**前端测试卷**

1. 单选题
2. 以下关于跨域说法错误的是（ ）

[A. Cookie，LocalStorage和IndexedDB都会受到同源策略的限制](https://www.nowcoder.com/question/javascript:void(0);)

[B. postMessage，JSONP，WebSocket都是常用的解决跨域的方案](https://www.nowcoder.com/question/javascript:void(0);)

[C. 跨域资源共享规范中规定了除了GET之外的HTTP请求，或者搭配某些MINE类型的POST请求，浏览器都需要先发一个OPTIONS请求](https://www.nowcoder.com/question/javascript:void(0);)

[D. http://www.bilibili.com和https://www.bilibili.com是相同的域名，属于同源](https://www.nowcoder.com/question/javascript:void(0);)

1. var foo = "Hello";

(function(){

var bar = " World";

alert(foo + bar);

})();

alert(foo + bar);

其中alert两次输出结果为（ ）

[A. hello world ; 报错](https://www.nowcoder.com/question/javascript:void(0);)

[B. hello world ; hello world](https://www.nowcoder.com/question/javascript:void(0);)

[C. hello world ; hello](https://www.nowcoder.com/question/javascript:void(0);)

[D. 报错](https://www.nowcoder.com/question/javascript:void(0);)

1. 多选题
2. 下列说法正确的是（ ）

[A. setTimeout(foo, 0) 这行代码可以等价替换为 foo()](https://www.nowcoder.com/test/question/javascript:void(0);)

[B. 使用 Object.assign(obj1, obj2) 可以实现对象的浅拷贝](https://www.nowcoder.com/test/question/javascript:void(0);)

[C. for...in 循环只能遍历对象上的可枚举属性](https://www.nowcoder.com/test/question/javascript:void(0);)

1. Object.keys() 可以遍历出对象原型链上的属性

2、下列结果返回true的是（ ）

[A. null == undefined](https://www.nowcoder.com/question/javascript:void(0);)

[B. null === undefined](https://www.nowcoder.com/question/javascript:void(0);)

[C. null === null](https://www.nowcoder.com/question/javascript:void(0);)

[D. NaN == null](https://www.nowcoder.com/question/javascript:void(0);)

[E. NaN === NaN](https://www.nowcoder.com/question/javascript:void(0);)

[F. Infinity + 1 !== Infinity](https://www.nowcoder.com/question/javascript:void(0);)

1. 简答题
2. 请描述一下 cookies，sessionStorage 和 localStorage 的区别。
3. prototype和\_\_proto\_\_的区别？
4. 什么是闭包？列举闭包的简单应用。

4、运行一下程序，控制台结果输出什么？

var peaks = ["Talac","Ralston","Rose","Marlette"]

var [last] = [...peaks].reverse()

var [first,...rest] = peaks

console.log(last)

console.log(peaks.join(','))

console.log(rest.join(','))

5、举例说明ES6的解构赋值和对象语义增强。

6、描述React组件的生命周期或VUE组件的生命周期。

7.清除浮动的方法有哪些？

参考答案：

1.父级div定义 height   
原理：父级div手动定义height，就解决了父级div无法自动获取到高度的问题。   
优点：简单、代码少、容易掌握   
缺点：只适合高度固定的布局，要给出精确的高度，如果高度和父级div不一样时，会产生问题

2，结尾处加空div标签 clear:both

原理：添加一个空div，利用css提高的clear:both清除浮动，让父级div能自动获取到高度   
优点：简单、代码少、浏览器支持好、不容易出现怪问题   
缺点：不少初学者不理解原理；如果页面浮动布局多，就要增加很多空div，让人感觉很不好

3，父级div定义 伪类:after 和 zoom

原理：IE8以上和非IE浏览器才支持:after，原理和方法2有点类似，zoom(IE转有属性)可解决ie6,ie7浮动问题   
优点：浏览器支持好、不容易出现怪问题（目前：大型网站都有使用，如：腾迅，网易，新浪等等）   
缺点：代码多、不少初学者不理解原理，要两句代码结合使用才能让主流浏览器都支持

4，父级div定义 overflow:hidden

原理：必须定义width或zoom:1，同时不能定义height，使用overflow:hidden时，浏览器会自动检查浮动区域的高度   
优点：简单、代码少、浏览器支持好   
缺点：不能和position配合使用，因为超出的尺寸的会被隐藏。

8. 如何进行网站性能优化

参考答案：

1. 从用户角度而言，优化能够让页面加载得更快、对用户的操作响应得更及时，能够给用户提供更为友好的体验。  
2. 从服务商角度而言，优化能够减少页面请求数、或者减小请求所占带宽，能够节省可观的资源。  
　　总之，恰当的优化不仅能够改善站点的用户体验并且能够节省相当的资源利用。  
　　前端优化的途径有很多，按粒度大致可以分为两类，第一类是页面级别的优化，例如 HTTP请求数、脚本的无阻塞加载、内联脚本的位置优化等 ;第二类则是代码级别的优化，例如 Javascript中的DOM 操作优化、CSS选择符优化、图片优化以及 HTML结构优化等等。另外，本着提高投入产出比的目的，后文提到的各种优化策略大致按照投入产出比从大到小的顺序排列。  
　　一、页面级优化  
1. JavaScript 压缩和模块打包  
2. 按需加载资源  
3. 在使用 DOM 操作库时用上 array-ids  
4. 缓存  
5. 启用 HTTP/2  
6. 应用性能分析  
7. 使用负载均衡方案  
8. 为了更快的启动时间考虑一下同构  
9. 使用索引加速数据库查询  
10. 使用更快的转译方案  
11. 避免或最小化 JavaScript 和 CSS 的使用而阻塞渲染  
12. 用于未来的一个建议：使用 service workers + 流  
13. 图片编码优化

1. 编程题

1、使用javaScript实现paddingNum方法，该方法功能为分割数字，每隔三位使用逗号分隔一次，返回值为分隔后的字符串。例如：

/\*\*

\* paddingNum(33) //return '33'

\* paddingNum(1234.56) //return '1,234.56'

\* paddingNum(123456789) //return '123,456,789'

\* paddingNum(987654.321) //return '987,654.321'

**\* paddingNum(98765.4321) //return '98,765.432,1'**

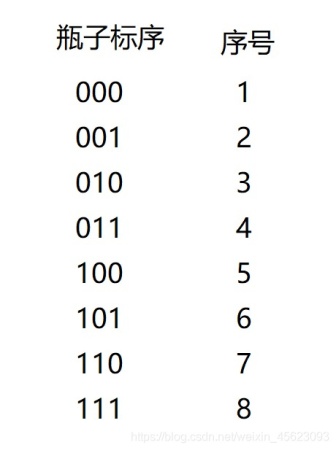
\* paddingNum(-987654.3) //return '-987,654.3'

\*/

1. 附加题

八瓶水中只有一个有毒，已经知道毒药发作时间为3小时，现有3只小白鼠，有三个小时时间，要知道那个杯子里有毒。

参考答案:



三只老鼠分别代表一位，用喝不喝代表01，组成八组数，八瓶水对应八组数。那个数成立了，就能找到那个数对应的那一瓶