WEB前端知识点汇总

1. var、let、const的区别？

答：(1)var声明的变量会提升，let/const不会

(2)相同作用域内，var可重复声明变量，let/const不能

(3)const声明常量时必须设置初始值且不能改变

(4)let/const声明的变量会形成块级作用域，而var不会

1. null、undefined的区别？

答：(1)类型不一样:console.log(typeOf null);//object

console.log(typeOf undefined);//undefined

(2)转化为值时不一样:console.log(Number(null));//0

console.log(Number(undefined));//NaN

(3) undefined===null;//false

undefined==null;//true

1. join、split的区别？

答：(1)join('-'):将数组中的元素按照指定字符转为字符串

(2)split(','):将字符串按照指定的字符切割为数组

1. 数组中push、pop、shift、unshift区别？

答：(1) push(value1,value2,...):往数组末尾添加元素，返回新数组长度

(2) pop():删除数组的最后一位元素，返回删除的元素

(3) shift():删除数组的第一位元素，返回删除的元素

(4) unshift(value1,value2,...):往数组开头添加元素，返回新数组长度

1. 谈谈你对闭包的理解？

答：(1)定义：受内层函数引用着无法释放的外层函数的作用域对象

(2)优点：可以避免全局变量的污染

缺点：闭包会常驻内存，会增大内存使用量，使用不当很容易造成内存泄露

(3)闭包有三个特性：

a.函数嵌套函数

b.函数内部可以引用外部的参数和变量

c.参数和变量不会被垃圾回收机制回收

1. 严格模式和非严格模式的区别？

答：(1)禁止给未声明的变量赋值：避免全局污染

(2)静默失败升级为错误：便于调试

(3)函数调用中的this不再指向window，而是undefined：杜绝通过this造成全局污染

(4)禁用了argument.callee：禁用递归

1. DOM的事件模型？

答：(1)从事件发生开始，到所有处理函数执行完，所经历的过程

(2)三个阶段：

a.捕获阶段：从顶级父元素<html>，到当前触发事件的元素，进行遍历。

记录各级父元素上绑定的事件处理函数

b.目标触发：优先触发目标元素上的事件处理函数

c.冒泡阶段：从当前目标元素向上，依次触发捕获阶段记录的各级父元素上的处理函数

1. js中由异步变同步的方法？

答：(1)promise：new Promise((resolve,reject)=>{ ... }).then(res=>{ ... }).catch(res=>{ ... })

(2)async-await：async用于声明一个function是异步的，await是等待一个异步方法执行完成

async function demo(){

let result=await Promise.resolve(123);

};

demo();

(3)generate函数：

(4)jQuery中的when()，then()：

1. ES6中的箭头函数和普通函数的区别？

答：(1)箭头函数是匿名函数，不能作为构造函数，不能使用new

(2)箭头函数不能绑定arguments，取而代之用rest参数...解决

(3)箭头函数没有原型对象

(4)箭头函数的this永远指向其上下文的this，没有办法改变其指向；普通函数的this指向调用它的对象

1. 列举异步使用场景以及目前js解决异步的方案？

答：

1. 在js中call、apply、bind有什么作用？区别？

答：(1)作用：改变this指向

(2)区别：①call、apply:临时性；apply可以打散数组

②bind:永久性

1. 在js中，继承有哪些方法？

答：(1)原型链继承：让新实例的原型等于父类的实例:Per.prototype=new Person();

(2)构造函数继承：用.call()和.apply()将父类构造函数引入子类函数(在子类函数中做了父类函数的自执行(复制))

(3)组合继承：结合了以上两种模式的优点，传参和复用

(4)原型式继承

(5)寄生式继承

(6)寄生组合继承

1. 浏览器对象模型BOM里常用的对象？window对象的常用方法？

答：(1)BOM：window、document、location、screen、history、navigator、event

(2)window：alert()、confirm()、prompt()、open()、close()

1. vue中$route与$router的区别？

答：(1)$route对象：表示当前的路由信息，包含了当前URL解析得到的信息(当前的路径、参数、query对象等)

(2)$router对象：是全局路由的实例，是router构造方法的实例

路由实例方法：

①push：this.$router.push('/index');//等同于<router-link :to='...'>

②go：this.$router.go(-1);//后退

③replace：this.$router.replace('/');//替换当前的页面，不会向history栈添加一个新的记录

1. ES6新特性？

答：(1)块级作用域：let、const

(2)箭头函数：sort((a,b)=>a-b);

(3)函数增强：给参数设置默认值:function add(a,b,c=0){...}

(4)模板字符串：`${JS表达式}`

(5)Spread/Rest操作符：...打散数组:console.log(...[1,2,3]);//1 2 3

(6)二进制和八进制字面量：console.log(0o10);//8 console.log(0b10);//2

(7)对象和数组解构：var student={name:'limao',age:18,sex:'男'}; var {name,age,sex}=student;

(8)对象超类：允许在对象中使用super方法：和this类似，但指向当前对象的原型对象

(9)for...of和for...in：前者用于遍历一个迭代器(数字下标)，后者用于遍历对象或关联数组

(10)promise

1. HTTP常见的状态码？

答：1XX:正在请求

200:请求成功

301:永久重定向 302:临时重定向 304:命中缓存

403:不允许访问 404:找不到资源 405:请求方法不被允许

500:服务器故障 502:网关故障

1. get、post的区别？

答：(1)get:明文提交，大小限制2kb，向服务器要数据的时候使用，无请求主体

(2)post:隐式提交，大小不限制，给服务器数据的时候使用，有请求主体

1. 盒子模型？

答：(1)未设置盒子模型：占地宽度=margin+border+padding+width(内容区域)

(2)设置了盒子模型：占地宽度=margin+width(内容区域+border+padding)

1. 解决浮动元素父元素高度坍塌的方法？

答：(1)父元素最后子元素位置处添加一个空的div，clear:both;

(2)父元素手动设置宽高

(3)父元素overflow:hidden;

(4)父元素也浮动起来

(5)父元素:after{

content:"";

display:block;

clear:both;

}

1. 页面上隐藏元素的方法？区别？

答：(1)display:none; //不占据页面空间

(2)visibility:hidden; //占据页面空间

(3)rgba(0,0,0,0); //占据页面空间，元素透明

1. rgba和opacity的区别？

答：opacity会继承父元素的opacity属性，而rgba只会对元素本身产生影响，无法继承

1. 元素在容器内居中？

答：(1)弹性布局：display:flex; justify-content:center; align-items:center;

(2)定位1---translate：父元素:position:relative;

子元素:position:absolute; top:50%; left:50%; transform:translate(-50%,-50%);

(3)定位2---margin：父元素:position:relative;

子元素:position:absolute; top:50%; left:50%; margin:-100px 0 0 -100px;

(4)定位3：父元素:position:relative;

子元素:position:absolute; top:0px; bottom:0px; left:0px; right:0px; margin:auto;

1. 图片在容器内居中？

答：(1)同定位3

(2)display:table-cell; vertical-align:middle; text-align:center;

1. 在CSS中定位的方式有哪些？区别？

答：(1)相对定位(relative)：没有脱离文档流，相对自己原来位置偏移，通常作为绝对元素的已定位祖先元素使用

(2)绝对定位(absolute)：脱离文档流，相对已定位的祖先元素的左上角偏移；如果没有，相对于body定位

(3)固定定位(fixed)：脱离文档流，永远相对于body定位

(4)浮动定位

(5)普通流定位

1. jquery中$的原理？

答：$本质上等效于new jquery()，创建jquery类型的子对象。

(1)$("选择器")：查找DOM元素，并保存进jquery对象中

(2)$(DOM元素)：比如$(this)、$(e.target)，直接将DOM元素保存进jquery对象中

(3)$(`HTML片段`)：用HTML片段创建新元素，保存进jquery对象中

(4)$(function(){...})：绑定DOMContentLoaded事件，在DOM内容加载后就提前自动执行js代码

1. 判断数据类型的方法？

答：(1)Object.prototype.toString.call()：console.log(Object.prototype.toString.call(123));//[object Number]

(2)typeof：console.log(typeof 123);//number

console.log(typeof [1,2,3]);//object

(3)instanceof：判断这个构造函数的原型是否在给定对象的原型链上

console.log(123 instanceof Number);//false

console.log([1,2,3] instanceof Array);//true

(4)constructor：console.log(arr.constructor === Array); //true

(5)\*判断是否为数组Array.isArray()：console.log(Array.isArray([1,2,3]));//true

1. Object.keys代替for...in遍历元素?

答：console.log(Object.keys({a:1,b:2,c:3}));//['a','b','c']

console.log(Object.values({a:1,b:2,c:3}));//[1,2,3]

1. 在h5中，webStorage和Cookie有什么区别？

答：(1)储存空间不同

1.webStorage能提供5M的存储空间(不同浏览器不同)，Cookie提供4K的空间

2.webStorage每个域(包括子域)都有独立的存储空间，每个存储空间是完全独立的，因此不会造成数据混淆

(2)与服务端交互

1.webStorage中数据仅仅是本地存储，不会与服务器发生任何交互

2.Cookie的内容会随着请求一并发送到服务器(每请求一个新的页面，Cookie都会被发送过去，造成宽带浪费)

(3)接口

1.webStorage提供了许多丰富易用的接口，拥有setItem，getItem，removeItem等方法，操作数据更简单

2.Cookie需要自己封装setCookie，getCookie等

1. localStorage和sessionStorage有什么区别？

答：(1)生命周期

localStorage的生命周期是永久的，关闭页面或浏览器之后localStorage中的数据也不会消失。localStorage除非主动删除数据，否则数据永远不会消失。

sessionStorage的生命周期是仅在当前会话下有效。sessionStorage在关闭了浏览器窗口后就会被销毁。同时独立的打开同一个窗口同一个页面，sessionStorage也是不一样的。

(2)存储大小

localStorage和sessionStorage的存储数据大小一般都是：5MB

(3)存储位置

localStorage和sessionStorage都保存在客户端，不与服务器进行交互通信

(4)存储内容类型

localStorage和sessionStorage只能存储字符串类型，对于复杂的对象可以使用ECMAScript提供的JSON对象的stringify和parse来处理

(5)获取方式

localStorage：window.localStorage

sessionStorage：window.sessionStorage

(6)应用场景

localStorage：常用于长期登录(+判断用户是否已登录)，适合长期保存在本地的数据

sessionStorage：敏感账号一次性登录

1. webSocket的工作原理？

答：webSocket的工作方式是广播和收听，用第三方模块socket.io通过事件绑定与事件触发完成数据的发送与接收，服务器端需要安装socket.io模块，而客户端需要下载socket.io.js文件，一端用.emit发送数据，另一端用.on接收数据。

1. 面向对象的三大特点？

答：(1)封装：创建一个对象保存一个事物的属性和功能，便于大量数据的管理和维护

(2)继承：父对象中的成员，子对象无需重复创建就可无条件直接使用，可节约内存

(3)多态：一个函数在不同情况下表现出不同的状态，包括2种：

①重载：根据传入参数的个数或类型不同，执行不同的逻辑

②重写：子对象对从父对象继承来的方法体进行修改或重写

1. 谈谈对promise的理解？

答：在promise之前解决异步函数的处理是使用回调函数，但是回调函数会造成深层嵌套，代码可读性差，容易造成回调地狱，ES6提出的promise解决了回调地狱。

(1).then()：通过同步链式结构来编写异步函数，但有2个需要注意的地方，使用.then()方法时，必须将new Promise return出函数外部；只要下一个任务函数返回的是new Promise，就可以继续使用.then()方法

(2)resolve：将promise的状态设置为完成，可以将其中的参数回调给.then()的下一个函数使用

reject：将promise的状态设置为失败，此时.then()方法执行失败时的回调

(3).catch()：1.可以用来指定reject的回调

2.接收err()回传的错误

3.当执行resolve方法时，若抛出异常，则会进入catch方法中，并且不会返回参数

1. this指向？

答：(1)obj.fun() this->点前的obj

(2)fun()或(function(){...})() this->window

(3)new Fun() this->正在创建的，但是暂时还没有创建完的这个新对象

(4)Student.prototype.intr=function(){… this …} this->将来调用这个函数的点前的某个子对象

1. ajax、axios的区别？

答：(1)传统ajax指的是XMLHttpRequest，最早出现的发送后端请求技术，但如果多个请求之间有先后关系的话，就会出现回调地狱。它本身是针对MVC的编程，不符合现在前端MVVM的浪潮；单纯使用ajax却要引入整个jquery非常的不合理

(2)axios是基于promise的专门发送http/xhr请求的函数库，它是promise的实现版本，支持promise API，符合最新的ES规范，并提供了一些并发请求的接口

1. 什么是MVVM？

答：MVVM是对前端代码三大部分的重新划分:

(1)界面(View)：增强版的HTML，提供了变量、分支、循环

(2)数据模型(Model)：专门保存页面所需的数据变量

(3)界面模型(ViewModel)：将数据模型和界面绑定在一起，可自动维护数据模型中的变量值与界面中显示内容始终保持一致

1. 什么是虚拟DOM树？为什么使用？

答：(1)仅保存页面中可能受影响的少量元素的简化版DOM树

(2)①遍历快：虚拟DOM树只包含受影响的个别元素

②效率高：仅更新本次受影响的少量DOM元素

③封装DOM操作，避免重复代码：

1. 深克隆、浅克隆的区别？

答：(1)浅克隆：新旧对象中的基本数据类型中的属性值会发生按值传递，但引用数据类型都会指向同一地址值

(2)深克隆：新对象创建副本时，保存与旧对象中同名的属性和值，但地址值不同

1. 防抖、节流？

答：(1)防抖：指触发事件后在n秒内函数只执行一次，如果在n秒内又触发了事件，则会重新计算函数执行时间

(2)节流：指连续触发事件但是在n秒中只执行一次函数

1. vue中的指令？

答：(1)元素的属性值需要变化：v-bind，简写为:

(2)一个元素显示隐藏：v-show

(3)二选一显示：v-if v-else

(4)多选一显示：v-if v-else-if v-else

(5)反复生成多个相同结构的元素：v-for

(6)绑定事件：v-on，简写为@

(7)绑定HTML代码片段：v-html

(8)防止用户短暂看到{{}}：v-cloak v-text

(9)只希望在首次加载时绑定一次：v-once

(10)防止内容中的{{}}被编译：v-pre

(11)只要绑定表单元素：v-model

1. 在HTML5中新增了哪些特性？

答：(1)语义标签：header、footer、nav、section、article、aside...

(2)增强型表单：color、date、datatime、email、month...

(3)视频和音频：video、audio

(4)canvas绘图：2D位图

(5)svg绘图：2D矢量图

(6)地理定位

(7)拖放API

(8)webworker：为js创造多线程环境，允许主线程创建 Worker 线程，将一些任务分配给后者运行

(9)webstorage：本地存储用户的浏览数据，分为sessionStorage和localStorage

(10)websocket：浏览器和服务器之间可以创建持久性的连接，并进行双向数据传输，如：多人聊天室

1. canvas和svg绘图有什么区别？

答：canvas是通过js来绘制的2D位图，svg是使用XML描述的2D矢量图

|  |  |
| --- | --- |
| **canvas** | **svg** |
| 依赖分辨率 | 不依赖分辨率 |
| 不支持事件处理器 | 支持事件处理器 |
| 文本渲染能力弱 | 最适合带有大型渲染区域的应用程序(比如谷歌地图) |
| 能够以.png或.jpg格式保存结果图像 | 复杂度高会减慢渲染速度  (任何过度使用 DOM 的应用都不快) |
| 最适合图像密集型的游戏，其中的许多对象会被频繁重绘 | 不适合游戏应用 |

1. 如何避免重排重绘？

答：(1)尽量用class来批量修改css样式

(2)如果必须用style修改，也可放在一句话中修改多个css属性：div.style.cssText="width:200px;height:100px"

1. DOM中的优化？

答：(1)查找：如果查找条件简单，首选按HTML特征查找；否则，就按选择器查找

(2)修改样式：尽量减少修改样式的次数，避免反复重排重绘

(3)添加元素：尽量减少查找DOM树的次数

①如果同时添加父、子，先将子加入父，再一次性将父添加到DOM树

②如果父已经在页面上，要添加多个平级子，先将子添加到文档片段，再一次性将文档片段添加到DOM树

1. 如何阻止事件冒泡？如何利用冒泡？如何阻止默认行为？

答：(1)e.stopPropagation()：不会阻止自己的代码执行，只会阻止自己的父元素上的处理函数执行

(2)事件委托：如果多个平级子元素需要绑定相同的事件时，只需将事件绑定在父元素上，所有子元素通过冒泡

共用父元素上的一个处理函数，用e.target代替this

(3)e.preventDefault()

1. 跨域？

答：(1)跨域：一个域名下的网站，引用或请求了另一个域名的网站的资源

(2)同源策略("协议+域名+端口"三者相同)：浏览器只允许当前网站的ajax请求自己域名下的接口地址

(3)解决跨域问题：4种

①跨域资源共享cors：app.use(cors({

origin:['http://localhost:8080',"http://127.0.0.1:5500"],//允许跨域的客户端网站地址

credentials:true

}))

②jsonp：非正式传输协议。核心原理：动态添加<script>标签来调用服务器提供的js脚本

<script>

function demo(result){

console.log(result);

}

</script>

<script src=”index.json?callback=demo”></script>

\*index.json文件中保存要被请求的数据

③Nginx反向代理：

④\*proxy(vue中常用)：

1. vue中组件间通信？

答：(1)父传子：子组件通过props属性，绑定父组件数据，实现双方通信

(2)子传父：将父组件的事件在子组件中通过$emit触发

(3)兄弟间传值：$on $emit

1. Vuex？

答：(1) Vuex 是一个专为 Vue.js 应用程序开发的状态管理模式

(2)包含5种基本对象：

①State：单一状态树，存储状态(变量)

②Getter：对数据获取之前的再次编译，可以理解为state的计算属性。我们在组件中使用$store.getters.fun() ③Mutation：修改状态，并且是同步的。在组件中使用$store.commit('',params)

④Action：类似于mutation，但是异步操作。在组件中使用$store.dispath('')

⑤Module：store的子模块，为了开发大型项目，方便状态管理而使用的

1. watch、computed的区别？

答：

|  |  |  |
| --- | --- | --- |
|  | **watch** | **computed** |
| **特性** | 监听事件 | 计算属性 |
| **应用** | 监听props，$emit或本组件的值执行异步操作 | 简化tempalte里面{{}}计算和处理props或$emit的传值 |
| **缓存** | 无缓存性，页面重新渲染时值不变化也会执行 | 具有缓存性，页面重新渲染值不变化,计算属性会立即返回之前的计算结果，而不必再次执行函数 |

1. vue的双向数据绑定的原理？

答：vue.js 是采用数据劫持结合发布者-订阅者模式的方式，通过Object.defineProperty()来劫持各个属性的setter，getter，在数据变动时发布消息给订阅者，触发相应的监听回调

1. webpack打包的流程？

答：

1. new Object()和Object.create()的区别？

答：(1)new Object()是创建新的对象，继承内置对象Object；

Object.create()是继承指定对象

(2)let obj1 = Object.create({x:1}); //obj1 = {}

let obj2 = new Object({x:1}); // obj2 = {x: 1}

1. 比较React与Vue？

答：(1)Vue组件分为全局注册和局部注册，在react中都是通过import相应组件，然后模板中引用

(2)Vue多了指令系统，让模板可以实现更丰富的功能，而react只能使用JSX语法

(3)Vue增加了语法糖computed和watch，而在react中需要自己写一套逻辑来实现

(4)react的思路是通过js来生成html、操作css；而Vue是把html，css，js组合到一起，html提供了模板引擎来处理

1. vue-router中的hash、history模式的区别？

答：(1)hash：指url尾巴后的#号以及后面的字符。hash不会被包含在http请求中，改变hash不会重新加载页面

(2)history：前端的url必须和实际后端发起请求的url一致，否则将返回404错误

1. 前端的性能优化？

答：(1)减少HTTP请求：精灵图、合并CSS和JS文件、懒加载

(2)控制资源加载优先级：将CSS文件放在head内部，JS文件放在body底部

(3)利用浏览器缓存

(4)使用CDN，即内容分发网络：依靠部署在各地的服务器，使用户就近获取所需内容

(5)减少重排：避免频繁操作DOM

1. 数组去重？

答：(1)//方法一:ES6中Set

function unique(arr){

   return Array.from(new Set(arr));

}

(2)//方法二:for循环+splice

    function unique(arr){

      for(var i=0;i<arr.length;i++){

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

          if(arr[i]==arr[j]){

            arr.splice(j,1);

            j--;

          }

        }

      }

      return arr;

    }

(3)//方法三:indexOf

    function unique(arr){

      var arr1=[];

      for(var i=0;i<arr.length;i++){

        if(arr1.indexOf(arr[i])===-1){

          arr1.push(arr[i])

        }

      }

      return arr1;

    }

(4)//方法四:sort()

    function unique(arr){

      arr=arr.sort();

      var arr1=[arr[0]];

      for(var i=1;i<arr.length;i++){

        if(arr[i]!==arr[i-1]){

          arr1.push(arr[i]);

        }

      }

      return arr1;

    }

1. v-for中key的作用？

答：提高渲染性能，避免数据混乱

1. 页面导入样式时，<link>和@import的区别？

答：(1)<link>属于XHTML标签，除了加载CSS外，还能用于定义RSS，定义rel链接属性等作用；

而@import是CSS提供的，只能用于加载CSS

(2)页面被加载时，<link>会同时被加载；

而@import引用的CSS会等到页面被加载完再加载

(3)<link>属于XHTML标签，无兼容性问题；

而@import是CSS2.1提出的，只在IE5以上才能被识别