

JS第八天

数组的方法（续）

- **sort () 方法** 数组排序；默认排序方式是按照unicode编码从小到大排序的。（两位数会分开读取）（默认）
 - **sort (函数 (a,b{ }))** 自定义排序，sort () 括号里加入一个函数，这个函数必须返回一个数字，是正数就交换位置，是负数就不交换位置。（a-b是从小到大排序。b-a是从大到小）
- **concat()方法**连接一个或者多个数组 `arr2.concat(arr3)` 返回值为新的数组，就算没有新的数组去连接，就等于克隆一份数组
- **slice (第几位开始 (包含) , 第几位结束 (不包含)) 方法** 截取数组，返回值是一个新的数组；

字符串的方法（string）

- length 字符串的length只可读，不可写；
- 字符串的方法基本上不改变原字符串；

截取类方法

- **substring (0,0) 方法**：从第几位开始截取，到第几位结束，但不包含结束位置（计数从0开始）。如果只传一个参数，就视为传的是开始位置，一直截取到最后；
- **substr (0,0) 方法**：从第几位开始，截取多少个；
- **split () 方法**
- **indexOf () 方法**找到指定字符首次出现的位置，并且返回出来。返回值是索引；找不到返回-1；可以放两个参数，（指定字符，从第几位开始找）
- **includes () 方法**包含，字符串中是否包含指定字符，包含就返回true。不包含就返回false；

应用类方法

- **toUpperCase()方法** 把小写英文转换成大写英文
- **toLowerCase()方法**把大写英文转换成小写英文
- **trim()方法**去掉字符串的前后空格；

时间对象

- 时间对象：用户本地的时间（可以改，不安全）
- **new Date ()** `let d = new Date()`
- 时间对象中可以放数字，月份以当前月份 - 1
 - 获取年：date.getFullYear(); `d.getFullYear()`
 - 获取月：date.getMonth()+1;(获取的月份从0计算，比当前月份少1；所以+1) `d.getMonth()+1`
 - 获取日：date.getDate(); `d.getDate()`
 - 获取周几：date.getDay();(如果是周日，为0)
 - 获取时：date.getHours()
 - 获取分：date.getMinutes()
 - 获取秒：date.getSeconds()
- **定时器**
 - **setInterval ()** 当间隔一段指定时间时，就执行一次函数；
 - **clearInterval**:清除指定的setInterval定时器
 - **setTimeout (函数，指定的时间（毫秒），第一个函数的实参)** 当到指定的时间时执行一次函数（只执行一次）；返回值为数字（即为定时器的编号）；
 - **clearTimeout(编号)**：清除指定的setTimeout定时器
- **倒计时**（未来的时间 - 现在的时间 = 剩下的时间）
 - `let d = new Date(2019,3,10,12,30)` ;设置未来的时间。自己固定
 - `let d2 = new Date()` 获取当前时间
 - `let d3 = d - d2` 获取剩余时间（剩余时间以毫秒计算）
 - 单位换算：（首先算出秒）
 - 天：Math.floor(s/86400)
 - 时：Math.floor(s/3600)
 - 分：Math.floor(s/60)
 - 秒：s = Math.floor (d - d2)%60/1000 ?

递归（函数自己调用自己）

一个函数里面有相应的运算或者逻辑处理，需要这个处理重复执行，所以才会重复调用自己；

注意递归容易死循环，所以需要写上递归终止条件

例子：输出的是15；

```
function fn(n){
    if(n==1)return 1;
    return fn(n-1) + n;
}
```

DOM (Document Object Model) 文档对象模型

通过document提供了一些api，能够赋予开发者操作页面的能力；

api：接口，方法；

当通过document提供的api获取到元素的时候，获取到的元素是个对象，他跟页面的标签是相映射的关系，也就是说，通过操作对象的属性，能够操作标签；

关系：dom树；（关系树）由节点组成；

节点：元素节点；整个HTML中全属于节点；

nodeType 查看节点类型，它的值是一个数字；

```
console.log(box.nodeType)
```

返回1.代表元素节点；

返回2，代表属性节点，

返回3，文本节点（文字，空格，换行）；

返回8，注释节点，

返回9，document；

- 父子关系
 - parentNode 父节点（上级元素）
 - children[索引]所有子节点（包括，元素、文本、注释、属性节点），索引第几个孩子
 - children 找到所有的子集元素节点
 - 第一个孩子：firstElementChild
 - 中间某个孩子
 - 最后一个孩子:lastElementChild
- 祖孙关系
- 兄弟关系
 - 上一个兄弟节点
 - .previousElementSibling
 - 下一个兄弟节点
 - .nextElementSibling
 - 没有就是空（null）

元素的增删改

- 创建一个元素节点：`document.createElement('div')`
- 添加元素(往元素的末尾添加元素) `box.parent.appendChild(child)`
- 添加元素（往元素的首位添加元素） `box.parent.insertBefore(新的元素名, 参照元素（放在谁的前边）)`
- 删除元素 `box.parent.removeChild(要删除的元素)`
- 克隆元素 `box.cloneNode()` 如果括号内添加true，不但能克隆元素本身，也能克隆它的所有子集；

重点捋清关系、背单词；