* 说说你对作用域链的理解？

当代码在一个环境中执行时，会创建变量对象的一个作用域链（scope chain，不简称sc）来保证对执行环境有权访问的变量和函数的有序访问。

* 变量提升？

JavaScript 仅提升声明，而不提升初始化。如果你先使用的变量，再声明并初始化它，变量的值将是 undefined。

1:所有的声明都会提升到作用域的最顶上去。

2:同一个变量只会声明一次，其他的会被忽略掉。

3:函数声明的优先级高于变量申明的优先级，并且函数声明和函数定义的部分一起被提升。

* 介绍js有哪些内置对象？

时间对象date，字符串对象string，数学对象Math，数值对象Number，数组对象Array，函数对象function,函数参数集合arguments,布尔对象Boolean，错误对象Error,基础对象Object

* 请解释JavaScript中的相等判断？
* 请解释什么叫做回调函数，并提供一个简单的例子？

详看xmind

* js有哪些基本数据类型？

Undefined、Null、Boolean、Number和String

* 闭包是什么？开发中常见的使用场景？

 函数外部无法访问函数内部的局部变量，但函数内部的函数可以访问本函数内的局部变量，故通过闭包实现函数外部访问函数内部局部变量。但容易造成内存泄漏，应当谨慎使用。

使用场景一:给对象设置私有变量并且利用特权方法去访问私有属性

使用场景二:采用函数引用方式的setTimeout调用

使用场景三:封装相关功能集

* 如果比较JavaScript中的两个对象？

function isDeepEqual(obj1, obj2, testPrototypes = false) {

if (obj1 === obj2) {//如果obj1和obj2是同一个对象，返回true

return true

}

if (typeof obj1 === "function" && typeof obj2 === "function") {

//如果obj1和obj2都是function，转换成字符串后比较

return obj1.toString() === obj2.toString()

}

if (obj1 instanceof Date && obj2 instanceof Date) {

//如果obj1和obj2都是Date实例，获取毫秒值比较

return obj1.getTime() === obj2.getTime()

}

if (

Object.prototype.toString.call(obj1) !==

Object.prototype.toString.call(obj2) ||

typeof obj1 !== "object"

) {

return false

}

//如果obj1是简单类型且toString后的值不相等，直接返回false

const prototypesAreEqual = testPrototypes

? isDeepEqual(

Object.getPrototypeOf(obj1),

Object.getPrototypeOf(obj2),

true

)

: true

//判断原型是否相等

const obj1Props = Object.getOwnPropertyNames(obj1)

const obj2Props = Object.getOwnPropertyNames(obj2)

return (

obj1Props.length === obj2Props.length &&

prototypesAreEqual &&

obj1Props.every(prop => isDeepEqual(obj1[prop], obj2[prop]))

)

//obj1和obj2自身属性长度相等

//原型相等

//递归调用isDeepEqual判断每一个属性值是否相等

}

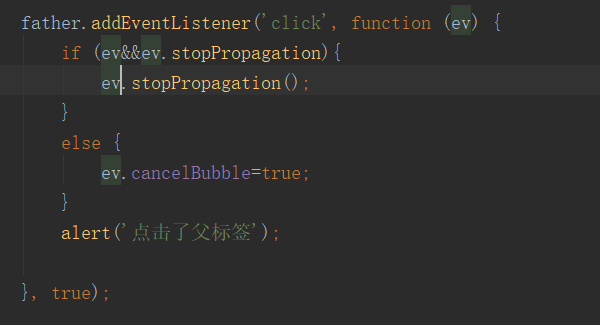
* JavaScript原型，原型链 ? 有什么特点？

<https://www.cnblogs.com/jiajialove/p/9059614.html>

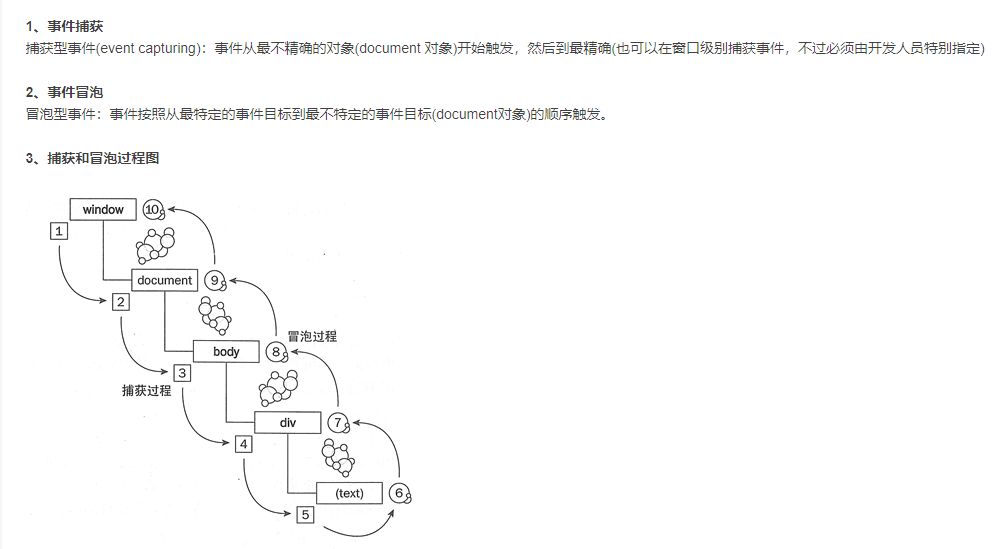
* Null和Undefined的区别、联系？



* 事件冒泡以及如何阻止它？



* 事件捕获原理？



* 匿名函数和命名函数的区别？
* 类和继承（es5实现方法 + es6实现方法）？

<https://blog.csdn.net/wdr2003/article/details/80662872>

* 谈谈This对象的理解？

理解：

1.this是js 的一个关键字，随着函数的使用场合的不同，this 的值会发生变化。

2.一个总原则：即this指的是调用函数的那个对象。

3.一般情况下，this 是全局对象，可以作为方法调用。

* 在JavaScript中如何创建私有变量？

<https://www.xinran001.com/frontend/205.html>

* new操作符具体干了什么呢?

1、创建一个空对象  
2、此空对象的\_\_proto\_\_指向构造的prototype  
3、借调构造将this指向这个对象并添加属性及方法  
4、返回这个对象

* 写一个方法遍历所有文档树所有节点？
* 事件模型？

<https://www.cnblogs.com/Leo_wl/p/3267503.html>

* sort排序相关算法？
* 数组和对象的深浅拷贝？

<https://www.cnblogs.com/shenjp/p/6950040.html>

* call、apply、bind？
* <https://www.runoob.com/w3cnote/js-call-apply-bind.html>
* setTimeout、setInterval和requestAnimationFrame的使用和区别？

<https://blog.csdn.net/qingyafan/article/details/52335753>

* JS中引起内存泄漏的原因？

<https://www.cnblogs.com/libin-1/p/6013490.html>

* 异步加载JS的方式有哪些？
* <https://www.cnblogs.com/1234wu/p/10184877.html>
* this关键字如何工作？请提供一些例子？

this 关键字表示的是当前类

比如 this.name 是指当前类的name属性bai，有些时候可以用于区分du同名属性，如你在类里有个属性为name，有个方法里也有个属性叫name，你在该方法里使用this.name是指类属性，直接使用name则是指方法的name属性，这样不会混淆；

可以用this.method()调用本类的method()方法；

可以调用父类的方法和属性，比如你继承了某个类，在该类中你不能直接看到父类的方法和属性，你需要打开父类去查看他的方法和属性，这样比较麻烦，你可以直接用this.的方式查看到父类的方法和属性；

使用this关键字调用类方法和属性通常比较方便，比如你的类比zhi较长，有时，你想调用某个方法或属性，你又忘了它的名称，直接使用this.的方式直接就能找到；

还有一点要注意，比如你有个类包含一个内部类，你在内部类使用this，则是指调用内部类的方法和属性，此时this不能调用到外部类的方法及属性。

* XML和JSON的区别？

xml：可扩展[标记语言](https://baike.baidu.com/item/%E6%A0%87%E8%AE%B0%E8%AF%AD%E8%A8%80" \t "https://www.cnblogs.com/jasonboren/p/_blank)，是一种用于标记电子文件使其具有结构性的[标记语言](https://baike.baidu.com/item/%E6%A0%87%E8%AE%B0%E8%AF%AD%E8%A8%80/5964436" \t "https://www.cnblogs.com/jasonboren/p/_blank)。

json：([JavaScript](https://baike.baidu.com/item/JavaScript" \t "https://www.cnblogs.com/jasonboren/p/_blank) Object Notation, JS 对象简谱) 是一种轻量级的数据交换格式。

二者区别：

它们都是一种数据交换格式。

1，xml是重量级的，json是轻量级的。

2，xml在传输过程中比较占带宽，json占带宽少，易于压缩。

3，xml和json都用在项目交互下，xml多用于做配置文件，json用于数据交互。

4，json可用jackson，gson等方法解析，xml可用dom，sax，demo4j等方式解析。

* javascript 代码中的”use strict”;是什么意思 ? 使用它区别是什么？

use strict是一种ECMAscript 5 添加的（严格）运行模式,这种模式使得 Javascript 在更严格的条件下运行,  
使JS编码更加规范化的模式,消除Javascript语法的一些不合理、不严谨之处，减少一些怪异行为。  
默认支持的糟糕特性都会被禁用，比如不能用with，也不能在意外的情况下给全局变量赋值;  
全局变量的显示声明,函数必须声明在顶层，不允许在非函数代码块内声明函数,arguments.callee也不允许使用；  
消除代码运行的一些不安全之处，保证代码运行的安全,限制函数中的arguments修改，严格模式下的eval函数的行为和非严格模式的也不相同;  
提高编译器效率，增加运行速度；  
为未来新版本的Javascript标准化做铺垫。

* 同步和异步的区别?



* js延迟加载的方式有哪些？

<https://www.cnblogs.com/8080zh/p/9232177.html>

* 如何为Array对象添加你自定义的函数，使得如下代码可以正常工作？



* attribute和property的区别是什么？



* JavaScript中提升(hoisting)是什么意思？

<https://www.cnblogs.com/qixingduanyan/p/11489567.html>

* 移动web相关知识点？

<https://www.cnblogs.com/wuqiuxue/p/8228878.html>

* http返回的请求有几种？分别代表什么含义？

<https://www.cnblogs.com/zhangsanfeng/p/9125029.html>

* ajax请求如何把异步改为同步？

Ajax请求默认的都是异步的

如果想同步 async设置为false就可以（默认是true）

var html = $.ajax({

url: “some.PHP”,

async: false

}).responseText;

或者在全局设置Ajax属性

$.ajaxSetup({

async: false

});

再用post，get就是同步的了