1. JS数据类型有哪些？（基本类型和运用类型）

（1）基本类型：**Nu**mber、String、Boolean、Undefined、Null；

1. 运用类型：Object。
2. 如何转化类型？

类型转换有两种：一种是显**式转换**，****即需要程序员手动写代码转换；另一种是隐匿转换，由JavaScript解释器转换。

1. 数组从小到大排序？

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>从小到大排序</title>

<script type="text/javascript">

var Arr=[13,7,10,76,97,100,35];

var min;

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

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

if(Arr[i]>Arr[j]){

min=Arr[j];

Arr[j]=Arr[i];

Arr[i]=min;

}

}

}

document.write(Arr.toString()+"<br />");

</script>

</head>

<body>

</body>

</html>

1. 数组从大到小排序？

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>从大到小排序</title>

<script type="text/javascript">

　　　　　　　//创建数组元素arr

var arr=[7,13,34,3,200,100,4];

　　　　　　　//创建每次循环存储最大值得变量

var max;

　　　　　　　//遍历数组，默认arr中的某一个元素为最大值，进行逐一比较

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

　　　　　　　　　　//外层循环一次，就拿arr[i] 和 内层循环arr.legend次的 arr[j] 做对比

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

if(arr[i]<arr[j]){

//如果arr[j]大就把此时的值赋值给最大值变量max

　　　　　　　　　　　　　　 max=arr[j];

arr[j]=arr[i];

arr[i]=max;

}

}

}

document.write(arr.toString()+"<br>");

　　　　　　　//200,100,34,13,7,4,3

</script>

</head>

<body>

</body>

</html>

1. 显示转换与隐式转换

显式是强制转换，而隐式是编译器自动转换的，一般同一数据类型（如byte,int,float,double），由低级转为高级为隐式（如int转为float或double），而高级转为低级为强制类型转换，即显式转换，须用户在代码中写的

1. Mouseover/mouseout与mouseenter/mouseleave的区别与联系
2. **mouseover/mouseout事件支持冒泡，绑定的事件会冒泡到子标签上。  
   2、mouseenter/mouseleave事件则只针对所绑定的标签，不会冒泡到子标签上。**
3. forEach和map和filter的区别

forEach()方法用于调用数组的每个元素，并将元素传递给回调函数。注意: forEach()对于空数组是不会执行回调函数的。

map()方法返回一个新数组，数组中的元素为原始数组元素调用函数处理后的值。map()方法按照原始数组元素顺序依次处理元素。

注意: map()不会对空数组进行检测。map()不会改变原始数组。

filter()方法创建一个新的数组，新数组中的元素是通过检查指定数组中符合条件的所有元素。注意: filter()不会对空数组进行检测。filter()不会改变原始数组。

1. offsetWidth/offsetHeight、clientWidth/clientHeight与scrollWidth/scrollHeight的区别

offsetWidth/offsetHeight返回值包含content + padding + border，如果有滚动条，也不包含滚动条clientWidth/clientHeight返回值只包含content + padding，如果有滚动条，也不包含滚动条

scroll Width/scrollHeight返回值包含content + padding＋溢出内容的尺寸

1. ==与===的区别

在C语言中，最容易产生混淆的操作符要属“=”与“==”。其中，“=”并不等于符号，而是赋值操作符，如x=3。除此之外，还可以在一个语句中向多个变量赋同一个值，即多重赋值。

1. Null和underfined的区别

null表示没有对象，即该处不应该有值

1)作为函数的参数，表示该函数的参数不是对象

2)作为对象原型链的终点

undefined表示缺少值，即此处应该有值，但没有定义

1）定义了形参，没有传实参，显示undefined

2)对象属性名不存在时，显示undefined

3)函数没有写返回值，即没有写return，拿到的是undefined

4)写了return，但没有赋值，拿到的是undefined

1. 字符串操作方法

字符串的基本操作方法中包含以下几种： 格式：int length = str.length (); 2、获取字符串中的第i个字符方法 ****charAt (i) 格式：char ch = str.charAt (i); //i为字符串的索引号，可得到字符串任意位置处的字符，保存到字符变量中 格式：char array [] = new char [80]****; //先要创建以一个容量足够大的char型数组，数组名为array str.getChars (indexBegin,indexEnd,array,arrayBegin); 2、 indexEnd: 需要复制的字符串的结束索引，indexEnd-1 3、array: 前面定义的char型数组的数组名

1. 数组操作方法

1、数组的创建

var arrayObj = new Array();//创建一个数组

var arrayObj = new Array([size]);//创建一个数组并指定长度，注意不是上限，是长度var arrayObj = new Array([element0[, element1[,..[, elementN]]]]);创建一个数组并赋值

要说明的是，虽然第二种方法创建数组指定了长度，但实际上所有情况下数组都是变长的，也就是说即使指定了长度为5，仍然可以将元素存储在规定长度以外的，注意:这时长度会随之改变。

2、数组的元素的访问

var testGetArrValue=arrayObj[1];//获取数组的元素值arrayObj[1]=“这是新值";//给数组元素赋予新的值

3、数组元素的添加

arrayObj. push([item1 [item2 [... [itemN ]l);//将一个或多个新元素添加到数组结尾，并返回数组新长度

arrayObj.unshift([item1 [item2 [... itemN ]I);//将一个或多个新元素添加到数组开始，数组中的元素自动后移，返回数组新长度

arrayObj.splice(insertPos,0,[item1[, item2[ . . . LitemN]]);//将一个或多个新元素插入到数组的指定位置，插入位置的元素自动后移，返回""。

4、数组元素的删除

arrayObj.pop();//移除最后一个元素并返回该元素值

arrayObj.shift();//移除最前一个元素并返回该元素值，数组中元素自动前移

arrayObj,splice(deletePos,deleteCount);//删除从指定位置deletePos开始的指定数量deleteCount的元素，数组形式返回所移除的元素

5、数组的截取和合并

arrayObj.slice(start, [end]);//以数组的形式返回数组的一部分，注意不包括end对应的元素，如果省略end将复制start之后的所有元素

arrayObj.concat([item1[, i tem2[ . .. L itemN]l);//将多个数组（也可以是字符串，或者是数组和字符串的混合）连接为一个数组，返回连接好的新的数组

6、数组的拷贝

arrayObj.slice(O);//返回数组的拷贝数组，注意是一个新的数组，不是指向arrayObj.concat();//返回数组的拷贝数组，注意是一个新的数组，不是指向

7、数组元素的排序

arrayObj.reverse();//反转元素(最前的排到最后、最后的排到最前)，返回数组地址arrayObj.sort();//对数组元素排序，返回数组地址

8、数组元素的字符串化

arrayObjoin(separator);//返回字符串，这个字符串将数组的每一个元素值连接在一起，中间用separator隔开。

toLocaleString . toString . valueOf:可以看作是join的特殊用法，不常用

1. JS中的定时器有哪些？他们的区别及用法是什

. 1.setInterval，按照指定间隔后把回调函数加入任务队列，js会逐个执行,并且会一直执行。

例如:var timer = setInterval( ()=> {alert("hi"'); },1000)

. 2.setTimeout，在指定的毫秒数后调用函数或计算表达式。

例如:var timer = setTimeout( ()=> {alert( "hi");},1000 )

. 3.requestAnimationFrame,与setTimeout相比requestAnimationFrame最大的优势是由系统来决定回调函数的执行时机，它能保证回调函数在屏幕每一次的刷新间隔中只被执行一次，这样就不会引起丢帧现象，常用于动画场景。

1. Document.write和innerHTML的区别

document.write是直接写入到页面的内容流，如果在写之前没有调用document.open,浏览器会自动调用open。每次写完关闭之后重新调用该函数，会导致页面被重写。

innerHTML则是DOM页面元素的一个属性，代表该元素的html内容。你可以精确到某一个具体的元素来进行更改。如果想修改document的内容，则需要修改

document.documentElement.innerElement,且innerHTML将内容写入某个DOM节点，不会导致页面全部重绘。

document.write是重写整个document,写入内容是字符串的htmlinnerHTML是HTMLElement的属性，是一个元素的内部html内容

所以两者都可向页面输出内容,innerHTML比document.write更灵活，而innerHTML很多情况下都优于document.write，其原因在于其允许更精确的控制要刷新页面的那一个部分

1. 解释一下DOM和BOM

js中bom和dom的区别之一:

1、BOM是Browser Object Model的缩写，即浏览器对象模型。

BOM和浏览器关系密切。浏览器的很多东西可以通过JavaScript控制的，例如打开新窗口、打开新选项卡(标签页)、关闭页面，把网页设为主页，或加入收藏夹，等等...这些涉及到的对象就是BOM

2、DOM是Document Object Model的缩写，即文档对象模型。

DOM和文档有关，这里的文档指的是网页，也就是HTML文档。网页是由服务器发送给客户端浏览器的，无论用什么浏览器，接收到的HTML都是一样的，所以DOM和浏览器无关，它关注的是网页本身的内容。由于和浏览器关系不大，所以标准就好定了。

js中bom和dom的区别之二:

1、BOM没有相关标准。

由于没有标准，不同的浏览器实现同一功能，可以需要不同的实现方式。对于上面说的功能，不同的浏览器的实现功能所需要的JavaScript代码可能不相同。

1. DOM是W3C的标准。

既然有标准了，大家就要按标准来了。

js中bom和dom的区别之三:

1、BOM的最根本对象是window。

2、DOM最根本对象是document(实际上是window.document)。

1. jQuery一个对象可以同时绑定多个事件，这是如何实现的

使用on绑定多个事件对应同一个事件处理

函数语法结构:jq对象.on(“事件类型1事件类型2",”"事件处理函数")

1. 针对jQuery的优化方法

1.使用选择器

尽量选用id选择器和元素选择器:因为遇到这些选择器的时候,jquery会自动调用浏览器的原生方法比如

根据图中可以看出ID选择器执行速度最快,其次是元素选择器,然后是class.所以优化jquery尽量使用

2.尽量不要使用".live()"方法

这是jquery1.3.1版本之后新增加的方法,这个方法的功能就是为新增DOM元素绑定事件.但对于效率来

3.使用data()

尽量使用data()方法储存变量,因为这是jquery的内部函数data()来存储状态．

1. jQuery常用选择器

1、基本选择器:

#id . element . .class . \* 、selector1,selector2,selectorN

2、层次选择器:

ancestor descendant . parent > child . prev + next、 prev ~ siblings3、基本过滤器选择器

:first . :last . :not . :even . :odd . :eq . :gt . :lt . :header .

:animated

4、内容过滤器选择器

:contains . :empty .:has . :parent

5、可见性过滤器选择器

:hidden . :visible6、属性过滤器选择器

[attribute] . [attribute=value] . [attribute!=value] 、[attribute^=value] 、[attribute$=value]、[attribute\*=value] . [attrSel1][attrSel2][attrSelN]

7、子元素过滤器选择器

:nth-child . :first-child . :last-child . :only-child

8、表单选择器

:input 、 :text 、 :password . :radio . :checkbox 、:submit 、:image . :reset. :button、:file .:hidden

9、表单过滤器选择器

:enabled . :disabled . :checked . :selected