# 做笔记

## 121.你如何优化自己的代码？

代码重用

避免全局变量（命名空间，封闭空间，模块化mvc..）

拆分函数避免函数过于臃肿

注释

## 122.请描述出下列代码运行的结果

function d(){

console.log(this);

}

d();//输出window对象

## 123.以下代码中end字符串什么时候输出

var t=true;

setTimeout(function(){

         console.log(123);

         t=false;

         },1000);

while(t){}

console.log(‘end’);

永远不输出

## 124.请尽可能详尽的解释ajax的工作原理

Ajax的工作原理相当于在用户和服务器之间加了—个中间层，使用户操作与服务器响应异步化。这样把以前的一些服务器负担的工作转嫁到客户端，利于客户端闲置的处理能力来处理，减轻服务器和带宽的负担，从而达到节约ISP的空间及带宽租用成本的目的。

简单来说通过XmlHttpRequest对象来向服务器发异步请求，从服务器获得数据，然后用javascript来操作DOM而更新页面。这其中最关键的一步就是从服务器获得请求数据。要清楚这个过程和原理，我们必须对 XMLHttpRequest有所了解。

## 125.什么是三元表达式？“三元”表示什么意思？

三元运算符:

三元如名字表示的三元运算符需要三个操作数。

语法是 条件 ? 结果1 : 结果2;. 这里你把条件写在问号(?)的前面后面跟着用冒号(:)分隔的结果1和结果2。满足条件时结果1否则结果2。

## 126.浏览器标准模式和怪异模式之间的区别是什么？

所谓的标准模式是指，浏览器按W3C标准解析执行代码；怪异模式则是使用浏览器自己的方式解析执行代码，因为不同浏览器解析执行的方式不一样，所以我们称之为怪异模式

## 127.下面这段代码想要循环输出结果01234，请问输出结果是否正确，如果不正确，请说明为什么，并修改循环内的代码使其输出正确结果

for(var i=0;i<5;++i){

         setTimeout(function(){

                            console.log(i+’’);

                   },100\*i);

}

## 128.关于IE的window对象表述正确的有：（ACD）

A. window.opener属性本身就是指向window对象

B. window.reload()方法可以用来刷新当前页面  应该是location.reload或者window.location.reload

C. window.location=”a.html”和window.location.href=”a.html”的作用都是把当前页面替换成a.html页面

D. 定义了全局变量g；可以用window.g的方式来存取该变量

## 129.下面正确的是  A

A: 跨域问题能通过JsonP方案解决

B：不同子域名间仅能通过修改window.name解决跨域   还可以通过script标签src  jsonp等h5 Java split等

C：只有在IE中可通过iframe嵌套跨域

D：MediaQuery属性是进行视频格式检测的属性是做响应式的

## 130.错误的是 B

A: Ajax本质是XMLHttpRequest

B: 块元素实际占用的宽度与它的width、border、padding属性有关，与background无关

C: position属性absolute、fixed、---relative---会使文档脱标

D: float属性left也会使div脱标

答案C：relative不会脱离文档流

## 131.变量的命名规范以及命名推荐

变量，函数，方法：小写开头，以后的每个单词首字母大写 （驼峰）

构造函数，class：每个单词大写开头

基于实际情况，以动词，名词，谓词来命名。尽量言简意骇，以命名代替注释

## 132.三种弹窗的单词以及三种弹窗的功能

1.alert

//弹出对话框并输出一段提示信息

    function ale() {

        //弹出一个对话框

        alert("提示信息！");

    }

2.confirm

    //弹出一个询问框，有确定和取消按钮

    function firm() {

        //利用对话框返回的值 （true 或者 false）

        if (confirm("你确定提交吗？")) {

            alert("点击了确定");

        }

        else {

            alert("点击了取消");

        }

    }

3.prompt

    //弹出一个输入框，输入一段文字，可以提交

    function prom() {

        var name = prompt("请输入您的名字", ""); //将输入的内容赋给变量 name ，

        //这里需要注意的是，prompt有两个参数，前面是提示的话，后面是当对话框出来后，在对话框里的默认值

        if (name)//如果返回的有内容

        {

            alert("欢迎您：" + name)

        }

    }

## 133.console.log( 8 | 1 ); 输出值是多少？

答案：9

## 134.jQuery框架中$.ajax()的常用参数有哪些？写一个post请求并带有发送数据和返回数据的样例

async是否异步

url请求地址

contentType发送信息至服务器时内容编码类型

data发送到服务器的数据

dataType预期服务器返回的数据类型

type请求类型

success请求成功回调函数

error请求失败回调函数

$.ajax({

            url: "/jquery/test1.txt",

            type: 'post',

            data: {

                id: 1

            },

            success: function(data) {

                alert(data);

            }

        }

## 135.JavaScript的循环语句有哪些？

  for,for..in,while,do...while

## 136.闭包：下面这个ul，如何点击每一列的时候alert其index？

<ul id="test">

    <li>这是第一条</li>

    <li>这是第二条</li>

    <li>这是第三条</li>

</ul>

//js

 window.onload = function() {

        var lis = document.getElementById('test').children;

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

            lis[i].onclick = (function(i) {

                return function() {

                    alert(i)

                };

            })(i);

        };

}

## 137.用正则表达式，写出由字母开头，其余由数字、字母、下划线组成的6~30的字符串？

^[a-zA-Z]{1}[\w]{5,29}$

## 138.列举浏览器对象模型BOM里常用的至少4个对象，并列举window对象的常用方法至少5个

对象：Window document location screen history navigator

方法：Alert() confirm() prompt() open() close()

## 139.写一个函数可以计算 sum(5,0,-5);输出0; sum(1,2,3,4);输出10;

function sum() {

    var result = 0;

    var arr = arguments;

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

        var num = arguments[i];

        if (typeof num=='number') {

            result += num;

        };

    };

    return result;

}

## 140.请写出一个程序，在页面加载完成后动态创建一个form表单，并在里面添加一个input对象并给它任意赋值后义post方式提交到：http://127.0.0.1/save.php

window.onload=function(){

    var form=document.createElement("form");

    form.setAttribute("method", "post");

    form.setAttribute("action", "http://127.0.0.1/save.php");

    var input=document.createElement("input");

    form.appendChild(input);

    document.body.appendChild(form);

    input.value="cxc";

    form.submit();//提交表单

}

## 141.用JavaScript实现冒泡排序。数据为23、45、18、37、92、13、24

//升序算法

function sort(arr){

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

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

            if(arr[j]>arr[j+1]){

                var c=arr[j];//交换两个变量的位置

                arr[j]=arr[j+1];

                arr[j+1]=c;

            }

        };

    };

    return arr.toString();

}

console.log(sort([23,45,18,37,92,13,24]));

## 142.前端代码优化的方法

var User = {

         count = 1，

         getCount：function（）{

                   return this.count;

         }

}

console.log(User.getCount());

var func = User.getCount;

console.log(func());

1 undefined（因为是window对象执行了func函数）;

## 143.下列JavaScript代码执行后，依次alert的结果是

(function test(){

      var a=b=5;

      alert(typeof a);

      alert(typeof b);

})();

alert(typeof a);

alert(typeof b);

答案：number

number

undefined

Number

## 145.下列JavaScript代码执行后，iNum的值是

var iNum = 0;

for(var i = 1; i< 10; i++){

     if(i % 5 == 0){

         continue;

    }

    iNum++;

}

答案：8

## 146.下列JavaScript代码执行后，依次alert的结果是

var obj = {proto: {a:1,b:2}};

function F(){};

F.prototype = obj.proto;

var f = new F();

obj.proto.c = 3;

obj.proto = {a:-1, b:-2};

alert(f.a);

alert(f.c);

delete F.prototype['a'];

alert(f.a);

alert(obj.proto.a);

答案：

1

3

undefined

-1

## 147.下列JavaScript代码执行后的li元素的数量是

<ul>

    <li>Item</li>

    <li></li>

    <li></li>

    <li>Item</li>

    <li>Item</li>

</ul>

var items = document.getElementsByTagName('li');

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

    if(items[i].innerHTML == ''){

        items[i].parentNode.removeChild(items[i]);

    }

}

## 148.程序中捕获异常的方法？

window.error

try{}catch(){}finally{}

## 149.给String对象添加一个方法，传入一个string类型的参数，然后将string的每个字符间价格空格返回，例如：addSpace(“hello world”) // -> ‘h e l l o ?w o r l d’

String.prototype.spacify = function(){

return this.split('').join(' ');

};

## 150.下列控制台都输出什么

**第**1题：

function setName(){

         name="张三";

}

setName();

console.log(name);

答案："张三"

第2题：

//考点：1、变量声明提升 2、变量搜索机制

var a=1;

function test(){

         console.log(a);

         var a=1;

}

test();

答案：undefined

第3题：

var b=2;

function test2(){

         window.b=3;

         console.log(b);

}

test2();

答案：3

第4题：

c=5;//声明一个全局变量c

function test3(){

         window.c=3;

         console.log(c);                 //答案：undefined，原因：由于此时的c是一个局部变量c，并且没有被赋值

         var c;

         console.log(window.c);//答案：3，原因：这里的c就是一个全局变量c

}

test3();

第5题：

var arr = [];

arr[0]  = 'a';

arr[1]  = 'b';

arr[10] = 'c';

alert(arr.length);   //答案：11

console.log(arr[5]);        //答案：undefined

第6题：

var a=1;

console.log(a++);           //答案：1

console.log(++a);           //答案：3

第7题：

console.log(null==undefined);    //答案：true

console.log("1"==1);              //答案：true，因为会将数字1先转换为字符串1

console.log("1"===1);            //答案：false，因为数据类型不一致

第8题：

typeof 1;                   "number"

typeof "hello";                 "string"

typeof /[0-9]/;                 "object"

typeof {};                  "object"

typeof null;              "object"

typeof undefined;         "undefined"

typeof [1,2,3];                  "object"

typeof function(){};       //"function"

第9题：

parseInt(3.14);                          //3

parseFloat("3asdf");               //3

parseInt("1.23abc456");

parseInt(true);//"true" NaN

第10题：

//考点：函数声明提前

function bar() {

    return foo;

    foo = 10;

    function foo() {}

    //var foo = 11;

}

alert(typeof bar());//"function"

第11题：

//考点：函数声明提前

var foo = 1;

function bar() {

         foo = 10;

         return;

         function foo() {}

}

bar();

alert(foo);//答案：1

第12题：

console.log(a);//是一个函数

var a = 3;

function a(){}

console.log(a);////3

第13题：

//考点：对arguments的操作

function foo(a) {

    arguments[0] = 2;

    alert(a);//答案：2，因为：a、arguments是对实参的访问，b、通过arguments[i]可以修改指定实参的值

}

foo(1);

第14题：

function foo(a) {

    alert(arguments.length);//答案：3，因为arguments是对实参的访问

}

foo(1, 2, 3);

第15题

bar();//报错

var foo = function bar(name) {

         console.log("hello"+name);

         console.log(bar);

};

//alert(typeof bar);

foo("world");//"hello"

console.log(bar);//undefined

console.log(foo.toString());

bar();//报错

第16题：

function test(){

         console.log("test函数");

}

setTimeout(function(){

         console.log("定时器回调函数");

}, 0)

test();

结果：

test函数

定时器回调函数​​​​