

## 5、字符串的新增方法

- 实例方法: includes(), startsWith(), endsWith()-- 子串查找方法, 返回布尔值。
- 实例方法: repeat()-- 方法返回一个新字符串, 表示将原字符串重复n次。
- 实例方法: padStart(), padEnd() -- 字符串补全长度的功能。
- 实例方法: trimStart(), trimEnd()--消除字符串头部或尾部的空格, 返回新串, 不修改原字符串。
- 实例方法: matchAll()-- 返回一个正则表达式在当前字符串的所有匹配。

### ① 实例方法: includes(), startsWith(), endsWith():

- JavaScript 只有indexOf方法, 可以用来确定一个字符串是否包含在另一个字符串中。
- ES6 又提供了三种新方法。
  - includes(): 返回布尔值, 表示是否找到了参数字符串。
  - startsWith(): 返回布尔值, 表示参数字符串是否在原字符串的头部。
  - endsWith(): 返回布尔值, 表示参数字符串是否在原字符串的尾部。

```
//定义一个url地址
let url = 'http://www.baidu.com/a/b.html';

//判断url字符串是否是以http开头
console.log(url.startsWith('http')); //true

//判断url字符串是否是以html结束
console.log(url.endsWith('html')); //true

//判断url字符串中是否包含baidu子串
console.log(url.includes('baidu')); //true
```

### ② 实例方法: repeat()

- repeat方法返回一个新字符串, 表示将原字符串重复n次。

```
//将原字符串重复指定次数
console.log('x'.repeat(3)); // 'xxx' 3次重复x

console.log('hello'.repeat(2)); // 'hellohello' 2次重复hello

console.log('na'.repeat(0)); // '' 0次重复, 故没有输出内容

console.log('na'.repeat(2.9)); // 'nana' 参数会自动舍去取整为2

console.log('na'.repeat(-1)); // 范围错误 RangeError: Invalid count value
```

### ③ 实例方法：padStart() / padEnd()

- ES2017 引入了字符串补全长度的功能。
- 如果某个字符串不够指定长度，会在头部或尾部补全。
  - padStart()用于头部补全
  - padEnd()用于尾部补全
- 这两个方法都有两个参数：
  - 第一个参数用来指定字符串的最小长度，
  - 第二个参数是用来补全的字符串，省略默认补空格。

```
console.log("123".padStart(5,"0")); //00123
console.log("123".padEnd(5,"0"));   //12300
console.log("123".padEnd(2,"0"));   //123

//补齐等宽编号
console.log("3".padStart(8,"20190000")); //20190003
console.log("23".padStart(8,"20190000")); //20190023
console.log("123".padStart(8,"20190000")); //20190123

//另一个用途是提示字符串格式（如下提示日期格式）。
console.log('12'.padStart(10, 'YYYY-MM-DD')); // "YYYY-MM-12"
console.log('09-12'.padStart(10, 'YYYY-MM-DD')); // "YYYY-09-12"
```

### ④ 实例方法：trimStart(), trimEnd()

- ES2019 对字符串实例新增了trimStart()和trimEnd()这两个方法。
- 它们的行为与trim()一致，trimStart()消除字符串头部的空格，trimEnd()消除尾部的空格。
- 它们返回的都是新字符串，不会修改原始字符串。

```
const s = "  abc  ";
s.trim()      // "abc"
s.trimStart() // "abc  "
s.trimEnd()   // "  abc"
```

- 浏览器还部署了额外的两个方法，trimLeft()是trimStart()的别名，trimRight()是trimEnd()的别名。

### ⑤ 实例方法：matchAll()

- matchAll()方法返回一个正则表达式在当前字符串的所有匹配。

```
let info = "<ul><li>北京</li><li>上海</li><li>广州</li></ul>";  
//传统匹配方式  
console.log(info.match(/<li>(.*?)</li>/g));  
//结果: ["<li>北京</li>", "<li>上海</li>", "<li>广州</li>"]  
  
//ES2020 增加了String.prototype.matchAll()方法,  
//可以一次性取出所有匹配。不过,它返回的是一个遍历器(Iterator),而不是数组。  
console.log([...info.matchAll(/<li>(.*?)</li>/g)]);  
//结果: [Array(2), Array(2), Array(2)]  
//其中["<li>北京</li>", "北京", index: 4, input: "<ul>...</ul>", groups: undefined]
```