

1. 关于回调函数说法正确的是：\_\_\_\_\_
  - A. 回调函数也是一个普通函数
  - B. 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
  - C. XMLHttpRequest 对象的 onreadystatechange 需要的就是一个回调函数
  - D. 回调函数即 Callback Function
2. 关于同步执行和异步执行的正确的说法是：\_\_\_\_\_
  - A. 同步执行即按顺序执行代码，一行执行之后再执行下一行
  - B. 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
  - C. 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
  - D. setTimeout 可以理解是异步执行的
3. 关于方法设计的说法正确的是：\_\_\_\_\_
  - A. 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
  - B. 方法的代码不应该太多，最好一个方法只完成一个小任务
  - C. 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
  - D. 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
4. 关于对象设计的说法正确的是：\_\_\_\_\_
  - A. 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
  - B. 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！
  - C. 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
  - D. 对象应该保持功能单一，最好一个对象只负责做好一件事
5. 关于对象方法的设计说法正确的是：\_\_\_\_\_
  - A. 对象的方法不亦过大，应适当拆分，以利于代码重用
  - B. 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
  - C. 将代码中可变的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
  - D. 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. 关于可视对象（带界面的对象）设计的说法正确的是：\_\_\_\_\_
  - A. 将样式等用代码实现是有好处的，使用对象的程序员只需要提供一些参数或设置一些属性就能实现界面的各种完美变化
  - B. 将样式等分离到单独的样式表中是有好处的，使用对象的程序员不用改程序代码，只需要定义样式就可以实现界面的调整和自定义
  - C. 将样式写在代码中也有一些缺点，例如，如果样式不符合要求，就需要修改程序代码
  - D. 将样式分离开也有缺点，例如，想实现一个简单的放大就要改好几处样式，而且还得找究竟要改哪些样式，挺不方便的

7. 关于 Ajax 说法正确的是：\_\_\_\_\_
- A . 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
  - B . j 表示 JSON，即大多数情况下是使用 JSON 格式传输数据的
  - C . x 表示 XML，XML 比 JSON 使用起来更方便
  - D . Ajax 通常需要使用 XMLHttpRequest 实现
8. 关于 Ajax 说法正确的是：\_\_\_\_\_
- A. 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
  - B. Ajax 技术是 H5 时代才产生的技术
  - C. Ajax 比使用表单复杂，因此并不常用
  - D. 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
9. 关于 XHR 说法正确的是：\_\_\_\_\_
- A. 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
  - B. 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
  - C. XHR 是通过回调函数 onreadystatechange 来通知网络请求状态变化的
  - D. 使用 XHR 只能提交简单数据，不能上传文件
10. 关于 Node.js 说法正确的是：\_\_\_\_\_
- A. Node.js 是服务端 JavaScript
  - B. 用 Node.js 可以实现 Web 服务器
  - C. 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
  - D. Node.js 的社区非常发达，有很多模块包可以直接安装使用
11. Node.js 安装 express 的命令是：\_\_\_\_\_
- Node.js 中 npm init 命令的作用是：\_\_\_\_\_
12. Node.js 中 npm uninstall multiparty 命令的作用是：\_\_\_\_\_
- 如果上述命令添加 --save 参数，会产生什么效果：\_\_\_\_\_
13. 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 == _____){
        console.log( '请求已完成！收到的文本数据是：' + xhr._____)
    }
}
xhr._____( 'POST' , url )
xhr.setRequestHeader( 'Content-Type' , 'application/x-www-form-urlencoded' )
xhr._____( 'name=常伟&password=123456' )    //发送请求体数据
```

14. 请在横线上备注以下代码的作用

```
const exp = require( 'express' ),  
    app = exp()  
  
app.use(exp.static( 'public' ))  
  
app.get( '/user/login' , (req, res) => {})  
app.listen(3000, () => {})
```

15. 关于 ES6 说法正确的是：\_\_\_\_\_

- A. ES6 是 JavaScript 的新标准，是浏览器正在努力实现的标准
- B. ES6 带来了很多新特性，如 const 常量，=>箭头函数等
- C. Node.js 几乎实现了所有的 ES6 新特性，因此 Node.js 可以使用 ES6
- D. 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

16. 关于 UML 中可见性的说法正确的是：\_\_\_\_\_

- A. public 表示到处可见可用
- B. private 表示只能在本类内部使用，限制最严格
- C. protected 可以在子类中使用
- D. package 表示不能在包外使用

17. 写代码将数组 arr 中的元素倒着 console.log()出来

18. 关于 XSS 攻击说法正确的是：\_\_\_\_\_

- A. 使用 JavaScript 攻击不会造成什么大的危害，最多影响用户体验
- B. XSS 通常是由于对用户输入没有过滤造成的
- C. XSS 是跨站脚本的意思
- D. 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

19. Node.js 命令 npm install -g grunt-cli 的作用是：\_\_\_\_\_  
grunt-contrib-concat 是：\_\_\_\_\_

20. 关于 grunt 的说法正确的是：\_\_\_\_\_

- A. Grunt 是前端构建工具，在 H5 开发中作用不大
- B. Grunt 可以自动执行命令，因此也称它们是自动化构建工具
- C. Grunt 可以帮我的干很多无聊的任务
- D. Grunt 可以大大节省我们的时间，提高 H5 开发效率

21. 备注以下代码的作用

```
module.exports = grunt => {  
    .....// 此处省略若干行代码
```

```
    grunt.registerTask( 'default' , [ 'concat' , ' uglify' ]) _____  
  }
```

22. 以下模块包的作用是什么

grunt-contrib-concat \_\_\_\_\_  
grunt-contrib-uglify \_\_\_\_\_

23. 请将以下代码改写成其它类型的循环语句

```
var arr = [1,2,3,4,5],  
    i = 0  
while(i < arr.length){  
    console.log(arr[i++])  
}
```

24. 列举任意 2 个服务端技术

25. 将多个代码文件合并成一个文件有什么好处？ \_\_\_\_\_  
将 js 压缩丑化有什么好处？ \_\_\_\_\_