

2022前端面试(一面面试题)

2022前端面试(一面面试题)

CSS

calc, support, media各自的含义及用法?

@support主要是用于检测浏览器是否支持CSS的某个属性，其实就是条件判断，如果支持某个属性，你可以写一套样式，如果不支持某个属性，你也可以提供另外一套样式作为替补。

calc() 函数用于动态计算长度值。 calc()函数支持 "+", "-", "*", "/" 运算;

@media 查询，你可以针对不同的媒体类型定义不同的样式。

css水平、垂直居中的写法，请至少写出4种?

这题考查的是css的基础知识是否全面，所以平时一定要注意多积累

水平居中

- 行内元素: `text-align: center`
- 块级元素: `margin: 0 auto`
- `position:absolute + left:50% + transform:translateX(-50%)`
- `display:flex + justify-content: center`

垂直居中

- 设置line-height 等于height
- `position: absolute + top:50% + transform:translateY(-50%)`
- `display:flex + align-items: center`
- `display:table+display:table-cell + vertical-align: middle;`

1rem、1em、1vh、1px各自代表的含义？

rem

rem是全部的长度都相对于根元素元素。通常做法是给html元素设置一个字体大小，然后其他元素的长度单位就为rem。

em

- 子元素字体大小的em是相对于父元素字体大小
- 元素的width/height/padding/margin用em的话是相对于该元素的font-size

vw/vh

全称是 Viewport Width 和 Viewport Height，视窗的宽度和高度，相当于 屏幕宽度和高度的 1%，不过，处理宽度的时候%单位更合适，处理高度的话 vh 单位更好。

px

px像素（Pixel）。相对长度单位。像素px是相对于显示器屏幕分辨率而言的。

一般电脑的分辨率有{1920*1024}等不同的分辨率

1920*1024 前者是屏幕宽度总共有1920个像素,后者则是高度为1024个像素

画一条0.5px的直线？

考查的是css3的transform

css 复制代码

```
height: 1px;  
transform: scale(0.5);
```

说一下盒模型？

盒模型是css中重要的基础知识，也是必考的基础知识

盒模型的组成，由里向外content,padding,border,margin.

在IE盒子模型中，width表示content+padding+border这三个部分的宽度

在标准的盒子模型中，width指content部分的宽度

box-sizing的使用

`box-sizing: content-box` 是W3C盒子模型

`box-sizing: border-box` 是IE盒子模型

rust 复制代码

box-sizing的默认属性是content-box

画一个三角形？

这属于简单的css考查，平时在用组件库的同时，也别忘了原生的css

css 复制代码

```
.a{
    width: 0;
    height: 0;
    border-width: 100px;
    border-style: solid;
    border-color: transparent #0099CC transparent transparent;
    transform: rotate(90deg); /*顺时针旋转90°*/
}

<div class="a"></div>
```

清除浮动的几种方式，及原理？

清除浮动简单，但这题要引出的是BFC，BFC也是必考的基础知识点

- `::after` / `
` / `clear: both`
- 创建父级 BFC (`overflow:hidden`)

- 父级设置高度

BFC（块级格式化上下文），是一个独立的渲染区域，让处于 *BFC* 内部的元素与外部的元素相互隔离，使内外元素的定位不会相互影响。

触发条件:

- 根元素
- `position: absolute/fixed`
- `display: inline-block / table`
- `float` 元素
- `overflow != visible`

规则:

- 属于同一个 *BFC* 的两个相邻 *Box* 垂直排列
- 属于同一个 *BFC* 的两个相邻 *Box* 的 `margin` 会发生重叠
- *BFC* 的区域不会与 `float` 的元素区域重叠
- 计算 *BFC* 的高度时，浮动子元素也参与计算
- 文字层不会被浮动层覆盖，环绕于周围

html

说一下标签的用法

label标签主要是方便鼠标点击使用，扩大可点击的范围，增强用户操作体验

遍历A节点的父节点下的所有子节点

这题考查原生的js操作dom,属于非常简单的基础题，但长时间使用mvvm框架，可能会忘记

xm1 复制代码

```
<script>
  var b=document.getElementById("a").parentNode.children;
```

```
console.log(b)
</script>
```

js

用js递归的方式写1到100求和?

递归我们经常用到，vue在实现双向绑定进行数据检验的时候用的也是递归，但我们要面试的时候手写一个递归，如果对递归的概念理解不透彻，可能还是会有一些问题。

csharp 复制代码

```
function add(num1,num2){
    var num = num1+num2;
    if(num2+1>100){
        return num;
    }else{
        return add(num,num2+1)
    }
}
var sum =add(1,2);
```

页面渲染html的过程?

不需要死记硬背，理解整个过程即可

浏览器渲染页面的一般过程：

1.浏览器解析html源码，然后创建一个 DOM树。并行请求 css/image/js在DOM树中，每一个HTML标签都有一个对应的节点，并且每一个文本也都会有一个对应的文本节点。DOM树的根节点就是 documentElement，对应的是html标签。

2.浏览器解析CSS代码，计算出最终的样式数据。构建CSSOM树。对CSS代码中非法的语法它会直接忽略掉。解析CSS的时候会按照如下顺序来定义优先级：浏览器默认设置 < 用户设置 < 外链样式 < 内联样式 < html中的style。

3.DOM Tree + CSSOM --> 渲染树（rendering tree）。渲染树和DOM树有点像，但是是有区别的。

DOM树完全和html标签一一对应，但是渲染树会忽略掉不需要渲染的元素，比如head、display:none的元素等。而且一大段文本中的每一个行在渲染树中都是独立的一个节点。渲染树中的每一个节点都存储有对应的css属性。

4.一旦渲染树创建好了，浏览器就可以根据渲染树直接把页面绘制到屏幕上。

以上四个步骤并不是一次性顺序完成的。如果DOM或者CSSOM被修改，以上过程会被重复执行。实际上，CSS和JavaScript往往会多次修改DOM或者CSSOM。

说一下CORS?

CORS是一种新标准，支持同源通信，也支持跨域通信。fetch是实现CORS通信的

如何中断ajax请求?

一种是设置超时时间让ajax自动断开，另一种是手动停止ajax请求，其核心是调用XML对象的abort方法，ajax.abort()

说一下事件代理?

事件委托是指将事件绑定到目标元素的父元素上，利用冒泡机制触发该事件

javascript 复制代码

```
ulEl.addEventListener('click', function(e){
  var target = event.target || event.srcElement;
  if (!!target && target.nodeName.toUpperCase() === "LI"){
    console.log(target.innerHTML);
  }
}, false);
```

target、currentTarget的区别?

currentTarget当前所绑定事件的元素

target当前被点击的元素

说一下宏任务和微任务?

1. 宏任务：当前调用栈中执行的任务称为宏任务。（主代码块，定时器等等）。
2. 微任务：当前（此次事件循环中）宏任务执行完，在下一个宏任务开始之前需要执行的任务为微任务。（可以理解为回调事件，`promise.then`，`process.nextTick`等等）。
3. 宏任务中的事件放在`callback queue`中，由事件触发线程维护；微任务的事件放在微任务队列中，由js引擎线程维护。

说一下继承的几种方式及优缺点？

说比较经典的几种继承方式并比较优缺点就可以了

1. 借用构造函数继承，使用`call`或`apply`方法，将父对象的构造函数绑定在子对象上
2. 原型继承，将子对象的`prototype`指向父对象的一个实例
3. 组合继承

原型链继承的缺点

- 字面量重写原型会中断关系，使用引用类型的原型，并且子类型还无法给超类型传递参数。

借用构造函数（类式继承）

- 借用构造函数虽然解决了刚才两种问题，但没有原型，则复用无从谈起。

组合式继承

- 组合式继承是比较常用的一种继承方法，其背后的思路是使用原型链实现对原型属性和方法的继承，而通过借用构造函数来实现对实例属性的继承。这样，既通过在原型上定义方法实现了函数复用，又保证每个实例都有它自己的属性。

说一下闭包？

闭包的实质是因为函数嵌套而形成的作用域链

闭包的定义即：函数 `A` 内部有一个函数 `B`，函数 `B` 可以访问到函数 `A` 中的变量，那么函数 `B` 就是闭包

export和export default的区别?

使用上的不同

javascript 复制代码

```
export default xxx
import xxx from './'

export xxx
import {xxx} from './'
```

说一下自己常用的es6的功能?

此题是一道开放题，可以自由回答。但要注意像let这种简单的用法就别说了，说一些经常用到并有一定高度的新功能

像module、class、promise等，尽量讲的详细一点。

什么是会话cookie,什么是持久cookie?

cookie是服务器返回的，指定了expire time（有效期）的是持久cookie,没有指定的是会话cookie

数组去重?

此题看着简单，但要想面试官给你高分还是有难度的。至少也要写出几种方法

js

ini 复制代码

```
var arr=['12','32','89','12','12','78','12','32'];
// 最简单数组去重法
function unique1(array){
  var n = []; //一个新的临时数组
  for(var i = 0; i < array.length; i++){ //遍历当前数组
    if (n.indexOf(array[i]) == -1)
      n.push(array[i]);
  }
}
```



```

        return n;
    }
    arr=unique1(arr);
    // 速度最快， 占空间最多（空间换时间）
    function unique2(array){
        var n = {}, r = [], type;
        for (var i = 0; i < array.Length; i++) {
            type = typeof array[i];
            if (!n[array[i]]) {
                n[array[i]] = [type];
                r.push(array[i]);
            } else if (n[array[i]].indexOf(type) < 0) {
                n[array[i]].push(type);
                r.push(array[i]);
            }
        }
        return r;
    }
    //数组下标判断法
    function unique3(array){
        var n = [array[0]]; //结果数组
        for(var i = 1; i < array.Length; i++) { //从第二项开始遍历
            if (array.indexOf(array[i]) == i)
                n.push(array[i]);
        }
        return n;
    }
}

```

es6

javascript 复制代码

es6方法数组去重

```
arr=[...new Set(arr)];
```

es6方法数组去重，第二种方法

```

function dedupe(array) {
    return Array.from(new Set(array)); //Array.from()能把set结构转换为数组
}

```

get、post的区别

此题比较简单，但一定要回答的全面

1.get传参方式是通过地址栏URL传递，是可以直接看到get传递的参数，post传参方式参数URL不可见，get把请求的数据在URL后通过？连接，通过&进行参数分割。psot将参数存放在

HTTP的包体内

2.get传递数据是通过URL进行传递，对传递的数据长度是受到URL大小的限制，URL最大长度是2048个字符。post没有长度限制

3.get后退不会有影响，post后退会重新进行提交

4.get请求可以被缓存，post不可以被缓存

5.get请求只URL编码，post支持多种编码方式

6.get请求的记录会留在历史记录中，post请求不会留在历史记录

7.get只支持ASCII字符，post没有字符类型限制

你所知道的http的响应码及含义？

此题有过开发经验的都知道几个，但还是那句话，一定要回答的详细且全面。

1xx(临时响应)

100: 请求者应当继续提出请求。

101(切换协议) 请求者已要求服务器切换协议，服务器已确认并准备进行切换。

2xx(成功)

200: 正确的请求返回正确的结果

201: 表示资源被正确的创建。比如说，我们 POST 用户名、密码正确创建了一个用户就可以返回 201。

202: 请求是正确的，但是结果正在处理中，这时候客户端可以通过轮询等机制继续请求。

3xx(已重定向)

300: 请求成功，但结果有多种选择。

301: 请求成功，但是资源被永久转移。

303：使用 GET 来访问新的地址来获取资源。

304：请求的资源并没有被修改过

4xx(请求错误)

400：请求出现错误，比如请求头不对等。

401：没有提供认证信息。请求的时候没有带上 Token 等。

402：为以后需要所保留的状态码。

403：请求的资源不允许访问。就是说没有权限。

404：请求的内容不存在。

5xx(服务器错误)

500：服务器错误。

501：请求还没有被实现。