1. 关于回调函数说法正确的是： A,B,D
2. 回调函数也是一个普通函数
3. 将回调函数分配给一个对象时通常不会立即调用，而是等到合适的时机时才调用
4. XMLHTTPRequest对象的onreadystatechange需要的就是一个回调函数
5. 回调函数即 Callback Function
6. 关于同步执行和异步执行的说法正确的是：­­­­ A,B,C
7. 同步执行即按顺序执行代码，一行执行之后再执行下一行
8. 异步执行时代码不会等待某些任务执行完成，而是开启任务后继续执行后面的代码
9. 同步执行通常意味着等待，使用不当会造成界面卡死，严重影响用户体验
10. setTimeout可以理解为是异步执行的
11. 关于方法设计的说法正确的是：B,C
12. 方法表示功能，方法的代码越多意味着功能越强大，所以方法的代码多一点好
13. 方法的代码不应该太多，最好一个方法只完成一个小任务
14. 方法太多会导致复杂的相互调用，难以阅读和维护，因此方法太小功能拆分的太细也不合适
15. 在面向对象编程中方法的参数和返回值用处不大，因为可以将方法使用的数据都保存到对象的属性中
16. 关于对象设计的说法正确的是： B,C,D
17. 与方法设计相同，对象中的方法和属性越多表示对象越强大，所以对象越大越好
18. 对象太大就失去了面向对象编程的意义，极端地说，一个程序用一个对象搞定，与不使用面向对象编程没啥区别！
19. 对象设计的重点在于设计属性和方法、每一个属性都应该认真考量是否必须，没有必要就不要设置属性
20. 对象应该保持功能单一，最好一个对象只负责做好一件事
21. 关于对象方法的设计说法正确的是： A,BCD
22. 对象的方法不亦过大，应适当拆分，以利于代码重用
23. 方法越大就意味着方法越笨重，可以使用的场合就越少，重用性就差
24. 将代码中可变的部分与不变的部分拆分成不同的方法有利于子类减少代码重复
25. 子类通常只需要重写变化的部分即可，其它不变的都可以直接继承过来

6.关于方法中的this关键字说法正确的是： A

1. 构造函数中的this表示当前正在构造的对象
2. 事件处理函数中的this通常表示当前监听事件的标签元素
3. 使用func.call(obj)或func.apply(obj)能够指定func函数内部的this为obj
4. 使用func.bind(obj)能够得到一个新func，这个func内部的this永远都是obj

7.说出以下两种定义属性的区别

代码一：

obj.age = 3

代码二：

Object.defineProperty(obj, ‘age’, {

value: 3,

enumerable: false,

writable: false,

configurable: false

})

答:1.第一个是给obj设置一个age属性并将3赋给age,

2.第二个是给odj添加一个age属性,同时对属性进行描述,属性值是3,不可以被枚举,不可以被修改,不可以被删除

8.关于构造函数说法正确的是： A,B,D

1. 构造函数的名字首字母应该大写，即Pascal命名法
2. 使用构造函数创建对象需要使用 new 关键字
3. 因为构造函数创建了一个对象，所以构造函数中需要使用return语句返回新对象
4. 使用构造函数的好处是可以简化对象的创建过程

9.以下代码有什么问题？至少列出2点

function Child(name, isMale){

var name = name

var isMale = isMale

this.eat = function(food){

console.log(this.name + ‘ eat ‘ + food)

}

}

答:这是一个构造函数,,没有调用,所以this.eat ,this.name 中this指的是window,而function(food)中food参数无法传值,console.log(this.name + ‘ eat ‘ + food)将打印不出想要的信息

10.以下代码的功能分别是什么？

Object.freeze(obj)

Object.preventExtensions(obj)

Object.seal(obj)

Object.getPrototypeOf(obj)

答:Object.freeze(obj)指冻结obj这个对象,对象冻结之后,只能查看和获取,不能添加删除修改对象的属性

Object.preventExtensions(obj)指阻止扩展这个对象,阻止之后,只能访问,获取,修改,删除对象的属性,不能添加对象的属性

Object.seal(obj)是密封obj这个对象,密封之后只能获取,查看,修改,不能添加和删除对象的属性

Object.getPrototypeOf(obj)获取obj的Prototype.

11.以下代码中{}中两行代码分别是什么作用？

function Student(name, isMale, course){

Child.call(this, name, isMale)

this.course = course

}

答:1,Child让this,这里指Student来代替自己调用自己的构造函数,同时把Student的构造函数内的参数传递给Child 的参数位

2,设置Student的一个属性course,把参数course的值传给该属性

12.以下代码2行代码的作用分别是什么？

Student.prototype = Object.create(Child.prototype)

Student.prototype.constructor = Student

答:1,创建一个新的对象,对象的属性是根据Child.prototype来设置的,再赋给Student.prototype(相当于把Child.prototype复制一份给Student.prototype)

2,把Student.prototype的constructor改为Student

13.以下代码在控制台的输出分别是什么？

var a = 3, b = 5

function test(a, c){

console.log(a + b)

b = c + a

}

console.log(a + b)

test(2, 6)

答:8,7

14.以下代码的输出分别是什么？

function test(c){

var a = 3, b = 5

b = c (b=3)

function func(a, c){

return a + b + c

}

return func

}

test(3)(1, 2)

答: func(1,2) , 6

15.关于Ajax说法正确的是：­ A,D

1. A表示Async，即异步的，也就是说默认情况下大家都用异步网络请求
2. j表示JSON，即大多数情况下是使用JSON格式传输数据的
3. x表示XML，XML比JSON使用起来更方便
4. Ajax通常需要使用XMLHttpRequest实现

16.关于Ajax说法正确的是：A,D

1. Ajax技术可以单独发起网络请求，而且不会导致页面刷新，所以用户体验很好
2. Ajax技术是H5时代才产生的技术
3. Ajax比使用表单复杂，因此并不常用
4. 如果说Form表单是用HTML标签进行网络请求，那Ajax就可以认为是用JavaScript进行网络请求

17.关于XHR说法正确的是： A

1. XHR是HTTP协议的API，通过XHR可以控制HTTP请求的各个细节，如请求行、请求头、请求体
2. XHR是由浏览器完成的，因此不能使用代码获取所有的响应头，响应头是由浏览器管理的
3. XHR是通过回调函数onreadystatechange来通知网络请求状态变化的
4. 使用XHR只能提交简单数据，不能上传文件

18.关于Node.js说法正确的是： A,B,D

1. Node.js是服务端JavaScript
2. 用Node.js可以实现Web服务器
3. Node.js像浏览器JavaScript一样功能很弱，如不能操作文件，也无法获取系统信息
4. Node.js的社区非常发达，有很多模块包可以直接安装使用

19.Node.js安装express的命令是： npm install express --save

Node.js中npm init命令的作用是： 初始化npm,会在项目目录中生成一个package.json的配置文件

20.Node.js中npm uninstall multer命令的作用是： 卸载 multer 模块包

如果上述命令添加 --save参数，会产生什么效果： 把 json中的 dependencies的数据也删除啦

21.以下代码使用XHR发起POST请求，请将代码补充完整

var xhr = new XMLHttpRequest()

xhr.onreadystatechange = function(){

if(xhr.readystate == 4 ){

console.log(‘请求已完成！收到的文本数据是：’ + xhr. responseText )

}

}

xhr. open (‘POST’, url )

xhr.setRequestHeader(‘Content-Type’, ‘application/x-www-form-urlencoded’)

xhr. send (‘name=常伟&password=123456’) //发送请求体数据

22.请在横线上备注以下代码的作用

const exp = require(‘express’), 用require加载express模块

app = exp()

app.use(exp.static(‘public’)) 使用use()方法设置该服务点的静态目录

app.get(‘/user/login’, (req, res) => {}) 使用 get()请求

app.listen(3000, () => {}) 监听分配的端口号,此处分配的为3000端口

23.关于ES6说法正确的是： A,B,C

1. ES6是JavaScript的新标准，是浏览器正在努力实现的标准
2. ES6带来了很多新特性，如const常量，=>箭头函数等
3. Node.js几乎实现了所有的ES6新特性，因此Node.js可以使用ES6
4. 目前浏览器对ES6的支持还不完善，因此不建议在浏览器的使用ES6
5. 给数组Array扩充方法，可以将元素倒序排列。

Array.prototype.daoxu=function(arr){  
 var Arr=[]  
 for(var i=arr.length-1;i>=0;i--){  
  
  Arr[arr.length-i-1]=arr[i]  
  
}  
return Arr  
}  
var li=new Array()

var result = li.daoxu(arr)  
console.log(result)

25.关于XSS攻击说法正确的是： BC

1. 使用JavaScript攻击不会造成什么大的危害，最多影响用户体验
2. XSS通常是由于对用户输入没有过滤造成的
3. XSS是跨站脚本的意思
4. XSS攻击方式的存在再一次证明了不能相信用户输入的每一个字符

26列举至少2个HTTP响应状态码及含义

状态码 200:ok,请求已成功,404:请求失败,请求所希望的资源不存在

1. 列出GET请求和POST请求的至少2点差异

1,get请求会把请求的数据直接追加在url的后边,post请求的数据是放在请求体中进行发送

2,get请求发送的内容有限,且发送的数据类型收到限制,post请求可以发送大量的数据,且可以发送各种文件类型

28.请描述你对http请求的理解，并写出HTTP响应的基本结构。

答:就是从客户端到服务器端的请求消息,通过浏览器发起一个到服务器上指定端口的请求,服务器收到请求会向浏览器端发回响应的消息.

HTTP响应的基本结构:响应行,响应头,响应体,响应行里有状态码,和状态码描述,响应头里有

Content-Type,Content-Encoding 等信息

29. Git是什么？为什么要使用Git？

答:Git 是分布式版本控制系统,因为Git功能强大,使用方便简单,由于在自己的电脑上有一个完整的版本库,可以在没有网络的情况下使用,只需等有网络的时候,将自己修改的版本推送到远程库即可.而且安全性能高,电脑断电,或者损坏,导致数据丢失,只需从别人那里复制一份即可.还可以快速恢复之前的版本,查看之前修改的内容,十分方便,所以使用Git.

30.请将该试题以及答案通过Git提交到远程仓库(GitHub或是OSChina均可，但须说明)。并写出你远程仓库的名称，

# 格式如下：[Roy9218PP](https://github.com/Roy9218PP)/[H5-1](https://github.com/Roy9218PP/H5-1) 。即Git账户名/仓库名。

Git账户名/仓库名: zhaoxisizou/luochao