一、引入css样式的方式有哪几种？

①内联方式

②直接写在style里面

③link引入

④@import引入

@import会在整个网页加载完成之后再渲染样式，导致事先显示无样式页面，闪烁以后再显示网页样式

二、position的所有值

①absolute ②fixed ③relative ④static默认 ⑤inherit继承父元素

三、box-sizing属性

①content-box border和padding不计算进width的宽度

②padding-box padding计算进width的宽度

③border-box padding和border都不计算进width

ie8+浏览器支持content-box和border-box

火狐支持三个

四、三列布局的css，左右两侧宽固定，中间自适应

中间元素设置margin-left=左边元素宽和margin-right=右边元素宽度

中间的元素需要放在浮动两个元素后面写，否则右浮动元素会被挤下去

五、box-shadow写单边阴影

box-shadow: h-shadow   v-shadow   blur   spread   color   inset;

(box-shadow: 水平阴影 垂直阴影 模糊距离 阴影大小 阴影颜色  内部阴影;)

单边阴影需要舍弃模糊距离，元素有多大，阴影色块就位于下方有多大，h-shadow仅是将他移动而已

六、var a = {n:4};

var b = a;

b.x = a = {n: 10};

console.log(a.x);//undefined

console.log(b.x);//{n:10}

七、cookie、sessionStorage、localStorage的区别

①kookie：储存在本地终端上的数据，主要是利用session判断客户登录状态

②sessionStorage 储存在客户端，浏览器器窗口关闭后。数据删除

③localstorage 储存在客户端，没有时间限制，长久储存

kookie数据始终在同源的http请求中携带，固数据不会超过4k

另外两个不会把数据发送给服务端，仅保存在本地，可保存容量5M或更大

sessionStorage不在不同的浏览器窗口中共享，即使是同一个页面，另外两个在同源窗口都是共享的

八、如何让一个元素相对于父元素居中（考虑到块级，内联，不定宽高）

水平居中：父元素display：flex just-content:space-around

垂直居中：父元素display：flex flex-flow：column just-content：space-around

九、如何利用rem写自适应网页

rem是文字的单位，可以根据屏幕大小自适应

十、call/apply/bind的区别

call和apply区别在于call传参时单个跟参数， apply传参可将多个参数放在一个数组里直接传

bind和其余两个的区别，bind是复制一个函数副本，执行副本函数，而另外两个是立即执行当前函数

一、伪数组、类数组转为数组

①遍历类数组放入新数组中

②arr=Array.Prototype.slice.call(类数组) 让类数组强制调用数组的方法

③arr\_\_proto\_\_=Array.prototype ie是没有提供\_\_proto\_\_

二、判断一个对象是否为数组

①判断原型对象

obj.\_\_proto\_\_==Array.prototype

Array.protetype.isPrototypeOf(obj) 数组的原型对象是否为obj原型对象

②判断构造函数

obj.constractor==Array obj的构造函数等于数组构造函数

obj instanceof Array obj是数组的一个实例吗

以上两种不严谨，可能存在改\_\_proto\_\_属性

③判断隐藏的class属性

Object.prototype.toString().call(obj) 强行调用最顶级的tostring方法

④Array.isArray(obj)

三、promise用法

new promise （ function（open，err）{ } ）.then/.catch 创建对象的同时，自动执行内部的函数

如果open则.then 如果err则catch

Es7

one two three都是promise的实例

(async function(){ try{ await one;await two ;await three } }).catch(err){ }

promise.all([ one ,two,three ])

四、原生实现事件委托

①直接在标签中onclick=“处理函数”

②document获取元素.onclick=处理函数 仅绑定一个

③document获取元素.addEventListener(“click”，处理函数) 可添加多个

五、状态码

2开头 请求成功，并处理了请求的状态

3 重定向

4 请求错误 妨碍了服务器处理

5 服务器错误

六、get post的区别

①传递数据量 post没限制 get 1kb

②get在浏览器回退的时候是无害的，post会再次提交

③get比post更不安全，参数会直接暴露在URL上

④get通过url传参，post通过request body传参

⑤对于GET方式的请求，浏览器会把http header和data一并发送出去，服务器响应200（返回数据）；

而对于POST，浏览器先发送header，服务器响应100 continue，浏览器再发送data，服务器响应200 ok（返回数据）。

七、页面性能优化

①尽量减少http的请求次数

将所有的背景图片放到一个图片文件中，利用background来显示不同的背景图片

②使用ajax实现异步更新

八、vue的生命周期

①beforeCreate() el:undefined data:undefined

②created() el:undefined data:{}

③beforeMount el:空的dom对象 data:{}

④mounted el:完整的dom对象 data:{}

⑤beforeUpdate

⑥undate() data里的值发生变化是触发

⑦beforeDistroy 摧毁前 提示你确认要删除xx

⑧distroyed 删除成功

九、this指向问题

this指向在函数创建时时决定不了的，this永远指向最后调用他的对象，也就是看他执行的时候是谁调用的

①如果这个函数被上一级对象所调用，那么this就指向上一级对象

②如果函数调用时是包含多个对象的，this也只是指向他的上一级对象

③构造函数中的this会因为关键词new而改变this的指向 this会指向实例

③另外我们也可以自行改变this的指向，利用call apply bind的方法

④如果一个函数中有return，假设return返回的是一个对象，那么this就指向这个对象；如果返回的不是一个对象，那么就指向函数的实例；注意：函数就是对象；null也是对象，但是null比较特殊，this不指向null

十、new的作用

①创建一个空的新对象

②让子对象继承构造函数的原型对象

③用新对象调用构造函数 这时如果构造函数中有this，自然就指向新对象

一、for循环的各种情况

①for(var i=0;i<=arr.length;i++) 最灵活

②arr.forEach((elem, I ,arr)=>{}) 无法控制遍历的顺序和步伐

③for(var elem of arr) 会以此获得元素值，保存在elem中，无法获得下标

④for(var key in arr) 专门遍历自定义下标名的关联数组和对象

二、js造成内存泄露的情况

内存泄露：已分配的堆内存由于某种原因程序未释放或无法释放，造成的系统内存浪费

①闭包 外层函数的局部作用域无法被释放

②意外全局变量 定义在函数内的全局变量，函数调用后，变量依然存在

解决：可用‘user strict’

③定时器 当不需要定时器时，定时器并没有被clear，里面的毁掉函数以及所依赖的变量都不能被回收

解决：clearTimeout(\*\*\*) clearInterval(\*\*\*)

④死循环

三、ES5新特性

① 严格模式 user strict

禁止给未声明的变量赋值

静默失败升级为错误 如修改只读属性，不报错也为成功

this不指向window而是指向undefined

②保护对象

value：属性值

writable：是否可修改value

enumerable：是否可遍历

configurable：是否可删除该属性 不可逆

③替换this的三种方法

④遍历API

forEach 每个元素执行相同的操作后再返回到原数组中

map 每个元素执行相同的操作后，放入到新数组中

filter 将符合条件的元素放入新数组中

reduce((prev,elem,I,arr)=>{ },初始值) 每次计算的值保存在prev中

四、ES6新特性

①遍历API forOf

②let 避免声明提前 添加块级作用域

③spread 打散数组专用function(…[ ])

④箭头函数 this的指向内外相同

⑤解构 利用关联数组或对象做形参

⑥Promise 控制异步函数的执行顺序 open则执行then err则执行catch

⑦class 将构造函数和他的原型对象封装为一个整体

五、