Js总结

总结总地址：https://yuchengkai.cn/docs/zh/frontend/

# 1. 说说你对作用域链理解

作用域链的作用是保证执行环境里有权访问的变量和函数是有序的，作用域链的变量只能向上访问，变量访问到window对象即被终止，作用域链向下访问变量是不被允许的。

简单的说，作用域就是变量与函数的可访问范围，即作用域控制着变量与函数的可见性和生命周期。

# 2. 请解释什么是事件代理

事件代理（Event Delegation），又称之为事件委托。是 JavaScript 中常用绑定事件的常用技巧。顾名思义，“事件代理”即是把原本需要绑定的事件委托给父元素，让父元素担当事件监听的职务。事件代理的原理是DOM元素的事件冒泡。使用事件代理的好处是可以提高性能。

可以大量节省内存占用，减少事件注册，比如在table上代理所有td的click事件就非常棒。

可以实现当新增子对象时无需再次对其绑定。

$(“#div”).on(“click”,”.a”,function(){

$(this).removeClass(“a”).addClass(“b”) ; // 这里的this实际就是被委托的事件；

})

[https://www.cnblogs.com/liugang-vip。、/p/5616484.html](https://www.cnblogs.com/liugang-vip/p/5616484.html) 详解网址

# 3.谈谈对this的理解

This是执行上下文的对象

1）函数自调用，this指向window

2）对象调用函数，this指向调用函数的对象

3）构造函数 new Person();this指向构造函数的实例

4）Call，apply强制绑定this

# 4. 事件模型

冒泡型事件：当你使用事件冒泡时，子级元素先触发，父级元素后触发

捕获型事件：当你使用事件捕获时，父级元素先触发，子级元素后触发

DOM事件流：同时支持两种事件模型：捕获型事件和冒泡型事件

阻止冒泡：在W3c中，使用stopPropagation（）方法；在IE下设置cancelBubble = true

阻止捕获：阻止事件的默认行为，例如click - <a>后的跳转。在W3c中，使用preventDefault（）方法，在IE下设置window.event.returnValue = false

## 1）事件机制

**e**.type事件类型 ; **e**.**target** 事件目标；**e**.currentTargrt====**this**. 事件侦听对象；  
  
 捕获阶段 目标阶段 冒泡阶段  
 外 内  
 | |  
 内 目标 外

## 2）点击事件

### （1）dom0

添加点击事件：Div.onclick=function(){};

删除点击事件：Div.onclick=null；

### （2）dom2

IE8及以上

添加点击事件：div.addEventlinster(“click”,clickHandler);

删除点击事件：div.removeEventLinster(“click”,clickHandler);

IE8以下

添加点击事件：div.attachEvent(“onclick”,clickHandler);

删除点击事件：div.detachEvent(“onclick”,clickHandler);

## 3）事件通用语法

### （1）dom0

IE8以上

阻止默认事件：return false；

阻止默认事件：e.returnValue=false;

取消冒泡事件：e.cancelBubble=true;

### （2）dom2

IE8以上

阻止默认事件：e.preventDeafult();

取消冒泡事件：e.stopPropagation();

### （3）开始全局捕获

div.setCapture();

### （4）关闭全局捕获

div.releaseCapture();

## 4）常用基本事件类型

1）select 选择触发事件

选择时触发。

2）scroll 滚动条事件

滚动滚轮时触发。

3）resize 窗口大小改变事件

窗口大小改变时触发。

4）焦点事件

Focus获得焦点时触发 blur 失去焦点时触发 ；主要针对input类型

## 5）鼠标滚轮事件

### （1）事件

1）click 点击事件

2）dblclick 双击事件

3）mousedown 鼠标按下

4）mouseup 鼠标抬起

5）mousemove 鼠标移动

6）mouseover 鼠标进入

7）mouseout 鼠标离开

8）mouseleave 鼠标首次移出时

9）mouseenter 鼠标首次移入时

### （2）鼠标位置坐标

1）e.clientX ,e.clientY;光标距视口的水平和垂直坐标；

2）e.pageX,e.pageY;光标距离页面的水平垂直坐标；

e.pageX=e.clientX+document.documentElement.scrollLeft;

e.pageY=e.clientY+document.documentElement.scrollTop;

3）e.screenX.e.screenY；光标相对于屏幕的坐标

4）e.offsetX,e.offsetY;光标相对于目标元素的坐标；

### （3）鼠标按键判断

1）IE8以后：button： 0代表鼠标左键，1代表中间键，2代表鼠标右键。

2）IE8以前：button： 1代表鼠标左键，4代表中间键，2代表鼠标右键。

### （4）滚轮事件

mousewheel

## 6）键盘事件

1）keydown 按下触发

2）keyup 抬起触发

3）keypress 按住一直触发

## 7）dom0和dom2的区别

1）dom0 适应于ie8以下及其以上，dom2适应ie8以上；

2）dom0 调用：div.onmousemove=function(){};

dom2调用：div.addEventlister(“mousemove”,function( ){ });

3）dom0移除： div.onmousemove=null;

Dom2移除：div.removeEventlister（）；

# 5. 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

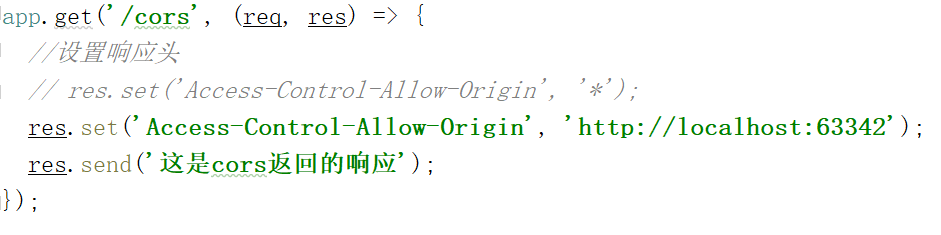
# 6.如何解决跨域问题?

## 1）jsonp

优点明显，但是只支持get请求。

<https://blog.csdn.net/sdnxyr/article/details/78707718> 访问上面的网站有详细的说明。

## 2）设置Access-Control-Allow-\*头



CORS（Cross Origin Resource Share）对方服务端设置响应头；

设置相应头：”Access-Control-Allow-Origin；

CORS请求默认不发送Cookie和HTTP认证信息。如果要把Cookie发到服务器，一方面要服务器同意，指定Access-Control-Allow-Credentials字段。

## 3）document.domain 跨子域

有局限性，一级域名一只才可以使用。

在请求“www.sojson.com/shiro” 的时候，写上如下代码：

if(document.domain !='sojson.com'){

document.domain = 'sojson.com';

}

然后在上传的地址“cdn.sojson.com” 的Iframe 文件内写上一样的代码。

if(document.domain !='sojson.com'){

document.domain = 'sojson.com';

}

这样上传就是在相同的域下了，有的人是少了第一步，其实仔细想想，正所谓跨域，就是指你使用功能的时候，当前域和使用的域不一样，所以要在前面就指定就可以了。故保证一致。

## 4）postMessage()

HTML5的postMessage()方法允许来自不同源的脚本采用异步方式进行有限的通信，可以实现跨文本档、多窗口、跨域消息传递。适用于不同窗口iframe之间的跨域

<https://blog.csdn.net/huijiaaa1/article/details/80952315> 此为详解网址。

Access-Control-Allow-Origin设置允许跨域的白名单，在白名单里的跨域请求是允许的。

# 7. 异步加载JS的方式有哪些？

## 1）promise对象

### 1）理解

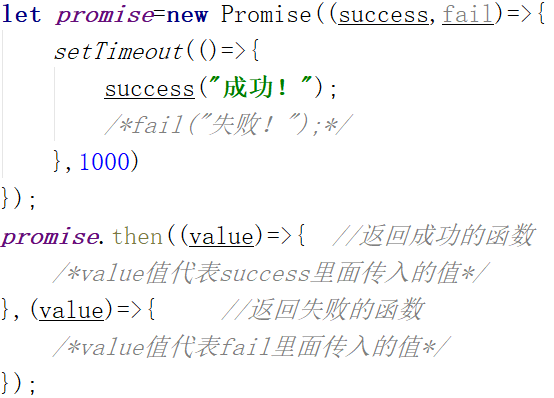
Promise对象;代表了未来某个将要发生的事件（通常是一个异步操作）；

有了promise对象，可以将异步操作以同步的流程表达出来，避免了层层嵌套的回调函数（俗称回调地狱）；

ES6的promise是一个构造函数，用来生成promise实例；

### 2）创建promise的步骤

如下代码：



### 3）核心思想

通过异步任务的执行结果去动态的修改promise对象的状态来决定调用成功或者失败的函数。

Promise对象then方法执行完有一个默认的返回值：promise对象（状态为默认成功，没有传参）

## 2）async函数（源自ES7）

### 1）概念

真正意义上去解决异步回调的问题，同步流程表达异步操作；

### 2）语法

Async function fun(){

Await 异步操作；

Await 异步操作；

}

特点：

当遇到await等待，等待当前的异步任务执行成功

无需调用任何方法，会根据上一次的执行结果自定判断是否需要执行下一次的任务；

Await返回值resolve（data）；

代码如下：



# 8. 那些操作会造成内存泄漏？

1）闭包

2）定义变量的时候不适用var

3）没有及时清除定时器

# 9. XML和JSON的区别？

## 1）数据体积方面

JSON相对于XML来讲，数据的体积小，传递的速度更快些。

## 2）数据交互方面

JSON与JavaScript的交互更加方便，更容易解析处理，更好的数据交互。

## 3）数据描述方面

JSON对数据的描述性比XML较差。

## 4）传输速度方面

JSON的速度要远远快于XML。

# 10. 常见web安全及防护原理

## 1）sql注入原理

就是通过把SQL命令插入到Web表单递交或输入域名或页面请求的查询字符串，最终达到欺骗服务器执行恶意的SQL命令

## 2）总的来说有以下几点：

永远不要信任用户的输入，要对用户的输入进行校验，可以通过正则表达式，或限制长度，对单引号和双"-"进行转换等

永远不要使用动态拼装SQL，可以使用参数化的SQL或者直接使用存储过程进行数据查询存取

永远不要使用管理员权限的数据库连接，为每个应用使用单独的权限有限的数据库连接

不要把机密信息明文存放，请加密或者hash掉密码和敏感的信息

## 3）XSS原理及防范

Xss(cross-site scripting)攻击指的是攻击者往Web页面里插入恶意html标签或者javascript代码。比如：攻击者在论坛中放一个看似安全的链接，骗取用户点击后，窃取cookie中的用户私密信息；或者攻击者在论坛中加一个恶意表单，当用户提交表单的时候，却把信息传送到攻击者的服务器中，而不是用户原本以为的信任站点

## 4）XSS防范方法

首先代码里对用户输入的地方和变量都需要仔细检查长度和对”<”,”>”,”;”,”’”等字符做过滤；其次任何内容写到页面之前都必须加以encode，避免不小心把html tag 弄出来。这一个层面做好，至少可以堵住超过一半的XSS 攻击

## 5）XSS与CSRF有什么区别吗？

XSS是获取信息，不需要提前知道其他用户页面的代码和数据包。CSRF是代替用户完成指定的动作，需要知道其他用户页面的代码和数据包。要完成一次CSRF攻击，受害者必须依次完成两个步骤

登录受信任网站A，并在本地生成Cookie

在不登出A的情况下，访问危险网站B

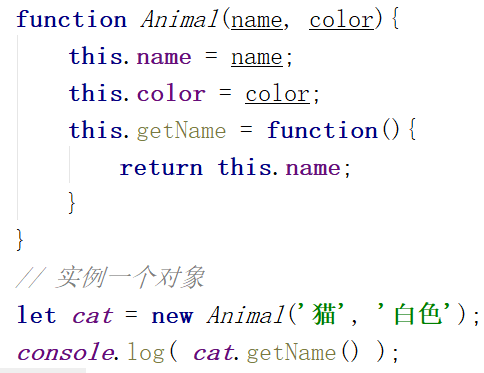
## 6）CSRF的防御

服务端的CSRF方式方法很多样，但总的思想都是一致的，就是在客户端页面增加伪随机数

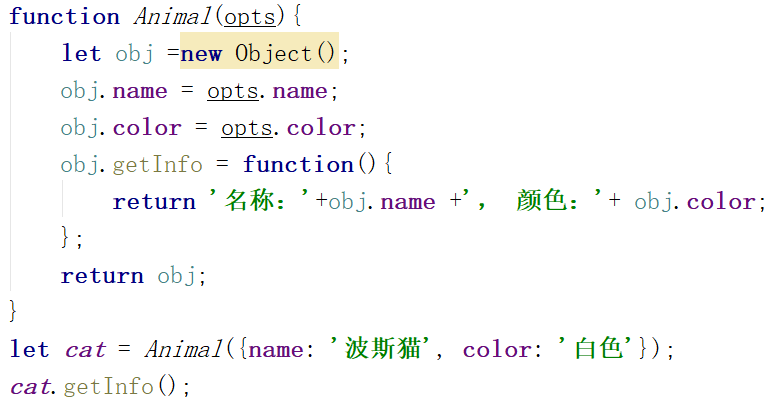
通过验证码的方法

# 11.用过哪些设计模式？

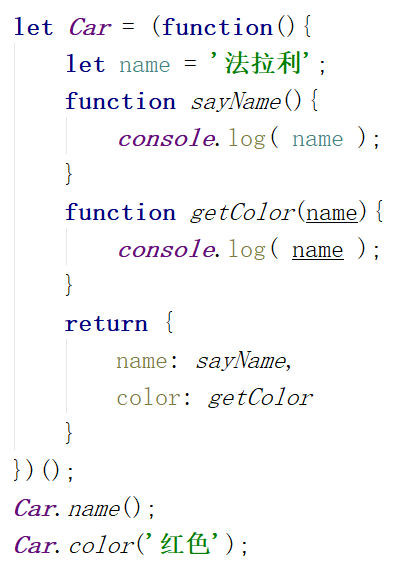
## 1）构造函数模式



## 2）工厂模式



## 3）模块模式



## 4）混合模式

## 5）单例模式

# 12. 为什么要有同源限制？

同源策略指的是：协议，域名，端口相同，同源策略是一种安全协议。

举例说明：比如一个黑客程序，他利用Iframe把真正的银行登录页面嵌到他的页面上，当你使用真实的用户名，密码登录时，他的页面就可以通过Javascript读取到你的表单中input中的内容，这样用户名，密码就轻松到手了。

# 13. javascript有哪些方法定义对象

对象字面量： var obj = {};

构造函数： var obj = new Object();

Object.create(): var obj = Object.create(Object.prototype);

# 14. 你觉得jQuery源码有哪些写的好的地方

jquery源码封装在一个匿名函数的自执行环境中，有助于防止变量的全局污染，然后通过传入window对象参数，可以使window对象作为局部变量使用，好处是当jquery中访问window对象的时候，就不用将作用域链退回到顶层作用域了，从而可以更快的访问window对象。同样，传入undefined参数，可以缩短查找undefined时的作用域链；

jquery将一些原型属性和方法封装在了jquery.prototype中，为了缩短名称，又赋值给了jquery.fn，这是很形象的写法；

有一些数组或对象的方法经常能使用到，jQuery将其保存为局部变量以提高访问速度；

jquery实现的链式调用可以节约代码，所返回的都是同一个对象，可以提高代码效率。

<https://www.jianshu.com/p/17a83794d031> 访问上面网址，可以查看jquery源码的简单分析，应付面试足矣。

# 15. 介绍js有哪些内置对象？

1）Object 是 JavaScript 中所有对象的父对象

2）数据封装类对象：Object、Array、Boolean、Number 和 String

3）其他对象：Function、Arguments、Math、Date、RegExp、Error

# 16. 说几条写JavaScript的基本规范？

1）不要在同一行声明多个变量

2）请使用===/!==来比较true/false或者数值

3）使用对象字面量替代new Array这种形式

4）不要使用全局函数

5）Switch语句必须带有default分支

6）If语句必须使用大括号

7）for-in循环中的变量 应该使用var关键字明确限定作用域，从而避免作用域污染

# 17. JavaScript有几种类型的值？你能画一下他们的内存图吗？

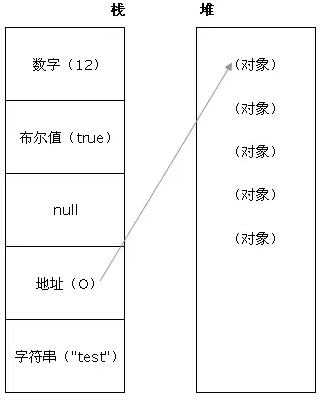
栈：原始数据类型（Undefined，Null，Boolean，Number、String）

堆：引用数据类型（对象、数组和函数）

两种类型的区别是：存储位置不同；

原始数据类型直接存储在栈(stack)中的简单数据段，占据空间小、大小固定，属于被频繁使用数据，所以放入栈中存储；

引用数据类型存储在堆(heap)中的对象,占据空间大、大小不固定,如果存储在栈中，将会影响程序运行的性能；引用数据类型在栈中存储了指针，该指针指向堆中该实体的起始地址。当解释器寻找引用值时，会首先检索其在栈中的地址，取得地址后从堆中获得实体。



# 18. 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

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）

由JSON字符串转换为JSON对象的时候可以用eval，var obj =eval('('+ str +')')

# 19. 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。

# 20. 同步和异步的区别?

同步：浏览器访问服务器请求，用户看得到页面刷新，重新发请求,等请求完，页面刷新，新内容出现，用户看到新内容,进行下一步操作；

异步：浏览器访问服务器请求，用户正常操作，浏览器后端进行请求。等请求完，页面不刷新，新内容也会出现，用户看到新内容。

# 21. 渐进增强和优雅降级

渐进增强 ：针对低版本浏览器进行构建页面，保证最基本的功能，然后再针对高级浏览器进行效果、交互等改进和追加功能达到更好的用户体验。

优雅降级 ：一开始就构建完整的功能，然后再针对低版本浏览器进行兼容。

# 22. 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。

# 23. DOM操作——怎样添加、移除、移动、复制、创建和查找节点?

## 1）创建新节点

createDocumentFragment() //创建一个DOM片段

createElement() //创建一个具体的元素

createTextNode() //创建一个文本节点

## 2）添加、移除、替换、插入

appendChild()

removeChild()

replaceChild()

insertBefore() //在已有的子节点前插入一个新的子节点

## 3）查找

getElementsByTagName() //通过标签名称

getElementsByName() //通过元素的Name属性的值(IE容错能力较强，会得到一个数组，其中包括id等于name值的)

getElementById() //通过元素Id，唯一性

# 24. 如何编写高性能的Javascript？

## 1）加载运行

javascript在浏览器中的性能，大概是开发者面对最重要的问题。因为javascript的阻塞特征，也就是在javascipt运行的过程中，其他的事情并不能被浏览器处理。所以javascript运行的越长，浏览器等待的时间就越长。所以应该

把脚本放在底部（通常就是标签的前面）。

由于每次http请求都会有性能负担，加载一个100KB的脚本要比加载4个25KB的快。所以把多个脚本合成一个脚本，通过打包工具来实现。

动态创建script标签，用它下载并执行代码。

## 2）数据

每一种数据都有读写的负担，全局变量，局部变量，数组，对象访问的代价都不一样

如果关心运行速度，尽量使用局部变量限制对象和数组的使用（原型链越深，搜索的速度越慢）。

作用域链搜索是从里往外的，在一个函数内部，首先是找局部变量，找不到，再去外面一层寻找变量。所以在函数的内部用全局变量，可以把值赋给局部变量。

不要使用with，with会改变作用域链，影响性能。

try-catch的catch表达式也是和with的效果一样，但是try-catch是一个非常有用的东西，所以在用try-catch的时候确保你了解可能发生的错误。

闭包，通常在一个函数激活对象和运行上下文一同销毁，但是闭包却不会。所以闭包意味着更多的内存开销。在大型网页应用的时候，需要注意这个问题。

属性嵌套，嵌套对象成员会造成 。

对于一些常用的对象，数组项存入局部变量中，访问局部变量的速度会大大高于那些元始的变量

## 3）DOM

在网页中，DOM操作的代价昂贵，通常是一个性能瓶颈。在对javascript和DOM之间，我们可以理解它们各自是一个房间，在DOM的房间的门外有一个守门人，每次从javascript到DOM需要缴纳过门费，所以操作DOM的次数越多，代价也就越昂贵，所以我们要尽量减少过门的次数，尽量在javascript门这边完成。

减少dom访问，在Javascript端做尽可能的事

对重复访问的dom对象使用局部变量保存

注意页面重绘和重排，把dom操作批量化（不要分步修改DOM，用一次性全部修改）

使用事件委托

## 4）选择算法

在javasctip中有很多种循环方法，while、do-whlie、for、for-in，forEach并且还有很多条件语句，if-else、switch，因为Javascript的资源有限，所以选择算法的时候优先选择最优的；

while、do-whlie、for 循环的性能相似，无所谓选择谁；

相对性能来说for-in是最浪费性能的一种（因为每次迭代都需要搜索原型的属性)，除非你要遍历一个属性不知道的对象，否则不要用。

switch比if-else要快；

forEach是基于函数的迭代，虽然很便利，但是基于函数的迭代是普通基于循环的八倍。

## 5）定时器和AJAX

当javascript和用户操作一起运行的时候，用户界面是不能响应操作的，原因是当脚本执行的时候，UI不会随着用户的交互而跟新，此时的用户交互被放进队列里面，当原先的脚本执行完之后才会执行这个交互。

定时器可以安排代码推迟去执行，可以把大的脚本分解成一个个小的任务，所以合理的使用定时器，可以提高用户的体验。

AJAX请求的数据尽量用json数据。

减少页面AJAX的请求次数。

# 25. jQuery和Zepto的区别？各自的使用场景？

区别：[文档](file:///C:\Users\MACHENIKE\AppData\Roaming\Microsoft\Word\zepto与jquery的区别.docx)；打开上面的文档即可。

## 2）Zepto的点透问题如何解决？

# 26. 什么是window对象? 什么是document对象?

## 1）window对象---

 代表浏览器中的一个打开的窗口或者框架，window对象会在<body>或者<frameset>每次出现时被自动创建，在客户端JavaScript中，Window对象是全局对象，所有的表达式都在当前的环境中计算，要引用当前的窗口不需要特殊的语法，可以把那个窗口属性作为全局变量使用，例如：可以只写document，而不必写window.document。同样可以把窗口的对象方法当做函数来使用，如：只写alert（），而不必写window.alert；

    window对象实现了核心JavaScript所定义的全局属性和方法。

    Window对象的Window属性和self属性引用都是他自己。

## 2）document对象

代表整个HTML文档，可以用来访问页面中的所有元素 。

每一个载入浏览器的HTML文档都会成为document对象。document对象使我们可以从脚本中对HTML页面中的所有元素进行访问。

document对象是window对象的一部分，

可以通过window.document属性对其进行访问

HTMLDocument接口进行了扩展，定义HTML专用的属性和方法，很多属性和方法都是HTMLCollection对象，其中保存了对锚、表单、链接以及其他可脚本元素的引用。

# 27.JSON

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

JSON字符串转换为JSON对象:

var obj =eval('('+ str +')');

var obj = str.parseJSON();

var obj = JSON.parse(str);

JSON对象转换为JSON字符串：

var last=obj.toJSONString();

var last=JSON.stringify(obj);

# 28.attribute和property的区别是什么？

attribute是dom元素在文档中作为html标签拥有的属性；

property就是dom元素在js中作为对象拥有的属性。

对于html的标准属性来说，attribute和property是同步的，是会自动更新的

但是对于自定义的属性来说，他们是不同步的



# 29. 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

# 30. 什么是“前端路由”?什么时候适合使用“前端路由”? “前端路由”有哪些优点和缺点?

## 1）什么是前端路由？

路由是根据不同的 url 地址展示不同的内容或页面；

前端路由就是把不同路由对应不同的内容或页面的任务交给前端来做，之前是通过服务端根据 url 的不同返回不同的页面实现的。

## 2）什么时候使用前端路由？

在单页面应用，大部分页面结构不变，只改变部分内容的使用；

## 3）前端路由有什么优点和缺点？

**优点**

用户体验好，不需要每次都从服务器全部获取，快速展现给用户；

**缺点**

使用浏览器的前进，后退键的时候会重新发送请求，没有合理地利用缓存；

单页面无法记住之前滚动的位置，无法在前进，后退的时候记住滚动的位置。

# 31.做的项目中，有没有用过或自己实现一些 polyfill 方案（兼容性处理方案）？



[代码文件链接](file:///C:\Users\MACHENIKE\AppData\Roaming\Microsoft\Word\jsown.js);大概总结了几个

# 32. 页面重构怎么操作？

网站重构：在不改变外部行为的前提下，简化结构、添加可读性，而在网站前端保持一致的行为。

也就是说是在不改变UI的情况下，对网站进行优化，在扩展的同时保持一致的UI。

**对于传统的网站来说重构通常是：**

表格(table)布局改为DIV+CSS

使网站前端兼容于现代浏览器(针对于不合规范的CSS、如对IE6有效的)

对于移动平台的优化

针对于SEO进行优化

深层次的网站重构应该考虑的方面

减少代码间的耦合

让代码保持弹性

严格按规范编写代码

设计可扩展的API

代替旧有的框架、语言(如VB)

增强用户体验

通常来说对于速度的优化也包含在重构中

压缩JS、CSS、image等前端资源(通常是由服务器来解决)

程序的性能优化(如数据读写)

采用CDN来加速资源加载

对于JS DOM的优化

HTTP服务器的文件缓存

# 33. 列举IE与其他浏览器不一样的特性？

## 1）表示发生事件：

（1）非IE浏览器下，事件对应的函数有一个隐藏的变量e，表示发生事件。

（2）IE下，不需要e变量，window.event表示发生事件。

解决方案：用e||window.event来兼容。

## 2）触发事件对象（触发事件的元素被认为是目标target）：

（1）IE下，window.event对象有srcElement属性，但没有target属性。

（2）Firefox下，e对象有target属性，但没有srcElement属性。

（3）Chrome下，e对象同时具有target和srcElement属性。

解决方案：event.srcElement ? event.srcElement : event.target来兼容。

## 3）按键码（字符代码）：

（1）IE下，window.event对象只有keyCode属性。

（2）FireFox下，e对象有which和charCode属性。

（3）Opera下，e对象有keyCode和which属性。

（4）Chrome下，e对象有keyCode、which和charCode属性。

解决方案：用e.keyCode || e.which || e.charCode来兼容。

## 4）阻止事件的默认行为：

（1）IE 中阻止事件的默认行为需要将window.event.returnValue属性设置为false。

（2）非IE阻止事件的默认行为需要调用 e.preventDefault() 方法。

解决方案：条件判断浏览器是否具有event.preventDefault再做相应处理。

## 5）阻止事件冒泡：

（1）IE阻止事件冒泡需要设置window.event.cancelBubble = true。

（2）非IE阻止事件冒泡需要调用e.stopPropagation()。

解决方案：条件判断浏览器是否具有event.stopPropagation再做相应处理。

## 6）IE中消除list的原点

仅需margin:0即可达到最终效果；FIrefox需要设置margin:0;padding:0以及list-style:none

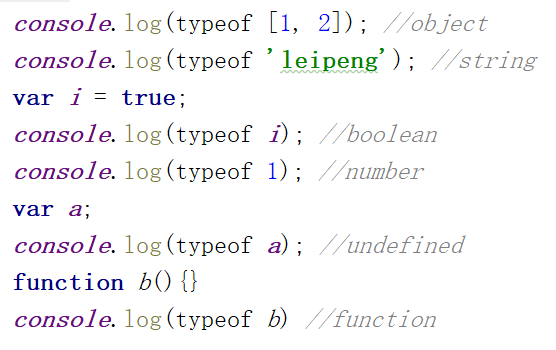
# 34. 是否了解公钥加密和私钥加密

第一种用法：公钥加密，私钥解密。---用于加密  
第二种用法：私钥加密，公钥解密。---用于签名  
有点混乱，不要去硬记，总结一下:  
**你只要想：**  
既然是加密，那肯定是不希望别人知道我的消息，所以只有我才能解密，所以可得出公钥负责加密，私钥负责解密；  
既然是签名，那肯定是不希望有人冒充我发消息，只有我才能发布这个签名，所以可得出私钥负责签名，公钥负责验证。  
**同一种道理，我在换种说法：**  
私钥和公钥是一对，谁都可以加解密，只是谁加密谁解密是看情景来用的：  
第一种情景是签名,使用私钥加密,公钥解密,用于让所有公钥所有者验证私钥所有者的身份并且用来防止私钥所有者发布的内容被篡改.但是不用来保证内容不被他人获得。  
第二种情景是加密,用公钥加密,私钥解密,用于向公钥所有者发布信息,这个信息可能被他人篡改,但是无法被他人获得。  
**比如加密情景：**  
如果甲想给乙发一个安全的保密的数据,那么应该甲乙各自有一个私钥,甲先用乙的公钥加密这段数据,再用自己的私钥加密这段加密后的数据.最后再发给乙,这样确保了内容即不会被读取,也不会被篡改.

# 35. WEB应用从服务器主动推送Data到客户端有那些方式？

参考网站：<https://blog.csdn.net/shuo1992/article/details/59477055>

# 36. javascript的typeof返回哪些数据类型

4

判断数据所有数据类型，并返回正确的结果,包括NaN。

let *a* = "string";  
let *b* = 111;  
let *c* = {};  
let *d* = [1, 2, 3];  
let *e* = function () {  
 *console*.log("eee");  
};  
let *f* = undefined;  
let *g* = null;  
let *h* = new Date();  
let *i* = /test/;  
let *j* = true;  
let *k* = NaN;  
let *l* = Infinity;  
function *judgeType*(option) {  
 let result= Object.prototype.toString.call(option);*//返回值[object Undefined]* result=result.split(" ")[1];  
 result=result.substring(0,result.length-1).toLowerCase();*//进行切割转换，如：undefined* if(result==="number"){  
 if(option<Infinity){ *// 只有数值才会小于无穷大数，NaN比较返回false* result="number";  
 }else if(option.toString()==="Infinity"){  
 result="Infinity";  
 }else {  
 result=NaN.toString();*//将NaN赋值给result，并转换为字符串* }  
 }  
 *console*.log(result);*//打印输出*}  
*judgeType*(*l*);

# 37. window.onload 和document ready的区别

window.onload 是在dom文档树加载完和所有文件加载完之后执行一个函数document.ready原生中没有这个方法，jquery中有 $().ready(function),在dom文档树加载完之后执行一个函数（注意，这里面的文档树加载完不代表全部文件加载完）。

$(document).ready要比window.onload先执行;

window.onload只能出来一次，$(document).ready可以出现多次。

# 38. JavaScript是一门什么样的语言，它有哪些特点？

javaScript一种直译式脚本语言，是一种动态类型、弱类型、基于原型的语言，内置支持类型。它的解释器被称为JavaScript引擎，为浏览器的一部分，广泛用于客户端的脚本语言，最早是在HTML网页上使用，用来给HTML网页增加动态功能。JavaScript兼容于ECMA标准，因此也称为ECMAScript。

**基本特点**

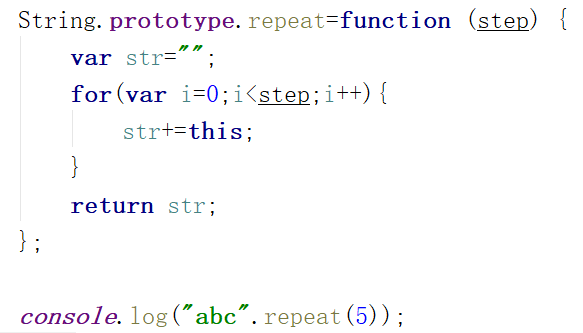
1）是一种解释性脚本语言（代码不进行预编译）。

2）主要用来向HTML（标准通用标记语言下的一个应用）页面添加交互行为。

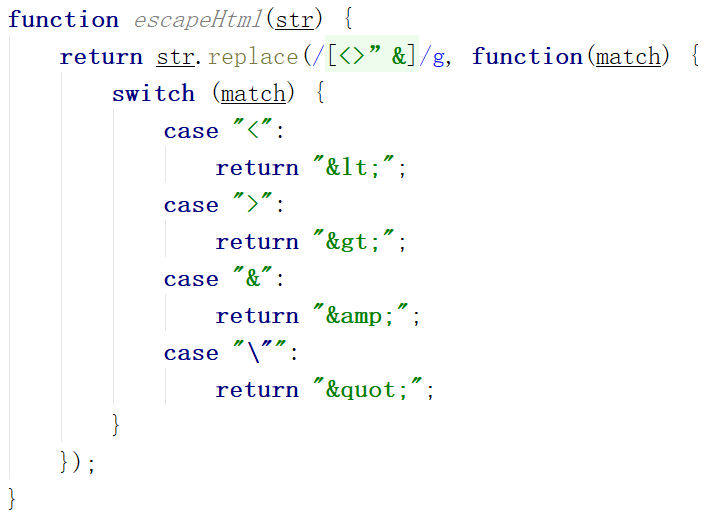
3）可以直接嵌入HTML页面，但写成单独的js文件有利于结构和行为的分离。

跨平台特性，在绝大多数浏览器的支持下，可以在多种平台下运行（如Windows、Linux、Mac、Android、iOS等）。

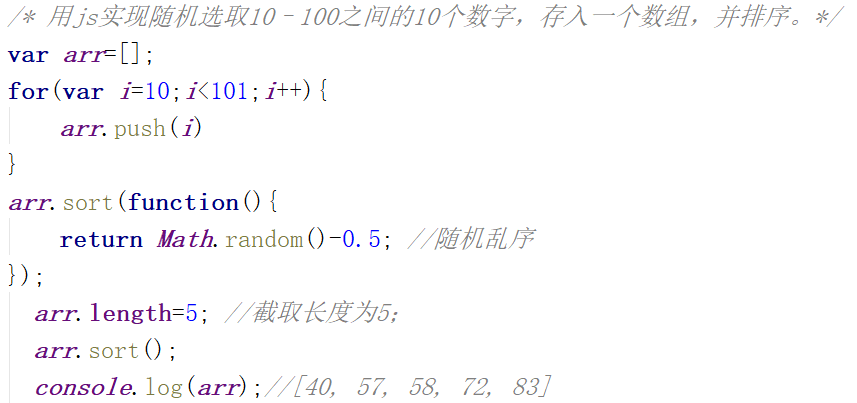
# 39. 在string对象上定义一个repeat函数，这个函数接受一个整数参数，来明确字符串需要重复几次，这个函数要求字符串指定重复的次数



# 40. 为了保证页面输出安全，我们经常需要对一些特殊的字符进行转义，请写一个函数escapeHtml，将<, >, &, “进行转义



# 41. 用js实现随机选取10–100之间的10个数字，存入一个数组，并排序



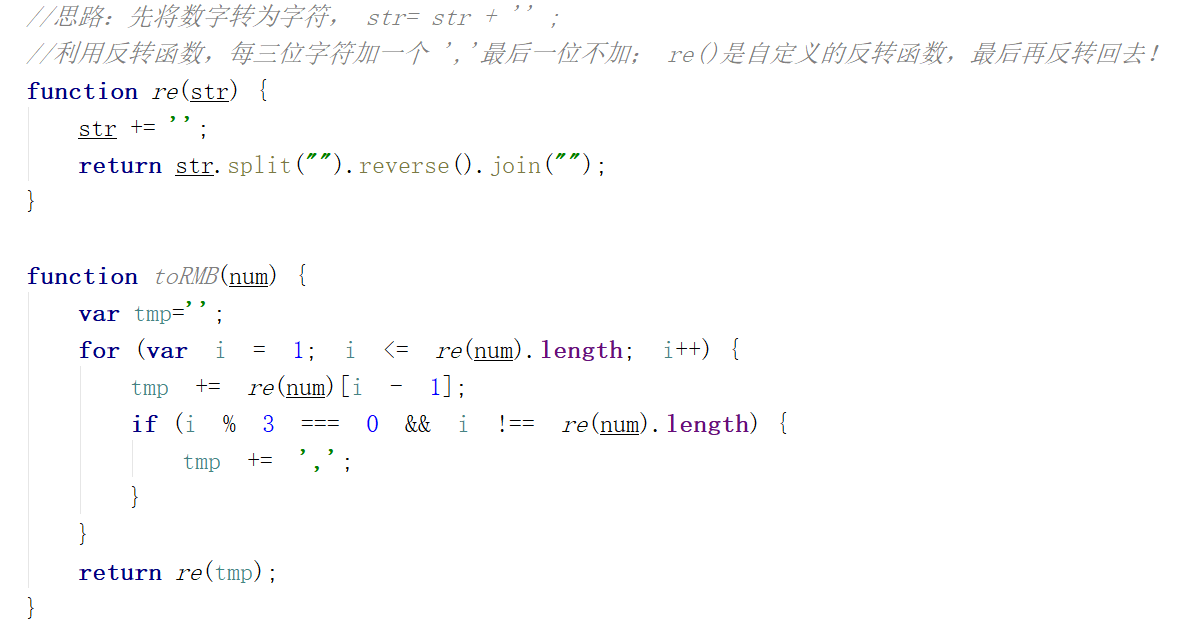
# 42.Data



# 43. 判断一个字符串中出现次数最多的字符，统计这个次数

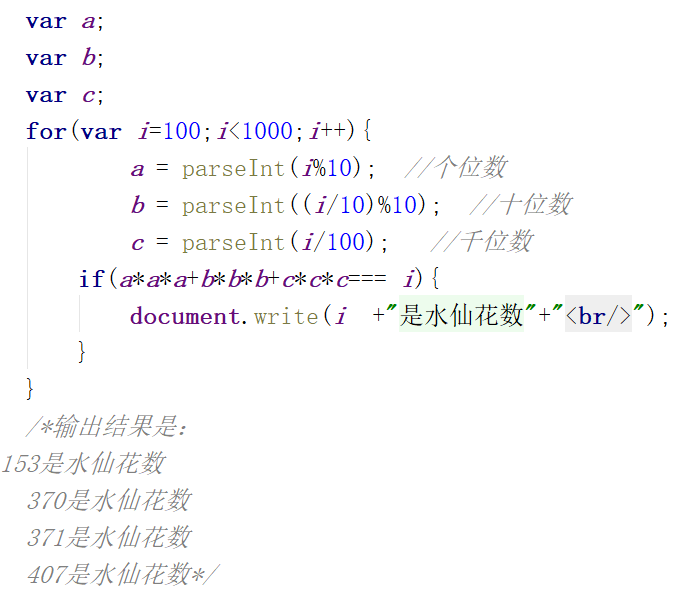


# 44. 将数字 12345678 转化成 RMB形式 如： 12,345,678



# 45. js实现：循环输出1000以内水仙花数

水仙花数是指一个 n 位数 ( n≥3 )，它的每个位上的数字的 n 次幂之和等于它本身。（例如：1^3 + 5^3+ 3^3 = 153）;



# 46.BOM

1）window.open（） ,弹出框，利用setintervel可以设置一定时间一个弹窗。

2）window.close ()关闭窗口。

3）window.location.reload(),重新加载，用在刷新页面。

4）window.location.href="";服务端动态返回地址，

5）window.location.assign();跳转地址替换的地址，有历史记录

6）window.location.repalce（）;跳转替换的地址，没有历史记录

7）window.location.search 获取地址栏？和后面的语句

8）window.location.hash 地址中#和后面的语句

# 47.js中的垃圾回收机制

<https://www.cnblogs.com/zhwl/p/4664604.html> 点击上述链接即可

# 48.你是如何优化自己的代码的

<https://blog.csdn.net/weixin_36401046/article/details/68937179> 点击上述链接即可；

# 49.为什么扩展javascript内置对象不是好的做法？

因为你不知道哪一天浏览器或javascript本身就会实现这个方法，而且和你扩展的实现有不一致的表现。到时候你的javascript代码可能已经在无数个页面中执行了数年，而浏览器的实现导致所有使用扩展原型的代码都崩溃了。

# 50. 在JS中有哪些会被隐式转换为false

Undefined、null、关键字false、NaN、零、空字符串。