**2019年最新经典web前端面试题**

**「 CSS篇 」**

1. CSS 盒子模型，绝对定位和相对定位

2. 清除浮动，什么时候需要清除浮动，清除浮动都有哪些方法

3. 如何保持浮层水平垂直居中

4. position 和 display 的取值和各自的意思和用法

5. 样式的层级关系，选择器优先级，样式冲突，以及抽离样式模块怎么写，说出思路，有无实践经验

6. css3动画效果属性，canvas、svg的区别，CSS3中新增伪类举例

7. px和em和rem的区别，CSS中link 和@import的区别是?

5. 了解过flex吗?

**「 JavaScript 篇 」**

**JavaScript 基础**

1. JavaScript 里有哪些数据类型，解释清楚 null 和 undefined，解释清楚原始数据类型和引用数据类型。比如讲一下 1 和 Number(1)的区别

2. 将一下 prototype 是什么东西，原型链的理解，什么时候用 prototype

3. 函数里的this什么含义，什么情况下，怎么用。

4. apply和 call 什么含义，什么区别?什么时候用。(我有篇文章 重点分析过)

5. 数组和对象有哪些原生方法，列举一下，分别是什么含义，比如连接两个数组用哪个方法，删除数组的指定项和重新组装数组(操作数据的重点)。

6. 怎样避免全局变量污染?ES5严格模式的作用，ES6箭头函数和ES5普通函数一样吗?

**JavaScript 的面向对象**

1. JS 模块包装格式都用过哪些，CommonJS、AMD、CMD。定义一个JS 模块代码，最精简的格式是怎样。

2. JS 怎么实现一个类。怎么实例化这个类。

3. 理解闭包吗?请讲一讲闭包在实际开发中的作用;闭包建议频繁使用吗?

4. 说一下了解的js 设计模式，解释一下单例、工厂、观察者。

5. ajax 跨域有哪些方法，jsonp 的原理是什么，如果页面编码和被请求的资源编码不一致如何处理?

**「 开源工具 」**

1)是否了解开源的架构工具 bower、npm、yeoman、gulp、webpack，有无用过，有无写过，一个 npm 的包里的 package.json 具备的必要的字段都有哪些(名称、版本号，依赖)

2)github常用不常用，关注过哪些项目

3)会不会用 ps 扣图，png、jpg、gif 这些图片格式解释一下，分别什么时候用。如何优化图像、图像格式的区别

4)说一下你常用的命令行工具

5)会不会用git，说上来几个命令，说一下git和svn的区别，有没有用git解决过冲突

**「 计算机网络基础 」**

1)说一下HTTP 协议头字段说上来几个，是否尽可能详细的掌握HTTP协议。一次完整的HTTP事务是怎样的一个过程?

2)cookies 是干嘛的，服务器和浏览器之间的 cookies 是怎么传的，httponly 的 cookies 和可读写的 cookie 有什么区别，有无长度限制

请描述一下cookies，sessionStorage和localStorage的区别

3)从敲入 URL 到渲染完成的整个过程，包括 DOM 构建的过程，说的约详细越好。

4)是否了解Web注入攻击，说下原理，最常见的两种攻击(XSS 和 CSRF)了解到什么程度。

5)是否了解公钥加密和私钥加密。如何确保表单提交里的密码字段不被泄露。验证码是干嘛的，是为了解决什么安全问题。

6)编码常识：文件编码、URL 编码、Unicode编码 什么含义。一个gbk编码的页面如何正确引

**「 前端框架 」**

1) 对 MVC、MVVM的理解

2) vue、angularjs等 相对于 jQuery在开发上有什么优点?

3)前后分离的思想了解吗?

4)你上一个项目都用到了那些方法优化js的性能?

5)angular的生命周期?

6)说一下你对vue和vuex的使用方法，vue的组件复用机制

**考察学习能力和方法**

1)你每天必须登录的网站(前端技术相关)是什么?

2)前端技术方面看过哪些书，有无笔记，都有哪些收获。

3)收藏了哪些代码片段?有想过开源自己的代码嘛?

4)怎么理解前端技术的大趋势?自己再做哪方面的知识储备?

5)是否了解或精通其他(后端)的编程语言?

6)做项目有没有遇到哪些印象深刻的技术攻关，具体遇到什么问题，怎么找答案的，最后怎么解的。

7)对以后自己的前端职业路线，怎么规划？

**开放性问题(重要)**

这些问题往往决定你是否最终被录用或者等到终轮面试，技术点回答错了不要紧，人脑不是机器，是可以恶补的。

但如果你没有思想和独到的思路，基础挖的再深，可能也打动不了面试官，因为比你基础好的一大堆，但每个人的个性思想却是不同的。

1. 如果需要你加班，你会加吗，抵触吗?

其实你肯定抵触，但你肯定要回答如果项目需要肯定会加。

2. 一个小项目让你自己负责搭建底层一些架构，你能胜任吗?

回答例如：我肯定愿意尝试，并做到最优的选择方案出来。

3. 如果项目拖太久，你情绪低落或者厌烦了怎么调节?

回答就是， 你结合自身挑着好听的说就行，就像聊天。

4. 你建议自己造轮子，还是利用开源的轮子?

回答：根据实际情况而定，如果开源完全满足 可以自己二次开发就好，大大缩短开发周期，

如果实在没有契合度很高的，可以花费几个工作日尝试造轮。