### 1、什么是javascript？

>JavaScript是一种具有面向对象能力的、解释型的程序设计语言。更具体一点，它是基于对象和事件驱动并具有相对安全性的客户端脚本语言。

>因为他不需要在一个语言环境下运行，而只需要支持它的浏览器即可。它的主要目的是，验证发往服务器端的数据、增加Web互动、加强用户体验度等

### 2、javascript组成和分别是做什么的？

+ ECMA 核心语法:版本 ECMA5.0 ~ ECMA2015（es6）1.语法2.变量和数据类型

.关键字和保留字 操作符 控制语句 对象

+ DOM document object model文档对象模型 描述处理网页内容的方法和接口

DOM就是把【文档】当作一个【对象】来看待  
       DOM的顶级对象是document  
       DOM主要学习的是操作页面元素  
       DOM是W3C标准规范

+ BOM browser object model浏览器对象模型 描述与浏览器进行交互的方法和接口

把【浏览器】当作一个【对象】来看待  
       BOM的顶级对象是window  
       BOM学习的是浏览器窗口交互的一些对象  
       BOM是浏览器厂商在各自浏览器上定义的，兼容性差

### 3、开发中你经常用什么 IDE:Integrated Development Environment(开发工具)

+ hbuilderx 适合写排版布局

+ vscode vue/react/javascript/jquery/html5/go/python/php

+ webstorm

### 4、浏览器的内核

+ webkit（googlechrome，safari,QQ,360急速，Opera ）

+ Gecko （Firefox）

+ Presto （opear）

+ Trident（IE）

### 5、开发者工具 || 控制面板 || debug || F12

+ Elements：查看结构样式，可以修改这些内容

+ Console ：查看输出结果和报错信息，javaScript调试

+ Network：查看网站向服务器发送的所有信息

+ Sources ：查看项目源码

+ Application：查看当前网站的数据存储和资源文件

+ performance(性能)

### 6、定义变量

+ 定义变量不赋值 默认是 undefined

### 7、定义变量的几种方式？

+ var

+ let

+ const

+ class 名称

```

class Person{}

```

+ import 名称

```

import A from "../index.js"

```

+ function 名称

```

function person(){}

```

### 8、使用var 和不使用var定义变量的区别？

+ 带var定义的变量是全局变量，存放在全局变量对象VO(G)中

+ 不带var 创建的不是变量，而是全局对象GO（global object）的一个属性。

* AO：Activive Object，即函数的活动对象。
* VO：Variable Object，即变量对象。

AO可以理解为VO的一个实例，也就是VO是一个构造函数，然后VO(Context) === AO，所以VO提供的是一个函数中所有变量数据的模板。

对于同一个函数分多次执行，那么里面的变量、形参和定义的函数肯定是不同的函数，所以每次执行都会产生一个AO对象，即VO是AO的一个实例，但是这个实例并不是new 出来的，而是在同一段执行代码执行的时候放进来的。

VO是不能访问的(除了全局上下文的VO可以间接访问)，但是可以访问AO的成员(属性)。

VO和AO其实是一个东西，只是处于不同的执行上下文生命周期。AO存在于执行上下文位于执行上下文堆栈顶部(就是上边说的’当控制进入函数代码的执行上下文时’)的时期。再粗暴点，就是函数调用时，VO被激活成了AO。

AO通过函数的arguments属性初始化，其值是一个ArgO，包括 callee、length、arg属性。其中arg属性就相当于下标,比如第一个参数对应arg = 0。

### 9、var||let||const 定义变量的区别？

+ 带var 和不带var 定义的变量是有区别的

+ let

+ 要先定义在使用，

+ 不能重复定义，

+ 默认值是undefined

+ 不存在变量提升

+ const

+ 一旦定义的基本类型是不能改变（但是对象和数组是可以改变内部属性或者值）

+ 不能重复定义

+ 也不存在变量提升

### 10、alert 特点

+ 输出的都是字符串

+ 会阻断代码执行

### 11、script 标签放在页面头部和尾部的区别？

>1.标签放置在标签内部时：将script放在里，浏览器解析HTML，发现script标签时，会先下载完所有这些script，再往下解析其他的HTML。讨厌的是浏览器在下载JS时，是不能多个JS并发一起下载的。不管JS是不来来自同一个host，浏览器最多只能同时下载两个JS，且浏览器下载JS时，就block掉解析其他HTML的工作。将script放在头部，会让网页内容呈现滞后，导致用户感觉到卡。

>2.标签放置在标签内部时：将script放在尾部的缺点，是浏览器只能先解析完整个HTML页面，再下载JS。而对于一些高度依赖于JS的网页，就会显得慢了。所以将script放在尾部也不是最优解，最优解是一边解析页面，一边下载JS。

>3.标签放置在标签之后时：首先声明。这在之后插入其他元素，从HTML 2.0起就是不合标准的。按照HTML5标准中的HTML语法规则，如果在后再出现

>因为在body以外写script也可能存在其他异常嘛。有什么理由能让开发者推断出后者会更安全呢？实际上在没有充分测试的前提下，如果要进行推断，那么可以推断出后者的风险更大。

### 12、html 中块级元素和内联元素都有哪些？

> display:block;display:inline-block;display:inline;

+ 块级元素：div,p,h1-h6,dl,dt,dd,ul,li,ol,li,table,main,article,nav,header,footer,figure,figcaption,form,section

+ 内联元素：span,a,img,b,strong,mark,address,time,em,i,s,del,textarea,input,select,checkbox,video,audio,button

### 13、定位有种？清除浮动？

> position:fixed,absolute,relatice,sticky;top/bottom;left/right;z-index:1-99999;

> clear:both,left,right;父级添加overflow:hidden;为元素 :before,:after

### 14、link和import 的区别

+ 1.link是HTML标签，@import是css提供的。

+ 2.link引入的样式页面加载时同时加载，@import引入的样式需等页面加载完成后再加载。

+ 3.link没有兼容性问题，@import不兼容ie5以下。

+ 4.link可以通过js操作DOM动态引入样式表改变样式，而@import不可以。

### 15、css 原先级别

> !important > 行内样式>ID选择器 > 类选择器 > 标签 > 通配符 > 继承 > 浏览器默认属性

### 16、javaScript的数据类型？

+ 基本数据类型

+ null

+ undefinde

+ string

+ number

+ boolean

+ 引用数据类型

+ object

+ Array

+ Date

+ regExp

+ Math

+ function

+ symbol

### Number 数值类型

> null=>0,非数值字符串 =>NaN,空字符串=>0

> 引用类型转换是先底层通过toString() 转换为字符串，在 Number 进行转换

```

console.log(Number({}))//"[object Object]"=>NaN

console.log(Number([]))//""=>0

console.log(Number([12]))//[12]=>"12"=12

console.log(Number([1,2]))//[1,2]=>"1,2"=>NaN

```

+ NaN 属于number 类型

+ NaN和任何值都不想等，包括自身

+ isNaN 判断变量是否是非数值

+ Number() 浏览器底层转换是通过Number 转换的

+ parseInt() 从左往又开始依次查找转换，只要遇到无法转换的就立即停止查找，把有效数值转换为整数

+ parseFloat()从左往又开始依次查找转换，只要遇到无法转换的就立即停止查找，把有效数值转换为浮点数

> parseInt 和parseFloat 转换null，undefined,空字符串,true,false,NaN =>NaN

>因为这两个方法 是将字符串转换为 有效数字，所以运行机制是先将 值转换为字符串，然后 从左到右依次查找有效数字，只要找到就开始相对应的转换，只要遇到非有效数字就停止查找（不管后边是否存在就不在查找）

###css3动画和JS动画的区别？

js实现的是帧动画

css实现的是补间动画

帧动画：使用定时器，每隔一段时间，更改当前的元素

补间动画：过渡（加过渡只要状态发生改变产生动画）动画(多个节点来控制动画)性能会更好

ps:

在支持css3的浏览器尽可能的使用css3动画（适用于移动端开发）

transition animation

transition 组合写法（transition: all 1s linear 1s）

###什么是css盒子模型

一个元素占有空间的大小由几个部分构成，其中包括

* ****元素的内容（content），****
* ****元素的内边距（padding），****内容与边框之间的距离
* ****元素的边框（border），****
* ****元素的外边距（margin），****边框与外部元素之间的距离

四个部分，这四个部分一起构成了****盒子模型****。

###什么是BFC

BFC，****块格式化上下文（Block Formatting Context，BFC）**** 是Web页面的可视化CSS渲染的一部分，是块盒子的布局过程发生的区域，也是浮动元素与其他元素交互的区域。

（1）浮动元素（2）position为absolute或fixed的（3）display为inline-block, table-cell, table-caption, table(table系的) 等（4）overflow 为hidden scroll auto inherit的（5）弹性布局(flex系)（6）网格布局(grid系)

###.v-show,v-if,以及v-if的优点

当v-if的判断条件为false时，包含v-if指令的的元素，根本不会存在dom中，而当条件为true时，它就会为我们在dom中创建这样的一个元素。

v-show，无论条件是否为真，还是为假，包含v-show的指令都会存在我们的dom中，只是它会基于条件的真假，给我们的元素增加一个行内样式:display:block或是display:none。

* 但需要在显示与隐藏之间进行频繁的切换操作时，就使用v-show。
* 当只有一次切换时，我们就使用v-if。

v-if：

优点：在初次页面渲染时，如果条件不满足（为假），将会缩短DOM的渲染时间。从而可以提高产品的用户体验度。

缺点：切换显示时，还需要重新渲染DOM，用户体验度比v-show较差些。

v-show：

优点：切换显示时，不会再重新渲染DOM，用户体验度更高些。

缺点：在初次渲染页面时，会将所有的代码块（元素）渲染完成，根据条件，判断是否显示的DOM树上。

###cookie、localStorage、sessionStorage的区别和使用？

cookie：是存储在本地的数据，有时候也用cookies，通常经过加密，应用最经典的就是判断注册用户是否已经登录过该网站。 localStorage：仅在客户端保存（即浏览器），不参与和服务器的通信；没有时间限制，即使浏览器关闭，数据依然存在；

创建和访问localStorage：

1）、设置数据：

var forgetData = {phone:vm.phone}; localStorage.setItem(“forgetData”,JSON.Stringfy(forgetData)); //forgetData是存储在localStorage里边的本地数据；JSON.Stringfy(forgetData)是将数据转化为字符串格式；

获取数据：

vm.forgetData=JSON.parse(localStorage.getItem(“forgetData”)); //将对象转化为json；

2）、设置：localStorage.name = “zhao”;

获取：localStorage.name //zhao localStorage.setItem(key,value);//设置数据 localStorage.getItem(key);//获取数据 localStorage.removeItem(key);//删除单个数据 localStorage.clear();//清除所有localStorage的数据

sessionStorage：当用户的浏览器窗口关闭时，数据会被清除；

共同点：都是保存在浏览器端，且同源的。

区别：

1. cookie数据始终在同源的http请求中携带即使不需要)，即cookie在浏览器和服务器之间来回传递；cookie数据还有路径的概念，可以限制cookie只属于某个路径下。存储大小限制也不同，cookie数据大小不能超过4K，同时因为每次http请求都会携带cookie，所以cookie只能保存很小的数据。

2.sessionStorage和localStorage不会自动把数据发给服务器，只在本地保存，虽然也有大小限制，但是要比cookie大得多，可以达到5M或者更大。

3.数据有效期不同，sessionStorage仅在当前浏览器窗口关闭前有效，不能持久保存；localStorage：始终有效，浏览器窗口关闭也一直保存；cookie：只在cookie设置的过期时间之前保存，即使浏览器窗口关闭。

4.作用域不同，sessionStorage在不同浏览器窗口的数据不能共享，即使是同一个页面；localStorage在所有的同源窗口中都是共享的；cookie也是在同源窗口中共享的，