<https://blog.angular-university.io/angular-2-smart-components-vs-presentation-components-whats-the-difference-when-to-use-each-and-why/>

### 智能组件和演示组件之间的典型交互

我们在这里看到的示例非常频繁，我们让智能组件通过将数据注入到演示组件@Input，并通过接收演示组件可能触发的任何操作@Output。

在这种情况下，我们使用自定义lesson事件来表明我们已经在列表中选择了给定的课程。

@Output通过清晰定义的界面，使用演示组件仍与智能组件保持隔离：

课程列表演示组件仅知道它发出了一个事件，但不知道该事件的接收者是什么，或者接收者将如何响应该事件

主屏幕智能组件订阅了lesson自定义事件并对事件做出反应，但是它不知道是什么触发了该事件。用户是双击课程列表还是单击视图按钮？这对于智能组件是透明的。

这样就很简单了，这里可能出什么问题了？

### 拆分智能组件与演示组件的清晰方法？

这样，我们可能会得出结论，构建我们的应用程序就像使所有顶级组件成为智能组件一样简单，并通过使用本地表示组件树来构建它们。

但事实是，有时并不是那么简单，因为自定义事件lesson不会冒出来。因此，例如，如果您有一棵深的组件树，并且希望多个级别的组件知道该事件，则该事件将不会冒泡。

### 什么问题导致自定义事件不会冒泡呢？

可以说，我们不但在课程列表和home组件之间仅嵌套了一层，还具有以下几层：课程列表位于选项卡面板中的可折叠面板内部。

课程列表仍希望通知home组件已通过lesson事件选择了课程。但是中间的两个组件TabPanel和CollapsiblePanel非特定于应用程序的表示组件。

想象它们是Angular Material库的组件！

这些仅用于Presentation的组件不知道lesson事件，因此它们无法冒泡。那么，我们该如何实现呢？为什么自定义事件不能简单地冒出来呢？

### 为什么自定义事件不冒出来，就像点击之类的DOM事件那样?

这不是偶然的，这是为了避免类似于服务总线的解决方案，比如AngularJs中的$scope.$emit()和$scope.$broadcast()意外创建的事件soup场景而设计的。

这些类型的机制往往会在应用程序的不同位置之间创建紧密的依赖关系，而这些位置不应该相互意识到，而且事件会被多次触发，或者以只查看一个文件时不明显的顺序触发。

因此，表示组件的自定义事件将只对其父组件可见，而不会在树的更上层可见。

如果由于某种原因我们确实需要冒泡行为，我们仍然可以使用element.dispatchEvent()。但是大多数时候这不是我们想要实现的。

### 那么，如何解决选项卡面板方案中可折叠面板中课程列表的情况？