# 寒冬winter的面试题

## 你对前端职业发展有何看法？

在我大一接触前端开发的时候，在论坛上和技术群里大家在讨论的还都是如何用div重构页面。后来大家逐渐的意识到语义化的重要性，开始审视自己的HTML标签**是否符合语义**。然后HTML5出现了，web app的兴起使得前端开发人员要做的工作不再局限于**切图和写页面**，更多的逻辑可以在客户端浏览器里运行，**javascript架构**得到了大部分开发者的重视。**前端代码分层**，每层各司其职变得尤为重要。MVC框架如backbone.js，MVVM框架如angular得到了越来越多开发者的青睐。

所以，从我短暂的前端开发之路的过程中来看，前端开发的发展越来越贴近「传统」的后端程序员，也许在未来的某个时间点上，前端和后端的隔阂将彻底的被打破，浏览器上的「工匠」将统称为web开发者。

但是，从我自己的角度上来看，我不希望自己的发展是随波逐流的。我接触前端开发的初衷，是前端开发可以将设计师脑海中的蓝图转化为现实。所以我更加希望自己的前端开发之路能越来越深入的提高用户体验。

这个方向更加的偏向设计师的工作——把展现给用户的界面精雕细琢，精确到每一个像素；更加注重文字的排版；与设计师的沟通要比跟后端程序员沟通的次数要多，跟设计师商讨每一个细节；页面元素各种状态以及页面在处于动态时，一些元素的效果。

在前端响应式开发大行其道的今天，如何把多终端的前端页面处理好也是前端职业发展不可绕过的一关，不同设备下不同元素的位置、布局、显隐虽有迹可循，但仔细琢磨后，又觉得有大道理没有被发现。前端探索的道路一直如小马过河，每个开发人员都有过河的资格和可能。

也许从大众角度来说，前端程序员未来的路越来越宽。

而对我来说，前端的路会越来越长。

## 前端和后端程序员应该如何合作？

终于挑了个闲暇时间来答这个问题。

身在一个小到芝麻粒儿一样的公司里，前后端在开发和调试过程中的交流比跟家里的媳妇儿交流的还要多（虽然我现在是跟一个后台哥合租）。

我已经经历过的开发过程中，前后端的配合和讨论多集中于：

* 数据交换时的约定
* 什么样的逻辑可以拿到前端进行处理，在安全可靠健壮的同时减少服务器端的压力。
* 数据交换方式（json,字符串,跳转页面时的参数传递等）

前端程序员就像一个「采购员」，后端程序员扮演的则是「供货商」。二者不必一定要知道对方是如何满足自己需求的，只要通过最快捷最方便的方式拿到自己想要的就可以皆大欢喜。但是任何一个有「野心」的前端程序员都会有「越界探索」的想法，探一探后端程序是如何运作的，它的架构有什么可以在前端开发中借鉴的。后端程序员也会有同样的「野心」，所以两种开发人员在合作的过程亦能学到彼此身上的知识，从而更加的了解对方所身处的环境，更能设身处地的为对方提供便利，形成一个良性循环。

## 讲几个你在项目中解决的让你印象最深的问题（难、匪夷所思、解决方案有趣都可以）

1.因为IOS6系统中safari内核浏览器对于ajax的post数据缓存问题导致的客户端与服务器端信息不对称。解决方案是在ajax请求头中关闭缓存。

2.使用jquery的dialog插件时，发现在IE7下，a标签里的javascript:void(0)会触发beforeunload事件。解决方案是用###代替javascript:void(0)。

**在JavaScript面向对象方面，你有什么体会和实践？**

任何技术都不应该为了实现而实现。在刚开始接触面向对象的时候，想把所有代码都写成面向对象的。现在想想，有一些功能和逻辑如果通过面向对象来实现，效率不一定会提高，甚至会降低很多。

1. 继承：javascript的继承使用链式继承，一个类与其父类和子类的关系都是通过原型链来追溯的。javascript中的函数是引用类型，这就说明如果一个父类的多个子类中的某一个子类通过引用修改了父类的函数，所有的子类都会受到影响。
2. 多态：Javascript是不支持多态的，但是我们可以通过一些设计模式来模仿多态，比如工厂方法，在函数入口判断传入参数调用不同的方法。调用某一个类的方法，会在原型链中该类的位置开始，逐渐向父类的方向查找，所以「重载」也就可以实现了，如果子类有该方法，就会调用子类的方法，若没有，会向上查找。
3. 封装：函数的作用域很好的保证了函数内部的变量不会被外部访问到。私有状态的变量和属性可以通过将变量闭包在构造函数内完成。这样，私有状态的变量会在每个对象上被创建，而不是从原型链进行继承。

function foo(){

this.pri = function(){/\*xxx\*/}

}

foo.prototype.pub = function(){

xxx

}

**谈谈JavaScript中的闭包，以及你的实践。**

我已经忘了闭包是什么，因为每一个函数都在闭包。它是私有变量的一个很好的创建方式。外部函数能够访问到这个变量的值，而缺点是，外部函数在引用这个值的时候注定了它无法被回收。所以，闭包使用过多会使越来越多的变量无法通过内存回收。

**说说 http://m.taobao.com 前端做的最烂的地方，以及你的改进。**  
1.大多数元素触摸时没有反馈。

其实已经做的很棒了，好难挑。

**谈几个有趣的html标签，以及它们的语义。**

1.img、iframe 这俩标签的语义我就不说了，前者是XSS攻击、盗取cookie和refer的利器。后者是跨域利器。

2.canvas标签，语义是镶嵌在网页中的画布元素。最近在做毕业设计，用Canvas做游戏，觉得这个标签包含了无数可能。

**讲一讲CSS的position/float/display都有哪些取值，它们相互叠加时的行为都是什么？**

position: static,absolute,relative,fixed(unavailable in IE6).

float: left,right,none.

display: none,block,inline,table-cell,inline-block,table,list-item...

**说几个你觉得有趣的CSS3选择器，以及他们有趣的用法。**

**自己问自己一道代表你水平的面试题，然后回答。**