## Vue和angular的区别

Vue相对于angular比较灵活

Vue是相对于html为模板进行操作，而angular是在原有的基础上添加功能

Vue性能好，容易优化，angular因为使用脏值检查（容易触发，运行次数多）

1. vue仅仅是mvvm中的view层，只是一个如jquery般的工具库，而不是框架，而angular而是mvvm框架。
2. vue的双向邦定是基于ES5 中的 getter/setter来实现的，而angular而是由自己实现一套模版编译规则，需要进行所谓的“脏”检查，vue则不需要。因此，vue在性能上更高效，但是代价是对于ie9以下的浏览器无法支持。
3. vue需要提供一个el对象进行实例化，后续的所有作用范围也是在el对象之下，而angular而是整个html页面。一个页面，可以有多个vue实例，而angular好像不是这么玩的。
4. vue真的很容易上手，学习成本相对低，不过可以参考的资料不是很丰富，官方文档比较简单，缺少全面的使用案例。高级的用法，需要自己去研究源码，至少目前是这样。
5. 相对于angular来说，vue.js更加简洁、更易于理解的API，能够快速上手

## Bootstrap应用过哪些组件

## vue生命周期

beforecreate：el 和 data 并未初始化 ,也就是所有都未初始化

created:完成了 data 数据的初始化，也就是data和data里的数据已经可以访问到了 el没有，

beforeMount：开始了 el的挂载，但数据并未渲染在页面上，也就是站了个位置，也就是虚拟dom 。 data 初始化

mounted ：完成挂载，页面也可以刷新数据

beforeUpdate：更新之前的数据，也就是页面中有数据或data中有数据变化时，这个就会触发

Updata：更新之后的事件函数，更新页面或data中的数据

beforeDestroy：实例销毁之前调用。在这一步，vue实例仍然完全可用。

Destroy：Vue 实例销毁后调用。调用后，Vue 实例指示的所有东西都会解绑定，所

有的事件监听器会被移除，所有的子实例也会被销毁。 该钩子在服务器端渲 染期间不被调用。

## 跨域有哪些方法

1 jsonp 动态创建script标签，通过src属性请求数据

2 通过**document.domain ,把他们的域名设为同一个域名**

**3 通过 document.name ，这个属性是所有页面都可以访问的，给他传入值，其他页面**

4反向服务器代理

是指以代理服务器来接受Internet上的连接请求，然后将请求转发给内部网络上的服务器；并将从服务器上得到的结果返回给Internet上请求连接的客户端，此时代理服务器对外就表现为一个服务器。

5 cors

 CORS定义一种跨域访问的机制，可以让AJAX实现跨域访问。CORS 允许一个域上的网络应用向另一个域提交跨域 AJAX 请求。实现此功能非常简单，只需由服务器发送一个响应标头即可。

## 什么是angular服务

在angular中 服务是一个函数或对象；为了不再给controller增加负担，就出现了服务

Angular 服务就是为angular组件啊 控制器啊 提供方法，数据的一个区域，每个组件模块都可以调用

## Vue用过哪些插件

Vue.router 路由

Vue.resource http请求

Vue-cil、 手脚夹 也就是搭建框架的

## 用那些实现移动端发

1用响应式开发

判断是视口的大小来进行布局

2 模块化开发

## ES6的特性

1 块级作用域 let声明变量 const声明常量

2 箭头函数

3 引入导出es6语法

1)**箭头操作符** inputs=>outputs: 操作符左边是输入的参数，而右边则是进行的操作以及返回的值

　　2) **支持类**, 引入了**class**关键字. ES6提供的类实际上就是JS原型模式的包装

　　3) **增强的对象字面量**.

　　　　1. 可以在对象字面量中**定义原型  \_\_proto\_\_**: xxx  //设置其原型为xxx,相当于继承xxx

　　　　2. 定义方法可以**不用function**关键字

　　　　3. **直接调用父类方法**

　　4) **字符串模板**: ES6中允许使用反引号 ` 来创建字符串，此种方法创建的字符串里面可以包含由美元符号加花括号包裹的变量${vraible}。

　　5) **自动解析数组或对象中的值**。比如若一个函数要返回多个值，常规的做法是返回一个对象，将每个值做为这个对象的属性返回。但在ES6中，利用解构这一特性，可以直接返回一个数组，然后数组中的值会自动被解析到对应接收该值的变量中。

　　6) **默认参数值:** 现在可以在定义函数的时候指定参数的默认值了，而不用像以前那样通过逻辑或操作符来达到目的了。

　　7) **不定参数**是在函数中使用命名参数同时接收不定数量的未命名参数。在以前的JavaScript代码中我们可以通过arguments变量来达到这一目的。不定参数的格式是三个句点后跟代表所有不定参数的变量名。比如下面这个例子中，…x代表了所有传入add函数的参数。

　　8) **拓展参数**则是另一种形式的语法糖，它允许传递数组或者类数组直接做为函数的参数而不用通过apply。

　　9) **let和const关键字:** 可以把let看成var，只是它定义的变量被限定在了特定范围内才能使用，而离开这个范围则无效。const则很直观，用来定义常量，即无法被更改值的变量。

　　10)**for of值遍历**每次循环它提供的不是序号而是值。

　　11) **iterator, generator**

　　12) **模块**

　　13) **Map, Set, WeakMap, WeakSet**

　　14) **Proxy**可以监听对象身上发生了什么事情，并在这些事情发生后执行一些相应的操作。一下子让我们对一个对象有了很强的追踪能力，同时在数据绑定方面也很有用处。

　　15)**Symbols**Symbol 通过调用symbol函数产生，它接收一个可选的名字参数，该函数返回的symbol是唯一的。之后就可以用这个返回值做为对象的键了。Symbol还可以用来创建私有属性，外部无法直接访问由symbol做为键的属性值。

　　16) **Math, Number, String, Object的新API**

17) **Promises**是处理异步操作的一种模式

如何让css只在当前组件中起作用

在每一个vue组件中都可以定义各自的css，js加上scoped就可以

<style scoped></style>

## 指令keep-alive

<component :is='curremtView' keep-alive></component>

<keep-alive>

<component></component>

</keep-alive>

当使用Keep-Alive模式（又称持久连接、连接重用）时，Keep-Alive功能使客户端到服务器端的连接持续有效，当出现对服务器的后继请求时，Keep-Alive功能避免了建立或者重

（服务器在请求完数据后会制动关闭）

新建立连接。(也就是说keep-alive可以在服务器请求数据以后不要在次请求，keep-alive会帮你缓存)

## 做过哪些浏览器兼容

IE （trident），火狐（gecko），谷歌（blink）；苹果（webkit）

常见bug

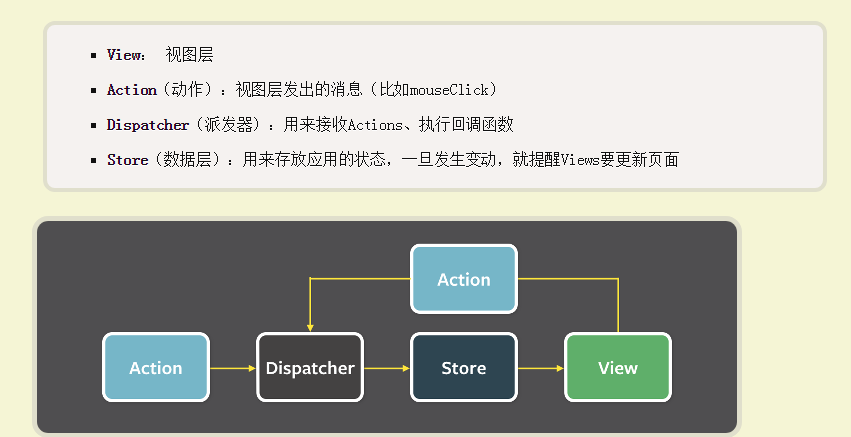
双倍margin display ： inlien

图片下的白边 display ： block 也可以浮动

## **Vue.js 和 React.js 有什么区别?**

React 支持服务端渲染，采用Flux结构（其实vue也是）

Flux 是一种架构思想，专门解决软件的结构问题。它跟[MVC 架构](http://www.ruanyifeng.com/blog/2007/11/mvc.html" \t "http://www.ruanyifeng.com/blog/2016/01/_blank)是同一类东西，但是更加[简单和清晰](http://www.infoq.com/news/2014/05/facebook-mvc-flux" \t "http://www.ruanyifeng.com/blog/2016/01/_blank)



在超大量数据的首屏渲染速度上，React 有一定优势，因为 Vue 的渲染机制启动时候要做的工作比较多，而且 React 支持服务端渲染。

个人认为vue和react.js最大的不同之处在于它们对DOM的渲染方式不同，vue可以直接在vue文件中使用html标签，数据绑定时类似angular，可以进行条件渲染，而react.js则采用了jsx语法（把html和css写进了javascript），运用虚拟DOM 的概念进行DOM对页面元素进行渲染，

React.js 和 Vue.js 确实有一些相似——它们都提供数据驱动、可组合搭建的视图组件。然而，它们的内部实现是完全不同的。React 是基于 Virtual DOM——一种在内存中描述 DOM 树状态的数据结构。React 中的数据通常被看作是不可变的，而 DOM 操作则是通过 Virtual DOM 的 diff 来计算的。与之相比，Vue.js 中的数据默认是可变的，而数据的变更会直接出发对应的 DOM 更新。相比于 Virtual DOM，Vue.js 使用实际的 DOM 作为模板，并且保持对真实节点的引用来进行数据绑定。 Virtual DOM 提供了一个函数式的描述视图的方法，这很 cool。因为它不使用数据观察机制，每次更新都会重新渲染整个应用，因此从定义上保证了视图通与数据的同步。它也开辟了 JavaScript 同构应用的可能性。

## **[你做的页面在哪些浏览器测试过？这些浏览器的内核是什么？经常遇到的浏览器的兼容性有哪些？怎么会出现？解决方法是什么？（至少说3个）](http://www.cnblogs.com/luckyXcc/p/5831094.html)**

IE内核浏览器：360，傲游，搜狗，世界之窗，腾讯TT

非IE内核浏览器：firefox opera safari chrome

1.就是ie6双倍边距的问题，在使用了float的情况下，不管是向左还是向右都会出现双倍，最简单的解决方法就是用display:inline;加到css里面去。

2.文字本身的大小不兼容。同样是font-size:14px的宋体文字，在不同浏览器下占的空间是不一样的，ie下实际占高16px，下留白3px，ff下实际占高17px，上留白1px，下留白3px，opera下就更不一样了。解决方案：给文字设定 line-height 。确保所有文字都有默认的 line-height 值。这点很重要，在高度上我们不能容忍1px 的差异。

3. ff(火狐)下容器高度限定，即容器定义了height之后，容器边框的外形就确定了，不会被内容撑大，而ie下是会被内容撑大，高度限定失效。所以不要轻易给容器定义height。

4. 还讨论内容撑破容器问题，横向上的。如果float 容器未定义宽度，ff下内容会尽可能撑开容器宽度，ie下则会优先考虑内容折行。故，内容可能撑破的浮动容器需要定义width。

5. 浮动的清除，ff下不清除浮动是不行的。

6. mirrormargin bug，当外层元素内有float元素时，外层元素如定义margin-top:14px，将自动生成margin-bottom:14px。padding也会出现类似问题，都是ie6下的特产，该类bug 出现的情况较为复杂，远不只这一种出现条件，还没系统整理。解决方案：外层元素设定border 或 设定float。

7. 吞吃现象，限于篇幅，我就不展开了。还是ie6，上下两个div，上面的div设置背景，却发现下面没有设置背景的div 也有了背景，这就是吞吃现象。对应上面的背景吞吃现象，还有滚动下边框缺失的现象。解决方案：使用zoom:1。这个zoom好象是专门为解决ie6 bug而生的。

## 前端开发的优化问题：

1 减少http请求，或合并

2 压缩js.css代码，

3使用懒加载，优先加载当前页面

4 图片尽量使用符合的图片格式，小图可以用精灵图或矢量图

5 减少全局变量 ，常用方法打包

6 利用本地储存数据，加快页面再次请求速度

7 减少dom操作

　 (1) 减少http请求次数：css spirit,data uri

(2) JS，CSS源码压缩

(3) 前端模板 JS+数据，减少由于HTML标签导致的带宽浪费，前端用变量保存AJAX请求结果，每次操作本地变量，不用请求，减少请求次数

(4) 用innerHTML代替DOM操作，减少DOM操作次数，优化javascript性能

(5) 用setTimeout来避免页面失去响应

(6) 用hash-table来优化查找

(7) 当需要设置的样式很多时设置className而不是直接操作style

(8) 少用全局变量

(9) 缓存DOM节点查找的结果

(10) 避免使用CSS Expression

(11) 图片预载

(12) 避免在页面的主体布局中使用table，table要等其中的内容完全下载之后才会显示出来，显示比div+css布局慢

## 你会哪些算法

冒泡 去重 出现次数 随机生成自定义字符 快速排列

## 关于项目的追问

你的项目用的哪些插件

你写项目用多少天

你上一家公司在哪里

在公司里做什么

接外面的活吗

你认为这个项目有哪些优点哪些缺点

## 状态码

100——客户必须继续发出请求

101——客户要求服务器根据请求转换HTTP协议版本

200——交易成功

201——提示知道新文件的URL

202——接受和处理、但处理未完成

203——返回信息不确定或不完整

204——请求收到，但返回信息为空

205——服务器完成了请求，用户代理必须复位当前已经浏览过的文件

206——服务器已经完成了部分用户的GET请求

300——请求的资源可在多处得到

301——删除请求数据

302——在其他地址发现了请求数据

303——建议客户访问其他URL或访问方式

304——客户端已经执行了GET，但文件未变化

305——请求的资源必须从服务器指定的地址得到

306——前一版本HTTP中使用的代码，现行版本中不再使用

307——申明请求的资源临时性删除

400——错误请求，如语法错误

401——请求授权失败

402——保留有效ChargeTo头响应

403——请求不允许

404——没有发现文件、查询或URl

405——用户在Request-Line字段定义的方法不允许

406——根据用户发送的Accept拖，请求资源不可访问

407——类似401，用户必须首先在代理服务器上得到授权

408——客户端没有在用户指定的饿时间内完成请求

409——对当前资源状态，请求不能完成

410——服务器上不再有此资源且无进一步的参考地址

411——服务器拒绝用户定义的Content-Length属性请求

412——一个或多个请求头字段在当前请求中错误

413——请求的资源大于服务器允许的大小

414——请求的资源URL长于服务器允许的长度

415——请求资源不支持请求项目格式

416——请求中包含Range请求头字段，在当前请求资源范围内没有range指示值，请求也不包含If-Range请求头字段

417——服务器不满足请求Expect头字段指定的期望值，如果是代理服务器，可能是下一级服务器不能满足请求

500——服务器产生内部错误

501——服务器不支持请求的函数

502——服务器暂时不可用，有时是为了防止发生系统过载

503——服务器过载或暂停维修

504——关口过载，服务器使用另一个关口或服务来响应用户，等待时间设定值较长

505——服务器不支持或拒绝支请求头中指定的HTTP版本

## 什么是reflow（回流） 与repaint（重绘）,如何减少

reflow指的是计算页面布局。某个节点reflow时会重新计算节点的尺寸和位置，而且还有可能触发其子节点、祖先节点和页面上的其他节点reflow。在这之后再触发一次repaint

repiant或者redraw遍历所有的节点检测各节点的可见性、颜色、轮廓等可见的样式属性，然后根据检测的结果更新页面的相应部分

下面情况会导致reflow发生

**1：改变窗口大小**

**2：改变文字大小**

**3：内容的改变，如用户在输入框中敲字**

**4：激活伪类，如:hover**

**5：操作class属性**

**6：脚本操作DOM**

**7：计算offsetWidth和offsetHeight**

**8：设置style属性**

那么为了减少回流要注意哪些方式呢？

**1：不要通过父级来改变子元素样式，最好直接改变子元素样式，改变子元素样式尽可能不要影响父元素和兄弟元素的大小和尺寸**

**2：尽量通过class来设计元素样式，切忌用style**

**3：实现元素的动画，对于经常要进行回流的组件，要抽离出来，它的position属性应当设为fixed或absolute**

**4：权衡速度的平滑。比如实现一个动画，以1个像素为单位移动这样最平滑，但reflow就会过于频繁，CPU很快就会被完全占用。如果以3个像素为单位移动就会好很多。**

**5：不要用tables布局的另一个原因就是tables中某个元素一旦触发reflow就会导致table里所有的其它元素reflow。在适合用table的场合，可以设置table-layout为auto或fixed，**

**6：这样可以让table一行一行的渲染，这种做法也是为了限制reflow的影响范围。**

**7：css里不要有表达式expression**

**8：减少不必要的 DOM 层级（DOM depth）。改变 DOM 树中的一级会导致所有层级的改变，上至根部，下至被改变节点的子节点。这导致大量时间耗费在执行 reflow 上面。**

**9：避免不必要的复杂的 CSS 选择器，尤其是后代选择器（descendant selectors），因为为了匹配选择器将耗费更多的 CPU。**

**10: 尽量不要过多的频繁的去增加，修改，删除元素，因为这可能会频繁的导致页面reflow，可以先把该dom节点抽离到内存中进行复杂的操作然后再display到页面上。**

## cookie 和session 的区别：

1、cookie数据存放在客户的浏览器上，session数据放在服务器上。

2、cookie不是很安全，别人可以分析存放在本地的COOKIE并进行COOKIE欺骗  
   考虑到安全应当使用session。

3、session会在一定时间内保存在服务器上。当访问增多，会比较占用你服务器的性能  
   考虑到减轻服务器性能方面，应当使用COOKIE。

4、单个cookie保存的数据不能超过4K，很多浏览器都限制一个站点最多保存20个cookie。

## meta viewport原理

Meta是苹果公司发明的

Viewport视口

视觉视口（看到的视口）

布局视口（屏幕的大小）

理想视口（根据屏幕大小判断视口）

<meta name="viewport"  
 content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">

**1、width :** 控制viewport的大小，可以指定一个值，如600， 或者特殊的值，如device-width为设备的宽度（单位是缩放为100%的CSS的像素）

**2、height :**和width相对应，指定高度

**3、initial-scale :** 初始缩放比例，页面第一次加载时的缩放比例

注意：initial默认值为1 width为320， 当值为2的时候，宽为160，

它是根据分辨率来判断的，像素不变，只是分辨率增加了，原来一个单位可以放两像素了

**4、maximum-scale :** 允许用户缩放到的最大比例，范围从0到10.0

**5、minimum-scale :** 允许用户缩放到的最小比例，范围从0到10.0

### 移动端请求头

<meta charset="UTF-8">  
<link rel="shortcut icon" href="favicon.ico" type="image/vnd.microsoft.icon" />  
<meta name="Generator" content="LIXIPHP (http://lixiphp.com)" />  
<meta name="viewport" content="width=device-width, initial-scale=1">  
<meta name="MobileOptimized" content="width">  
<meta name="HandheldFriendly" content="true">  
<meta name="apple-mobile-web-app-capable" content="yes">  
<meta http-equiv="cleartype" content="on">  
<meta http-equiv="X-UA-Compatible" content="IE=edge, chrome=1">

1、编码Unicode UTF-8

2、浏览器图标地址和类型

3、Generator 生产者

4、网页缩放比例为1:1

5、控制手机布局，width默认为屏幕大小，也可以为具体数字如240、480，Windows Mobile推荐设置为240

6、控制手机放大缩小网页显示

7、控制全屏显示(Apple Safar)

8、ClearType主要是针对LCD液晶显示器设计，可提高文字的清晰度。

9、兼容性

## 什么是内存泄露

不在用到的内存没有得到及时的释放就是内存泄露，也就是声明的变量被闭包存储，无法被垃圾回收机制回收，也不被引用，如果想要解决需要var a = null

## **xhtml和html有什么区别** HTML是一种基本的WEB网页设计语言，XHTML是一个基于XML的置标语言 最主要的不同： XHTML 元素必须被正确地嵌套。 XHTML 元素必须被关闭。 标签名必须用小写字母。 XHTML 文档必须拥有根元素。

## 介绍一下你对浏览器内核的理解？

主要分成两部分：渲染引擎(layout engineer或Rendering Engine)和JS引擎。

　　渲染引擎：负责取得网页的内容（HTML、XML、图像等等）、整理讯息（例如加入CSS等），以及计算网页的显示方式，然后会输出至显示器或打印机。浏览器的内核的不同对于网页的语法解释会有不同，所以渲染的效果也不相同。所有网页浏览器、电子邮件客户端以及其它需要编辑、显示网络内容的应用程序都需要内核。

　　JS引擎则：解析和执行javascript来实现网页的动态效果。

最开始渲染引擎和JS引擎并没有区分的很明确，后来JS引擎越来越独立，内核就倾向于只指渲染引擎。

## 浏览器对localStorage的兼容问题 ；还有哪些本地存储？

localStorage是浏览器用于本地储存的一个对象，浏览器支持还是不错的，ie 从IE8开始支持；

cookie

cookie会随着每次HTTP请求头信息一起发送，无形中增加了网络流量，另外，cookie能存储的数据容量有限，根据浏览器类型不同而不同，IE6大约只能存储2K。

Flash ShareObject

这种方式能能解决上面提到的cookie存储的两个弊端，而且能够跨浏览器，应该说是目前最好的本地存储方案。不过，需要在页面中插入一个Flash，当浏览器没有安装Flash控件时就不能用了。所幸的是，没有安装Flash的用户极少。

缺点：需要安装Flash插件。

Google Gear

Google开发出的一种本地存储技术。

缺点：需要安装Gear组件。

userData

IE浏览器可以使用userData来存储数据，容量可达到640K，这种方案是很可靠的，不需要安装额外的插件。缺点：它仅在IE下有效。

sessionStorage

使用于Firefox2+的火狐浏览器，用这种方式存储的数据仅窗口级别有效，同一个窗口（或者Tab）页面刷新或者跳转，都能获取到本地存储的数据，当新开窗口或者页面时，原来的数据就失效了。

缺点：IE不支持、不能实现数据的持久保存。

globalStorage

使用于Firefox2+的火狐浏览器，类似于IE的userData。

localStorage

localStorage是Web Storage互联网存储规范中的一部分，现在在Firefox 3.5、Safari 4和IE8中得到支持。

缺点：低版本浏览器不支持。

结论：  
Flash shareobject是不错的选择，如果你不想在页面上嵌入Flash，可以结合使用userData(IE6+)和globalStorage(Firefox2+)和localStorage(chrome3+)实现跨浏览器。

## 怎样解决移动端1px这个问题

border-1px {

position: relative;

&:after{

display: block;

position: absolute;

left: 0;

bottom: 0;

width: 100%;

border-top: 1px solid #000;

content: '';

}

}

## Vue中响应对象的添加删除方法

用set (添加) delete（删除）



## Js和其他语言有什么不同

Js是一种弱类型（函数的参数个数不定），动态类型，基于原型的一种语言

ECMAscript提供核心功能

Js的特点

1 js是一种解释型脚本语言

2 可以和html 发生交互行为

3 可以镶嵌在html，也可以引用

4 大多数平台都支持

怎样调整工作中遇到的错误（他给的建议抓包 查看报文，在检查的Network - > headers -> status-code）

你请求的报文是什么400 401 404

200 OK：客户端请求成功。

　　 400 Bad Request：客户端请求有语法错误，不能被服务器所理解。

　　 401 Unauthorized：请求未经授权，这个状态代码必须和 WWW-Authenticate报头域一起使用。

　　 403 Forbidden：服务器收到请求，但是拒绝提供服务。

　　 404 Not Found：请求资源不存在，举个例子：输入了错误的URL。

　　 500 Internal Server Error：服务器发生不可预期的错误。

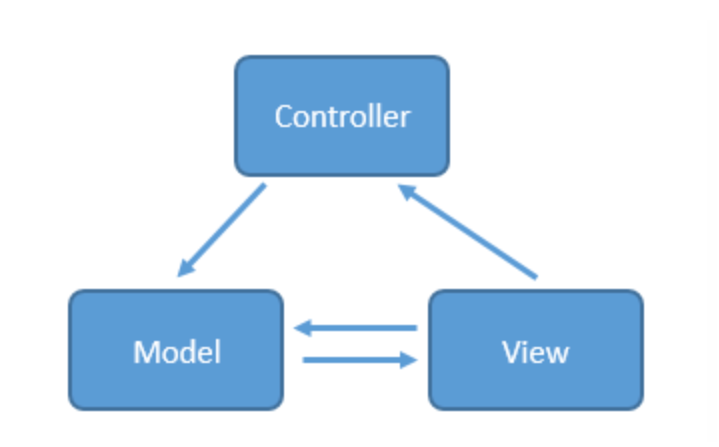
## Mvc和mvvm的区别

MVC框架 (model view controller)

通过controller的控制去操作model层的数据，返回给view层去展示

工作原理：

当用户发事件到的时候，view层会发送指令到controller层，接着controller去通知model层更新数据，model层更新完数据以后直接显示在view层上，这就是MVC的工作原理

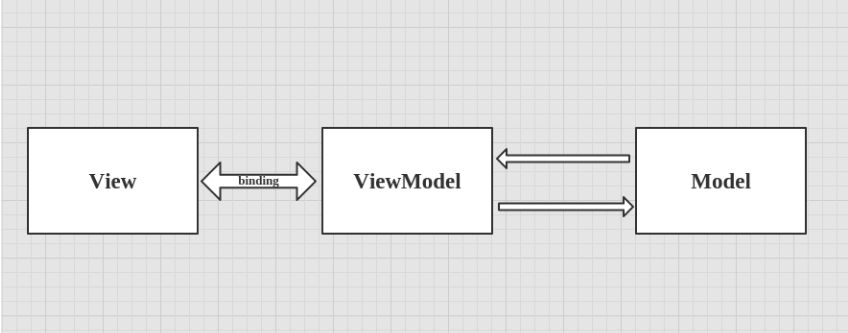


MVVM （model view viewModel）最早是微软提出来的

对数据储存 页面展示 对数据进行加工

View层和viewModel层相互绑定的,根据信息去model中更新数据，然后返回给model

ViewModel是Vue的核心，它是Vue的一个实例。Vue实例时作用域某个HTML元素上的，这个HTML元素可以是body，也可以是某个id所指代的元素。 DOM Listeners和Data Bindings是实现双向绑定的关键。DOM Listeners监听页面所有View层DOM元素的变化，当发生变化，Model层的数据随之变化；Data Bindings监听Model层的数据，当数据发生变化，View层的DOM元素随之变化。



## 前后端分离

也就是说SPA单页应用程序

优点 页面刷新无跳转 无白屏 缺点 不利于搜索引擎优化

我说我们公司一般是json格式，

职责划分

后端 (JAVA)，专注于服务层数据格式、数据稳定业务逻辑 MVC

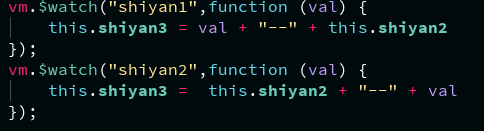
前端，专注于UI层 控只逻辑、渲染逻辑 交互、用户体验 MVC

## computed和watch有什么区别，用computed做过什么

computed



watch



Watch必须监听到具体元素，而computed只要包括（也就是用到了的数据发生变化时）

Computed的元素（如：shiyan4）在data中不存在，若存在则不会起作用

## ajax你了解多少，为什么要跨域？

Ajax 里的get post jsonp

为什么要跨域

这里考到了同源策略，在同一个路径下，可以用get，post请求

如果是其他地址的数据，就要用到跨域了

## Ajax的优缺点

**Ajax的优点：**  
    1、 最大的一点是页面无刷新，在页面内与服务器通信，给用户的体验非常好。  
    2、 使用异步方式与服务器通信，不需要打断用户的操作，具有更加迅速的相应能力。  
    3、 可以把从前一些服务器负担的工作转嫁到客户端，利用客户端限制的能力来处理， 减轻服务器和带宽的负担，节约空间和宽带租用成本，并且减轻服务器的负担，Ajax 的原则是"按需取数据"，可以最大承德的减少冗余的请求和相应对服务器造成的负担。  
    4、 基于标准化的并被广泛支持的技术，不需要下载插件或者小程序。

**Ajax的缺：**

1 ajax破坏了后退按钮，破坏浏览器机制。

2 不安全，会暴露数据

3 对浏览器搜索引擎支持较弱

4 一些手持设备（如手机、PDA等）现在还不能很好的支持ajax，

5 破坏了程序的异常机制

## Vuex是什么，什么时候会用到

Vuex专门为vue开发的状态管理模式，是一个框架

如果您不打算开发大型单页应用，使用 Vuex 可能是繁琐冗余的。确实是如此——如果您的应用够简单，您最好不要使用 Vuex。一个简单的 **[global event bus](https://cn.vuejs.org/v2/guide/components.html" \l "非父子组件通信" \t "https://vuex.vuejs.org/zh-cn/_blank)** 就足够您所需了。但是，如果您需要构建是一个中大型单页应用，您很可能会考虑如何更好地在组件外部管理状态，Vuex 将会成为自然而然的选择

## Angular2的新特性：

　　1、移除了 controller +$scope的设计，改用组件式开发。（更容易理解和上手）

　　2、性能更好（渲染更快，变化检测效率更高）

　　3、优先为移动应用设计（Angular Mobile Toolkit ）

　　4、更加贴合未来的标准（如es6/7、WebComponent）

## **[HTML页面加载和解析流程](http://blog.csdn.net/xifeijian/article/details/10813339" \t "http://blog.csdn.net/xiaozhuxmen/article/details/_blank)** 1. 用户输入网址（假设是个html页面，并且是第一次访问），浏览器向服务器发出请求，服务器返回html文件；  2. 浏览器开始载入html代码，发现＜head＞标签内有一个＜link＞标签引用外部CSS文件；  3. 浏览器又发出CSS文件的请求，服务器返回这个CSS文件；  4. 浏览器继续载入html中＜body＞部分的代码，并且CSS文件已经拿到手了，可以开始渲染页面了；  5. 浏览器在代码中发现一个＜img＞标签引用了一张图片，向服务器发出请求。此时浏览器不会等到图片下载完，而是继续渲染后面的代码；  6. 服务器返回图片文件，由于图片占用了一定面积，影响了后面段落的排布，因此浏览器需要回过头来重新渲染这部分代码；  7. 浏览器发现了一个包含一行Javascript代码的＜script＞标签，赶快运行它；  8. Javascript脚本执行了这条语句，它命令浏览器隐藏掉代码中的某个＜div＞ （style.display=”none”）。突然少了这么一个元素，浏览器不得不重新渲染这部分代码；  9. 终于等到了＜/html＞的到来，浏览器泪流满面……  10. 等等，还没完，用户点了一下界面中的“换肤”按钮，Javascript让浏览器换了一下＜link＞标签的CSS路径；  11. 浏览器召集了在座的各位＜div＞＜span＞＜ul＞＜li＞们，“大伙儿收拾收拾行李，咱得重新来过……”，浏览器向服务器请求了新的CSS文件，重新渲染页面。

## DOM和BOM的区别

Dom主要负责document，也就是多html进行操作

Bom主要对应window，

## 反问

1. 问一下之前没有回答上来的问题
2. 你们公司目前项目是什么状态，进度如何
3. 你们的技术站是什么
4. 你们公司有没有定期培训学习
5. 你们公司福利待遇怎样
6. 你们公司现在做的项目兼容哪些浏览器

## 关于上家公司的问题

1. 你们公司的工作流程（从开始到上线）

项目经理提出什么需求，就做什么

产品出需求，同时出ui页面

接口处接口， 前端开始写页面开发  
 提交测试  
 上线

1. 你们公司的主要做哪些项目
2. 项目你负责那一块
3. 你工作常遇到哪些问题（我觉得最好不要说兼容性问题）

主要是一些兼容性问题(大牛说能决绝的都不是问题)

接口问题

1. 工作实验哪些浏览器兼容

我一般不问那么笼统的问题。要问的话（通常前提是面试对象在简历里写了或在交谈中提及自己熟悉/擅长/精通兼容性处理），我会让他讲讲自己遇到的最严重的兼容性case，需求是什么，一开始怎么做的，怎么发现有问题，怎么找到问题的原因，有哪些解决方案，最后选了哪一种，为什么选这种方式，这个过程里学到了什么，再做一次类似case会有什么改变，将来怎么避免等等。

作者：贺师俊  
链接：https://www.zhihu.com/question/28954391/answer/42681493  
来源：知乎  
著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。

1. 手机兼容性问题

1、流式布局

     流式布局的解决方案有不少弊端，虽然可以让各种屏幕都适配，但是显示的效果差强人意，因为只有几个尺寸的手机能够完美的显示出ui设计和交互设计师想要的效果。在页面布局的时候都是通过百分比来定义宽度，高度大都是用px来固定住，所以在大屏幕的手机下 显示效果会变成有些页面元素宽度被拉的很长，但是高度还是和原来一样，非常的不协调，这就是流式布局的缺点．

2、固定宽度做法

      宽度固定为３２０然后居中，但是不可取，会造成大屏手机留白

  3、响应式布局

  工作量大，维护不易，好处是一处源码可适应多个终端

 4、设置viewport进行缩放

(设置屏幕密度为高频，中频，低频自动缩放，禁止用户手动调整缩放)

<metaname="viewport"content="width=device-width,target-densitydpi=high-dpi,initial-scale=1.0, minimum-scale=1.0, maximum-scale=1.0, user-scalable=no"/>

 5、rem等比例适配

    rem是通过根元素进行适配的，网页中的根元素指的是html我们通过设置html的字体大小就可以控制rem的大小。设置元素高度、宽度、字体大小等属性的单位为rem,在不同屏幕分辨率下只设置html的字体大小就可以控制其它元素来达到兼容性的效果。可以通过响应式或JS来动态设置html字体 大小。

1. 如何优化代码

