**Passive Event Listeners - 被动事件监听器**

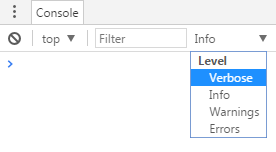
在写webapp页面的时候，Chrome 提醒

[Violation] Added non-passive **event** listener **to** a scroll-blocking 'touchstart' **event**. Consider marking **event** handler **as** 'passive' **to** make the page more responsive.

翻译过来如下：

违反：没有添加被动事件监听器来阻止'touchstart'事件，请考虑添加事件管理者'passive'，以使页面更加流畅。

出现如上提示这可能是由于console的过滤器选择了Verbose



* Verbose-冗长,就是事无具细的把所有log显示出来
* Info-显示开发者自定义的log及错误信息
* Warnings-显示危险信息
* Errors-显示错误信息

一般默认的是Info,不会提示这个提醒，虽然这并不影响代码的正常运行，但　是既然提醒了还是弄明白是什么原因的好一些。

这是因为Chrome51版本以后，Chrome增加了新的事件捕获机制－**Passive Event Listeners**

Passive Event Listeners就是告诉前页面内的事件监听器内部是否会调用preventDefault函数来阻止事件的默认行为，以便浏览器根据这个信息更好地做出决策来优化页面性能。当属性passive的值为true的时候，代表该监听器内部不会调用preventDefault函数来阻止默认滑动行为，Chrome浏览器称这类型的监听器为被动（passive）监听器。目前Chrome主要利用该特性来优化页面的滑动性能，所以Passive Event Listeners特性当前仅支持mousewheel/touch相关事件

以前的事件捕获代码如下：

document.addEventListener("click", fn, false/true)

第三个参数决定了fn函数是在冒泡还是捕获阶段触发。  
现在第三个参数不但可以是布尔值，还可是一个对象

**document**.addEventListener("**mousewheel**", **fn**, {passive: true})

由于passive对象只在Chrome浏览器中支持，所以这里需要做一个兼容处理

**var** passiveSupported = false;

**try** {

**var** options = Object.defineProperty({}, "passive", {

get: **function**() {

passiveSupported = true;

}

});

window.addEventListener("test", null, options);

} **catch**(err) {}

**function** **fn**() {

console.log("fn")

}

document.addEventListener("mousewheel", fn, passiveSupported ? { passive: true } : false)

再看Chrome调试会发现，Chrome已经不要提醒让头疼的提醒。