1. Oppo面试题
2. 现有代码如下：

let name = ‘aaa’;

let obj = {

name:’bbb’,

dose:function(){

this.name = ‘ccc’;

}

} ;

let does = obj.dose;

does();

console.log(obj.name);//bbb

问题：写出此时打印出的值

1. 编写一个函数解析 Query String ,并返回解析后的对象。

如输入：a=b&c=d

则输出对象为：

{

a:’b’,

c:’d’

}

let str ="a=b&c=d";

function getQueryString(qs){

let args={}

let items = qs.length?qs.split("&"):[];

// console.log(items)

let item = null,

name = null,

value = null

for(let i=0;i<items.length;i++){

item = items[i].split("=");

name = item[0];

value = item[1];

if(item.length){

args[name]=value

}

}

return args;

}

console.log(getQueryString(str))

1. 给下面列表中的li节点绑定点击事件，点击时弹出其在列表中的顺序（页面不跳转）

<ul id=”nav”>

<li><a href=”http://qq.com”>QQ</a></li>

<li><a href=”http://baidu.com”>百度</a></li>

<li><a href=”http://jd.com”>jd</a></li>

<li><a href=”http://taobao.com”>淘宝</a></li>

</ul>

如：点击QQ列则弹出1，点击百度列则弹出2，一次类推

var aLi = document.getElementsByTagName('li');

var aArr = document.getElementsByTagName('a');

for(let i=0;i<aLi.length;i++){

aLi[i].onclick=function(){

alert(i)

// aArr[i].setAttribute("href","javascript:void(0)")

return false

}

}

1. 什么是防抖节流，请实现一个防抖函数、节流函数

<https://www.jianshu.com/p/c8b86b09daf0>

1. 实现一个函数判断变量是Array，Object，String还是Number

typeof instanceof constructor

var a = [1,2,3];

var b = {"lili":"name"};

var c = "aaaa";

var d = 123;

//(arr instanceof Array); instanceof 与constructor 类同

//typeOf 如果你只是用typeof来检查该变量，不论是array还是object，都将返回‘object'。

function judgeType(obj){

if(obj.constructor === Array){

console.log("数组");

}else if(obj.constructor === Object){

console.log("对象");

}else if(obj.constructor === String){

console.log("字符串");

}else if(obj.constructor === Number){

console.log("数值");

}

}

judgeType(a);

judgeType(b);

judgeType(c);

judgeType(d);

6.

for(var a=0;a<10;a++){

setTimeout(function(){

console.log(a);

},1000)

}

输出是什么，为什么

答案是10个10，为什么

7.css实现左侧固定，右侧自适应布局

<https://www.jb51.net/css/641183.html>

1. 写出实现低水平垂直居中的几种方法

<https://www.cnblogs.com/zyzkroos8/p/aceymomo.html>

<https://blog.csdn.net/dengdongxia/article/details/80398430>

1. 使用正则表达式实现连字符命名转换为驼峰命名

<https://www.cnblogs.com/hxc555/p/5932093.html>

1. 实现一个父类Parent，它可以接受一个名字name，同时还有一个eat，一个子类继承这个父类，同时子类覆盖父类的eat方法。

//父级对象

function Person(name){

this.name = name;

this.say = function(){

console.log('我的名字：'+this.name);

}

}

//创建子类

function Teacher(name){

this.name = name;

//继承父类

Person.apply(this,[name]);

//如果子类里的函数名和父类里的函数名一致时叫做将父类的函数重写了,我们调用时调用的是子类中的函数

this.say = function(){

console.log(this.name +'喜欢');

}

}

// 实例化

var teacher1 = new Teacher('花儿','小岛');

teacher1.say();

其他面试题：

1. 实现点击下面的按钮出现提示框，再次点击其他地方提示框消失

提示框

Html:

<div class="box">

<div class="box1">

提示框

</div>

</div>

Js:

<button class="btn">按钮</button>

var obox1 =document.getElementsByClassName("box1")[0];

var obtn =document.getElementsByClassName("btn")[0];

obtn.onclick =function(ev){

obox1.style.display ="none";

ev.stopPropagation();

}

window.onclick =function(){

obox1.style.display="block"

}

1. 请详细说一下你对vue生命周期的理解

总共分为8个阶段创建前/后，载入前/后，更新前/后，销毁前/后。

创建前/后： 在beforeCreated(在创建之前)阶段，vue实例的挂载元素$el和\*\*数据对象\*\*data都为undefined，还未初始化。在created(创建)阶段，vue实例的数据对象data有了，$el还没有。

载入前/后：在beforeMount阶段，vue实例的$el和data都初始化了，但还是挂载之前为虚拟的dom节点，data.message还未替换。在mounted(安装)阶段，vue实例挂载完成，data.message成功渲染。

更新前/后：当data变化时，会触发beforeUpdate(更新前)和updated(更新)方法。

销毁前/后：在执行destroy(破坏)方法后，对data的改变不会再触发周期函数，说明此时vue实例已经解除了事件监听以及和dom的绑定，但是dom结构依然存在

转自17题：<http://www.bslxx.com/p/3187.html>  
Vue2.0 探索之路——生命周期和钩子函数的一些理解:<https://segmentfault.com/a/1190000008010666>

1. 编写冒泡排序

var arr =[4,2,18,20,16];

for(var i=0; i<arr.length-1;i++){

for(var j=0;j<arr.length-i-1;j++){

if(arr[j]>arr[j+1]){

var temp =arr[j];

arr[j]=arr[j+1];

arr[j+1]=temp

}

}

}

console.log(arr)

1. vue 的data，methods,computed,watch之间有什么区别

Methods:方法，只要调用就会重新执行一次，相应的有一些触发条件，在某些时候methods和computed看不出来具体的差别，但是一旦在运算量比较复杂的页面中，就会体现出不一样。

Watch:监听， watch用于观察和监听页面上的vue实例，当然在大部分情况下我们都会使用computed，但如果要在数据变化的同时进行异步操作或者是比较大的开销，那么watch为最佳选择。watch为一个对象，键是需要观察的表达式，值是对应回调函数。值也可以是方法名，或者包含选项的对象。和computed不一样的是：如果在data中没有相应的属性的话，是不能watch的。

Computed：计算，当页面中有某些数据依赖其他数据进行变动的时候，可以使用计算属性。需要注意的是，computed是具有缓存的，这就意味着只要计算属性的依赖没有进行相应的数据更新，那么computed会直接从缓存中获取值，多次访问都会返回之前的计算结果。

1. vuex有哪几种属性

state => state就是存放数据的地方，类似一个仓库 , 特性就是当mutation修改了state的数据的时候，他会动态的去修改所有的调用这个变量的所有组件里面的值（ 若是store中的数据发生改变，依赖这个数据的组件也会发生更新 ）

getters => getter用来获取数据

mutations => 提交更改数据的方法，同步操作！

actions => Action 类似于 mutation，不同在于：

Action 提交的是 mutation，而不是直接变更状态。

Action 可以包含任意异步操作

modules =>使用单一状态树，导致应用的所有状态集中到一个很大的对象。但是，当应用变得很大时，store 对象会变得臃肿不堪。为了解决以上问题，Vuex 允许我们将 store 分割到模块（module）。每个模块拥有自己的 state、mutation、action、getters、甚至是嵌套子模块——从上至下进行类似的分割：

1. 行内元素有哪些？块级元素有哪些？行内元素和块级元素的区别

块级元素：div p h1 h2 h3 h4 ul 行内元素: a b br i span input

区别：

块级元素(block)特性： 总是独占一行，表现为另起一行开始，而且其后的元素也必须另起一行显示; 宽度(width)、高度(height)、内边距(padding)和外边距(margin)都可控制;

行内元素(inline)特性： 和相邻的行内元素在同一行; 宽度(width)、高度(height)、内边距的 top/bottom(padding-top/padding-bottom)和外边 距的 top/bottom(margin-top/margin-bottom)都不可改变（也就是 padding 和 margin 的 left 和 right 是可以设置的），就是里面文字或图片的大小。

1. css3新增伪类有哪些

p:first-of-type 指定父元素的首个 p 元素。

 p:last-of-type  指定父元素的最后一个 p 元素。

 p:only-of-type属于父元素的特定类型的唯一子元素的每个 p 元素。

p:only-child    选择属于其父元素的唯一子元素的每个 <p> 元素。

p:nth-child(2)  选择属于其父元素的第二个子元素的每个 <p> 元素。

 :enabled、:disabled 控制表单控件的禁用状态。

:checked，单选框或复选框被选中。

1. css优先级算法如何计算

<https://www.cnblogs.com/love-sea520/p/5903196.html>

标签 0 0 0 1

Class 0 0 1 0

Id 0 1 0 0

！Important

1. http状态码都有哪些含义

<https://www.cnblogs.com/xflonga/p/9368993.html>

200:请求成功

304：未修改

400：错误请求

403：服务器拒绝请求

500：服务器内部错误

503：服务器不可用

还不全

1. http的定义、特点、请求、请求头、相应响应头等内容，请参见http详解

<https://blog.csdn.net/weixin_38087538/article/details/82838762>

<https://www.jianshu.com/p/34d8c1abf38a>

1. js面向对象中继承的实现方法

原型链继承 call和apply 组合（构造函数+原型链继承）

<https://www.cnblogs.com/Grace-zyy/p/8206002.html>

1. 请实现，鼠标点击页面中任意标签，alert该标签的名称

document.onclick=function(e){

e = e||window.event;

let select = e.target||e.srcElement;

alert(select.tagName)

}

1. 浏览器内核有哪些？分别有什么代表的浏览器

    IE类型的浏览器基本上是使用  trident内核

    非IE浏览器主流有：firefox、safari、opera、chrome及内核分别为：

       firefox：gecko内核

       Safari：webkit内核

       opera：以前是用presto内核，现在已经改用google chrome的blink内核

       chrome：blink（基于webkit，是google与opera software共同开发的

1. 在什么场景下会出现外边距合并？如何合并？如何不让相邻元素外边距合并？给个父子外边距合并的范例

<https://blog.csdn.net/songyi160/article/details/87891222>

1. 如何解决前端浏览器缓存问题
2. 如何实现移动端手势方法（左滑右滑）

<https://www.cnblogs.com/justSmile2/p/10616658.html>

ontouchStart ontouchMove ontouchEnd

1. 请问console.log输出什么

var a= 1;

var b = 3;

var c = 6;

console.log(++a+b+++c+++--c);

17

1. 使用原生js筛选数组当中的质数（素数）[15,19,499,2,36,401,798,3,156,819,64]

var arr=[15,19,499,2,36,401,798,3,156,819,64];

var arr1=[];

for(var i=0;i<arr.length;i++){

var type=true

for(var j=2;j<arr[i];j++){

if(arr[i]%j==0){

type=false

break;

}

}

if(type){

arr1.push(arr[i])

}

}

console.log(arr1)

1. 用纯css 手写一个动画效果，一个盒子边缘淡入进来，进入到页面正中间转一圈，然后离开页面，循环且重复
2. websocket 实现检测链接和断线重连的方法有哪些
3. 讲讲你的前端性能优化心得

优化心得：

### **[减少http请求次数](http://developer.yahoo.com/performance/rules.html" \l "num_http" \t "https://www.cnblogs.com/developersupport/p/_blank)**

80%的响应时间花在下载网页内容(images, stylesheets, javascripts, scripts, flash等)。**减少请求次数**是缩短响应时间的关键！

### **[延迟加载](http://developer.yahoo.com/performance/rules.html" \l "postload" \t "https://www.cnblogs.com/developersupport/p/_blank)**

延迟加载需要我们知道我们的**网页最初加载需要的最小内容集**是什么。剩下的内容就可以推到延迟加载的集合中。

Javascript是典型的可以延迟加载内容。一个比较激进的做法是开发网页时先确保网页在没有Javascript的时候也可以基本工作，然后通过延迟加载脚本来完成一些高级的功能。

### **[提前加载](http://developer.yahoo.com/performance/rules.html" \l "preload" \t "https://www.cnblogs.com/developersupport/p/_blank)**

与延迟加载目的相反，提前加载的是为了提前加载接下来网页中访问的资源，下面是提前加载的类型

无条件提前加载：当前网页加载完成后，马上去下载一些其他的内容。例如google会在页面加载成功之后马上去下载一个所有结果中会用到的image sprite。

### **[减少DOM元素数量](http://developer.yahoo.com/performance/rules.html" \l "min_dom" \t "https://www.cnblogs.com/developersupport/p/_blank)**

网页中元素过多对网页的加载和脚本的执行都是沉重的负担，500个元素和5000个元素在加载速度上会有很大差别。

### **[避免CSS表达式](http://developer.yahoo.com/performance/rules.html" \l "css_expressions" \t "https://www.cnblogs.com/developersupport/p/_blank)**

CSS表达式的问题在于它被重新计算的次数远比我们想象的要多，不仅在网页绘制或大小改变时计算，即使我们滚动屏幕或者移动鼠标的时候也在计算，因此我们还是尽量避免使用它来防止使用不当而造成的性能损耗。

**[使用外部Javascirpt和CSS文件](http://developer.yahoo.com/performance/rules.html" \l "external" \t "https://www.cnblogs.com/developersupport/p/_blank)**

使用外部Javascript和CSS文件可以使这些文件被浏览器缓存，从而在不同的请求内容之间重用。

同时将Javascript和CSS从inline变为external也减小了网页内容的大小。

使用外部Javascript和CSS文件的决定因素在于这些外部文件的重用率，如果用户在浏览我们的页面时会访问多次相同页面或者可以重用脚本的不同页面，那么外部文件形式可以为你带来很大的好处。但对于用户通常只会访问一次的页面，例如microsoft.com首页，那inline的javascript和css相对来说可以提供更高的效率。

### **[精简Javascript和CSS](http://developer.yahoo.com/performance/rules.html" \l "minify" \t "https://www.cnblogs.com/developersupport/p/_blank)**

精简就是将Javascript或CSS中的空格和注释全去掉，统计表明精简后的文件大小平均减少了21%

以下求势科技笔试题

1. css盒子模型，绝对定位和相对定位

标准盒模型： 内容，border ,margin，padding

怪异盒子模型：盒子设置的宽高已经包括了内容，border，padding 总的大小为宽高加margin

https://www.cnblogs.com/leibeiwang/p/6164650.html

绝对定位和相对定位：

相对定位：

a.不影响元素本身的特性

b.不脱离文档流

c.如果没有定位偏移量，对元素本身没有任何影响

绝对定位：

a、使元素完全脱离文档流；

b、如果有定位父级相对于定位父级发生偏移，没有定位父级相对于整个文档发生偏移；

c、相对定位一般都是配合绝对定位元素使用；

1. 清除浮动，什么时候需要清除浮动，清除浮动有哪些方法

<https://www.cnblogs.com/nxl0908/p/7245460.html>

1. 如何保持浮动层水平垂直居中

.parent{position:relative;width:600px;height: 400px; border:1px solid red;}

.children{position: absolute;width: 100px;height: 50px;top: 50%;left: 50%;

-webkit-transform:translate(-50%,-50%);background: pink;}

<div class="parent">

　　 <div class="children">dsfadsf</div>

　　</div>

<https://blog.csdn.net/fuohua/article/details/75734957>

1. Position和display的取值和各自的用法

position

    1、position属性取值：static(默认)、relative、absolute、fixed、inherit。

    2、postision：static；始终处于文档流给予的位置。看起来好像没有用，但它可以快速取消定位，让top，right，bottom，left的值失效。在切换的时候可以尝试这个方法。

    3、除了static值，在其他三个值的设置下，z-index才会起作用。（确切地说z-index只在定位元素上有效）

    4、position：relative和absolute都可以用于定位，区别在于前者的div还属于正常的文档流，后者已经是脱离了正常文档流，不占据空间位置，不会将父类撑开。定位原点relative是相对于它在正常流中的默认位置偏移，它原本占据的空间任然保留；absolute相对于第一个position属性值不为static的父类。所以设置了position：absolute，其父类的该属性值要注意，而且overflow：hidden也不能乱设置，因为不属于正常文档流，不会占据父类的高度，也就不会有滚动条。

    5、position：fixed 旧版本IE不支持，定位原点相对于浏览器窗口，而且不能变。常用于header，footer，或者一些固定的悬浮div，随滚动条滚动又稳定又流畅，比JS好多了。fixed可以有很多创造性的布局和作用，兼容性是问题。

    6、position：inherit。规定从父类继承position属性的值。但是任何版本的IE都不支持该属性值。

    display

    1、display属性取值：none、inline、inline-block、block、flex、inherit。

    2、display属性规定元素应该生成的框的类型。文档内任何元素都是框，块框或行内框。

    3、display：none和visiability：hidden都可以隐藏div，区别有点像absolute和relative，前者不占据文档的空间，后者还是占据文档的位置。

    4、display：inline和block，又叫行内元素和块级元素。表现出来的区别就是block独占一行，在浏览器中通常垂直布局，可以用margin来控制块级元素之间的间距；而inline以水平方式布局，垂直方向的margin和padding都是无效的，大小跟内容一样，且无法设置宽高。inline就像塑料袋，内容怎么样，就长得怎么样；block就像盒子，有固定的宽和高。

    5、inline-block就介于两者之间。

    6、display: flex 意为"弹性盒布局模型"，用来为盒状模型提供最大的灵活性。任何一个容器都可以指定为flex布局。设为Flex布局以后，子元素的float、clear和vertical-align属性将失效。采用flex布局的元素，称为flex容器。它的所有子元素自动成为容器成员，称为flex项目（flex item）。

    容器的属性：以下6个属性设置在容器上。

        flex-direction    决定主轴的方向（项目的排列方向）。

            row（默认值）：主轴为水平方向，起点在左端。

            row-reverse：主轴为水平方向，起点在右端。

            column：主轴为垂直方向，起点在上沿。

            column-reverse：主轴为垂直方向，起点在下沿。

        flex-wrap   默认情况下，项目都排在一条线上。flex-wrap属性定义，如果一条轴线排不下，如何换行。

            nowrap（默认）：不换行。

            wrap：换行，第一行在上方。

1. 样式的层级关系，选择器优先级，样式冲突，以及抽离样式模块怎么写，说出思路（红色的只能按自己理解来说），

1.继承：

如果样式是被继承下来的，优先使用最近祖先元素的样式

2.继承样式<指定样式

3.指定样式冲突，权重高的获胜,权重可以叠加

标签 1

class 10

id 100

内联 1000

行内 10000

伪元素/伪对象(::after) 1

伪类(:link,:hovor) 10

4.权重相同，后者获胜

5.!important 样式属性不被覆盖，不能滥用

1. Css3动画效果属性，canvas，svg区别，css3中新增伪类举例
2. Px,em,rem的区别，css中的link和@import的区别
3. Js中有哪些数据类型，解释清楚null和undefined，解释清楚原始数据类型和引用数据类型，比如讲一下，1和Number(1)的区别
4. 讲一下prototype是什么 东西，原型链的理解，什么时候用prototype

js它没有"子类"和"父类"的概念，也没有"类"（class）和"实例"（instance）的区分，全靠一种很奇特的"原型链"（prototype chain）模式，来实现继承。

原型链:

    每一个对象都有一个\_\_proto\_\_属性,指向构造该对象的构造函数的原型对象

    每个构造函数都有一个prototype属性,这个属性指向它的原型对象,简称"原型"

1. 函数里的this什么含义，什么情况下，怎么用

<https://blog.csdn.net/splendid_can/article/details/53351577>

var box1 = document.getElementById('box1')

// var box = ()=>{

// console.log(this)

// }

// box()

box1.onclick=()=>{

console.log(this)

}

1. call和apply什么含义，什么区别，什么时候用
2. 数组和对象由哪些原生方法，列举一下，分别是什么含义，比如链接两个数组，用哪个方法，删除数组的指定项和重新组装数组（操作数据的重点）
3. 怎样避免全局变量污染，es5严格模式的作用，es6箭头函数和es5普通函数一样么？
4. Vue2.5的双向绑定原理，vue3.0的双向绑定原理

2.5的绑定原理：<https://www.cnblogs.com/wangjiachen666/p/9883916.html>

<https://www.cnblogs.com/zhenfei-jiang/p/7542900.html>

3.0的绑定原理：https://www.jianshu.com/p/68c3e8f2e785

1. Vuex流程，项目中哪些功能使用了vuex
2. React生命周期
3. React的高阶组件

https://www.cnblogs.com/libin-1/p/7087605.html

https://www.cnblogs.com/mengff/p/9657232.html

1. React路由拦截和路由守卫

进入组件的守卫使用的是Redirect

离开的守卫使用的Prompt

登录的路由守卫使用了axios的拦截器和Redirect 一起做的，并进行了封装

1. Redux流程，使用redux时遇到什么问题，都怎么解决的
2. Redux的中间件都有哪些
3. Redux-thunk 和redux-saga的区别
4. 怎样解决地狱回调问题
5. 数组和列表有什么区别
6. 线程-异步线程-事件循环-任务队列都是什么怎么理解

https://www.cnblogs.com/cangqinglang/p/8967268.html

https://www.cnblogs.com/zhengyufeng/p/10895895.html

https://www.cnblogs.com/sunmarvell/p/9564815.html

1. 简述深拷贝和浅拷贝之间的区别，深拷贝的方式都有哪几种距离说明
2. 请用一段代码简述闭包，闭包是什么，为什么要用闭包，一般什么情况下使用，闭包产生的问题，怎样解决
3. Js实现递归算法

function digui(n){

if(n==1){

return 1

}

return n\*digui(n-1)

}

console.log( digui(5))

1. 一个页面从输入url到页面加载显示完成，这个过程都发生了什么
2. 简述vue的传值方式并且详细说明具体过程
3. git流程，git的商用命令，怎样创建分支、切换分支、合并分支
4. vue-cli 如何添加自定指令
5. 简述 let和const的区别

let 定义变量，有局部作用域

const 定义常量

1. 双向绑定和vuex是否冲突，为什么？
2. 请写出下面代码的运行结果，分析一下为什么，（涉及到线程，异步线程，宏任务和微任务）

setTimeout(()=>{

console.log(1);

})

console.log(2);

new Promise((res)=>{

console.log(3);

res();

}).then(()=>{

console.log(4);

})

new Promise((res)=>{

console.log(5);

res();

}).then(()=>{

console.log(6);

})

setTimeout(()=>{

console.log(8);

})

console.log(9);

1. 移动端自适应怎么做？rem为什么能够适应所有的手机屏幕宽度?

1、使用meta标签，这也是普遍使用的方法，理论上讲使用这个标签是可以适应所有尺寸的屏幕的，但是各设备对该标签的解释方式及支持程度不同造成了不能兼容所有浏览器或系统。

首先解释该标签的含义：

<meta name="viewport" content="width=device-width,initial-scale=1.0, minimum-scale=1.0, maximum-scale=1.0, user-scalable=no"/>

2、第二种自适应屏幕尺寸的方法是将页面做成980宽度，在没有viewport标签的情况下，移动设备屏幕范围会显示页面980的宽度，如果页面大于980，则在屏幕范围内显示页面一部分，如果页面小于980，则页面居中两侧显示空白，那么你猜，如果页面宽度等于980会出现什么情况呢？

3、百分比法， 首先应明确一个概年，CSS中的百分比中的百指的是什么，我告诉你指的是父元素，所有百分比都是这样的。子元素宽度50%，那么父元素的宽度就是百，子元 素的padding-left：50%，父元素的宽度是百，子元素的margin-top:20%，那么父元素的高是百。所以body默认宽度是屏幕宽度 （PC中指的是浏览器宽度）子孙元素按百分比定位（或指定尺寸）就可以了，这只适合布局简单的页面，复杂的页面实现很困难。

4、使用css3单位rem， 有人这样解释rem，root-em，就是根部的em，想必em大家都懂的，那么rem就是将根节点html的font-size的值作为整个页面的基准 尺寸，默认html的font-size是16px，即1rem=16px，如果某div宽度为32px你可以设为2rem。当我们把html的 font-size设为20px时，1rem=20px，那么32px=1.6rem了。到这里我们也就了解了rem的用法了，那么怎么用rem来实现不 同尺寸屏幕的自适应呢？在页面载入开始时首先判断window的宽度（是window的宽度($(window).width())，不是屏幕分辩率的宽 度（screen.width），两者的差别请自行查阅），假设宽度为W，一个div在宽度为640px的设计稿的下的宽度为dW1，这样如果html的 font-size为100px，那么这个div的宽度用rem表示是多少呢？计算：div宽度dW2=dW1/100,px与rem之间很好换算，除以 100就可以，这是假定屏幕宽度为640的，而不同宽度的屏幕怎么处理，为了能保证换算容易那就要为html设置一个合适的font-size，计 算：100 / 640 = fontSize / W, fontSize = W / 640 \* 100 = W / 6.4;

解释一下为什么把100作为一个缩放比例，用10或1不是更方便吗，这是因为大多数浏览器font-size的最小值为12px，所以只能用100了。

5、媒体查询， 媒体查询也是css3的方法，我们要解决的问题是适应手机屏幕，这个媒体查询正是为解决这个问题而生，媒体查询的功能就是为不同的媒体设置不同的css样 式，这里的“媒体”包括页面尺寸，设备屏幕尺寸等，比如我们要为宽度小于480px的页面中的class="icon"的元素设置样式，可以这样 写，@media screen and (max-width=480px) {.icon{ some styles }};这里仅介绍这种思路，关于媒体查询的详细用法请参阅css手册

1. Echars的使用，项目中常遇到的问题

<https://blog.csdn.net/bluefish_flying/article/details/85259784>

1. es6的新特性有哪些，es7,es8呢？
2. 箭头函数为什么可以保留this的指向
3. Promise的三种状态，promise的使用，promise中的catch的使用
4. 跨域的解决方式
5. Ajax和jsonp的区别
6. 写出以下的输出结果

console.log('1'+2+3); //123

console.log(''==false);//true

console.log(0==false);//true

console.log(undefined == false);//false

console.log(1+2+'3');//33

console.log('0'==false);//true

console.log(-1==false);//false

console.log(!!undefined == false);//true

//!undefined true

console.log(!!undefined)

1. 写出打印结果

['Google','Apple','Microsoft'].sort();//Apple Google Microsoft

['Google','apple','Microseft'].sort();//Google Microseft apple

[10,20,1,2].sort();//1 10 2 20

打印结果为双斜杠后面内容，原因是 sort 默认按字符的ascii码的顺序进行排序，如果第一个ascii相同则比较第二个ascii的值，以此类推

//如想按数值输出排序结果，则需按下面的函数进行操作

function aa(a,b){

return a-b

}

console.log([10,20,1,2].sort(aa))

1. 请列举一些vue指令并写出他的用法

v-model v-html v-for v-bind v-show v-if v-on

1. 什么是虚拟dom，为什么虚拟dom可以优化性能
2. vuex和redux的区别
3. 强制缓存和协商缓存

强制缓存：是在第一次访问服务器取到数据之后，在过期时间之内不会再去重复请求；要实现强制缓存这个流程的核心是“如何知道当前的时间是否超过了过期时间，”；

在 http 1.0 版本中，强制缓存通过 Expires 响应头来实现；

在 http 1.1 版本中，强制缓存通过 Cache-Control 响应头来实现

强制缓存只有首次请求才会跟服务器通信，读取缓存资源时不会发出任何请求，资源的 Status 状态码为 200

协商缓存：协商缓存每次读取数据时都需要跟服务器通信，并且会增加缓存标识；第一次是拿数据和标识的过程，第二次开始，就是浏览器询问服务器资源是否有更新的过程；

在http的1.0版本和http1.1版本中都是通过响应头来实现的。

两种缓存的两个版本的实现优先级都是1.1高于1.0

1. 小程序的支付流程，登录流程，支付或登录失败返回什么
2. Asnyc/await 多条数据，获取耗时分别是t1,t2,t3,怎么让耗时小于三者之和，只能用async/await
3. 什么是优雅降级与渐进增强

优雅降级：Web站点在所有新式浏览器中都能正常工作，如果用户使用的是老式浏览器，则代码会检查以确认它们是否能正常工作。由于IE独特的盒模型布局问题，针对不同版本的IE的hack实践过优雅降级了,为那些无法支持功能的浏览器增加候选方案，使之在旧式浏览器上以某种形式降级体验却不至于完全失效.

渐进增强：从被所有浏览器支持的基本功能开始，逐步地添加那些只有新式浏览器才支持的功能,向页面增加无害于基础浏览器的额外样式和功能的。当浏览器支持时，它们会自动地呈现出来并发挥作用。

1. 小程序生命周期

页面生命周期:

https://developers.weixin.qq.com/miniprogram/dev/framework/app-service/page-life-cycle.html

组件生命周期

https://developers.weixin.qq.com/miniprogram/dev/framework/custom-component/lifetimes.html

1. 响应式布局与自适应的区别

响应式布局就是在不同屏幕不同分辨上实现不同的展示方式。响应式布局能使网站在手机和平板电脑上有更好的浏览体验，也就就是说一个网站能后兼容多个终端，而不是为了每一个终端做一个特定的版本。

什么是自适应布局

自适应布局就是能忘了使网页自适应的显示在不同大小终端设备上的新网页设计方式及技术，需要开发多套界面来适应不的终端。

响应式布局与自适应布局的区别是什么

1、自适应布局是通过检测视口分辨率来判断当前访问的设备是pc端还是平板或者是手机，从而请求服务层返回不同的页面；

响应式布局是通过检测视口分辨率针对不同客户端在客户端做代码处理，来展现不同的布局和内容

2、自适应布局需要开发多套界面，而响应式布局只需要开发一套界面就可以了

3、自适应对页面做的屏幕适配是在一定范围：比如pc端一般要大于1024像素，手机端要小于768像素，而响应式布局是一套页面全部适应。

4、自适应布局如果屏幕太小会发生内容过于拥挤，响应式布局可以自动识别屏幕宽度并作出相应的调整页面设计

1. 怎样解决http协议的无连接性
2. 两边固定中间自适应，使用flex布局

.box1{

height: 200px;

display: flex;

}

.box2{

width: 200px;

background: yellow;

height: 200px;

}

.box4{

width: 200px;

background: red;

height: 200px;

}

.box3{

flex:1 0 0;

height: 200px;

background: salmon;

}

<div class="box1">

<div class="box2">左边</div>

<div class="box3">中间</div>

<div class="box4">右边</div>

</div>

1. 后端如何识别cookie，cookie的结构，cookie详解，cookie是否能跨域，
2. 抓包工具 fiddler charles
3. 做的项目中遇到哪些问题，怎么解决的（时常被问）
4. 浏览器缓存都有哪些，泣别是什么
5. 如果浏览器关闭后，不保存本地缓存，要怎么实现
6. Vue的项目怎么搭建

1. 查看node版本

node -v

v10.16.1

2. 查看vue版本

vue -V (Vue不是内部或外部命令...)--执行步骤3

3. 安装@vue/cli脚手架

npm i -g @vue/cli

4. 创建新项目

vue create maoyan

1.Please pick a preset:

Manually select features

Check the features needed for your project:(上下键移动, 空格键选择, 选完之后按回车确定)

Babel Router Vuex CSS Pre-processors

Use history mode for router?

y

Pick a CSS pre-processor

Sass/SCSS(width node-sass)

Where do you prefer placing config for Babel, PostCSS, ESLint, etc.? (Use arrow keys)

In package.json

5. 进入项目目录下, 并启动服务

cd maoyan 进入项目目录

npm run serve 启动服务

1. React项目怎么搭建

1.node环境

2.全局安装脚手架m install --save g create-react-app

3.进入想要创建的项目的目录

E:

np

cd www

4.创建项目

create-react-app 项目名

5.进入项目

cd 项目名

6.运行项目

npm start

7.安装 react-router-dom的依赖

npm install --save react-router-dom

8.打包

npm run build

1. 关闭浏览器会不会执行vue的destory钩子函数
2. Iframe怎么实现父子页面之间的传参
3. 平时有逛过技术论坛么，有看过什么书籍么，

#### 有空或者打代码打累的时候会去逛逛，主要是逛简书，掘金，思否，CSND博客等这几个论坛吧，看的书籍大都是以前学原生js的时候看的，记得书名的就是《**Web前端开发最佳实践》、《Javascript设计模式与开发实践》**

1. Typeof 检测null和undefined会得到什么结果
2. 数组的那些方法会改变自身结构

var arr=[8,3,4,5,9,2,7,1]

//splice,reverse,push,pop,shift,unshift,fill,sort,copyWithin

// console.log(arr.copyWithin(1,2,5))

1. 组件里面的data为什么要return数据
2. Vue的修饰符都有哪些

<https://www.cnblogs.com/aaronthon/p/9225943.html>

1. Vue的mixin是做什么的
2. 在不使用循环的情况下写出一个值与下标相同，长度为100的数组，[0,1,2,3,...,99]

var d = Array.apply(null,{length:100})

var e = Object.keys(d)

console.log(e)

1. 实现一个类似bootstrapde Modal弹出框
2. H5有哪些新标签和新特性
3. Css3动画有多少种方式，各有什么特点
4. 传输数据用哪两种方式，有什么区别
5. 数组倒转，数组降序，各写一个函数，

原数组：[3,6,2,4,1,5]

1. var arr = [“那”,“你”,“很”,“棒”,“棒”,“哦”,“！”]如何把这个数组转换为一句话？

var arr = ['你',"很","棒","棒","哦","！"]

var b = arr.join("")

console.log(b)

var c= arr.toString().replace(/,/g ,"")

console.log(c)

109、字符串反转，如将 '12345678' 变成 '87654321'；

//思路：先将字符串转换为数组 split()，利用数组的反序函数 reverse()颠倒数组，再利用 jion() 转换 为字符串

var str = '12345678'; str = str.split('').reverse().join('');

1. 如何获取当前页面的cookie

function getCookie(name){

var strcookie = document.cookie;//获取cookie字符串

var arrcookie = strcookie.split("; ");//分割

for ( var i = 0; i < arrcookie.length; i++) {

var arr = arrcookie[i].split("=");

if (arr[0] == name){

return arr[1];

}

}

return "";

}

1. display：none与visibility：hidden的区别

1.display:none是彻底消失，不在文档流中占位，浏览器也不会解析该元素；visibility:hidden是视觉上消失了，可以理解为透明度为0的效果，在文档流中占位，浏览器会解析该元素；

2.使用visibility:hidden比display:none性能上要好，display:none切换显示时visibility，页面产生回流（当页面中的一部分元素需要改变规模尺寸、布局、显示隐藏等，页面重新构建，此时就是回流。所有页面第一次加载时需要产生一次回流），而visibility切换是否显示时则不会引起回流。

所以我使用visibility:hidden，在页面渲染时第二个tab页中的轮播图就可以获取宽度做自适应了。

1. 定义一个函数，使用闭包，每调一次，返回值加1
2. 写出下面代码块执行的结果

var bar = 1;

function test(){

console.log(bar);

var bar = 2;

console.log(bar);

}

test();

1. 写出以下代码执行的结果

function fn(){

this.a = 0;

this.b = function(){

console.log(this.a);

}

}

fn.prototype = {

b:function (){

this.a = 20;

console.log(this.a);

},

c:function(){

this.a = 30;

console.log(this.a);

}

}

var obj = new fn();

obj.c();

obj.b();