## 2019/7/12

1.请问cookie，localStorage，sessionStorage的区别

2.COOKIE和SESSION的区别和用法

## 2019/7/11

程序出现bug了，你是如何调试的

平时工作中是如何进行性能优化的

都有哪些方式可以实现两个页面间的通信。

在javascript对象上定义一个repeatify函数,

这个函数接受一个整数参数,来明确子字符串需要重复几次,

这个函数要求字符串重复指定的次数,

比如:'abc'.repeatify(3);//"abcabcabc"

1.//定义函数:

String.prototype.repeatify=function(n){ 2.//给string对象添加一个新函数

var str=this;

var newStr="";

for(var i=1;i<=n;i++){

newStr+=str;

}

return newStr;

}

"abc".repeatify(3); //abcabcabc

## 2019/7/8

for(var i=0;i<3;i++){setTimeout(()=>console.log(i),1);} //三个3

const shape={

radius:10,

diameter(){

return this.radius\*2;

},

perimeter:()=>2\*Math.PI\*this.radius

}; 原题没有打印，只调用了。所以没有结果

console.log(shape.diameter()); //20

console.log(shape.perimeter()); NaN

var foo={

bar:function(){ return this.baz;},

baz:1

}

f=foo.bar;

console.log(f()); //undefined

## 2019/7/5

解释jsonp的原理，以及为什么不是真正的ajax

你用过的es6有哪些

说说你常用的伪类有哪些，分别有什么用

## 2019/7/4

1. 页面有一个按钮button id为 button1，通过原生的js如何禁用？最少写出一种办法

2. 请编写一段JavaScript脚本生成下面红色这段DOM结构。并添加到#a1上

<div id="a1">

<div id="D1">DOM</div>

</div>

3. 双向绑定和单向数据绑定的优缺点？

双向绑定:

优点: 方便，页面和data其中一个有变化，另一个即时更新;

缺点: 不便于控制，不知道数据何时因何改变，只能通过watch监视；无法追踪局部状态的变化，增加了debug的难度；数据变化来源入口不上一个，容易因为一处错误造成应用雪崩。

单向绑定:

优点: 所有的状态变化都可以被记录，跟踪，状态变化通过手动调用通知，原题易追溯；组件只有唯一的入口和出口，程序直观便于理解和维护。

缺点: 数据单方向修改，不方便；代码量大，繁琐。

## 2019/7/3

1.VUE组件之间如何通信？

(父子，平级)

基本五种: props $emit/$on .sync $attrs和$listeners provide/inject

其它: EventBus Vuex $parent/$children $root broadcast/dispatch(vuel.0)

2.vue是怎么实现跨域的 具体用的什么方法 ?

cors: 设置origin配置

3.VUE 组件之间传值怎么传 口述

子组件: 设置props声明

父组件: 通过绑定同名属性传递父组件的data值级子组件

## 2019/7/2

1请解释 JavaScript 中 this 是如何工作的。

this默认总是指向它的直接调用者;如果没有找到，则指向window

遇到return，如果返回的是对象，则this指向返回对象，否则指向函数实例

使用call/apply/bind绑定的，thsi指向绑定对象

定时器(匿名函数)内没有默认的宿主对象，所有this指向window

箭头函数内部没有this，this指向外层最近的调用者

1.箭头函数在调用时，不会生成自身作用域下的this和arguments\

2.不像普通函数一样在调用时自动获取this，而是沿着作用域链向上查找，找到最近的外部一层作用域的this，并获取

3.在定义对象的方法/具有动态上下文的回调函数/构造函数中都不适用this会被new操作符和call/apply/bind改变指向。

笔记里:

什么是this: 自动获得正在调用当前函数的.前的对象主语。——.前是谁？this自动就变成谁！

何时: 只要对象自己的方法，要使用自己的属性时，必须加this.前缀！

比如:

console.log(`I'm ${this.sname},I'm ${this.sage}`)

1. obj.fun() fun()中的this->.前的obj

2. fun() fun()中的this->默认指window =window.fun()

3. new fun() 中的this->新对象

4. Array.prototype.sum=function(){ this-> 将来调用sum的.前的任意数组对象！}

2.请指出 JavaScript 宿主对象 (host objects) 和原生对象 (native objects) 的区别？

其实有三类 内置(Build-in)对象、宿主对象(Host)、原生对象(Native)

原生:独立于宿主环境的ECMAScript实现提供的对象，与宿主无关，在js,nodejs等中均有这些对象，在运行过程中动态创建的对象，需要new

Object Function Array String Boolean Number Date RegExp Error...

内置:ECMAScript中的宿主是网页的运行环境，即操作系统和浏览器，宿主对象基于运行环境。

比如:

浏览器中:window及其下的子对象

在nodejs下:则是global及其下的子对象。

3.描述以下变量的区别：null，undefined 或 undeclared？该如何检测它们？

null:表示没有对象，即不应该有值，转为数值时为0

1.作为函数的参数表示该参数不是对象。

2.作为原型链终点。

undefined:表示缺少值，声明未定义

1.变量声明未赋值

2.调用函数时缺参

3.访问对象未赋值的属性

4.函数没有返回值时

Undeclared:js语法错误，没有声明直接使用，js无法找到对应的上下文。

检测:

null作为js原始类型之一，使用typeof类型判断结果是"object"

undefined使用typeof判断的结果是"undefined"

Undeclared是报错

## 2019/7/1

vue是怎么实现响应式的？检测数据变化有哪些需要注意的地方？

vue-router实现原理？

vuex解决了什么问题？

答: 数据管理(全局)

vue源码包含哪些目录，里面的内容是？

## 2019/6/18

1.call() 和 .apply() bind（）的含义和区别？

相同: 都是用来替换函数中不正确的this

何时: 只要函数中的this不是想要的，都可随意替换

如何: 2种:

1. 在调用函数时，临时换一次: call 打电话，程序中: 调用函数

函数.call(替换this的对象, 实参值列表...)

比如: calc.call(lilei, 10000,1000,2000)

1. call是调用的意思，先调用calc函数

2. call将第一个参数对象替换calc中的this

3. call会将后续实参值列表注射入calc中传给形参

问题: 如果不确定要注入的实参值个数，可用数组向函数中注入实参值列表。

如果传入的实参值是放在一个数组中的，可将call()换成apply()，

2.apply()可先打散数组为单个值，再分别传入要调用的函数中。

3. bind:

什么是: 基于原函数，创建一个和原函数一模一样的新函数，并永久绑定this指定的对象：

何时: 如果函数中的this，需要反复被替换，就用bind创建一个新的专属的函数。永久绑定this。

如何: var 新函数=原函数.bind(对象, 要绑死的实参值,...)

var newFun=原fun.bind(替换this的对象)

原理: 1. bind会创建一个和原函数一幕一样的新函数副本——只创建，并未调用

2. 将新函数中的this，永久绑定为指定的对象

结果: 从此，调用新函数时，不用再每次都临时替换this

总结: 只要函数调用时，其中的this不是想要的，就可用call/apply/bind替换

1. 如果只在本次调用时，临时替换一次，就用call()

如果给定的实参值是放在数组中的，则用apply()先打散，再调用函数，并传入。

2. 如果希望永久绑定this时，就用bind()，创建一个专属的新函数，永久绑定this为指定的对象，同时可绑定部分固定的参数值。

强调: 用bind创建的新函数中的this，不可能再被别人用call替换。因为bind后的函数中的this已经被替换成固定的对象了。所以再用call/apply已无this可换

2.new的过程都发生了什么？

## 关于nth

这个东西 可以尝试多用用 可以少起很多类名

div :nth-child(n) ---->选中某个元素，该元素必须是某个父元素下的第n个子元素；

**div p:nth-child(n)---->选中p元素，且该p元素必须是某个父元素(div)下的第n个子元素**

注意：n是从1开始的

如下代码，p:nth-child(1)，只会选中第二个div中第一个子元素p;

不会选中第一个div中的第一个p，因为第一个div中第一p元素不是第一个子元素

复制代码

<style>

p:nth-child(1){

color:red

}

</style>

<div style="border:1px solid">

<span>div span中第一个段落。</span>

<p>div 中第一个段落。</p>

<p>div 中的最后一个段落。</p>

</div><br>

<div style="border:1px solid">

<p>另一个 div 中第一个段落。</p>

<p>另一个 div 中的最后一个段落。</p>

</div>

复制代码

**正方向范围**

li:nth-child(n+6)

选中从第6个开始的子元素

**负方向范围**

:nth-child(-n+9)

选中从第1个到第9个子元素。使用 :nth-child(-n+9) ，就相当让你选中第9个和其之前的所有子元素

**前后限制范围**

:nth-child(n+4):nth-child(-n+8)

选中第4-8个子元素。使用 nth-child(n+4):nth-child(-n+8) 我们可以选中某一范围内子元素，上面的例子里是从第4个到第8个子元素

**奇数、偶数位**

:nth-child(odd)

:nth-child(even)

## 隔选择子元素

:nth-child(3n+1),

选择1,4,7,10

**范围高级用法**

nth-child(n+2):nth-child(odd):nth-child(-n+9)

使用 nth-child(n+2):nth-child(odd):nth-child(-n+9) 我们将会选中的子元素是从第2位到第9位，并且只包含奇数位。

## 关于:nth-child和:nth-of-type的区别。

:nth-child可以选择父元素下的字元素，:nth-of-type也可以。但是它们到底有什么区别呢？

其实区别很简单：

:nth-of-type为什么要叫:nth-of-type？因为它是以”type”来区分的。

也就是说：ele:nth-of-type(n)是指父元素下第n个ele元素，

而ele:nth-child(n)是指父元素下第n个元素且这个元素为ele，若不是，则选择失败。

<div>

<ul class="demo">

<p>1</p>

<li>2</li>

<li>3</li>

</ul>

</div>

上面这个例子，.demo li:nth-child(2)选择的是<li>3</li>节点，

而.demo li:nth-of-type(2)则选择的是<li>2</li>节点。

## 2019/5/25

1给一个200\*200的元素顺垂直居中有几种方式

2 图片和文字在一行 ，怎么样让他们都垂直居中；

3. 有两个并列的div块为A何B，加入A块宽50px并且带浮动，

如果我想要讲B的左外简介设置调整10px，解决方案有哪些？

4 <style>

ul{width:300px;height:50px;}

li{width:50px;height:50px;background:blue;display:inline-block}

</style>

<body>

<ul>

<li></li>

<li></li>

</ul>

</body>

本代码执行后发现li标签之间有间隙，该怎么修改？

5设置margin的上下和设置宽高无效，其他有效的元素有哪些？

6：写一个数组去重的方法

7：请写一个表格以及对应的 CSS，使表格奇数行为白色背景，偶数行为灰色背景，鼠标移 上去时为黄色背景

8：一个高40px的div里面有两个div，要求左边固定宽200px显示，右边自适应宽度，学出代码

9：脱离文档流的方式有哪些

10页面上有一个div，想让他在页面上看不见有几种实现的方法

## 2019/5/24

1.Sass less是什么？为何要使用？

SCSS:

是一款强化css的辅助工具

在css的语法上，增加了变量，嵌套，混合，导入，函数等高级功能。

这些拓展令css更加强大与优雅

有助于更好的管理样式文件，以及更高效的开发项目。

LESS:

是一门CSS预处理语言，

它扩充了CSS语言，增加了诸如变量，混合，函数等功能，让CSS更易维护，方便制作主题，扩充;

less可以运行在node或浏览器端。

2.sass混合器怎么使用

把重用的样式封闭到混合器中:

关键字@mixin混合器名称{重用的样式}

使用:

关键字@include混合器名称

3.模糊属性值匹配有哪些?

[attr^=value]匹配attr的值以value开关的元素

[attr$=value]匹配attr的值以value结尾的元素

[attr\*=value]匹配attr的值含有value的元素

[attr~=value]匹配attr的值有value这个独立的元素。

4.伪类选择器有哪些？作用是什么？

已学: :link :visited :hover :active :focues

1.目标伪类(锚点)

:target{}

elem:target{}

示例: div:target{display:block;}

2.结构伪类

:first-child{} 匹配父元素的第一个子元素

:last-chile{} 匹配父元素的最后一个子元素

:nth-child(n){} 匹配父元素的第n个子元素

注意:结构伪类是两个条件，可以e:nth e:nth..多层寻找

3.匹配空元素

:empty{} 空元素:没有文本，没有空格，没有其他子元素(没有内容)

示例: :empty{}空的任意元素

div:empty{}空的div

4.唯一

:only-child{} 匹配父元素的唯一子元素

5.否定伪类

:not(selector){}匹配该选择器下，符合条件以外的元素，相当于匹配selector的反选

示例:

a:not(first-child){color:#00f}

除了第一个a标签，其他a标签都变蓝色

## 2019/05/14

//第11题

Math.round(11.5)等于多少？ 12

Math.round(-11.5)等于多少？ 11

var a=1;

a=="1"成立吗？成立 只要值相等就是true

a==="1"成立吗？ 不成立 不仅值要相等，类型也要一样，所以false

var i=1; var b=++i

b的值为？ 2

i的值为？ 1

当代码运行到一半时，抛出了异常，后面的代码还会运行吗？

catch(捕获)会继续执行

throw(抛出)不会

var a=26; var b=25; alert("a+b="+a+b)将显示什么？

答: a+b=2625

//第12题

var a=10;

sayHi();

function sayHi(){

a=a+10;

console.log(a);

return a;

}

console.log(a);

console.log(sayHi()+10); // 打印结果20 20 30 40

=======================================================

var a=10;

sayHi();

function sayHi(){

var a=a+10;

console.log(a);

return a;

}

console.log(a);

console.log(sayHi()+10); //打印结果 NaN 10 NaN NaN

## 2019/6/5

1.

var a=0,b=0

function A(a){

A=function(b){console.log(a+b++)}

console.log(a)

}

A(1)

A(12)

A(100)

2.var m=true;setTimeout(function(){m=false},3000);while(m){}alert("ok");

请问多久能弹出OK

3.

var i=3

function fn(){

i\*=2;

retur function(n){

console.log(n\*(++i))

}

}

var f=fn()

f(3)

fn()(3)

f(4)

fn()(3)

## 2019/6/13

答案在aa.js

1.数组降维 将二维数组将为一维数组

var arr=[

[0,1,2,3,4],

[5,6,7,8,9],

[11,12,13,14,15],

[16,17,18,19,"jj"]

]

2.如何判断arr是不是数组

## 2019/6/17

如何获取浏览器URL中查询字符串中的参数？

location: 是专门保存地址栏中url信息的对象，并提供了打开新链接的方法。

何时: 1. 获得url地址栏中各个部分的参数时

2. 通过操作地址栏中的地址实现跳转和刷新时

属性/方法: 1.可分段获得url的各个组成部分:

测试地址为：

http://www.runoob.com/jquery/misc-trim.html?channelid=12333&name=xiaoming&age=23

## 2019/6/18

1.call() 和 .apply() bind（）的含义和区别？

2.new的过程都发生了什么

## 2019/6/19

1 Javascript如何实现继承？

答: 利用原型链: js中的继承都是自动继承原型对象。

2 documen.write和 innerHTML 的区别？

答:document.write DOM方法，会重写所有网页内容

innerHTML DOM属性，可以片段的修改网页内容

3 哪些操作会造成内存泄漏？

答: 内存泄漏:程序不再需要占用内存时，却无法释放。

1.意外的全局变量

2.定时器或者回调函数

3.脱离DOM的引用

4.闭包

5.子对象的重复方法定义在构造函数中而不是原型对象时。

4 件委托是什么

多个平级子元素需要绑定相同事件时，只给共同的父元素绑定一次事件即可

子元素会冒泡到父元素执行处理函数

5 关于window.location.search() 返回的是什么？

答: 查询字符串?xxx=xxx&xxx=xxx..

6 关于window.location.hash 返回的是什么？

答: #锚点

7 window.location.reload() 作用？

答: 刷新页面

## 2019/6/21

实现Array.prototype.map函数

实现Function.prototype.bind函数

## 2019/6/25

说出至少4种vue当中的指令和它的用法？

1.v-bind:绑定属性的值—>简写:属性名="表达式"

<img v-bind:src="表达式"> -->简写为 <img :src="pm25<100?img/1.png....">

2.v-show: 根据表达式决定当前一个元素显示隐藏-->实质是display:none

<button v-show="i<count" @click="next">下一页</button>

2.2 v-if v-else-if v-else :多选一 删除添加元素的方式

<img src="img/1.png" v-if="pm25<100">

<img src="img/2.png" v-else-if="pm25<200">

<img src="img/3.png" v-else-if="pm25<300">

<img src="img/4.png" v-else>

3.v-for:根据一个数组中的元素，动态反复生成多段相同的页面内容

<li v-for="(elem,i) of tasks" :key="i" class="c1">{{i}}--{{elem}}</li>

4.v-on:绑定事件 -->简写　@且可以传参，和使用事件e

<button @click="change(-1)"> - </button>

<button v-on:click="change(+1)"> + </button>

5.v-cloak和v-text :暂时隐藏{{}}

在页面的<style>中，定义属性选择器:

[v-cloak]{display:none;} vue只定义了v-cloak但是没给设置display

new Vue加载完成后，会自动查找v-cloak属性， 移除该属性。

v-text:

<h1 v-text="`${uname}`"></h1> 加载完new Vue之前看不见v-text

6.v-html:绑定html片段

<h1 v-html="details"></h1>

//===================

data:{details:'<h1 style="color:lightBlue">HHHH</h1>'}

纯文本内容，可使用{{}}或v-text

7.v-once:仅在首次Vue加载时，绑定一次，不会再变化

<h1 v-once>{{new Date(data).toLocaleString()}}</h1>

8.v-pre:保持内容中的{{}}不被Vue编译 -->和转义类似，会显示原样

<h1 v-pre>vue的绑定语法是:{{变量}}</h1>

js代码 一到判断的时候 你是用三目运算的多 还是if else的多 为什么

数组和对象有哪些原生方法，列举一下？

## 2019/6/27

1.$route和$router的区别

$router属性负责组件之间跳转

$route 属性负责接收参数

2.v-on 可以绑定多个方法吗？

v-on即@可以使用v-on绑定多个方法，顺序执行

<div v-on:click="fn1" v-on:click="fn2">

或者用 v-on="{ click:fn1,keyup:fn2... }"

3.v-if 和 v-show 区别

v-show采用display:none/block方式控制显示隐藏，不修改DOM树上的节点——效率高

v-if是多选一，通过删除添加元素的方式控制显示隐藏--效率低

4.怎么让css只在当前组件起作用

用前缀区分:在每个vue组件中都可以定义各自的css,js,如果想写的css只对当前组件起作用，

则:<style scoped></style>

则会给生个样式添加一个唯一标识[data-v-06890860]{...}

## 4.10

写出增删改查sql语句

Select \* from table

Insert into table values ()

Dalete from table where id=？

Update table set name=“xx”，sex=“0”where id=？

## 4.11

有几种原始类型分别是什么，用什么判断

String Number boolean undefined null typeof

## 4.12

打印倒三角

for (var i=10;i>=1;i--){

for (var j=i,str='';j>=1;j--){

str+='\*'

}

console.log(str);

}

## 4.15

写一个方法，求0到n之间的偶数和（可以写m到n之间的偶数和）

function fun(n,m){

var sum=0

for(var i=n;i<m;i++){

if(i%2==0){

sum+=i

}

}

console.log(sum)

}

## 04.22

Var a=”0123456789” 输出0246813579 0918273645

function fun(str){

var sum1=""

var sum2=""

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

if(a[i]%2==0){

sum1+=a[i]

}else{

sum2+=a[i]

}

}

console.log(sum1+sum2)

}

function fun(str){

var sum1=""

var sum2=""

for(var i=0,j=str.length-1;i<str.length/2;i++,j--){

sum1+=a[i]+a[j]

}

console.log(sum1)

}

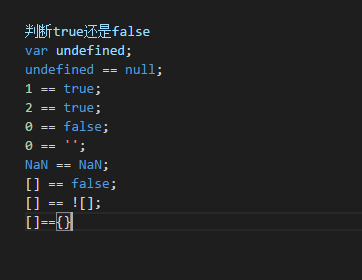
04.23

0423文件

04.24

Js基础小测

## 04.25

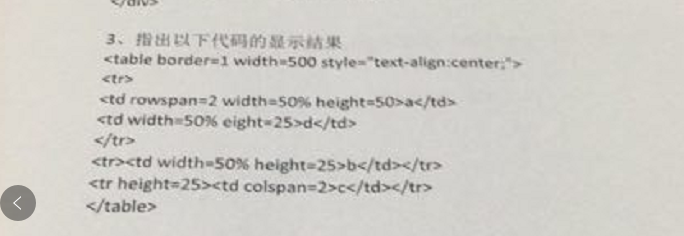


true true false true true false true true false

04.29:

html的空标签有哪些

04.30:



05.05:

1． 如何理解html的语义化

根据内容的结构化（内容语义化），选择合适的标签（代码语义化）便于开发者阅读和写出更优雅的代码的同时让浏览器的爬虫和机器很好地解析。

2． 锚点的作用是什么？如何创建锚点

锚点是文档中某行的一个记号，类似于书签，用于链接到文档中的某个位置。当定义了锚点后，我们可以创建直接跳至该锚点（比如页面中某个小节）的链接，这样使用者就无需不停地滚动页面来寻找他们需要的信息了

<a name=”d1”>锚点一</a>

<a href=”#d1”>回到锚点一</a>

3． 行内元素和块及元素有哪些

05.08:

同步和异步的区别？

请求发出后，是否需要等待结果，才能继续执行其他操作。

同步需要等待，异步步需要等待

简述ajax的过程

实例化XMLHttpRequest对象

var xhr = new XMLHttpRequest();

// 监听状态

xhr.onreadystatechange = () => {

// 数据请求完毕             判断状态码

if (xhr.readyState===4&&xhr.status===200) {

// 将数据转化成json在、数组

JSON.parse(xhr.responseText)

}

}

// 打开数据请求   请求方式，路径，同步false 异步true(默认)

xhr.open('get/post', "路径", "true/false");

// 发送数据

xhr.send(null)

05.09:

ajax的优缺点有哪些？

1、最大的一点是页面无刷新，用户的体验非常好。

2、使用异步方式与服务器通信，具有更加迅速的响应能力。

3、可以把以前一些服务器负担的工作转嫁到客户端，利用客户端闲置的能力来处理，减轻服务器和带宽的负担，节约空间和宽带租用成本。并且减轻服务器的负担，ajax的原则是“按需取数据”，可以最大程度的减少冗余请求，和响应对服务器造成的负担。

4、基于标准化的并被广泛支持的技术，不需要下载插件或者小程序。

ajax的缺点

1、ajax不支持浏览器back按钮。

2、安全问题 AJAX暴露了与服务器交互的细节。

3、对搜索引擎的支持比较弱。

4、破坏了程序的异常机制。

5、不容易调试。

05.10:

添加css样式有几种方式，分别是什么

三种 内部 外部 内联

你所知道的选择器有哪些

1.标签选择器2.类选择器3.ID选择器4.通用选择器5.群组选择器6.后代选择器7.伪类选择器8.复杂选择器9.元素选择器10.子代选择器

5.13:

css优先级算法如何计算？

为什么要初始化样式？

5.14:

Math.round(11.5)等于多少？Math.round(-11.5)等于多少

11.5 -11.5

var a=1；a==”1“成立？a==="1"成立？

不成立

var b=0;var c=2;b?c=3:c=5;c的值为？

5

var i=1;var b=++i b和i的值为

2,2

当代码运行到一半时，抛出了异常，后面的代码还会运行吗？

不会

var a=26;var b=25; alert("a+b="+a+b)将显示什么

a+b=2625

var a=10;

sayHi();

function sayHi(){

a=a+10;

console.log(a)

return a

}

console.log(a)

console.log(sayHi()+10)

// 20 20 30 40

var a=10;

sayHi();

function sayHi(){

var a=a+10;

console.log(a)

return a

}

console.log(a)

console.log(sayHi()+10)

// NAN 10 NAN NAN

5.15:

什么是盒子模型？

盒子模型=内边距+外边距+边框+内容区域

清除浮动的方式有哪些?

1.父元素也浮动，弊端，影响父元素后面的非浮动元素

2.直接给父元素写高度 弊端，不是每次都能知道具体高是多少

3.overflow：hidden 弊端，会让真正要溢出不能显示

4.在父元素中追加一个块级元素，这个块级元素，没有内容，不写高。只写clear:both。就可以让父元素在文档流中找到内容的高度

.after:after,.a:before{content: "";display: table;}

.a:after{clear: both;}

.b{\*zoom: 1;}

<div class="a b">

<div></div>

<div></div>

</div>

5.17: 如何利用一个标签实现类似“——————————左右两条线——————————”这种效果

<div class="s1">左右两条线</div>