# 渐进式 JavaScript 框架

在我看来，渐进式代表的含义是：主张最少。  
每个框架都不可避免会有自己的一些特点，从而会对使用者有一定的要求，这些要求就是主张，主张有强有弱，它的强势程度会影响在业务开发中的使用方式。  
比如说，Angular，它两个版本都是强主张的，如果你用它，必须接受以下东西：  
- 必须使用它的模块机制- 必须使用它的依赖注入- 必须使用它的特殊形式定义组件（这一点每个视图框架都有，难以避免）

所以Angular是带有比较强的排它性的，如果你的应用不是从头开始，而是要不断考虑是否跟其他东西集成，这些主张会带来一些困扰。

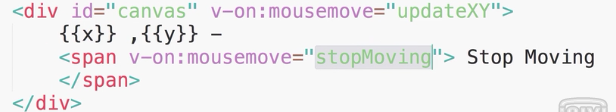
比如React，它也有一定程度的主张，它的主张主要是函数式编程的理念，比如说，你需要知道什么是副作用，什么是纯函数，如何隔离副作用。它的侵入性看似没有Angular那么强，主要因为它是软性侵入。

Vue可能有些方面是不如React，不如Angular，但它是渐进的，没有强主张，你可以在原有大系统的上面，把一两个组件改用它实现，当jQuery用；也可以整个用它全家桶开发，当Angular用；还可以用它的视图，搭配你自己设计的整个下层用。你可以在底层数据逻辑的地方用OO和设计模式的那套理念，也可以函数式，都可以，它只是个轻量视图而已，只做了自己该做的事，没有做不该做的事，仅此而已。  
渐进式的含义，我的理解是：没有多做职责之外的事。

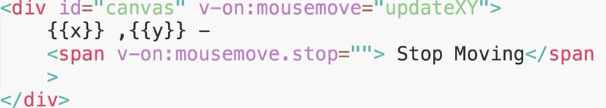
# 事件修饰符

参照vue官网， 事件处理->事件修饰符

-. 阻止事件冒泡







<div id="canvas" v-on:contextmenu="contextMenu"></div>

contextMenu(event) {

console.log(event);

event.preventDefault();

}

<div id="canvas" v-on:contextmenu.prevent="contextMenu"></div>

Vue1.0 vue2.0

1. 生命周期
2. 代码片段
3. For循环
4. 过滤器