如果面试题按类型来分，主要涉及到“技术”与“非技术”两大类，技术类别下涉及到的子类别有：

1. 移动 & PC端布局类
2. JavaScript 核心基础类
3. 衍生框架类
4. 项目应用类

这四大类别的面试题若按出现频率来划分，则面试时 100% 会问到的题型有：“移动端&PC端布局类、JavaScript 核心基础类”。本次选择这两类中难度更高一些的 “JavaScript 核心基础类” 面试题，进行了分析和解答，供面试前准备。

**特别说明：**

假如你在面试时被这些题型卡住了，就必须正视自己的知识短板，并且要心悦城服的刻苦钻研、直至学会为止！毕竟想要成为一个合格的 “WEB前端工程师”，掌握以下这些 JavaScript 核心的基础知识，才担得上“合格”这两个字，否则就是“冒牌”的哦~

**一、JavaScript 的 typeof 返回哪些数据类型？**

**分析：**

这道题检验的是 JS 基本功，只要对 typeof 运算符是了解的，就没有太大难度。具体在回答的时候，再结合理论知识和编码过程中实际情况进行回答即可。另外，考虑到面试时的严谨性， ES2015 中新增的类型也并入回复中。

**建议回复：**

首先，JavaScript 中一共有两大数据类型：

1. 基础类型
2. 引用类型

    ·  基础类型包括：Number、String、Boolean、Null、Undefined、Symbol（该类型位 ES2015 中新增类型）

    ·  引用类型包括：Object typeof 运算符把类型信息以字符串形式返回，需要注意的是 typeof 返回的类型和 JavaScript 定义的类型有细微的差异。 typeof 返回七种可能的值：“number”、“string”、“boolean”、“object”、"symbol"、“function”和“undefined”。

**二、请写出以下代码运算结果：**

[?](https://www.jb51.net/article/108375.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | alert(**typeof** **null**);  alert(**typeof** undefined);  alert(**typeof** NaN);  alert(NaN == undefined);  alert(NaN == NaN);  **var** str = "123abc";  alert(**typeof** str++);  alert(str); |

**分析：**

这道题与“题目一”是连环扣，当“题目一”回答完后，通过此题再一次强化运算符和数据类型的基本功。

**建议回复：**

本题主要是考察 typeof 判断值的类型，它们输出的结果依次是：

[?](https://www.jb51.net/article/108375.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | alert(**typeof** **null**); // object  alert(**typeof** undefined); // undefined  alert(**typeof** NaN); // number  alert(NaN == undefined); // false  alert(NaN == NaN); // false  **var** str = "123abc";  alert(**typeof** str++); // number  alert(str); // NaN |

**三、例举至少 3 种强制类型转换和 2 种隐式类型转换?**

**分析：**

类型转换听起来可能有点宽泛，但这道题明确给出了回答的范围。

**建议回复：**

1. 强制类型转换： 明确调用内置函数，强制把一种类型的值转换为另一种类型。强制类型转换主要有：Boolean、Number、String、parseInt、parseFloat

2. 隐式类型转换： 在使用算术运算符时，运算符两边的数据类型可以是任意的，比如，一个字符串可以和数字相加。之所以不同的数据类型之间可以做运算，是因为 JavaScript 引擎在运算之前会悄悄的把他们进行了隐式类型转换。隐式类型转换主要有：+、–、==、!

**四、JavaScript 的事件流模型都有什么？**

**建议回复：**

事件流描述的是从页面中接收事件的顺序。 DOM 结构是树形结构，当页面中的某一个元素触发了某个一个事件，事件会从最顶层的 window 对象开始，向下传播到目标元素，途径的祖先节点都会触发对应的事件，如果当前节点的该事件绑定了事件处理函数的话，则会执行该函数当事件达到目标元素并执行绑定函数（如果有绑定的话）后，事件又会向上传播到 window 元素，途径的祖先节点都会触发对应的事件（如果绑定事件处理函数的话）

**事件流包含三个阶段：**

* 事件捕捉阶段
* 处于目标阶段
* 事件冒泡阶段

      事件捕捉阶段：事件开始由顶层对象触发，然后逐级向下传播，直到目标的元素；

      处于目标阶段：处在绑定事件的元素上；

      事件冒泡阶段：事件由具体的元素先接收，然后逐级向上传播，直到不具体的元素；

**五、BOM 对象有哪些，列举 window 对象？**

**建议回复：**

1. window 对象，是 JS 的最顶层对象，其他的 BOM 对象都是 window 对象的属性；
2. document 对象，文档对象；
3. location 对象，浏览器当前URL信息；
4. navigator 对象，浏览器本身信息；
5. screen 对象，客户端屏幕信息；
6. history 对象，浏览器访问历史信息；

**六、请简述 AJAX 及基本步骤？**

**分析：**

对于这种纯概念题，建议面试前再写一遍原生 AJAX 函数的封装过程，一定要在理解的基础上去背，才会在面试时有良好发挥。

**建议回复：**

简述 AJAX：AJAX即“Asynchronous Javascript And XML”（异步 JavaScript 和 XML），是指一种创建交互式网页应用的网页开发技术。通过在后台与服务器进行少量数据交换，AJAX 可以使网页实现异步更新。这意味着可以在不重新加载整个网页的情况下，对网页的某部分进行更新。

**AJAX 基本步骤：**

1. 初始化ajax对象
2. 连接地址，准备数据
3. 发送请求
4. 接收数据（正在接收，尚未完成）
5. 接收数据完成

[?](https://www.jb51.net/article/108375.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | //初始化ajax对象  **var** xhr = xhr = **new** XMLHttpRequest();  //连接地址，准备数据  xhr.open(“方式”,”地址”,是否为异步);  //接收数据完成触发的事件  xhr.onload =**function**(){}  //发送数据  xhr.send(); |

**七、HTTP 状态消息 200 302 304 403 404 500 分别表示什么？**

**分析：**

“带着理解去记忆” 是对付概念题最好的法宝，此外多看看延展的知识点，也是有益无害。

**建议回复：**

* 200：请求已成功，请求所希望的响应头或数据体将随此响应返回。
* 302：请求的资源临时从不同的 URI 响应请求。由于这样的重定向是临时的，客户端应当继续向原有地址发送以后的请求。只有在 Cache-Control 或 Expires 中进行了指定的情况下，这个响应才是可缓存的。
* 304：如果客户端发送了一个带条件的 GET 请求且该请求已被允许，而文档的内容（自上次访问以来或者根据请求的条件）并没有改变，则服务器应当返回这个状态码。304 响应禁止包含消息体，因此始终以消息头后的第一个空行结尾。
* 403：服务器已经理解请求，但是拒绝执行它。
* 404：请求失败，请求所希望得到的资源未被在服务器上发现。
* 500：服务器遇到了一个未曾预料的状况，导致了它无法完成对请求的处理。一般来说，这个问题都会在服务器端的源代码出现错误时出现。

**八、同步和异步的区别?**

**分析：**

又是概念题。

**建议回复：**

* 首先同步异步于阻塞非阻塞并没有关系。同步异步主要是事情做完以后，如何进行处理、或者说关注的是一种消息通信机制。
* 同步的情况下，是由处理消息者自己去等待消息是否被触发；
* 而异步的情况下是由触发机制来通知处理消息者；

**举例：** 比如在上课时，你问老师一个问题，这个问题可能需要花费一点时间去思考，这个时候老师可能：

* 思考，思考……，好了，有答案了；
* 这个问题需要一点时间，你先做点别的，等我想好了，去找你。

第一种就是同步，第二种就是异步。所以同步异步可以说是对被请求方来说的，被请求者使用什么方式来告知处理结果。

* 阻塞非阻塞，主要是对于请求者而言的。
* 阻塞：发出请求等待结果返回，然后再处理后续的事情；
* 非阻塞：发出请求不等待结果返回，可以接着做后续的事情；

举例，还是上一个例子：

* 老师在使用同步思考的时候，你可以静静的等待老师给出答案，也可以边做自己的事情边等待老师的回答，当然这时候你需要时刻去关注老师是否已经想好了，在程序中需要进行轮询了。乀(ˉεˉ乀)
* 老师使用异步的方式，这个时候老师告诉你可以先去做别的，好了就通知你，那么你可以去做点别的，然后监听事件就行，当然你也可以很轴，我就不做别的！我要一直等着老师“想好了”的事件发生。
* 所以同步可以是阻塞的也可以是非阻塞的，异步也是如此。

**九、GET和POST的区别，何时使用POST？**

**分析：**

像是数据交互类的概念题在面试时被问到的机率之多，也反应出它在工作中的实用价值。所以没别的，好好学吧，好好背！

**建议回复：**

GET和POST的区别：

GET：一般用于查询数据，使用URL传递参数，由于浏览器对地址栏长度有限制，所以对使用get方式所发送信息的数量有限制，同时浏览器会记录（历史记录，缓存）中会保留请求地址的信息，包括地址后面的数据。get 只能发送普通格式（URL 编码格式）的数据。

POST：一般用于向服务器发送数据，对所发送的数据的大小理论上是没有限制，浏览器会缓存记录地址，但是不会记录 post 提交的数据。post 可以发送纯文本、URL编码格式、二进制格式的字符串，形式多样。

在以下情况中，请使用 POST 请求：

1. 以提交为目的的请求（类似语义化，get 表示请求，post 表示提交）；
2. 发送私密类数据（用户名、密码）（因为浏览器缓存记录特性）；
3. 向服务器发送大量数据（数据大小限制区别）；
4. 上传文件图片时（数据类型区别）；

**十、 AJAX 的局限性?**

**建议回复：**

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。不会执行你的 JS 脚本，只会操作你的网页源代码；
4. 跨域请求有一定限制。解决方式：jsonp；

**十一、new 操作符具体干了什么呢?**

**分析：**

原理概念题。

**建议回复：**

当使用 new 操作符调用构造函数，函数实际会经历如下步骤：

1. 创建一个新对象；
2. 把函数中上下文（作用域）对象this指向该对象；
3. 执行代码，通过this给新对象添加属性或方法；
4. 返回对象；

**十二、null 和 undefined 的区别？**

**分析：**

经典的概念题。

**建议回复：**

null： null表示空值，转为数值时为0；

undefined：undefined表示"缺少值"，就是此处应该有一个值，但是还没有定义。

1. 变量被声明了，但没有赋值时，就等于undefined。
2. 对象没有赋值的属性，该属性的值为undefined。
3. 函数没有返回值时，默认返回undefined。

**十三、JavaScript 原型，原型链 ? 有什么特点？**

**建议回复：**

JavaScript 原型： 每创建一个函数，函数上都有一个属性为 prototype，它的值是一个对象。 这个对象的作用在于当使用函数创建实例的时候，那么这些实例都会共享原型上的属性和方法。

原型链： 在 JavaScript 中，每个对象都有一个指向它的原型（prototype）对象的内部链接（proto）。这个原型对象又有自己的原型，直到某个对象的原型为 null 为止（也就是不再有原型指向）。这种一级一级的链结构就称为原型链（prototype chain）。 当查找一个对象的属性时，JavaScript 会向上遍历原型链，直到找到给定名称的属性为止;到查找到达原型链的顶部（Object.prototype），仍然没有找到指定的属性，就会返回 undefined。

**十四、实现对数组进行乱序**

**分析：**

这道题检验了数组的 sort() 方法，因为是乱序，所以还需要用到 Math.random() 产生随机数，打乱排序规律。

**建议回复：**

在写代码前，先把实现原理介绍一下，然后再开始写代码：

[?](https://www.jb51.net/article/108375.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5 | **var** a = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];  **var** sign = 1;  a.sort(**function**(a, b) {  **return** Math.random() - 0.5  }); |

**十五、实现一个函数 clone()，可以对 JavaScript 中的5种主要的数据类型（包括 Number、String、Object、Array、Boolean）进行值复制。**

**分析：**

这道题考察了以下知识点：

1. 使用 typeof 判断值得类型；
2. 使用 toString 区分数组和对象；
3. 递归函数的使用；

建议回复：

[?](https://www.jb51.net/article/108375.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | function clone(obj) {   //判断是对象，就进行循环复制   if (typeof obj === 'object' && typeof obj !== 'null') {    // 区分是数组还是对象，创建空的数组或对象    var o = Object.prototype.toString.call(obj).slice(8, -1) === "Array" ? [] : {};    for (var k in obj) {     // 如果属性对应的值为对象，则递归复制     if(typeof obj === 'object' && typeof obj !== 'null'){      o[k] = clone(obj[k])     }else{      o[k] = obj[k];     }    }   }else{ //不为对象，直接把值返回    return obj;   }   return o;  } |