1.javascript的typeof返回哪些数据类型

2.例举3种强制类型转换和2种隐式类型转换?

3.split() join() 的区别

4.数组方法pop() push() unshift() shift()

5.事件绑定和普通事件有什么区别

6.IE和DOM事件流的区别

7.IE和标准下有哪些兼容性的写法

8.ajax请求的时候get 和post方式的区别

9.call和apply的区别

10.ajax请求时，如何解释json数据

11.b继承a的方法

12.事件委托是什么

13.闭包是什么，有什么特性，对页面有什么影响，请写出一个闭包示例

14.如何阻止事件冒泡和默认事件

16.添加 删除 替换 插入到某个接点的方法

17.解释jsonp的原理，以及为什么不是真正的ajax

18.javascript的本地对象，内置对象和宿主对象

19.document load 和document ready的区别

20.”==”和“===”的不同

21.javascript的同源策略

编写一个数组去重的方法

cookie 、localStorage、sessionStorage的区别是什么？

什么是Web Workers？为什么我们需要他们？

25.Web Worker线程的限制是什么？

26.我们如何在JavaScript中创建一个worker线程？

27.如何中止Web Worker?

28.我们如何从本地存储中添加和移除数据？

29.本地存储的生命周期是什么？

本地存储和cookies（储存在用户本地终端上的数据）之间的区别是什么？

什么是事务存储？我们如何创建一个事务存储？

本地存储和事务存储之间的区别是什么？

什么是WebSQL？

WebSQL 是HTML5的一个规范吗？

我们如何使用WebSQL？

36.JavaScript原型，原型链 ? 有什么特点？

原型对象也是普通的对象，是对象一个自带隐式的 \_\_proto\_\_ 属性，原型也有可能有自己的原型，如果一个原型对象的原型不为null的话，我们就称之为原型链。

原型链是由一些用来继承和共享属性的对象组成的（有限的）对象链。

JavaScript的数据对象有那些属性值？

　 writable：这个属性的值是否可以改。

configurable：这个属性的配置是否可以删除，修改。

enumerable：这个属性是否能在for…in循环中遍历出来或在Object.keys中列举出来。

value：属性值。

当我们需要一个属性的时，Javascript引擎会先看当前对象中是否有这个属性， 如果没有的话，就会查找他的Prototype对象是否有这个属性。

function clone(proto) {

　　 function Dummy() { }

　　 Dummy.prototype = proto;

　　 Dummy.prototype.constructor = Dummy;

　　 return new Dummy(); //等价于Object.create(Person);

}

function object(old) {

   function F() {};

    F.prototype = old;

    return new F();

}

var newObj = object(oldObject);

37.eval是做什么的？

它的功能是把对应的字符串解析成JS代码并运行；

避免使用eval，不安全，非常耗性能（2次，一次解析成js语句，一次执行）。

写一个通用的事件侦听器函数

// event(事件)工具集，来源：https://github.com/markyun

markyun.Event = {

// 页面加载完成后

readyEvent : function(fn) {

if (fn==null) {

fn=document;

}

var oldonload = window.onload;

if (typeof window.onload != 'function') {

window.onload = fn;

} else {

window.onload = function() {

oldonload();

fn();

};

}

},

// 视能力分别使用dom0||dom2||IE方式 来绑定事件

// 参数： 操作的元素,事件名称 ,事件处理程序

addEvent : function(element, type, handler) {

if (element.addEventListener) {

//事件类型、需要执行的函数、是否捕捉

element.addEventListener(type, handler, false);

} else if (element.attachEvent) {

element.attachEvent('on' + type, function() {

handler.call(element);

});

} else {

element['on' + type] = handler;

}

},

// 移除事件

removeEvent : function(element, type, handler) {

if (element.removeEnentListener) {

element.removeEnentListener(type, handler, false);

} else if (element.datachEvent) {

element.detachEvent('on' + type, handler);

} else {

element['on' + type] = null;

}

},

// 阻止事件 (主要是事件冒泡，因为IE不支持事件捕获)

stopPropagation : function(ev) {

if (ev.stopPropagation) {

ev.stopPropagation();

} else {

ev.cancelBubble = true;

}

},

// 取消事件的默认行为

preventDefault : function(event) {

if (event.preventDefault) {

event.preventDefault();

} else {

event.returnValue = false;

}

},

// 获取事件目标

getTarget : function(event) {

return event.target || event.srcElement;

},

// 获取event对象的引用，取到事件的所有信息，确保随时能使用event；

getEvent : function(e) {

var ev = e || window.event;

if (!ev) {

var c = this.getEvent.caller;

while (c) {

ev = c.arguments[0];

if (ev && Event == ev.constructor) {

break;

}

c = c.caller;

}

}

return ev;

}

};

38.Node.js的适用场景

高并发、聊天、实时消息推送

39.创建一个对象

function Person(name, age) {

this.name = name;

this.age = age;

this.sing = function() { alert(this.name) }

}

40.谈谈This对象的理解。

this是js的一个关键字，随着函数使用场合不同，this的值会发生变化。

但是总有一个原则，那就是this指的是调用函数的那个对象。

this一般情况下：是全局对象Global。 作为方法调用，那么this就是指这个对象

41.事件、IE与火狐的事件机制有什么区别？ 如何阻止冒泡？

1. 我们在网页中的某个操作（有的操作对应多个事件）。例如：当我们点击一个按钮就会产生一个事件。是可以被 JavaScript 侦测到的行为。

2. 事件处理机制：IE是事件冒泡、火狐是 事件捕获；

3. ev.stopPropagation();

42.什么是闭包（closure），为什么要用？

待完善

执行say667()后,say667()闭包内部变量会存在,而闭包内部函数的内部变量不会存在.使得Javascript的垃圾回收机制GC不会收回say667()所占用的资源，因为say667()的内部函数的执行需要依赖say667()中的变量。这是对闭包作用的非常直白的描述.

function say667() {

// Local variable that ends up within closure

var num = 666;

var sayAlert = function() { alert(num); }

num++;

return sayAlert;

}

var sayAlert = say667();

sayAlert()//执行结果应该弹出的667

43.如何判断一个对象是否属于某个类？使用instanceof

（待完善）

if(a instanceof Person){

alert('yes');

}

44.new操作符具体干了什么呢?

1、创建一个空对象，并且 this 变量引用该对象，同时还继承了该函数的原型。

2、属性和方法被加入到 this 引用的对象中。

3、新创建的对象由 this 所引用，并且最后隐式的返回 this 。

var obj = {};

obj.\_\_proto\_\_ = Base.prototype;

Base.call(obj);

45.JSON 的了解

JSON(JavaScript Object Notation) 是一种轻量级的数据交换格式。它是基于JavaScript的一个子集。数据格式简单, 易于读写, 占用带宽小

{'age':'12', 'name':'back'}

46.ajax 是什么?ajax 的交互模型?同步和异步的区别?如何解决跨域问题?

待完善

1. 通过异步模式，提升了用户体验

2. 优化了浏览器和服务器之间的传输，减少不必要的数据往返，减少了带宽占用

3. Ajax在客户端运行，承担了一部分本来由服务器承担的工作，减少了大用户量下的服务器负载。

Ajax的最大的特点是什么。

Ajax可以实现动态不刷新（局部刷新）

readyState属性 状态 有5个可取值： 0=未初始化 ，1=正在加载 2=以加载，3=交互中，4=完成

ajax的缺点

1、ajax不支持浏览器back按钮。

2、安全问题 AJAX暴露了与服务器交互的细节。

3、对搜索引擎的支持比较弱。

4、破坏了程序的异常机制。

5、不容易调试。

跨域： jsonp、 iframe、window.name、window.postMessage、服务器上设置代理页面

47.模块化怎么做？

[立即执行函数](http://benalman.com/news/2010/11/immediately-invoked-function-expression/),不暴露私有成员

var module1 = (function(){

　　　　var \_count = 0;

　　　　var m1 = function(){

　　　　　　//...

　　　　};

　　　　var m2 = function(){

　　　　　　//...

　　　　};

　　　　return {

　　　　　　m1 : m1,

　　　　　　m2 : m2

　　　　};

　　})();

48.对Node的优点和缺点提出了自己的看法：

（优点）因为Node是基于事件驱动和无阻塞的，所以非常适合处理并发请求，

因此构建在Node上的代理服务器相比其他技术实现（如Ruby）的服务器表现要好得多。

此外，与Node代理服务器交互的客户端代码是由javascript语言编写的，因此客户端和服务器端都用同一种语言编写，这是非常美妙的事情。

\*（缺点）Node是一个相对新的开源项目，所以不太稳定，它总是一直在变，而且缺少足够多的第三方库支持。看起来，就像是Ruby/Rails当年的样子。

49.异步加载的方式

(1) defer，只支持IE

(2) async：

(3) 创建script，插入到DOM中，加载完毕后callBack

documen.write和 innerHTML的区别

document.write只能重绘整个页面

innerHTML可以重绘页面的一部分

50.告诉我答案是多少？

(function(x){

delete x;

alert(x);

})(1+5);

函数参数无法delete删除，delete只能删除通过for in访问的属性。

当然，删除失败也不会报错，所以代码运行会弹出“1”。

51.JS中的call()和apply()方法的区别？

例子中用 add 来替换 sub，add.call(sub,3,1) == add(3,1) ，所以运行结果为：alert(4);

注意：js 中的函数其实是对象，函数名是对 Function 对象的引用。

function add(a,b)

{

alert(a+b);

}

function sub(a,b)

{

alert(a-b);

}

add.call(sub,3,1);

52..Jquery与jQuery UI 有啥区别？

\*jQuery是一个js库，主要提供的功能是选择器，属性修改和事件绑定等等。

\*jQuery UI则是在jQuery的基础上，利用jQuery的扩展性，设计的插件。

提供了一些常用的界面元素，诸如对话框、拖动行为、改变大小行为等等

53.jquery 中如何将数组转化为json字符串，然后再转化回来？

jQuery中没有提供这个功能，所以你需要先编写两个jQuery的扩展：

$.fn.stringifyArray = function(array) {

return JSON.stringify(array)

}

$.fn.parseArray = function(array) {

return JSON.parse(array)

}

然后调用：

$("").stringifyArray(array)

JavaScript中的作用域与变量声明提升？

http状态码有那些？分别代表是什么意思？

100-199 用于指定客户端应相应的某些动作。

200-299 用于表示请求成功。

300-399 用于已经移动的文件并且常被包含在定位头信息中指定新的地址信息。 400-499 用于指出客户端的错误。

400 1、语义有误，当前请求无法被服务器理解。

401 当前请求需要用户验证

403 服务器已经理解请求，但是拒绝执行它。

500-599 用于支持服务器错误。 503 – 服务不可用

你所知道的页面性能优化方法有那些？

除了前端以外还了解什么其它技术么？你最最厉害的技能是什么？

AMD（Modules/Asynchronous-Definition）、CMD（Common Module Definition）规范区别？

谈谈你认为怎样做能是项目做的更好？

你对前端界面工程师这个职位是怎么样理解的？它的前景会怎么样？

加班的看法

加班就像借钱，原则应当是------救急不救穷

平时如何管理你的项目，如何设计突发大规模并发架构？

先期团队必须确定好全局样式（globe.css），编码模式(utf-8) 等

编写习惯必须一致（例如都是采用继承式的写法，单样式都写成一行）；

标注样式编写人，各模块都及时标注（标注关键样式调用的地方）；

页面进行标注（例如 页面 模块 开始和结束）；

CSS跟HTML 分文件夹并行存放，命名都得统一（例如style.css）

JS 分文件夹存放 命民以该JS 功能为准英文翻译；

图片采用整合的 images.png png8 格式文件使用 尽量整合在一起使用方便将来的管理

哪些操作会造成内存泄漏？

内存泄漏指任何对象在您不再拥有或需要它之后仍然存在。

垃圾回收器定期扫描对象，并计算引用了每个对象的其他对象的数量。如果一个对象的引用数量为 0（没有其他对象引用过该对象），或对该对象的惟一引用是循环的，那么该对象的内存即可回收。

setTimeout 的第一个参数使用字符串而非函数的话，会引发内存泄漏。

闭包、控制台日志、循环（在两个对象彼此引用且彼此保留时，就会产生一个循环）

你说你热爱前端，那么应该WEB行业的发展很关注吧？ 说说最近最流行的一些东西吧Node.js、Mongodb、npmM、MVVM、MEAN

作为一名前端工程师，无论工作年头长短都应该必须掌握的知识点有：

1、DOM结构 —— 两个节点之间可能存在哪些关系以及如何在节点之间任意移动。

2、DOM操作 ——如何添加、移除、移动、复制、创建和查找节点等。

3、事件 —— 如何使用事件，以及IE和标准DOM事件模型之间存在的差别。

4、XMLHttpRequest —— 这是什么、怎样完整地执行一次GET请求、怎样检测错误。

5、严格模式与混杂模式 —— 如何触发这两种模式，区分它们有何意义。

6、盒模型 —— 外边距、内边距和边框之间的关系，及IE8以下版本的浏览器中的盒模型

7、块级元素与行内元素 —— 怎么用CSS控制它们、以及如何合理的使用它们

8、浮动元素——怎么使用它们、它们有什么问题以及怎么解决这些问题。

9、HTML与XHTML——二者有什么区别，你觉得应该使用哪一个并说出理由。

10、JSON —— 作用、用途、设计结构。  
71.用js写个原生的ajax过程

请实现，鼠标点击页面中的任意标签，alert 该标签的名称（注意兼容性）

73.请给出异步加载js方案，不少于两种。

请写出jquery绑定事件的方法，不少于两种。

ajax 有那些优缺点?如何解决跨域问题?

GET和POST的区别，何时使用POST？

请解释一下 JavaScript 的同源策略

Flash、Ajax各自的优缺点，在使用中如何取舍？

javascript里面的继承怎么实现，如何避免原型链上面的对象共享

ajax过程

80.嵌入JS应该放在什么位置？

81.请写一个简单的幻灯效果页面

82.data-属性的作用是什么？

83.解释下JavaScript中this是如何工作的。

84.AMD vs.CommonJS？

85.什么是哈希表？

86.解释下为什么接下来这段代码不是IIFE(立即调用的函数表达式)：function foo(){}();.

87.描述以下变量的区别：null，undefined或undeclared？

88.什么是闭包，如何使用它，为什么要使用它？

89.请举出一个匿名函数的典型用例？

90.解释“JavaScript模块模式”以及你在何时使用它。

91.你是如何组织自己的代码？是使用模块模式，还是使用经典继承的方法？

92.请指出JavaScript宿主对象和原生对象的区别？

93.call和.apply的区别是什么？

94.在什么时候你会使用document.write()？

95.请尽可能详尽的解释AJAX的工作原理。

96.请解释JSONP的工作原理，以及它为什么不是真正的AJAX。

97.请解释变量声明提升

98.请描述下事件冒泡机制。

99."attribute"和"property"的区别是什么？

100.为什么扩展JavaScript内置对象不是好的做法？

101.为什么扩展JavaScript内置对象是好的做法？

102.请指出document.onload和document.ready两个事件的区别。

103.==和===有什么不同？

104.你如何从浏览器的URL中获取查询字符串参数

105.请描述一下JavaScript的继承模式。

106.如何实现下列代码：[1,2,3,4,5].duplicator();//[1,2,3,4,5,1,2,3,4,5]。

107.描述一种JavaScript中实现memoization(避免重复运算)的策略。

108.什么是三元表达式？“三元”表示什么意思？

109.JavaScript里函数参数arguments是数组吗？

110.什么是"use strict";?使用它的好处和坏处分别是什么？

111.解释"chaining"。

112.解释"deferreds"。

113.你知道哪些针对jQuery的优化方法？

114.请解释.end()的用途。

115.你如何给一个事件处理函数命名空间，为什么要这样做？

116.请说出你可以传递给jQuery方法的四种不同值。

117.什么是效果队列？

118.请指出.get()，[]，eq()的区别。

119.请指出.bind()，.live()和.delegate()的区别。

120.请指出$和$.fn的区别，或者说出$.fn的用途。

121.请写出一个函数实现N!的计算。N取很大的值时，该怎么办？

122.modulo(12,5) //2 ；问题：实现满足上述结果的modulo函数。

123."i'm a lasagna hog".split("").reverse().join("");问题：上面的语句的返回值是什么？

124.(window.foo||(window.foo="bar"));问题：window.foo的值是什么？

125.两个平级界面块 a 和 b，如果 a 中触发一个事件，有哪些方式能让 b 知道？详述原理

126.一个 angular 应用应当如何良好地分层？

127.angular 的缺点有哪些？

128.angular的数据绑定采用什么机制？详述原理

129. compile和link的区别：性能和能力

130.ng-if 跟 ng-show/hide 的区别有哪些？

131.什么是Stub？举个使用场景

132.、window.onload 和 document.ready的区别？

133.判断一个字符串出现次数最多的字符，统计这个次数并输出

134、编写一个方法，去掉一个数组的复重元素

135、module(12,5)//2 实现满足这个结果的modulo函数

136、FF与IE中如何阻止事件冒泡，如何获取事件对象，以及如何获取触发事件的元素

137、s可否实现面向对象编程，如果可以如何实现js对象的继承