1.js数据类型

1.1种类

基础数据类型：boolean、number、string、null、undefined、symbol

symbol ES6新加，初始化时不需要使用new操作符

复杂数据类型：object

1.2怎么判断数据类型

1.2.1.判断基础数据类型：

typeof一元运算符(一元运算符优先级高，后面有与/或运算等可以连着，不需要使用括号括起来),后面的括号可以省略,不是函数

null和object返回的是object,其余都是本身字符串，string->string,number->number,boolean->boolean,undefined->undefined,symbol->symbol,function->function

1.2.2.判断对象

eg判断数组：

(1) [].constructor === Array (是function类型)

(2) Array.isArray([])方法 (IE9+,Firefox4+,Safari5+,Opera10.5+,Chrome):当不传递参数或者传递参数为基本数据类型时，返回false。

Array.prototype指向Array函数的原型对象-是数组；Object.prototype的原型对象是对象，没有\_\_proto\_\_属性。

问题：如果一个页面存在两个iframe，且在两个iframe之间判断的话，则constructor和isArray可能会出现问题。两个页面有两个window对象，contructor构造函数在两个不同的window对象上，这时使用这两种方式判断就会出现问题？(每一个window对象都会创建一个Object对象等)可以使用第三种方式

(3) Object.prototype.toString.call([]) === '[object Array]' (识别内置对象时往往十分有用，但对于自定义对象请不要使用这种方法:是指参数中不要识别自定义对象，否则结果都为’[object Object]’)

Eg:Object.prototype.toString返回的是判断的内置函数类型，如[object Object];

Array.prototype.toString.call([])返回的是””空字符串；

Date.prototype.toString.call(new Date())返回的是”” "Sun Oct 14 2018 00:57:41 GMT+0800 (中国标准时间)";因为Array和Date函数的原型对象重写了toString的方法。

自定义构造函数a，a.prototype.toString.call([])===’[object Array]’,a.prototype是一个对象，没有重写toString方法，即调用Object.prototype的toString方法。

(4)instanceof:检测对象的原型链是否指向构造函数的prototype对象的。(同在不同的iframe和windows间检测失效)

(5)duck type

比如我们不知道一个对象是否是数组，我们可以判断这个对象的length是否是数字，是否有join,push等数组的函数方法，通过特征判断对象是否属于某些类型，有时也会用到。(但是使用duck type判断基本类型时会有问题：基本类型有对应的封装对象的属性和方法，这时判断不准确。)

1.3数组

1.3.1.数组都有哪些方法

push,pop,shift,unshift,reverse,sort,concat,slice,splice,indexOf,lastIndexOf,forEach,map,filter,every,some,reduce,reduceRight

keys,values,entries,copyWithin,fill,find,findIndex,flat,flatMap,includes,join,toString,toLocalString

1.3.2.数组的方法哪些对原数组是有影响的

push,pop,shift,unshift,reverse,sort,splice

1.3.3.数组如何去重

(1).创建一个空数组，循环原数组，使用indexOf判断值在新数组中是否存在，不存在则push到新数组中。

(2).直接在原数组中进行去重，在原数组中比较是否有重复，使用splice截取原数组。

(3).使用对象属性不能重复的特点，创建一个对象，将数组中的值一次扔到对象中。

(4).先对原数组排序，然后循环，相邻的数组有重复的，则去除

(5).用一行代码去重数组：[...a]=new Set([原数组])

1.3.4.特别的数组-伪数组：伪数组有哪些？怎么转换为数组？

dom选择器选出的节点列表

函数中的arguments对象

jquery选择器选出的jquery对象也是伪数组

转换：(1)创建一个数组，将伪数组中的每一项依次push进去，得到新数组

(2) 使用Array.prototype.slice.call(args)

(3) [].slice.call(args)

slice源码的处理方式？？？

1. 判断第二个参数end是否为undefined，是则复制为length
2. 定义一些存储的变量size，数组clone等
3. 设置start数值，判断start是否大于0，小于0则Math.max(0,length-start)
4. 设置upTo数值，判断是否为数值类型，取值Math.min(lenght,end)
5. 判断end是否小于0，小于则end=end+length
6. 求数组长度size=end-start
7. 判断size是否小于0，小于则返回空数组
8. 使用charAt判断this对象是否为字符串，循环取值this.charAt(i)加入数组
9. 为数组或类数组时循环使用this[i]加入数组

splice源码的实现？？

V8实现？？

1. 判断参数是否==2，等于2则只删除不添加。使用slice方法截取前部分和后部分，使用concat连接成新数组替换原来的数组，返回删除的数组
2. 参数个数<2，则原数组=[0,index]，返回index到最后的数组。
3. 参数个数>2，创建新数组为添加的元素，当len=0时，原数组=前部分+新添加数组+后部分；len>0，原数组=前部分+新添加数组+后部分(后部分第一个值要+删除数组的长度)，并返回删除的数组。

对比：(1) splice() 方法会直接对数组进行修改。

slice() 不对数组进行修改，原数组不会改变。

(2)splice() 返回值为被操作的值。slice() 以新的字符串返回被提取的部分。

(3) splice先执行删除操作，删除指定个数的元素，然后再插入elements（元素或数组）,他的每次删除都涉及大量元素的重新排列，而在插入元素时引入队列来管理。所以splice()的效率不高。

1.4.字符串

1.4.1.字符串有哪些常见方法？

charAt,concat,fromCharCode,indexOf,lastIndexOf,match,repalce.search.slice,split,substring,substr,toLowerCase,toUpperCase,toString,valueOf,toLocalLowerCase,toLocalUpperCase

1.4.2.substring、substr、slice有什么区别，有哪些相同点和不同点？

听详细讲解

slice(start,end)：截取字符串或数组；参数为负数时使用数组长度+参数，第一个参数绝对值大于第二个参数绝对值则返回空数组。

substring(start,end)：只截取字符串；参数为负数时值为0，并且数值小的数为截取开始的参数，自动换位置。

substr(start,len)：只截取字符串；第一个参数可以为负，长度+参数；第二个参数为负时不转换，截取出空字符串。

1.5数字

1.5.1.如何判断是不是NaN

isNaN()可以判断，更准确的用途不是判断是不是NaN,而是判断传进去的参数是否可以转换为数字。对于不能转为数字的参数结果会为false，所以首先判断是不是数字，再使用isNaN。

更好更简单的判断方式：使用全等和自身进行判断。(NaN与任何数都不相等)

使用ES6的is()方法判断：Object.is(NaN,NaN)===true(此方法不能判断数据是NaN格式，因为比较两个字符串本身也是相等的)

1.6 布尔类型

1.6.1.在JS中有哪些转换成boolean类型是false？

"",null,undefined,+0/-0,NaN,false(只有这几种)，其他都为true

对象转为boolean类型也为true

eg：new Boolean(false)转为布尔类型是true(是对象)

1.6.2.如何将一个值快速转为boolean类型？

使用!!两个非操作符

2.new操作符

2.1.new操作符的执行过程？

共四步：(1)创建一个空对象

(2)修改this指针，将this指针指向创建出来的对象

(3)运行构造函数中所有的代码

(4)将创建出来的对象作为返回值返回

2.2.使用new操作符时，如果没有参数，括号是可以省略的

2.3.对构造函数来讲，里面的返回值应该是什么？

new操作符创建的一定是对象，

构造函数中return的值如果不是对象，则会被忽略

return的是对象的话，则会返回return的对象。

(单例模式经常使用这种结构)

3. JS中的继承是如何实现的(看js高级程序设计的详解)

(1).原型链继承：将父类的实例作为子类的原型。

有对象A和B，希望B继承A

B.prototype = new A() ???

缺点：只能继承原型对象上的属性，且属性共享。只要对其中一个进行改变，相关的都会改变。没有办法实现多继承。

(2).构造函数继承：在B中将A的所有代码运行一遍

在B的构造函数中写A.apply(this,arguments)

优点：可以实现多继承；

(子类实例共享父类引用属性; 创建子类实例时，可以向父类传递参数)

缺点：但是只能继承构造函数中的属性，没办法继承原型对象上的属性；

所有继承的属性都是相互独立的，不共享，占内存。

(原型链和构造函数继承都有存在的问题，所以常用组合式继承)

(3).组合式继承(伪经典继承)

优缺点：综合原型链继承和构造函数继承的优点。但是父类的构造函数被执行两次，第一次prototype时执行了一次，第二次子类调用构造函数的时候又执行了一次，如果构造函数很复杂则影响效率；相同的属性在子类实例中存在，在父类的原型对象上又存在一份，调用时会存在同名屏蔽问题，多占用内存。

(4).原型式继承

创建一个对象B，让B直接继承A对象

ES5中新增create：解决问题:直接让一个对象继承另一个对象

原型链、构造函数、组合式继承都是让两个类之间相互继承关系，而不是对象

(5).寄生式继承

原型式继承和寄生式继承不是实现两个构造函数间的继承，是实现两个对象间的继承的。

(6).组合寄生式继承(完美继承)

各种继承方式的优缺点，应用场景？

4.defer和async的区别？

Defer是延迟执行，同时并行加载多个js，然后按照顺序依次执行。

Async是异步加载，同defer一样会异步加载js，但是下载完之后立即执行，不一定按照写的引入顺序执行。

考点：(1)Js代码的加载和执行会阻塞页面的html渲染

(2)正常的js下载是同步的依次下载，但是这两个属性会改用异步的方式加载外部的js文件。

(3)async是在外部js加载完成后，如果浏览器空闲，并且load事件触发的时候，则会在load事件触发之前执行。Defer是在js加载完成后，整个文档解析完成后才会执行。执行时间不一样。(如果将一个defer标签放在head里，则和放在body后没有区别，只是会节省下载时间，但是执行时间没有办法确定)

5.ES6中的promise

5.1promise实现和方法？

创建promise对象：new promise()，传两个参数，一个成功时的回调函数，一个失败时的回调函数。

使用时调用then或者catch方法。

5.2当页面有多个promise时，怎么才能等到所有的promise都完成后才继续调用后面的东西？

在promise中有个特殊的方法all

5.3 promise中还有rice方法，all和rice的区别？

参数都是数组，数组都是promise对象。

传给回调函数的值不同？？？

5.4 promise中then和catch的返回值是什么？

返回值都是新的promise的实例，可以链式调用

5.5 在promise的then或catch中显示的返回一个非promise对象，这个时候如何执行？

返回值还是一个promise对象，但是这种情况下then会立即执行，而不会等到promise中的语句执行完。

5.6在promise中异常的捕获时如何实现的，和try catch有什么不同？

对于try catch来讲，是尝试后面的代码能否运行，是一种异常的捕获机制；但对于promise里面存在了两个状态，一个是resolve状态，一个是reject状态，只要发生了异常和错误，promise会执行进入reject决议，直接进入catch进行处理。Try catch不能捕获异步的异常，如不能捕获ajax异常，只能捕获回调函数中的异常(或者可以将ajax包在try catch中)；promise可以捕获异步异常。

6.es6新增的哪些，项目中用到了哪些？

Import和export使用，模块化

解构赋值

Let和const

Class和extend

函数的拓展：箭头函数，传参数的默认值，rest参数，双冒号的使用。

字符串的拓展：字符串模版，增加的方法

数组的拓展：from(),of(),拓展运算符…[]转为成将数组分割出的每一项

对象的拓展：属性的简洁表达式，解构赋值，新方法is、keys、values、ass、entries

新增class类

新增promise对象

7.babel的实现：项目中没办法使用es6，要使用babel转换，babel每一个转换是怎么实现的？

怎么将es6转位es5？

Let和const在赋值的时候怎么转换？箭头函数怎么转换？解构和拓展怎么转换？

8.箭头函数和普通函数有什么区别？

没有arguments对象，this指向发生了转变

9.JS中的事件

9.1事件是怎么实现的？

(1)在标签中嵌入dom

(2)先获得节点，将事件的绑定从html中挪到了js中

(3)事件监听-先获得节点，使用addEventlistener对某个事件进行监听，将处理的函数传给function。

9.2.三种事件绑定的实现都什么问题？

第三种事件监听存在浏览器兼容的处理。写通用事件监听的方法

9.3.事件委托

一个列表，上面每个列表上都有类似的事件，如何提升性能，使用事件委托。

9.4.事件流

一个完整的js事件流是从window开始再回到window的过程，3个阶段：捕获-目标-冒泡。

9.4.1如果对于同一个dom元素，既在捕获中注册了同一个事件，又在目标和冒泡中注册了同一个事件，当事件被触发时，整体的事件执行顺序是怎样？

正常事件是按照事件的dom流的执行顺序处理的，但是如果事件处于目标阶段，这时事件的执行顺序决定了绑定事件是书写的顺序。

9.5.stopPropofation()和preventDefault()的区别？

阻止事件冒泡和阻止默认行为。

在IE下要处理兼容性问题。

9.6.target和currentTarget的区别？

Target指的是属性指向的目标dom节点；currentTarget指向当前对象的dom节点。

10.网络请求

10.1 http的状态码？

分为5类：1\*\*，表示接受到请求，正在处理。

2\*\*，成功。如200,ok。

3\*\*，服务器重定向响应，主要是为了完成某些特别的指令指定的一些动作完成后续的处理。

4\*\*，客户端错误。

5\*\*，服务端错误。

常见的状态码什么意思？

301，304，400，401，403，404，500，502，503，504

10.2说明在网页中输入url敲回车都发生了什么事？

(1)浏览器查域名的ip地址：dns解析-浏览器的缓存、系统缓存、路由缓存

(2)解析完，浏览器会向服务器发起请求。--服务器会进行重定向的响应，指浏览器会向输入的地址的重定向的响应—浏览器会跟踪重定向的响应取到地址—服务器处理浏览器发送的地址并返回http的响应，包含html的代码—浏览器接受到html代码，可能包括很多东西如图片音频cssjs等，浏览器渲染(涉及异步请求的发送js的解析等)-页面呈现给用户。

(具体的重定向中的协议及缓存怎么处理等过程？？)

10.3跨域的原理和实现方式？

(1)jsonp跨域：通过script标签实现。

只能get请求，存在安全隐患。从外部直接加载js填充到内部

(2)修改document.domain，再通过iframe的方式。在一个页面中嵌入iframe，然后两个页面都通过js强制将document.domain改成主域。

(3)设置location.hash,通过三个页面实现跨域。使用window.hashChange方法监听。

(4)使用window.name和iframe配合

Name属性值在不同的页面中一旦加载就会一直存在，而且name支持很长很长的的值，最大2M。

(5)postMessage不仅能实现跨域，还能实现多窗口的通讯。传输数据最好使用JSON.stringify序列化转成字符串，有些浏览器不支持其它格式的数据；设置域的时候/表示当前窗口同源，\*表示任何域。

(6)CORS:主要是在后端进行设置，但是如何前台请求带cookie的话，使用js和jquery都需要在前端设置一个属性值？

(7)Ngix反向代理：也是在服务端完成。

在配置Ngix时做一个代理服务器，把它做成跳板，比如现在要访问domain2，但是只能访问domain1，则在Ngix上自动指向Domain2。(实现方式)

(8)中间件的代理，使用node搭服务器，和Ngix相似。

代理的原理：同源策略其实是浏览器的安全策略，而不是http协议的一部分，在服务器调用接口只是走http协议，所以就不存在跨域的问题。

(9)Websocket:HTML5提供的新的交互方式。

常见的通讯都是基于websocket。

Js原生写起来很麻烦，可以使用websocket.js框架。

(10)image跨域：请求图片，可以往服务器发送信息，没办法接受信息。

怎么通过css进行数据的统计？当使用加载一张图片时，如何图片不存在，则浏览器不会请求这张图片，只不过在代码中存在。只要当图片能显示或display属性变化时，浏览器才会请求图片。如何请求的事跨域的图片，服务器自然能收到跨域的信息，也就能收到数据的统计。所以使用Image跨域，将图片设置到背景图中，在背景图中切换隐藏和显示方式，让浏览器发送请求，再通过服务器将请求的信息统计出来就实现了数据统计。

问题：

1对于三大框架，生命周期是什么样的？

2.虚拟dom事如何实现的？虚拟dom有什么缺点？

3.对于这些数据绑定的框架mvvm，双向数据绑定是如何实现的？

4.jquery框架的构造函数是怎么写的？ajax方法本身的返回值是什么？jquery的dom选择器如何进行优化？和原生的选择器最后实现的性能有多大？

5.工具函数lodash-防抖函数和截流函数是如何实现的？fp是如何实现的？

6.xss攻击的防御？

7.webworker和websocket的使用有什么问题？解决什么问题？应用场景？

8.http和https有什么区别？

9.amd模块和cmd模块，umd模块各有什么优缺点？

10.前段的性能优化？移动端的性能优化？是如何实现的？应该注意哪些问题？

11.etag是干嘛的？

12.在js中如何实现快排？

13.http2.0有哪些新特性？用到了哪些功能？

14.闭包的实现？作用？

15.cookies使用？弊端？怎么删除？

16.在浏览器本地各种缓存怎么使用？有什么区别？

17.html5提供了什么新标签？干嘛用的？

18.哪些操作会造成内存的泄露？

19.call和apply有什么区别？

20．pc网页开发，列举ie和其他浏览器有什么不一样的特性？

21.想创建一个web应用，从服务端推送数据到客户端有哪些实现方式？

22.原生ajax怎么实现？

(1)创建XMLHttpRequest对象用于在后台与服务器交换数据，使用ActiveXObject兼容IE浏览器。

(2)调用open方法请求服务器

(3)发送post方法时设置setRequestHeader添加http头，设置发送信息的内容编码类型

(4)调用onreadystatechange监听服务器相应状态(readyState服务器相应的状态，status服务器返回的状态码，responseText获得字符串形式的相应数据)

(5)调用send方法发送数据。

23.ajax有什么优缺点？在IE下有什么问题？get和post有什么区别？

24.对于html，attribute和property有什么区别

25.在网络协议中7/5层的网络模型？每一层都实现了什么功能？在网络模型中304缓存如何实现？如何使用？

26.在html5中为什么要使用!DOCTYPE HTML5

问题：{}.toString.call([])在chrome中报错可能是因为chrome将{}以为代码块，不是对象类型。