

1、typeof 能出来几个值 6 es6 中是 7

2、var name="abc";

Console.log("name="+name==="abc"?a:b) 答案是 b

3、var a={"name":"cao","age":18};

Var b=a; 改变 b 会不会改变 a 的值 会 怎么解决 将 b 转化为字符串

4、弹性布局和盒模型 bootstrap 中 row 的作用 清除浮动

5、transform 和 animation 的使用场景

6、http 请求

是指从客户端到服务器端的请求消息。

7、冒泡排序

8、js 二分法

```
var arr = [1,4,6,8,9,10,56];
```

```
function binarySearch(arr,findval,leftIndex,rightIndex){
```

```
    var midIndex = Math.floor((leftIndex+rightIndex)/2);
```

```
    var midval = arr[midIndex];
```

```
    if(midval>findval){
```

```
        //左边找
```

```
        binarySearch(arr,findval,leftIndex,midIndex);
```

```
        alert("left");
```

```
    }else if(midval<findval){
```

```
        //右边找
```

```
        binarySearch(arr,findval,midIndex,rightIndex);
```

```
        alert("right");
```

```
    }else {
```

```
        document.write("找到下标为" +midIndex);
```

```
    }
```

```
}
```

```
//测试:
```

```
binarySearch(arr,10,0,arr.length-1);
```

9、vue 中计算属性和定义方法的区别 计算属性会有缓存，大量的计算需要用计算属性

10、vue 设计模式 mvvm

MVVM 是 Model-View-ViewModel 的简写。它本质上就是 MVC 的改进版。MVVM 就是将其中的 View 的状态和行为抽象化，让我们将视图 UI 和业务逻辑分开。当然这些事 ViewModel 已经帮我们做了，它可以取出 Model 的数据同时帮忙处理 View 中由于需要展示内容而涉及的业务逻辑

- Model（模型）表示应用程序核心（比如数据库记录列表）。
- View（视图）显示数据（数据库记录）。
- Controller（控制器）处理输入（写入数据库记录）。
- Model（模型）是应用程序中用于处理应用程序数据逻辑的部分。
通常模型对象负责在数据库中存取数据。
- View（视图）是应用程序中处理数据显示的部分。
通常视图是依据模型数据创建的。
- Controller（控制器）是应用程序中处理用户交互的部分。
通常控制器负责从视图读取数据，控制用户输入，并向模型发送数据。

11、微信小程序生命周期

Object 页面的初始数据

onLoad	Function	生命周期函数--监听页面加载
onReady	Function	生命周期函数--监听页面初次渲染完成
onShow	Function	生命周期函数--监听页面显示
onHide	Function	生命周期函数--监听页面隐藏
onUnload	Function	生命周期函数--监听页面卸载
onPullDownRefresh	Function	页面相关事件处理函数--监听用户下拉动作
onReachBottom	Function	页面上拉触底事件的处理函数
onShareAppMessage	Function	用户点击右上角转发
onPageScroll	Function	页面滚动触发事件的处理函数
onTabItemTap	Function	当前是 tab 页时，点击 tab 时触发

12、画布

13、设置头部 ETag 是什么意思

14、斐波那契数列

```
function fibonacci(n) { if (n < 0) { throw new Error('输入的数字不能小于 0'); }  
if (n == 0) { return 0; } if (n == 1) { return 1; } return fibonacci(n-1) +  
fibonacci(n-2); }
```

```
function fibonacci4 (n) { if (n < 0) throw new Error('输入的数字不能小于 0'); let  
dataMinusTwo= 0, dataMinusOne = 1, data; if (n == 0) return dataMinusTwo;  
if (n == 1) return dataMinusOne; for (var i=2;i<=n;i++) { data =  
dataMinusOne + dataMinusTwo; dataMinusTwo = dataMinusOne;  
dataMinusOne = data; } return data; }
```

15、数据类型 es6 Map 是一组键值对的结构，查找速度快。 Set

Set 和 Map 相似，Set 的特点是不会存储 value，只会存储不重复的 key

Symbol 类型是保证每个属性的名字都是独一无二的，对于一个对象由对个模块构成的情况非常有用

字符串反转：

```
1、 var str = "abcdef";  
   console.log( str.split("").reverse().join("") );  
  
2、 var str="abcdef"  
   var i=str.length;  
   i=i-1;  
   for (var x = i; x >=0; x--)  
   {  
     document.write(str.charAt(x));  
   }
```

16、v-if 和 v-show 的区别

v-if

作用：判断是否加载固定的内容，如果为真，则加载；为假时，则不加载。

用处：用在权限管理，页面条件加载。

语法：v-if="判断表达式"

特点：控制元素插进来或者删除，而不是隐藏。

so，一般的，v-if 有更高的切换消耗，而 v-show 有更多的初始化渲染消耗。

so，如果需要频繁的切换而对安全性无要求，使用 v-show。如果在运行时，条件不可能改变，则使用 v-if 较好。

17、Vue 常见指令：

v-if//v-show//v-else//v-for//v-bind//v-on

nodeJs 异步操作转化为同步操作

首先: `npm install async`

其次在 js 文件中引入: `var async = require('async');`

函数有: `series waterfall parallel parallelLimit ...`

`series` 函数 串行执行

它的作用就是按照顺序一次执行。

18、数组去重

常规 js `for.....in` 循环 es6 中 `set`

```
var set1 = Array.from(new Set([1,1,2,2,33,'33',44,'44']))
```

19、文件缓存怎么处理:

1、通过 `query` 的方式添加时间戳或者版本号 (`<script type="text/javascript" src="index.js?v=1.0.0.0"></script>`)

2、直接修改文件的 URL, 而不是在其后添加 `query` 利用 `fis` (`<script type="text/javascript" src="index_a5dae5b.js"></script>`)

20、vue 实现双向数据绑定 : 数据劫持

1、监听 `Observer` 利用 `Object.defineProperty()` 来监听属性变动

2、解析 `compile` 主要做的事情是解析模板指令, 将模板中的变量替换成数据

3、实现一个 `Watcher`, 作为连接 `Observer` 和 `Compile` 的桥梁, 能够订阅并收到每个属性变动的通知, 执行指令绑定的相应回调函数, 从而更新视图

4、`mvm` 入口函数, 整合以上三者

21、webpack 开发环境和生产环境。

当运行项目时, `cmd` 中敲命令: `npm run dev/npm run build:prod/npm run build:st/npm run lint`, 就会运行不同环境下的代码, 如开发环境、生产环境等等。