**null和undefined的区别？**

null是一个表示"无"的对象，转为数值时为0

undefined是一个表示"无"的原始值，转为数值时为NaN

当声明的变量还未被初始化时，变量的默认值为undefined

null用来表示尚未存在的对象，常用来表示函数企图返回一个不存在的对象

undefined表示 “缺少值”，就是此处应该有一个值，但是还没有定义。典型用法是：

1. 变量被声明了，但没有赋值时，就等于 undefined

2. 调用函数时，应该提供的参数没有提供，该参数等于 undefined

3. 对象没有赋值的属性，该属性的值为 undefined

4. 函数没有返回值时，默认返回 undefined

null表示“没有对象”，即该处不应该有值。典型用法是：

1. 作为函数的参数，表示该函数的参数不是对象

2. 作为对象原型链的终点

**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);

**什么是JavaScript 原型，原型链 ? 有什么特点？**

1. 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

2. 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里面的继承怎么实现，如何避免原型链上面的对象共享**

用构造函数和原型链的混合模式去实现继承，避免对象共享可以参考经典的extend()函数，很多前端框架都有封装的，就是用一个空函数当做中间变量

**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。

**javascript对象的几种创建方式**

1. 工厂模式 2. 构造函数模式 3. 原型模式 4. 混合构造函数和原型模式 5. 动态原型模式 6. 寄生构造函数模式

7. 稳妥构造函数模式

**javascript继承的6种方法**

1. 原型链继承 2. 借用构造函数继承 3. 组合继承(原型+借用构造) 4. 原型式继承 5. 寄生式继承6. 寄生组合式继承

**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？**

Ajax 的作用：

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：

1. Ajax可以实现动态刷新（局部刷新）

2. readyState 属性 状态 有5个可取值： 0 = 未初始化，1 = 启动， 2 = 发送，3 = 接收，4 = 完成

Ajax 同步和异步的区别:

1. 同步：提交请求 -> 等待服务器处理 -> 处理完毕返回，这个期间客户端浏览器不能干任何事

2. 异步：请求通过事件触发 -> 服务器处理（这是浏览器仍然可以作其他事情）-> 处理完毕

ajax.open方法中，第3个参数是设同步或者异步。

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

**使用ajax的流程**

1. 创建XMLHttpRequest对象,也就是创建一个异步调用对象

2. 创建一个新的HTTP请求,并指定该HTTP请求的方法、URL及验证信息

3. 设置响应HTTP请求状态变化的函数

4. 发送HTTP请求

5. 获取异步调用返回的数据

6. 使用JavaScript和DOM实现局部刷新

**Flash、Ajax各自的优缺点，在使用中如何取舍？**

Flash：1. Flash适合处理多媒体、矢量图形、访问机器 2. 对CSS、处理文本上不足，不容易被搜索

Ajax： 1. Ajax对CSS、文本支持很好，支持搜索 2. 多媒体、矢量图形、机器访问不足

共同点：

1.与服务器的无刷新传递消息 2.可以检测用户离线和在线状态 3. 操作DOM

**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、firefox同时支持两种事件模型，也就是：捕获型事件和冒泡型事件

3. ev.stopPropagation(); 注意旧ie的方法：ev.cancelBubble = true;

**Node有什么优点和缺点**

优点：

1. 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。

2. 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。

适用场景：1. 高并发 2. 聊天 3. 实时消息推送

**call() 和 .apply() 的作用？**

动态改变某个类的某个方法的运行环境

**如何判断当前脚本运行在浏览器还是node环境中？**

通过判断 Global 对象是否为window，如果不为window，当前脚本没有运行在浏览器中。即在node中的全局变量是global ,浏览器的全局变量是window。 可以通过该全局变量是否定义来判断宿主环境

**js延迟加载的方式有哪些？**

1. defer和async
2. 动态创建DOM方式（创建script,插入到DOM中，加载完毕后callback）
3. 按需异步载入js

**eval是做什么的？**

1. 它的功能是把对应的字符串解析成JS代码并运行

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