

1: javascript几部分组成?

1. 核心 (ECMAScript)
2. 文档对象模型 (DOM)
3. 浏览器对象模型 (BOM)

2: js数据类型有哪些? 判断数据类型方法?

基本数据类型

```
1:string  
2:number  
3:boolean  
4:null  
5:undefined
```

引用类型

```
1:object  
2:function  
3:Array  
4>Date  
5:RegExp
```

判断数据类型方法

1:typeof

```
//判断基本数据类型  
console.log(typeof 1);
```

2:instanceof

```
//引用类型通过instanceof基于原型链来判断变量是否某个对象的实例
var a=[];
console.log(a instanceof Array)
```

3:constructor

constructor 返回对象相对应的构造函数 可以用于来判断数据类型和引用类型

```
console.log([].constructor == Array);
console.log({}.constructor == Object);
```

4:Object.prototype.toString.apply([])

```
Object.prototype.toString.apply([]) === '[object Array]'
```

3: Undefined和Null区别

- 当声明的变量还未被初始化时，变量的默认值为undefined。
- null用来表示尚未存在的对象，常用来表示函数企图返回一个不存在的对象。

4:事件冒泡和事件捕获

事件冒泡：从当前触发的元素一级一级向上触发 直到document结束
事件捕获：从根节点到目标节点 并且触发各个节点事件

5: == 和 ===区别

== 转换类型比较，=== 先比较类型 在比较值

6: js本地对象有哪些？ 内置对象有哪些？ 宿主对象有哪些？

本地对象

```
1:string
2:number
3:boolean
4:null
5:undefined
6:object
7:function
8:Array
9>Date
10:RegExp
11>Error
```

内置对象 也属于本地对象

内置对象也就是我们不必显示地区实例化，直接就可以使用了，因为它们已经实例化了。

1:Global对象

Global对象是一个全局对象。所有在全局作用域中定义的属性和函数，都是Global对象的属性，比如：isNaN()、isFinite()、parseInt()以及parseFloat()，实际上都是Global对象的方法，它还包括了其他一些方法：

```
1:encodeURIComponent
2:encodeURIComponent
3:encodeURIComponent
4:eval
```

2:Math对象

Math对象中保存了数学公式和信息。

Math.ceil() 执行向上舍入，即它总是取最接近数值且大于数值的整数

Math.floor() 执行向下舍入，即它总是取最近数值且小于数值的整数。
Math.round() 执行标准舍入，即它总是将数值四舍五入为最接近的整数。
Math.random()
方法返回介于0和1之间的一个随机数（不包括0和1）。

宿主对象

1:BOM

2:DOM

```
setTimeout()  
setInterval()  
window.location  
document.location  
window.history      //返回历史页面  
window.navigator    //获取浏览器信息  
window.screen       //获取分辨率
```

7：字符串操作方法有哪些？数组操作方法有哪些？

字符串操作方法

```
s.charAt(index)    //返回指定位置的字符  
s.concat(s2)      //用于连接两个字符串  
s.substring(start,end) //用于从原字符串取出子字符串并返回，不改变原字符串。第一个参数表示子字符串的开始位置，第二个位置表示结束位置。  
s.substr(start,length) //用于从原字符串取出子字符串并返回，不改变原字符串。第一个参数是子字符串的开始位置，第二个参数是子字符串的长度。如果第一个参数是负数，表示倒数计算的字符位置。如果第二个参数是负数，将被自动转为0，因此会返回空字符串。  
s.indexOf(s)       //返回给定元素在字符串中第一次出现的位置，如果没有出现则返回-1。可以接受第二个参数，表示搜索的开始位置  
s.trim()           //用于去除字符串两端的空格，返回一个新字符串  
s.toLowerCase()    //用于将一个字符串全部转为小写，返回一个新字符串，不改变原字符串。
```

```
s.toUpperCase() //全部转为大写
s.search() //返回值为匹配的第一个位置。如果没有找到匹配，则返回-1。
s.replace(oldValue,newValue) //用于替换匹配的子字符串，一般情况下只替换第一个匹配（除非使用带有g修饰符的正则表达式）。
s.split() //按照给定规则分割字符串，返回一个由分割出来的子字符串组成的数组。还可传入第二个参数，决定了返回数组的成员数。
```

数组操作方法

```
a.valueOf() //返回数组本身
a.toString() //返回数组的字符串形式
a.push(value,value....) //用于在数组的末端添加一个或多个元素，并返回添加新元素后的数组长度。
pop() //用于删除数组的最后一个元素，并返回该元素
join() //以参数作为分隔符，将所有数组成员组成一个字符串返回。如果不提供参数，默认用逗号分隔。
concat() //用于多个数组的合并。它将新数组的成员，添加到原数组的尾部，然后返回一个新数组，原数组不变。
shift() //用于删除数组的第一个元素，并返回该元素。
unshift(value) //用于在数组的第一个位置添加元素，并返回添加新元素后的数组长度。
reverse() //用于颠倒数组中元素的顺序，返回改变后的数组
slice(start_index, upto_index); //用于提取原数组的一部分，返回一个新数组，原数组不变。第一个参数为起始位置（从0开始），第二个参数为终止位置（但该位置的元素本身不包括在内）。如果省略第二个参数，则一直返回到原数组的最后一个成员。负数表示倒数第几个。
sort() //对数组成员进行排序，默认是按照字典顺序排序。排序后，原数组将被改变。如果想让sort方法按照自定义方式排序，可以传入一个函数作为参数，表示按照自定义方法进行排序。该函数本身又接受两个参数，表示进行比较的两个元素。如果返回值大于0，表示第一个元素排在第二个元素后面；其他情况下，都是第一个元素排在第二个元素前面。
map() //对数组的所有成员依次调用一个函数，根据函数结果返回一个新数组。
forEach() //遍历数组的所有成员，执行某种操作，参数是一个函数。它接受三个参数，分别是当前位置的值、当前位置的编号和整个数组。
reduce() //依次处理数组的每个成员，最终累计为一个值。从左到右处理（从第一个成员到最后一个成员）
indexOf(s) //返回给定元素在数组中第一次出现的位置，如果没有出现则
```

8.DOM 0级和2级事件的写法、及优缺点？

DOM0同一个标签的同一个事件只能绑定一个函数，如果绑定多个，则后绑定的会覆盖前边绑定的函数。DOM2级事件处理，可以给同一个标签的同一个事件绑

定多个函数，触发事件时按照绑定的顺序执行相应的函数。

- DOM0 把事件直接绑定到标签或者直接绑定click事件都是dom0事件
- DOM2 是通过addEventListener绑定事件ie里面通过attachEvent