

《京程一灯》精英班第十周笔试题 姓名：

请按要求完整作答。

- 1. 请描述在你项目中你使用到了哪些设计模式？（5分）

- 答：

（以下这些是我课上讲的你可以自己加的）🍄 本题考点分为如下：

1.单例模式

2.代理模式

3.命令模式

4.发布-订阅模式

5.职责链模式

- 2.请描述如何处理XSS与Csrp？（5分）

📖 高频考题：

1.CSRF跨站请求伪造 它也被称为 “One Click Attack” 或者Session Riding，通常缩写为CSRF或者XSRF，是一种对网站的恶意利用。相对来说更加难以防范。4. 终级防范采用强验证码+动态Token请求。


2.XSS跨站脚本攻击，为了不和层叠样式表CSS混淆，故将跨站脚本攻击缩写为XSS。恶意攻击者往Web页面里插入恶意Script代码，当用户浏览该页之时，嵌入其中Web里面的Script代码会被执行，从而达到恶意攻击用户的目的。分为反射型（被动的非持久性XSS。诱骗用户点击短型URL，服务器解析后响应，在返回的响应内容中隐藏和嵌入攻击者的XSS代码，从而攻击用户）、持久型（也叫存储型XSS——主动提交恶意数据到服务器，当其他用户请求后，服务器从数据库中查询数据并发给用户受到攻击。）、DOM型（DOM通过html一个结构执行事件脚本。 ）。

3.XSS漏洞防范-payload 输入转义、输出转义、Cookie HTTP Only、CSP（Content-Security-Policy：default-src ‘self’ ）

其余的攻击方式还有网页挂马和WebShell，也可以拓展一下。

- 3.请书写在前端中有哪些混淆办法并如何进行反混淆。 (10分)

- 答：

 这个其实是个实战的需求，不过很多面试官也会问到：

1.通过tool.lu/js/ 这种网站生成的代码直接去掉eval就好了


2.aaencode/jjencode 字符混淆以及逻辑混淆

3.一锤落地直接解决

```
Number.prototype.constructor.constructor = function(a) {
    console.log('-----');
    console.log(a);
    return Function.apply(null, arguments);
}
```

4.加密混淆神器 一般反混淆过程很艰难jscrambler

- 4.请列举常见的加密算法并描述各自的特点 (5分)

 扯到了这个你也可以往区块链上连接一下：

1.base64：Base64用于传输8Bit字节代码的编码方式之一，可用于在HTTP环境下传递较长的标识信息。编码具有不可读性

2.md5 MD5的应用是对一段信息产生信息摘要，以防止被篡改，不可逆，原因是其是一种散列函数，使用的是hash算法，在计算过程中原文的部分信息是丢失了的。

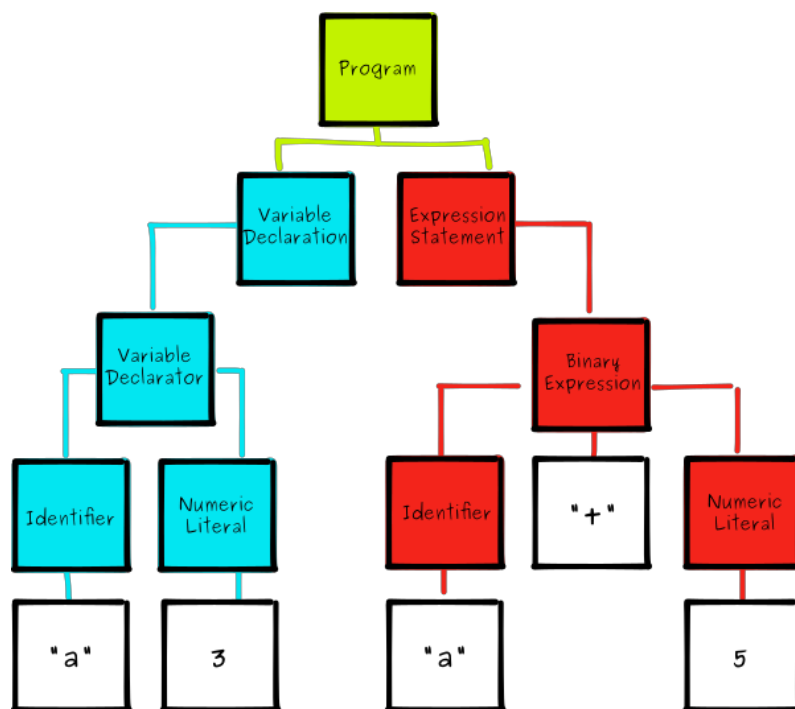
3.安全哈希算法 (Secure Hash Algorithm) 主要适用于数字签名标准 (Digital Signature Standard DSS) 里面定义的数字签名算法 (Digital Signature Algorithm DSA) 。


4.Crypto模块作为nodeJS已经稳定下来的模块在整个node中具有举足轻重的地位，一切app都需要加密解密，那么crypto就是一个提供加密功能的模块。在这个模块中已经打包了OpenSSL hash, HMAC (哈希信息验证码) , cipher (加密) ,decipher (解密) ,sign (签名) 以及verify (验证) 的功能。

5.其余加密技术手段 MD5、SHA-1、SHA-256、AES、Rabbit、MARC4、HMAC、HMAC-MD5、HMAC-SHA1、HMAC-SHA256、PBKDF2

- 5.你知道什么是AST么, 什么情况下会使用AST呢?(10分)

- 答:



 Babel 加密 混淆 Less编译工具啥的的都是这个原理：

抽象语法树 (Abstract Syntax Tree) 也称为AST语法树, 指的是源代码语法所对应的树状结构。也就是说, 对于一种具体编程语言下的源代码, 通过构建语法树的形式将源代码中的语句映射到树中的每一个节点上。

```

var UglifyJS = require("uglify-js");

var code = "var a = 1;";


var toplevel = UglifyJS.parse(code); //toplevel就是语法树

var transformer = new UglifyJS.TreeTransformer(function (node) {
  if (node instanceof UglifyJS.AST_Number) { //查找需要修改的叶子节点
    node.value = '0x' + Number(node.value).toString(16);
    return node; //返回一个新的叶子节点 替换原来的叶子节点
  }
});
  
```

```
toplevel.transform(transformer); //遍历AST树
var ncode = toplevel.print_to_string(); //从AST还原成字符串
console.log(ncode); // var a = 0x1;
```

- 6. 请描述你对ASMJS的认知 (10分)

- 答：

 我猜一般的面试官知道的少一些：


Asm.js 来自于 JavaScript 应用的一个新领域: 编译成JavaScript的C/C++应用. 它是 JavaScript 应用的一个全新流派, 由 Mozilla 的 Emscripten项目催生而来。

Emscripten 将 C/C++ 代码传入 LLVM, 并将 LLVM生成的字节码转换成 JavaScript (具体的, Asm.js, 是 JavaScript 的一个子集).

它具有的特性包括JavaScript子集 (为性能优化而生, 特别是那些需要编译成JavaScript的应用。)、限制操作对象 (保证性能、能够直接转换成汇编代码。)、虚拟机的抽象 (可有效负载和存储的大型二进制堆、整型和浮点运算、高阶函数定义、函数指针等。), 适合场景 (游戏、图像、处理语言翻译和库。)

- 7.请描述WebAssembly的具体应用和原理。(10分)

- 答：

 这个技术确实有点火 (动动你的小手从头敲一个demo行不皇上🎺)：

1.定义一个可移植, 体积紧凑, 加载迅捷的二进制格式为编译目标, 而此二进制格式文件将可以在各种平台 (包括移动设备和物联网设备) 上被编译, 然后发挥通用的硬件性能以原生应用的速度运行。

1.WebAssembly 主要试图解决现有技术的一些问题：

- (1) JavaScript：性能不够理想, 以及语言本身的一堆坑 (这个大家都懂)
- (2) Flash：私有技术 (而且漏洞一堆), 并且是纯二进制格式
- (3) Silverlight：私有技术, 并且是纯二进制格式
- (4) 各种插件 (Plug-in)：安全性问题, 平台兼容问题

- 8.请问如何将代码压缩进图片并如何将代码解释出来。(15分)

- 答：



🍄 这个题是让你提升你在面试官面前的高度的：

1.写进去 `imageData.data[j] = arr[i];`

2.读出来 `var imageData = context.getImageData(0, 0, width, height);`
`str += imageData.data[i]!==0?hexToString(imageData.data[i]):' '; eval(str);//执行`

3.原理：字符可以转为16进制，与图片RGB的一个R/G/B相对应，即一个像素点可容纳3个字符

- 9.你如何看待PWA AMP 等等新鲜的技术。(15分)

- 答：

🍄 谷歌的东西向来是引领技术的潮流。：

1.AMP , Accelerated Mobile Pages , 译意大致是“加速的移动页面”，是Google去年10月份推出的一个提高移动页面访问速度的技术

2.Progressive Web Apps 是 Google 提出的用前沿的 Web 技术为网页提供 App 般使用体验的一系列方案。这篇文章里我们来完成一个非常简单的 PWA 页面。

- 10.你了解过微前端么？基于微前端的架构怎么设计呢。(15分)

- 答：

- 见直播课代码演示。基于YOG2基于System.js

