# 面试/鄙视（笔试）题

目录

[面试/鄙视（笔试）题 1](#_Toc21957)

[1. 如何让元素垂直居中 2](#_Toc14756)

[2. 定时器的执行顺序： 2](#_Toc26355)

[3. 刷新 3](#_Toc26541)

[4. 作用域与作用域链 3](#_Toc20799)

[5. 闭包是如何形成的（什么是闭包）： 3](#_Toc59)

[6. 闭包笔试题： 4](#_Toc25438)

[7. 面向对象-构造函数的原型-添加兼容IE8的indexOf方法 4](#_Toc4129)

[8. 对象、封装、继承、多态 4](#_Toc21510)

[9. ES6新特性，函数参数内可以设置默认初始值 5](#_Toc957)

[10. GET与POST区别 6](#_Toc13459)

[ Nodjs 6](#_Toc2958)

[11. 面试题:如何自学一门新的编程语言 6](#_Toc3541)

[12. 阿里面试题:用户在浏览器中输入 www.taobao.com直到看到页面之间发生了什么? 7](#_Toc26182)

[13. 页面优化？ 7](#_Toc8272)

[14. json格式的理解 7](#_Toc17551)

[15. 常见浏览器的内核名称是什么，各自有哪些特性？ 7](#_Toc19312)

[16. 增删改查和创建元素有哪些？ 8](#_Toc23520)

[17. 事件的三个阶段 8](#_Toc31624)

[18. 请描述cookies、sessionStorage和localStorage的区别 9](#_Toc10517)

[19. http://item.taobao.com/item.html?a=1&c=2&d=xxx&e,请写一段js程序提取url中的各个get参数（参数名和参数个数不确定），将其按key-value形式返回到一个json结构中，如（a:”1”, b:”2”, c:””, d:”xxx”, e:”undefined”）; 9](#_Toc15506)

[20. js内置对象有哪些 9](#_Toc10404)

[21. $scope和$rootScope的关系/区别 9](#_Toc16382)

[22. Doctype？严格模式与混杂模式如何触发？区分它们有何意义？移动端如何设置doctype？ 10](#_Toc14763)

[23. 行内元素有哪些，块级元素有哪些？描述一下CSS的盒子模型和伪类和内容生成； 11](#_Toc22054)

[24. 字符串与数组转化的函数有哪些？ 12](#_Toc18979)

[25. 详细解释数组方法pop() push() unshift() shift() 12](#_Toc8211)

[26. Document load和document ready的区别？ 12](#_Toc7678)

[27. 阻止事件冒泡和阻止默认行为 12](#_Toc9401)

[28. 编写一个数组去重的方法： 13](#_Toc20484)

[29. 使用正则表达式去首尾空格、验证手机号、验证邮箱 13](#_Toc3292)

[30. 请描述一下cookies，sessionStorage和localStorage的区别，html5如何设置哪些文件需要离线缓存？ 13](#_Toc25244)

[31. Px 、em 和 rem的区别 14](#_Toc9008)

[32. 对与前端工程师这个职位你是怎么理解的？如何融入公司才能提高开发效率。 14](#_Toc10222)

[33. 一个ng模块可以包含哪些？ 14](#_Toc25208)

[34. Git 和 svn的区别 14](#_Toc17404)

[35. 请写出一下也行结果 14](#_Toc27062)

[36. BOM对象有哪些 列举window对象 15](#_Toc24744)

[37. AJAX基本步骤 15](#_Toc11937)

[39. setTimeout、Promise、Async/Await 的执行顺序 16](#_Toc10943)

[40. ES6数组去重 16](#_Toc18681)

[41. 三次握手四次挥手 17](#_Toc10366)

## 如何让元素垂直居中

1. display：table
2. 绝对定位/固定定位
   1. 父元素position:relative;
   2. 子元素position:absolute;
   3. 上移子元素的一半margin-top：50%;
   4. 再下移父元素的一半top:50%;
3. 位移：

**#start**{  
 **text-align**: **center**;   
 **position**:**absolute**;  
 **top**:50%;**left**:50%;  
 **transform**:**translate**(-50%,-50%);  
}

1. 绝对居中

position:absolut;

top:0;

left:0;

right:0;

bottom:0;

margin:auto;

## 定时器的执行顺序：

var a;

setTimeout(function(){console.log(a) }, 10) //100

a=100;

var b=10;

setTimeout(function(){console.log(a) }, 0) //100

b=100;

定时器的回调函数不能直接加入主队列，需要放到回调队列，等待主程序执行完后再执行。

## 刷新

history.go(0)

location.reload()

Location.reload(true)强制刷新

## 作用域与作用域链

什么是作用域: 一个变量的可用范围

专门保存变量的对象

什么是作用域链: 由各级作用域，逐级引用形成的链式结构

2个作用:

1. 存储了所有的变量

2. 控制着变量的使用顺序

## 闭包是如何形成的（什么是闭包）：

外层函数的函数作用域对象无法释放，导致外层函数的局部变量被保存下来—可重用

意义：避免全局污染，每调用一次函数，创建一个闭包。

缺点：变量不会主动释放，会造成内存泄露。

表现形式：函数里面嵌套函数，返回函数。函数内可以引用局部变量，局部变量不会被垃圾回收。

**张东：**

鄙视:

什么是闭包：即重用一个变量，又保护变量不被污染的一种机制

闭包形成的原因:

外层函数的函数作用域对象无法释放

导致: 外层函数的局部变量被保存下来——可重用

## 闭包笔试题：

显示点击第几个按钮；

记录每个按钮点击的次数；

## 面向对象-构造函数的原型-添加兼容IE8的indexOf方法

**let *arr***=[1,2,3,4,5,4,3,2,1];  
*//判断当前浏览器数组类型的原型对象中是否包含indexOf***if**(typeof ***Array***.prototype.indexOf!=**"function"**){  
 *//向数组类型的原型对象中添加indexOf方法* ***Array***.prototype.indexOf=**function** (val,fromi) {  
 *//this->将来调用该方法的“.”前的对象  
 //如果fromi没有，则默认为0* fromi=fromi||0;  
 *//从fromi开始遍历数组中每个元素；* **for**(**let** i=fromi;i<**this**.length;i++){  
 **if**(**this**[i]===val){  
 **return** i;  
 }  
 }  
 **return** -1;  
 };  
}  
**let *i***=***arr***.indexOf(3);

## 对象、封装、继承、多态

**1. 什么是对象:**

对象: 集中存储多个数据的存储空间

本质: 其实对象底层就是关联数组

**2. 封装:**

什么是: 创建一个对象，描述现实中一个具体事物的属性和功能

1. 用对象直接量:

var obj={//创建一个新对象的意思

属性名:值,

... : ... ,

方法名(){...}, //~~方法名:function(){...}~~

... (){...},

}

2. 用new:

如何: 2步:

1. 先创建空对象: var obj={}; //new Object();

2. 向空对象中添加新属性: obj.新属性名=值;

js中的对象，可随时强行添加新属性和方法

3、遍历成员: for...in

for(var key in obj){

key //当前属性名

obj[key]//当前属性值

}

**3. 用构造函数**

什么是构造函数: 描述一类对象，统一结构的函数

何时: 只要想反复创建同一类型的多个对象时, 都要先用构造函数定义统一的结构

如何: 2步:

1. 定义构造函数: 定义同一类型对象的统一结构

function 类型名(属性参数,...){

this.属性名=属性参数;

... =属性参数;

this.方法名=function(){

... this.属性名 ...

}

}

2. 调用构造函数，反复创建新对象:

var obj=new 类型名(属性值,...);

new: 4件事:

1. 创建一个新的空对象

2. 让新对象自动继承构造函数的原型对象

3. 调用构造函数, 自动将函数中的this指向新对象

4. 返回新对象的地址保存在变量中

## ES6新特性，函数参数内可以设置默认初始值

function aa(a,b=3,c=2){

return a+b+c;

}

console.log(aa(5)) //10

console.log(aa(5,5)) //12

### GET与POST区别

|  |  |  |
| --- | --- | --- |
|  | GET | POST |
| 语义 | 客户端想获取指定资源 | 客户端想传递数据给服务器 |
| 如何发起 | 浏览器地址输入回车  标签 href src  超链接跳转  JS跳转  表单GET提交  AJAX-GET | 表单POST  AJAX POST |
| 安全级别 | 不安全 | 不安全 |
| 数据长度 | 1KB(汉字几个字) | 没有限制 |
| 是否需要编码 | 需要 | 不需要 |
|  |  |  |

## Nodjs

### 面试题:如何自学一门新的编程语言

(1)了解背景：历史、现状、发展、应用领域

(2)搭建环境：写helloworld

1:启动 apache

2:创建文件 01.php 并且保存在 C:\xampp\htdocs

3:运行唯一方式: 打开浏览器在地址栏

http://127.0.0.1/01.php 回车

(3)数据类型

(4)变量常量

(5)运算符

(6)逻辑结构

(7)通用小程序

(8)函数对象

(9)常用函数库，类库

(10)实用项目

### 阿里面试题:用户在浏览器中输入 www.taobao.com直到看到页面之间发生了什么?

(1)操作系统访问网络上的DNS服务器 把域名转ip地址

(2)浏览器发起HTTP请求消息

(3)web服务器接收并解析请示消息，查找指定的资源，可能访问数据库，构建并返回HTTP响应消息

(4)浏览器接收并解析响应消息

(5)浏览器缓存接收到响应内容，并解析和渲染响应内容

### 页面优化？

减少http请求次数，减少外部资源引用，数据库优化，代码封装，，；减少DOM操作次数，相同页面使用引用方式引入（头部和尾部）；使用精灵图（雪碧图）；使用CDN公共库（jq,bt）,代码压缩，

### json格式的理解

json是一个字符串，支持目前主流语言，比xml简单；数据压缩，传送快；客户端解析方便

### 常见浏览器的内核名称是什么，各自有哪些特性？

一、Trident内核代表产品Internet Explorer，又称其为IE内核。Trident（又称为MSHTML），是微软开发的一种排版引擎。使用Trident渲染引擎的浏览器包括：IE、傲游、世界之窗浏览器、Avant、腾讯TT、Netscape 8、NetCaptor、Sleipnir、GOSURF、GreenBrowser和KKman等。

二、Gecko内核代表作品Mozilla FirefoxGecko是一套开放源代码的、以C++编写的网页排版引擎。Gecko是最流行的排版引擎之一，仅次于Trident。使用它的最著名浏览器有Firefox、Netscape6至9。

三、WebKit内核代表作品Safari、Chromewebkit 是一个开源项目，包含了来自KDE项目和苹果公司的一些组件，主要用于Mac OS系统，它的特点在于源码结构清晰、渲染速度极快。缺点是对网页代码的兼容性不高，导致一些编写不标准的网页无法正常显示。主要代表作品有Safari和Google的浏览器Chrome。

四、Presto内核代表作品OperaPresto是由Opera Software开发的浏览器排版引擎，供Opera 7.0及以上使用。它取代了旧版Opera 4至6版本使用的Elektra排版引擎，包括加入动态功能，例如网页或其部分可随着DOM及Script语法的事件而重新排版。

### 增删改查和创建元素有哪些？

**增**：parent.appendChild(新元素);

**删：**parent.removeChild(child【要移除的子元素】)；

**改：**parent.replaceChild(新元素，现有元素【要替换的元素】)；

Parent.insertBefore/After(新元素，现有元素)；

**查：** document.getElementById(“box”)、document.getElementsByClassname(“color”);

Document.querySelect(); =>找一个

var elems=parent.querySelectorAll("selector"); =>找多个

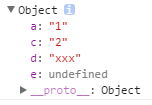
创建：var a=document.createElement("a");

### 事件的三个阶段

1. 事件捕获阶段：当使用事件捕获时，父级元素先触发，子元素后触发
2. 处于目标阶段
3. 事件冒泡阶段：当使用事件冒泡时，子级元素先触发，父元素后触发

### 请描述cookies、sessionStorage和localStorage的区别

### <http://item.taobao.com/item.html?a=1&c=2&d=xxx&e>,请写一段js程序提取url中的各个get参数（参数名和参数个数不确定），将其按key-value形式返回到一个json结构中，如（a:”1”, b:”2”, c:””, d:”xxx”, e:”undefined”）;

**var *url***=**"http://item.taobao.com/item.html?a=1&c=2&d=xxx&e"**;  
**function** *getData*(url){  
 **var** result={};  
 **var** temp=url.split(**"?"**)[1].split(**"&"**);  
 **console**.log(temp);  
 **for** (**var** i **in** temp){  
 **var** s=temp[i].split(**"="**);  
 **console**.log(s);  
 result[s[0]]=s[1];*//xx[]为对象属性添加值* }  
 **return** result;  
}  
**console**.dir(*getData*(***url***)) //

### js内置对象有哪些

内置对象: ES标准中规定的，浏览器厂商已经实现的对象

包括: 11个:

Number String Boolean -> 包装类型

Array RegExp Date Math

Error

Function Object

Global (在浏览器中被window代替)

### $scope和$rootScope的关系/区别

面试：$scope与$rootScope的关系

①不同的控制器的$scope是不一样的，不能直接进行通信

②$scope的id是由2开始的，随着控制器的加载的先后顺序依次向上递增 2、3、4.。。

③$rootScope是在ng初始化的时候 ，id是1，随后所创建的控制器对应的$scope都是$rootScope的子元素

<**div ng-controller="myCtrl01"**>  
 <**p**>{{"myCtr01 num is "+num}}</**p**>  
</**div**>  
  
<**div ng-controller="myCtrl02"**>  
 <**p**>{{"myCtr02 num is "+num}}</**p**>  
</**div**>  
<**script**>  
 *//创建模块* **var *app*** = ***angular***.module(**'myApp'**, [**'ng'**]);  
 *//创建一个控制器* ***app***.**controller**(  
 **'myCtrl01'**,  
 **function** ($scope,$rootScope) {  
 *//指定一个测试语句* ***console***.log(**'in ctrl01 func'**);  
 $scope.num = 10;  
 ***console***.log($scope);  
 ***console***.log($rootScope);  
 }  
 );  
 *//创建一个控制器* ***app***.**controller**(  
 **'myCtrl02'**,  
 **function** ($scope) {  
 *//指定一个测试语句* ***console***.log(**'in ctrl02 func'**);  
 ***console***.log($scope);  
 }  
 );  
</**script**>

### [Doctype？严格模式与混杂模式如何触发？区分它们有何意义？](http://www.cnblogs.com/wuqiutong/p/5986191.html)移动端如何设置doctype？

（1）<!DOCTYPE>声明位于文档中的最前面，处于<html>标签之前。用来告诉浏览器用哪种文档类型规范来解析这个文档。

（2）如果Doctype不存在或者其形式不正确那么默认为混杂模式。如果XHTMLl文档中包含完整的Doctype，那么它一般以标准模式呈现。

（3）严格模式的排版和 JS 运作模式是  以该浏览器支持的最高标准运行。

（4）在混杂模式中，页面以宽松的向后兼容的方式显示。模拟老式浏览器的行为以防止站点无法工作。

（5）移动端设置doctype：在文档中的最前面，处于<html>标签之前以<!DOCTYPE html>声明。

### 行内元素有哪些，块级元素有哪些？描述一下CSS的盒子模型和伪类和内容生成；

行内元素：span a b i sub sup input button img select label

块级元素：div section nav article footer h1-h5 ul ol li dl dt p

盒子模型：定义了元素处理(计算)尺寸，边框，内边距和外边距的 一种方式

伪类：

目标伪类：

:target

结构伪类：

1、：first-child

2、：last-child

3、:nth-child()

4、:empty

5、:only-child

否定伪类：

:not(select){}

伪元素选择器：

:first-letter

:first-line

::selection

内容生成：

:before :after

Content

div:before{

content:"子曰:";

color:red;

font-weight:bold;

margin-right:10px;

}

### 字符串与数组转化的函数有哪些？

Var arr=str.split(“”);

Var str=arr.join(“”);

### 详细解释数组方法pop() push() unshift() shift()

Pop()：数组结尾出栈

Push()：数组结尾进栈

Unshift()：数组开头入栈；

Shift()：数组开头出栈

### Document load和document ready的区别？

1. load是当页面所有资源加载完成后（包括DOM树，css文件，js文件，图片资源等），执行一个函数。
2. ready是当DOM树加载完成后执行一个函数（不包含图片和css等），这样会比load较快执行
3. 在原生的jS中不包括ready()这个方法，只有load方法就是onload事件

### 阻止事件冒泡和阻止默认行为

阻止冒泡/蔓延(w3c标准)：

Function fun(e){

e.stopPropagation();

}

阻止默认行为：

Function fun(e){

e.preventDefault();

}

阻止冒泡/蔓延（兼容模式）：

Function fun(e){

Window.event?window.event.cancelBubble=true : e.stopPropagation();

}

阻止默认行为：

Function fun(e){

Window.envnt? window.event.returnValue=false : e.preventDefault();

}

### 编写一个数组去重的方法：

**let *arr***=[1,2,3,1,2,3,4,8,4,];  
**function** *noRepeat*(arr) {  
 **let** arr1=[];  
 **let** hash={};  
 **for**(**let** i = 0 ;i<arr.**length**;i++){  
 **if**(hash[arr[i]]==**undefined**){  
 arr1.push(arr[i]);  
 hash[arr[i]]=1  
 }  
 }  
 **return** arr1;  
}  
***console***.log(*noRepeat*(***arr***))

### 使用正则表达式去首尾空格、验证手机号、验证邮箱

手机：(\+86|0086)?\s\*1[34578]\d{9}

邮箱：^[a-zA-Z0-9\_-]+@[A-Za-z0-9\_-]+(\.[ A-Za-z0-9\_-]+)$

**字符描述：**  
^ ：匹配输入的开始位置。  
\：将下一个字符标记为特殊字符或字面值。  
\* ：匹配前一个字符零次或几次。  
+ ：匹配前一个字符一次或多次。  
(pattern) 与模式匹配并记住匹配。  
x|y：匹配 x 或 y。  
[a-z] ：表示某个范围内的字符。与指定区间内的任何字符匹配。  
\w ：与任何单词字符匹配，包括下划线。

{n,m} 最少匹配 n 次且最多匹配 m 次  
$ ：匹配输入的结尾。

### 请描述一下cookies，sessionStorage和localStorage的区别，html5如何设置哪些文件需要离线缓存？

都是用来保存用户数据，但是cookies存储容量小，一般4k以下，且会参与服务器通信，数据始终会在同源的http请求中携带来回传输，影响速度，而sessionStorage和localStorage是H5提出来的新特性，不会自动把数据发送给服务器，容量也提升到5M或更大。另外cookie保存的数据是在设置的过期时间前有效，sessionStorage存储的数据是在浏览器窗口关闭前有效，localStorage保存的数据永久有效。

### Px 、em 和 rem的区别

Px 是像素单位，是相对显示器分辨率而言的。

Em是相对长度单位，相对继承上一级字体大小。

Rem 是C3新增一个单位，也叫根em，是相对于body的字体大小，没有继承关系。

如：

### 对与前端工程师这个职位你是怎么理解的？如何融入公司才能提高开发效率。

前端工程师是一个具有创造力和逻辑能力的职位，也是一个服务工程师职位。这就要求我们平时留意身边事物，提高代码熟练度，同时了解公司业务，了解工作流程，服从领导工作安排，与团队友好相处，才能更好地沟通，协作，从而提高开发效率，提升客户满意度。

### 一个ng模块可以包含哪些？

1. 控制器：处理业务逻辑
2. 服务：把需要反复用到的功能封装，方便复用
3. 指令：扩展html标签
4. 过滤器：实现对数据的筛选，过滤、格式化

### Git 和 svn的区别

github与svn都属于版本控件系统，但是两者不同于，github是分布式的，svn是属于集中式的。

### 请写出一下也行结果

1. alert(typeof null) // object
2. alert(typeof undefined) // undefined
3. alert(typeof NaN) //number
4. var str = “123abx

alert(typeof str++) //number

1. alert(srt) //NaN
2. alert(NaN == NaN) // false
3. alert(NaN == undefined) //false
4. alert(null == undefined) //false

### BOM对象有哪些 列举window对象

screen location innerWidth resizeTo onresezi href open go

### AJAX基本步骤

**var** http = **null**;  
**if** (window.XMLHttpRequest) {  
 //1 创建xml对象  
 http = **new** XMLHttpRequest();  
}  
**else** {  
 // 兼容ie  
 http = **new** ActiveXObject("Microsoft.XMLHTTP")  
}  
//2 绑定事件  
http.onreadystatechange = () => {  
 **if** (http.readyState === 4) {  
 console.log('完成');  
 **if**(http.status === 200){  
 console.log(2, http.responseText);;  
 }  
 }  
 **else**{  
 console.log(http.status);  
 }  
};  
// 3打开连接  
http.open('GET', './test.php', **true**);  
//4 发送请求  
http.send();

### setTimeout、Promise、Async/Await 的执行顺序

<https://blog.csdn.net/Umbrella_Um/article/details/100698686>

<https://www.cnblogs.com/cristina-guan/p/11487403.html>

### ES6数组去重

***// 利用Map对象和数组的filter方法*** *// 1.Map对象是ES6提供的一个新的数据结构，其中has的办法是返回一个布尔值，表示某个值是否存在当前的Mp对象之中，set的办法是给Map对象设置key/value。  
 // 2.filter() 方法创建一个新的数组，新数组中的元素是通过检查指定数组中符合条件的所有元素。  
 // 所以说，Map对象结合filter方法可以达到数组去重的效果~***let *arr*** = [2,2,2,**null**,**null**]  
**const *map*** = **new *Map***()  
**let *newArr*** = ***arr***.filter(item => !***map***.has(item) && ***map***.set(item, 1))  
***console***.log(***newArr***)

***// 利用from和 set去重*** *// 1.Set是ES6新提供的数据结构，类似于数组，但是本身没有重复值。  
// 2.Array.from方法用于将两类对象转为真正的数组：类似数组的对象（array-like object）和可遍历（iterable）的对象（包括 ES6 新增的数据结构 Set 和 Map）。  
// 所以set结合Array.from同样可以达到数组去重的效果。不过需要注意的是，主流浏览器像Chrome,Firfox,Opera，Safari，包括微软的Edge,都是支持的，但是唯独IE系列不支持。***let *arr*** = [1,2,2,1,**undefined**,**undefined**]  
**let *newArr*** = ***Array***.from(**new *Set***(***arr***))  
***console***.log(***newArr***)

***// 利用set + 扩展运算符...*****let *arr*** = [1, 1,3,3,3,***NaN***,***NaN***]  
**let *newArr*** = [...new ***Set***(***arr***)]  
***console***.log(***newArr***)

### 三次握手四次挥手

* **三次握手**

1.背景：TCP位于传输层，作用是提供可靠的字节流服务，为了准确无误地将数据送达目的地，TCP协议采纳三次握手策略。

2.原理：

1）发送端首先发送一个带有SYN（synchronize）标志地数据包给接收方。

2）接收方接收后，回传一个带有SYN/ACK标志的数据包传递确认信息，表示我收到了。

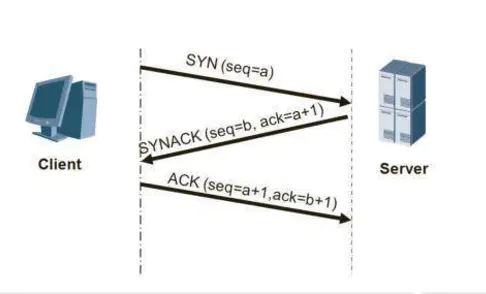
3）最后，发送方再回传一个带有ACK标志的数据包，代表我知道了，表示’握手‘结束。

通俗的说法

1）Client：嘿，李四，是我，听到了吗？

2）Server：我听到了，你能听到我的吗?

3）Client：好的，我们互相都能听到对方的话，我们的通信可以开始了。



* **四次挥手**

**1.意义：**当被动方收到主动方的FIN报文通知时，它仅仅表示主动方没有数据再发送给被动方了。但未必被动方所有的数据都完整的发送给了主动方，所以被动方不会马上关闭SOCKET,它可能还需要发送一些数据给主动方后，再发送FIN报文给主动方，告诉主动方同意关闭连接，所以这里的ACK报文和FIN报文多数情况下都是分开发送的。

2.原理：

 1）第一次挥手：Client发送一个FIN，用来关闭Client到Server的数据传送，Client进入FIN\_WAIT\_1状态。

 2）第二次挥手：Server收到FIN后，发送一个ACK给Client，确认序号为收到序号+1（与SYN相同，一个FIN占用一个序号），Server进入CLOSE\_WAIT状态。

 3）第三次挥手：Server发送一个FIN，用来关闭Server到Client的数据传送，Server进入LAST\_ACK状态。

 4）第四次挥手：Client收到FIN后，Client进入TIME\_WAIT状态，接着发送一个ACK给Server，确认序号为收到序号+1，Server进入CLOSED状态，完成四次挥手

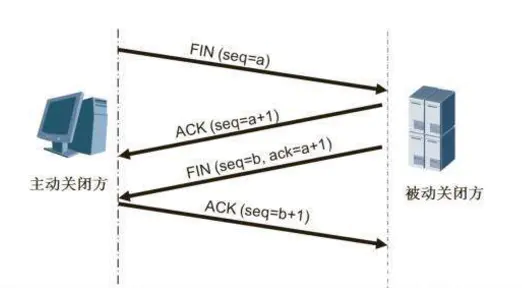
通俗的说法

1）Client：我所有东西都说完了

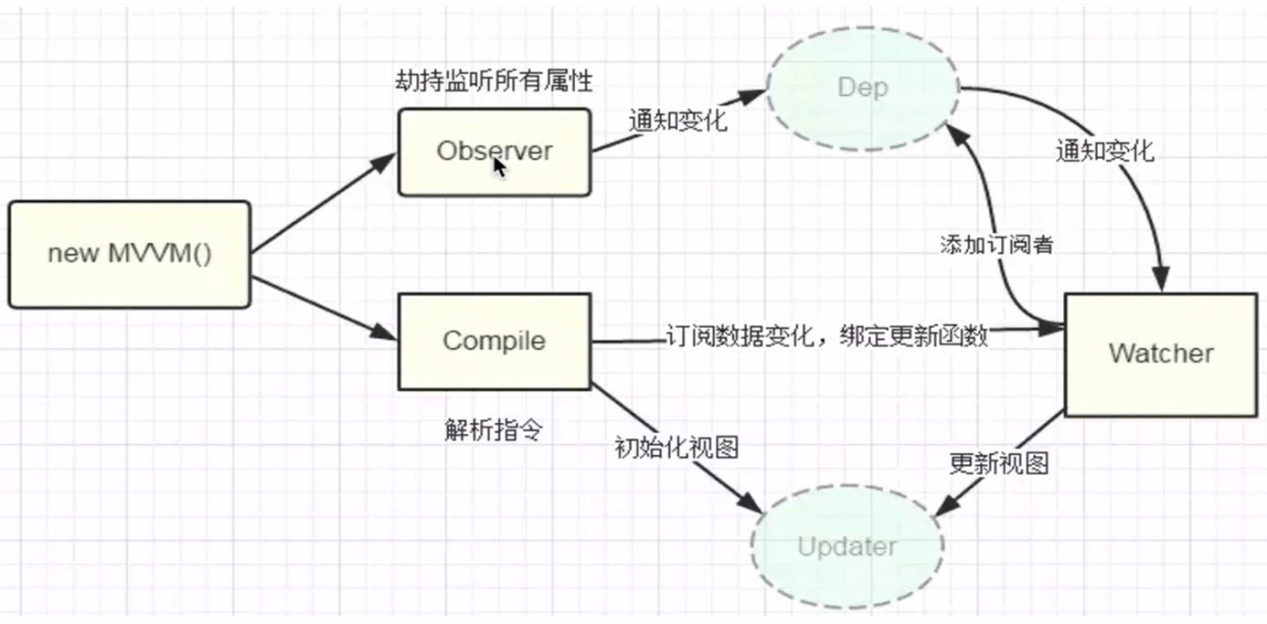
2）Server：我已经全部听到了，但是等等我，我还没说完

3）Server：好了，我已经说完了

4）Client：好的，那我们的通信结束l



### 阐述vue MVVM响应式原理



Vue 是采用数据劫持配合发布者和订阅者的方式，通过Object.definerProperty()去劫持各个数据属性的setter和getter， 在数据变动时发布消息给依赖收集器，并通知观察者，观察者做出相应回调去更新视图。

MVVM作为绑定入口，整合Observer、Compile和Watcher三者，通过Observer来监听model数据变化，通过Compile来解析编译模板指令，最终利用Watcher搭起Observer和Compile之间的通信桥梁，达到数据变化 => 视图更新， 视图交互变化 => model变更的双向绑定效果。