览众数据前端开发工程师（初级）招聘试题

**姓名：韩畅畅 日期：2020/12/26 得分：**

**第一部分：HTML+CSS**

1. 行内元素有哪些？块级元素有哪些？CSS 的盒模型？

行内元素：a, span, i, em

块级元素：div, section, header, main, nav, footer

1. 你如何对网站的文件和资源进行优化？期待的解决方案包括哪些？

①静态资源CDN

②合理使用缓存减少重复的请求

③使用webp格式的图片代替png与jpg格式的图片，尽量使用css代替图片

④尽量使用css动画代替js动画减少重排与重绘

1. 清除浮动的几种方式，各自优缺点?

①额外标签：添加了额外的标签，对SEO不好

②伪元素：最好的方法

1. flex常见的属性有哪些？

①justify-content，align-items，flex，flex-direction

**第二部分：JAVASCRIPT**

1. 例举 3 种强制类型转换和 2 种隐式类型转换?

强制类型转换：Number(‘20’), String(20), Boolean(0)

隐式类型转换：’20’/20, ‘20’/’20’

1. 写出call 和 apply 的区别例子代码？
2. fn.call(this, 10, 20,30);
3. fn.apply(this, [10, 20, 30]);
4. b 继承 a 的方法。（能写几种写几种）

①原型继承

②构造函数继承

1. 写出事件委托的例子？

<ul id="ul1">

<li>111</li>

<li>222</li>

<li>333</li>

<li>444</li>

</ul>

window.onload = function(){

var oUl = document.getElementById("ul1");

oUl.onclick = function(){

alert(123);

}

}

1. 手写闭包例子？

*function* Father() {

*var* fo = 20;

            return *function* Son() {

*console*.log(fo);

                return fo;

            }

        }

        Father();

1. 编写一个数组去重的方法。

*function* uniqueArray(array) {

*const* result = {};

            return *array*.filter(element *=>* {

                return *result*.hasOwnProperty(typeof element + element) ? false : (result[typeof element + element] = true);

            });

        };

**第三部分：选择题**

1、 let promise = new Promise(function(resolve, reject) {

console.log('Promise');

resolve();

});

promise.then(function() {

console.log('resolved.');

});

console.log('Hi!');

运行以上代码，以下正确输出的的是：（D）

A：// resolved// Hi!// Promise

B：// Promise// resolved// Hi!

C：// Hi!// Promise// resolved

D：// Promise// Hi!// resolved

写出以上所选答案的运行原理：

2、 function boo (a) {return a \* 3}

function foo (b) {return boo(4) \* 2}

console.log(foo(3))

运行以上代码，以下正确输出的的是：（B）

A：36

B：24

C：6

D：12

写出以上所选答案**线程**的运行原理：

同步代码先在主线程上执行，然后再执行微任务。

**第四部分：综合**

1. 根据JS对象的原型原理，写出具有如下要求的代码例子

要求：1）扩展原型

* + 1. 原型链式的继承

1. 写一个深度拷贝的算法

*function* deepCopy(object, hash = new WeakMap()) {

            if (object instanceof RegExp) {

                return RegExp(object);

            };

            if (object instanceof Date) {

                return Date(object);

            };

            if (object === null || typeof object !== 'object') {

                return object;

            };

            if (*hash*.has(object)) {

                return object;

            };

*const* result = new object.constructor();

*hash*.set(object, result);

            for (*const* key in object) {

                if (*object*.hasOwnProperty(key)) {

                    result[key] = deepCopy(object[key], hash);

                };

            };

            return result;

        };

1. 写出用setTimeout模拟setInterval的代码

*function* setInterval(fn, delay){

*function* interval(){

                setTimeout(interval, delay);

                fn();

            }

            setTimeout(interval, delay)

        }

1. vue里面哪儿不会用到双向绑定?

nextTick

1. CSRF、XSS他们分别的攻击手段与防御手段

XSS: 在客户端插入一段JS恶意脚本。

防御：对用户的输入过滤。使用HttpOnly。使用CSP白名单策略。

CSRF: 利用用户的登陆状态来做坏事。

防御：验证Referer或者Origin。使用SameSite。使用 CSRF Token

1. 写出一段js回收机制原理的代码，体现在哪地方进行回收，回收的是什么？

*function* foo(name) {

*console*.log(name);

*function* h(name) {

*console*.log(name + 'h');

            }

        };

        foo('hcc');

foo入栈，h入栈。执行完成以后，h出栈， foo出栈。回收的是foo与h函数声明所占用的栈内存，以及函数体在堆上的内存