

javascript 内置对象

Array
创建
.length
栈方法
push()
pop()
shift()
join()
reverse()
sort()
concat()
slice()
splice()
indexOf()
lashIndexOf()
String
stringObject.charAt(index)
stringObject.charCodeAt(index)
stringObject.slice(start,end)
stringObject.substring(n1,n2)
stringObject.substr(start,len)
stringObject.split(separator)
stringObject.replace(regexp/substr,replacement)

Array

创建

`var colors = new Array(3);`//数组内有三个变量，可以不指定

`var numbers = new Array(1,3,6,9);`//数组内有1, 3, 6, 9

`var cols = ["red","yellow","green"];`//字面量表示法

`var infos = [6,"marry",true,{email:"marry@163.com"}];`

- 索引: 0, 1,

- 超出索引会返回`undefined` 但可以添加超出索引的部分 `length`自动增加

.length

`array.length`

- 通过更改`length`可以把数组的值移除

栈方法

push()

- 把值添加到数组的最尾

`var colors = new Array("red","green");`

`colors.push("blue");`//返回添加数据之后的数组的值

`colors.push("yellow","black");`

unshift()

- 把值添加到数组的最头

`var nums=[2,7,8,6];`

`nums.unshift(1,2);`

pop()

-把值从尾部删除

-返回删除的元素

shift()

-把值从头部删除

-返回删除的元素

join()

`var nums=[2,4,5]`

```
var str=nums.join();

#str为2,4,5

var words=['border','left','color'];

var wordstr=words.join("");

#值之间没有连接符号

var wordstr=words.join("-"):

#border-left-color
```

reverse()

#颠倒数组中的顺序

```
nums.reverse()

console.log(nums)

#[5,4,2]
```

sort()

#默认方法toString()以第一位数字排序

```
var arr=[9,23,15,88,12];

arr.sort(function(a,b){return b-a});

console.log(arr);

#降序排列

arr.sort(function(a,b){return a-b});

console.log(arr);

#升序排列
```

concat()

```
var arr1=["a","b","c"],
    arr2=["d","e",1,3],
    arr3;

arr3=arr1.concat(arr2,['m',99,8])
```

slice()

#start 必须 规定从何处开始选取，如是负数，从数组尾部开始算起

#end 可选 规定从何处结束选取，是数组片断结束处的数组下标

```
var colors=["red","green","blue","yellow","orange"];
```

```
var newColors=colors.slice(1,3);
```

```
#green blue
```

```
var newColors2=colors.slice(-4,3);
```

```
#green blue
```

splice()

删除

```
var arr=["a","b","c","d","e","f"];
```

```
var delArr=arr.splice(2,2);
```

```
#[ "a","b","e","f"]
```

插入

```
var insertArr=arr.splice(2,0,"g","h");
```

```
#[ "a","b","c","g","h","d","e","f"]
```

替换

```
var replaceArr=arr.splice(1,2,"x","y","z");
```

```
#[ "a","x","y","z","d","e","f"]
```

indexOf()

#searchvalue 必需 要查找的项

#startIndex 可选 起点位置的索引

#返回找到的value的位置， 没有的时候返回-1

```
var nums=[1,7,5,7,8,1,6,9];
```

```
var pos=nums.indexOf(7);
```

```
#1
```

```
var pos=nums.indexOf(99);
```

```
#-1
```

```
var pos=nums.indexOf(7,3);
```

#3 从索引3开始查找

#===严格相等

#也可以查找字串的位置

lastIndexOf()

#从末尾查找

String

stringObject.charAt(index)

#返回字符串本身

#没有找到返回-1

#也可检测字符串

stringObject.charCodeAt(index)

#返回字符串的字符编码

#没有找到返回-1

#也可检测字符串

stringObject.slice(start,end)

stringObject.substring(n1,n2)

#自动转换小的数字为开始位置，较大的数字为结束位置

#负数自动转换为0

#切片包括结束位置

stringObject.substr(start,len)

str.substr(-5,4) //(6,4)

str.substr(3,-4) //什么都截取不到

stringObject.split(separator)

var str='welcome-to-beijing';

var arr=str.split("-");

stringObject.replace(regex/substr,replacement)

var tel='010-62971268,4000-100-9098,010-86789889';

var newTel=tel.replace(',',' ');

string.toUpperCase()

#生成字符串的副本

string.toLowerCase()

Math

Math.min(num1,num2,...)

var min=Math.min(5,-4,0)

#只要出现一个非数字就返回NaN

Math.max(num1,num2,...)

Math.ceil(num)

#向上取整 返回大于num的最小整数

Math.floor(num)

#向下取整

Math.abs(num)

Math.random()

#生成0-1之间的随机数 不生成1

Math.floor(Math.random()*(m-n+1)+n);

#生成n-m之间的数 m>n

Date

创建

var today = new Date()

var date = new Date(year,moth,day,hours,minutes,seconds)

获取方法

getFullYear(): 4位数年份

getMonth(): 0-11

getDate(): 返回月份中的天数

getDay(): 返回星期0-6

getHours():

getMinutes():

getSeconds():

getTime(): 返回1970年1月1日到现在的毫秒数

year=today.getFullYear()

设置方法

get变为set

#50天之后是星期几

today.setDate(today.getDate+50)

today.getDay()