大家好，我是蕴川 (王松博)，这是我加入Opportunity的第四周。以下是我本周学习的总结汇报，请大家指正。

一、工作学习内容

**1.事件的基本处理**

（1）事件的绑定一般采用v-on:来进行绑定，其简写为@

（2）必须指出，在vue中执行事件回调时，即用绑定事件进行绑定时，在vue实例中必须采用methods的定义方式，否则会进行不必要的数据代理。

感想：当使用vue进行事件绑定时，必须要采用把回调函数写在vue实例中的办法，否则无法在绑定容器中使用。此外，绑定的回调函数如果需要传参的话，直接在之后接（）即可，如果在此过程中需要保留event参数，用$event即可承接。

**2. 事件修饰的使用**

（1）当使用@符号进行事件绑定时，有必要对触发事件进行修饰时，vue提供了方便的方式。

（2）常用的几种修饰符：.self,.prevent,.stop.once。

（3）对于多种的修饰符来说，.self是只有event.target是当前操作的元素时才触发事件，.Prevent会阻止触发默认事件，.stop会阻止事件冒泡，.once会使事件只触发一次。

感想：其实对于事件修饰的相关内容，它对于之前的相关原生知识有一定的要求，在这种方面我又感受到了自己的不足。。。不过对于其相关知识的理解还是可以做到。

**3.键盘事件**

（1）keyup和kedown,分别指在按键的键位抬起和按下时触发事件。

（2）vue为常用的一些键盘按键提供了别名，如delete之类的。最大的区别其实可以看一看大小写。

（3）对于vue并没有提供别名的按键，可以通过—的方式进行自定义键位，注意为小写，再用eg：cap-slk进行使用。

（4）vue 提供了自定义的别名方式，vue.config.keycodes=自定义绑定的按键的数码。

感想： 对于tab键等特殊的键位，会与其本身所带有的事件进行冲突。对于键盘本身就要搭配其他键位的使用也会又类似的情况（如ctrl和fn）

二．问题与解决：

Q1：回到了原生知识的掌握不足上。对于很多的相关的问题，如在事件修饰的部分，感觉有点吃力。

A1:没有更好的办法了，只能去补其他的原生知识，在别的地方处理好了才能搞定vue。

Q2:学习时间投入。没错我已经把它放到问题里了。。。最近一周的时间投入确实不太高，写在这来警醒自己！

A2:努力！

## 三、下周工作计划

* 学习相关的原生知识，对于html的进一步理解。
* 学习css，然后把鸽了很久的监视属性弄上去。