# JS面试题

1. 获取一堆name名相同的div，点击弹出显示第几个div

var boxes = document.getElementsByName("div");

for(var i = 0; i < boxes.length; i++) {

boxes[i].index = i;

boxes[i].onclick = function() {

alert(this.index);

}

}

1. abcdefgsdds 获取里面字面最多的字母

var str = “abcdefgsdds”;

var obj = {};

var arr = [];

var letter;

for(var i = 0,len = str.length;i<len;i++){

letter = str[i];

if(!obj[letter]){

obj[letter] = 1;

}else{

obj[letter]++;

}

}

var max\_key,max\_num=0;

for(key in obj){

if(max\_num <obj[key]){

max\_num = obj[key];

max\_key = key;

}

}

document.write(“字母：”+max\_key+”　次数：”+max\_num);

1. abcdefghi 获取ghi怎么实现

"abcdefghi".substring(6)

1. ‘asdjhklzxc’ 截取‘asd’

"asdjhklzxc".substring(3,-1)

1. 翻转字符串

"asd".split("").reverse().join("")

1. 删除数组第一个元素

arr.shift()

1. [1,2,1,3,2,3,5,6,5] 去除重复元素

var arr = [1,2,1,3,2,3,5,6,5] ;

var newArr = [];

for(var i = 0; i < arr.length; i++) {

if(newArr.indexOf(arr[i]) == -1) {

newArr.push(arr[i]);

}

}

alert(newArr)

1. 正则表达式
2. JS中有几种数据类型分别是哪几种？

Number,String,Boolean,Object,Null,Undefined

1. 说一下拖拽的原理？怎么样实现多个DIV在拖拽时有相撞让它们改变属性与属性值和他们的位置？

mousedown,mousermove,mouseup事件

e.clientX,e.clientY 鼠标位置

碰撞检测

1. 用JS创建一个a标签，在body中显示，点击在新窗口打开

var alink = document.createElement("a");

alink.innerHTML = "google";

alink.href = "http://www.google.com";

alink.target = "\_blank";

document.body.appendChild(alink);

1. 鼠标滚轮动画

function mousewheel(element,up,down) {

element.onmousewheel = fn;

if(window.addEventListener) {

element.addEventListener("DOMMouseScroll",fn,false)

}

function fn(e) {

var e = e || window.event;

console.log(e);

if(e.wheelDelta > 0 || e.detail < 0) {

up();

}else {

down();

}

e.cancelBubble = true;

e.stopPropagation();

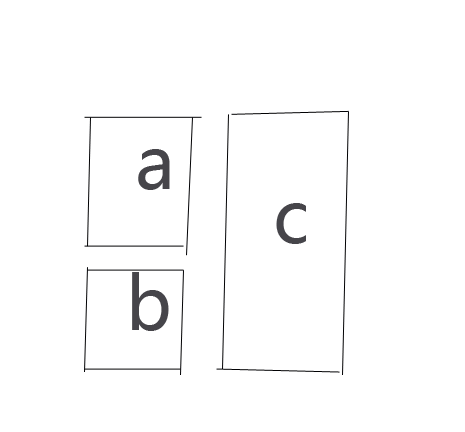
}

}

mousewheel(document,function(){},function(){})//函数调用

1. 定位右下角广告

使用css position:fixed

1. 鼠标经过A A变大 其他的位置不动

A绝对定位，脱离文档流即可

1. 简述javascript的历史 （略）
2. JS中避免命名冲突的三个方法

命名空间，闭包，匿名函数

1. JS在HTML中的引入方式有哪些？又有什么优缺点？

a外部引入 b内部嵌套 c行内代码 例：<button onclick=”alert(‘hello’)”>按钮</button> 对比css引入

1. 写出5条标识符的命名规范 （）

1.区分大小写；

2.第一个字符必须是一个字母、下划线（\_）或一个美元符号（$）

3..不能含有空格

4.不能以关键字或保留字命名,不能以数字开头

1. 从JS的角度，写出8条以上优化代码的方法？

http://www.haorooms.com/post/web\_xnyh\_jscss

1. 按优先级，对这几种运算符进行排序：！ && \* = >

！ && \*> =

1. 在循环中，break和continue语句的功能有何不同

break 终止整个循环，continue终止当次循环

1. 解释事件冒泡、事件捕获的区别

事件流解释即可

1. 说说什么叫做事件委托

利用事件冒泡的原理，子元素的事件会冒泡到父元素，可以只给父元素添加事件，通过事件目标判断元素。优点：节省内存，动态添加的子元素也包含事件

1. 列举不少于6条的IE与FF脚本兼容性问题，需要写出命令

(1) window.event：

表示当前的事件对象，IE有这个对象，FF没有，FF通过给事件处理函数传递事件对象

(2) 获取事件源

IE用srcElement获取事件源，而FF用target获取事件源

(3) 添加，去除事件

IE：element.attachEvent(“onclick”, function) element.detachEvent(“onclick”, function)

FF：element.addEventListener(“click”, function, true) element.removeEventListener(“click”, function, true)

(4) 获取标签的自定义属性

IE：div1.value或div1[“value”]

FF：可用div1.getAttribute(“value”)

(5) document.getElementByName()和document.all[name]

IE；document.getElementByName()和document.all[name]均不能获取div元素

FF：可以

(6) input.type的属性

IE：input.type只读

FF：input.type可读写

(7) innerText textContent outerHTML

IE：支持innerText, outerHTML

FF：支持textContent

(8) 是否可用id代替HTML元素

IE：可以用id来代替HTML元素

FF：不可以

1. 构造函数有何优缺点？

构造函数可以创建多个对象，

缺点：容易跟普通函数一样调用

1. 用自己的话描述一下“闭包“

函数内部又定义了一个函数，内部函数引用外部函数的变量，就构成了闭包

1. 用自己的话描述一下对“异步“和”同步“的理解

同步：按顺序执行，同一时刻只能执行一个事件

异步：不按顺序执行，同一时刻可以执行多个任务

1. 写倒计时 距离2018年还有 多少天多少小时多少分钟多少秒

var time = document.getElementById("time");

function clock() {

//设置目标日期

var targetDate = new Date(2018,0,1);

var currentDate = new Date();

var remainTime = targetDate - currentDate;

//天数

remainDay = parseInt(remainTime/1000/60/60/24);

//小时数

remainHours = parseInt(remainTime/1000/60/60%24);

//分钟

remainMinutes = parseInt(remainTime/1000/60%60);

//秒

remainSeconds = parseInt(remainTime/1000%60);

time.innerHTML = remainDay + "天"+remainHours+"小时"+remainMinutes+"分钟"+remainSeconds+"秒";

}

setInterval(clock,1000);

1. alert怎样换行

转义字符,换行符\n

alert("大家好\n大家好")

1. Tab切换 （略）
2. 瀑布流 （略）
3. 左侧内容可上下滚动，右侧是滚动条，如何实现布局

固定高度，overflow-y:scroll

1. 向上滚动动画如何实现

document.body.scrollTop = "xxx";

document.documentElement.scrollTop = "xxx";

1. form表单提交

默认提交，form.onsubmit = function(){}可在表单数据提交之前验证

1. javaScript与ECMAScript的区别？

javascript是具体实现，ECMAScript是标准，大纲

1. ==与===的区别？

==等于，判断值是否相等，会有类型转换

===全等 ，判断类型跟值是否相等，类型不会转换

1. 什么是DOM？

文档对象模型

1. Undefined与null的区别？

Undefined无定义，null空

1. 浏览器的兼容性问题 （css 以及js）
2. DOM是什么？谈谈你对它的理解？(略)
3. 有两个对象，var o = {num:1};var p = {num:2};要求书写一段代码，使得o+p===3为true。

Object.prototype.toString = function(){

return this.num;

}

1. 如果给你这么一个页面：顶部有logo什么的，下来是导航条，中上部有图片切换，右边是登录注册，左边有简单的信息展示，中下部是一些新闻，底部是一般的那些联系我们呀什么的传统页尾。这样一个布局加上所有的js效果和新闻链接包括所有的兼容，你大概能多久完成（新鸿儒）；
2. 讲解一下this （作用域）

this的指向：普通函数内的this指向全局变量

构造函数内部this指向新创建出来的对象

对象方法内的this指向的是调用该方法的对象

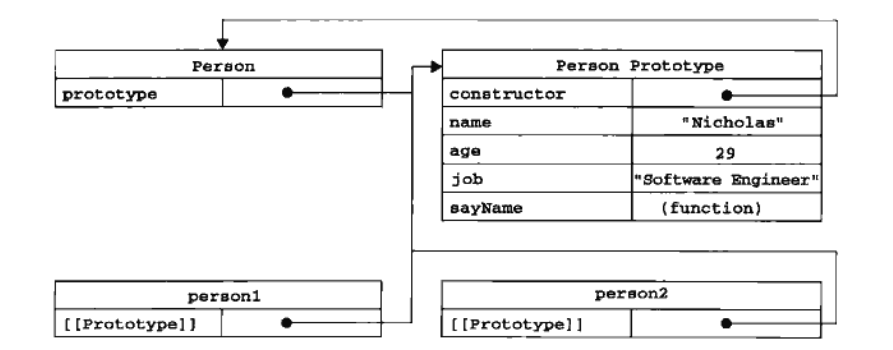
call,apply,bind可以改变this的指向

1. 构造函数

new关键字后紧接的函数即为构造函数，约定首字母大写

1. tween.js (略)
2. 原型，原型链的理解

ECMA5之前没有类的概念，js的继承是基于原型链的



1. 说说前端优化从哪方面着手，具体说明有哪些？
2. html 语义化，符合标准，减少嵌套层数，加快浏览器DOM解析速度
3. css 合并，压缩 少写冗余css代码，使用精灵图 ...
4. js 减少DOM操作，使用变量缓存数据，少用全局变量，做动画的元素都绝对定位...
5. 代码说明call和apply方法的区别

这两个都是函数的方法，可以改变this的指向，fn.call(obj,param1,param2,...)

call传入单个参数

fn.apply(obj,[param1,param2,...])

apply传入一个数组

1. function f(x){

alert(x);(1)

arguments[0]=null;

alert(x);(2)

}

f(5);

1. \_\_\_(2)\_\_\_

答案：5，null

1. var a=new Array(3);

a[0]=function(x){return x\*x;}

a[1]=20;

a[2]=a[0](a[1]);

a[2]的结果\_\_\_400\_\_\_

1. var fn = function(i){

return function(){

return i++;

}

}(10);

var a = fn();

var b = fn();

alert(a);(1)

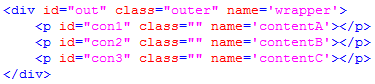
alert(b);(2)

(1)\_10\_\_(2)\_\_11\_

1. 对于js语言，下列说法中（ C ）是正确的？

A.一种标记语言规范的版本名 B.前端编程中，新增特性的一种泛指和统称

C.一种脚本语言 D.可以实现跨平台开发的技术

1. 通过三种方法获取到div中的第2个p标签，并弹出p标签中的内容 （DOM获取元素的接口）
2. 用两种方法，将第上题中的第一个p标签里的内容的颜色设置为红色。

element.style.color = “red”

element.className = “red” .red{color:red}

1. 如何测算数组的长度？ array.length
2. 滚动类效果中，会涉及到的定位属性有哪些，同时写出每个命令的含义。

left,top,right,bottom,scrollLeft,scrollTop

1. 在放大镜效果中，获取鼠标位置的方法，同时说明鼠标位置是针对谁获取的？

e.clientX,e.clientY 相对于窗口坐标

1. 书写简单代码。实现一个效果：有两个按钮，每单击“开始弹框“就会开始弹框，在单击后每2秒钟出现一个弹框，弹框内容是“文本“。如果单击”停止弹框“，则停止弹出。使用两种方法实现。注意防止出现持续性弹框。

（注意在点击按钮之前清掉计时器即可）

1. 事件种类有哪些？分别举出不少于4个的例子？

鼠标事件，键盘事件，触屏事件

1. 访问到tween算法中的Bounce的easeInOut方法？

var startPos = 0;//确定起点位置

var endPos = 500;//确定终点位置

var everyPos = 20;//确定每步移动距离

var step = 0;//确定当前步数

var stepMax = 25;//确定总步数

var tween = {

Bounce:{

easeIn:xxxxx;

easeOut:xxxxx;

easeInOut:xxxx;

}

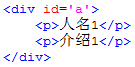
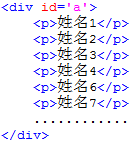
}

tween.Bounce.easeInOut(step,startPos,(endPos-startPos),stepMax)

1. Date对象中getDay()和getDate()分别是什么含义？取值范围是多少？

getDay()获取星期几 0~6

getDate() 获取日期 1~31

1. 设置当前时间为2004年12月12日的代码书写方法？（分写合写两种方式）
2. new Date(2004,11,12)
3. var d = new Date(); d.setFullYear(2004),d.setMonth(11);d.setDate(12);
4. 使用原生DOM对象的方法，实现如下操作：将内容为“人名1”的p标签替换为h1 标签，内容为“姓名” (略)
5. 使用原生DOM对象的方法，实现如下操作：将内容为“姓名5”的p标签插入到“姓名6”的p标签上面，同时删除最后一个p标签（注，不确定共有多少个p标签）（略） 
6. 写出不少于5种Math对象的方法，分别说明含义以及语法

Math.random()随机数0~1之间

Math.abs()绝对值

Math.floor() 向下取整

Math.ceil()向上取整

Math.min();Math.max().....

1. 使用任意一种方法随机生成一个从100-300的整数。

function rnd(min, max) {return parseInt(Math.random() \* (max - min)) + min}

//不包含300

rnd(100,300)

1. 运用原生js，使用闭包的知识点实现Tab切换效果 （略）
2. 运用回调的方法实现大图滚动 （略）
3. 团购倒计时效果的制作 （略）
4. 拖拽效果的实现 （略）
5. 碰壁反弹的思路描述 （略）

**二、代码填空**

1. tab切换效果 代码填空

<style type="text/css">

.show{display:block;}

.select{background:#fcc;}

div {display:none}

</style>

<body>

<h2>

<span id='span1' class='select'>标题1</span>

<span id='span2'>标题2</span>

<span id='span3'>标题3</span>

</h2>

<div id='con1' class='show'>内容1</div>

<div id='con2'>内容2</div>

<div id='con3'>内容3</div>

</body>

<script type="text/javascript">

var cons = document.getElementsByTagName('div');

var spans = document.getElementsByTagName('span')

for (var i=0; i< spans.length ; i++) {

spans[i] .onclick=function () {

for (var j=0; j< cons.length ; j++) {

if ( this == spans[j] ) {

cons[j]. className = “show” ;

spans[j]. className = “select” ;

}

else{

cons[j]. className = “” ;

spans[j]. className = “” ;

}

}

}

}

</script>

1. 用JS截取标签中的abc

<body>

<p id='con'>abcdef</p>

</body>

<script type="text/javascript">

var con = document.getElementById('con');

var string = con. innerText ;

var result = string. substring(0,3) ;

alert(result);

</script>

1. 用js如何获取一个上传文件的扩展名

<input type="file" id="file";

var file = document.getElementById(“file”)

alert(file.files[0].name)

1. 大图滚动全功能代码编写 （略）
2. 用js实现类、继承（继承构造函数内部的属性方法及原型的属性和方法）、私有变量

function Car() {

//构造函数

this.name = "benz"; //属性

var price = "1000000"; //私有变量，外部不能访问，只能通过下面的getPrice方法调用

this.getPrice = function() {

return price;

}

}

Car.prototype.drive = function() {console.log("drive")}//给原型添加drive方法，创建出来的实例都有该方法

var c = new Car();//创建实例对象

1. 图片轮播效果（格式：上边一张大图，下边五张缩略图 1.每个一秒切换一张图片，鼠标移上暂停滚动，鼠标移开继续滚动 2.点击缩略图显示对应的大图）（略）
2. 写一个trim函数，兼容IE firefox chrome（正则）

if(typeof String.prototype.trim !== "function") {

String.prototype.trim = function() {

return this.replace(/^\s\* | \s\*$/g,"");

}

}

直接调用即可，例：var str = “ sdf ”;str.trim();

1. 如何提升网页加载速度

前端角度：合并压缩css,js减少http请求数以及大小，使用精灵图，减少dom操作...

1. 列举事件绑定的方法（原生+JQ addEventlistener与attachEvent的区别）

addEventlistener现代浏览器都支持，ie从ie9开始支持，可以绑定多个事件

attachEvent ie专有，可以绑定多个事件，但this指向全局对象

1. setInterval 与 setTimeout 的区别 (略)
2. split() join() 的区别

split()字符串方法，将字符串转换成数组

join()数组方法，将数组合并成字符串

1. addEventlistener和通过on这种方式绑定的事件（例：ele.onclick = funciton(){}）有什么区别

addEventlistener可以绑定多个事件

on这种方式只能绑定一次，之后绑定的事件会覆盖前面绑定的，实质是对象属性重新赋值

1. DOM事件流

事件捕获，事件目标，事件冒泡

1. ajax请求时，如何解析json数据

三种方式 1.JSON.parse(string)

1. new Function(“return ” + string)()
2. eval(“(”+string+”)”)
3. 写一个获取非行间样式的函数

//获取非行间样式，兼容现代浏览器跟低版本ie

function getStyle(element,property) {

if(window.getComputedStyle) {

return window.getComputedStyle(element,null)[property];

}else {

return element.currentStyle[property];

}

}

1. 闭包是什么，有什么特性，对页面有什么影响

闭包简单理解就是：函数内部有嵌套定义里一个函数，内部函数调用外部函数的局部变量的时候就构成闭包。能保存局部变量，函数执行完不销毁。消耗内存资源

1. 解释jsonp的原理，以及为什么不是真正的ajax

原理：通过script标签的跨域功能，而非XMLHttpRequest

1. window load 和document ready的区别

document ready 实质是DOMContentLoaded事件

所有的节点渲染完毕，就执行

load事件要等到所有的资源加载完毕，如图片，视频，js脚本等

1. 如何使用原生 Javascript 代码深度克隆一个对象（注意区分对象类型）

//对象的深复制

function deepCode(oldObj,newObj) {

var newObj = newObj || {};//如果存在newObj 就newObj,否则用空对象

for(var i in oldObj) {

if(oldObj.hasOwnProperty(i)) {

if(typeof oldObj[i] === "object") {//判断属性是否引用类型的

var c = (oldObj[i].constructor === Array)?[]:{};//如果判断值是否是数组

newObj[i] = deepCode(oldObj[i],c);

//递归

}else {

newObj[i] = oldObj[i];

}

}

}

return newObj;

}