1. 关于回调函数说法正确的是： ABD
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：­­­­ ABC
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： BC
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： BC
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： ACD
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： ABD

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:第一种是直接创建一个obj对象,再对obj对象添加一个age属性.

第二种是Object调用它的defineProperty()方法,在方法里设置对象obj,属性age,以及属性值,包括属性的描述,即有是否可以被遍历到,是否可以修改,是否可以删除等描述信息

8.关于构造函数说法正确的是： ABD

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:(1)这是一个构造函数,对他在内部扩充属性的时候,应该用this关键字,创建实例,而这里创建的属性(name,isMale)属于私有属性,(2)在实例对象中输出的时候,应该在构造函数外部用new关键字调用;(3)由于food无法传值,所以console.log(this.name + ‘ eat ‘ + food)打印不出结果

10.以下代码的功能分别是什么？

Object.freeze(obj), 答:让对象obj被冷冻,即obj对象不可以添加,修改,删除属性

Object.preventExtensions(obj) 答:阻止obj对象扩展属性,即不可以再添加属性,但是可以修改,删除

Object.seal(obj) 答:密封obj对象,密封之后,不可以添加,删除属性,可以查看,获取,修改

Object.getPrototypeOf(obj)答:用来获取obj对象的原型

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:第一句:让另一构造函数Student继承Child构造函数的name,isMale属性

第二句:给构造函数Student扩充coures属性,属性值为传的第三个参数course

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:第一句:通过把Child的原型拷贝一下,在赋值给构造函数Student,使Student可以继承Child构造函数用过原型扩充的静态属性和方法

第二句:第一句之后,Student指向的构造函数会改变,所以应该通过这一句把constructor再改为Student本身,否则,可能会在该构造函数扩充静态属性的时候造成双向继承的现象

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:结果是:8和 7

14.以下代码的输出分别是什么？

function test(c){

var a = 3, b = 5

b = c

function func(a, c){

return a + b + c

}

return func

}

test(3)(1, 2)

答:func(1,2) 和 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 B

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： ABD

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node包管理,生成一个json格式的文件

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

如果上述命令添加 --save参数，会产生什么效果： 会把json文件中的despences的数据也删除

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’), 加载一个express功能模块,便于搭建一个服务站点

app = exp()

app.use(exp.static(‘public’)) use是node的中间件,创建一个静态目录,他的内部静态文件不经过服务器处理,浏览器可以自己处理

app.get(‘/user/login’, (req, res) => {}) 向服务器提交的数据方式为get方式

app.listen(3000, () => {}) 添加一个监听事件,以及一个端口号,便于访问浏览器

23.关于ES6说法正确的是： AB 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

24.给数组Array扩充方法，可以将元素倒序排列。

Array.prototype.reverseEle = function(arr){

Var result = [ ]

For(var i = arr.length - 1 ; i >= 0;i--){

Result[arr.length - 1 - i ] = arr[i]

}

Return result

}

25.关于XSS攻击说法正确的是： B C

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

答:404,请求失败,访问的资源不存在

501:服务器不支持当前某一个功能

1. 列出GET请求和POST请求的至少2点差异

答:(1);GET请求的参数直接添加在url后面,而post请求参数是在请求体中的,参数返回给服务器的方式不同,

1. ;get请求提交少量的文本,文件 post请求可以提交很多的文本,文件
2. 请描述你对http请求的理解，并写出HTTP响应的基本结构。

答:http请求包括4个部分,有请求行,请求体,空行,请求头,是浏览器请求和服务器建立连接

Http响应的基本结构:响应头,响应行,响应体,空行

29. Git是什么？为什么要使用Git？

答:git是分布式控制管理系统, 因为git相对于其他管理系统,查询历史版本快,并且不需要联网,可以对代码不备份,而且git版本库占用空间小等优点

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。