a input

* + 1. 这里有两个方法
       1. A.focus() 让A获取焦点
       2. A.blur() 让A失去焦点
    2. 有两个表单焦点事件
       1. A.onfoucus = funciton(){} 当A获取焦点的时候 做什么事情
       2. A.onblur = function() {} 当A失去焦点的时候 作什么事情
  1. 搜索框案例 （练习焦点事件）
  2. 下拉框选中案例 （练习selected属性）
  3. 兴趣爱好案例（练习checked属性）
  4. 账号长度验证案例（练习添加类className）

补充：input事件 （当文本框中的内容发生改变的时候 就会触发该事件）

可以用于随着文字内容的输入，在输入过程中及时判断字符长度规则

aInp[0].oninput = function() {

……

}

* 1. 全选案例（练习checked属性以及简单逻辑）

1. 样式操作：
   1. 通过style属性来设置样式

var box = document.getElementById('box');

box.style.width = '100px';

box.style.height = '100px';

box.style.backgroundColor = 'red';（js中遇到复合属性 font-size=> fontSize）

//border-top-color ==> borderTopColor

* 1. 通过定义类，在js中给元素添加类的方式实现设置样式

var box = document.getElementById('box');

box.className = 'show';

* 1. 自定义属性操作

getAttribute() 获取标签行内属性

setAttribute() 设置标签行内属性

removeAttribute() 移除标签行内属性

与element.属性的区别: 上述三个方法用于获取任意的行内属性

元素.常规属性 这种方式书写简便 但是不能获取页面上的自定义属性

上面三个方法 书写繁琐 但是可以操作行间的任意属性

1. 案例：
   1. tab选项卡切换

难点：如何解决事件处理函数里面的i获取不到的问题（通过建立自定义属性，将自己的下标存起来）

* 1. This指向问题总结一下
     1. 函数直接调用 this指向window
     2. 方法被对象所调用，那么方法里面的this指向该对象
     3. 构造函数被new所调用，函数里面的this其实是一个隐式的虚拟对象，初始化模型，在new调用过程中，根据this实例化出来一个实例对象
     4. 事件处理函数被事件所调用，函数里面的this指向触发该事件前面的那个元素

1. 自我感悟

<div id="box">box</div>

<script>

function fn(){

console.log(this);

}

var oBox = document.getElementById("box");

oBox.onclick=fn;//加括号之后点击为什么就没用了，不加括号是什么意思？

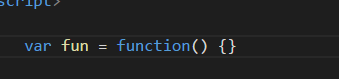
</script>

解释：oBox.onclick 意思就是 点击了之后调用 后面那个函数

要是你在后面 写（） 意思是直接调用函数了。

都没有点击 这个操作直接就执行这个函数了。。

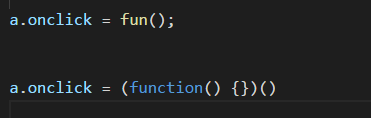
调用函数还有直接写一个函数名的情况？只有这种 事件这种情况



前面是一个函数名

后面是一个 匿名函数

所以你写个 函数名 就是跟 写个匿名函数一样，没有调用，你光写个匿名函数 也没有调用



这两个才是一个意思

只写一个函数名和写一个匿名函数都不会执行，点击之后会执行