**字符串的扩展**

一：字符的Unicode表示法

1：JavaScript 允许采用\uxxxx形式表示一个字符，其中xxxx表示字符的 Unicode 点。

但是，这种表示法只限于码点在\u0000~\uFFFF之间的字符。超出这个范围的字符，必须用两个双字节的形式表示。

console.log("\uD842\uDFB7");

console.log("\u20BB7");



2：在ES6中只要将码点放入大括号，就能正确解读该字符。

console.log("\u{41}\u{42}\u{43}");

==》ABC

二：codePointAt()

1：codePointAt方法会正确返回 32 位的 UTF-16 字符的码点。对于那些两个字节储存的常规字符，它的返回结果与charCodeAt方法相同。

2：codePointAt方法返回的是码点的十进制值，如果想要十六进制的值，可以使用toString方法转换一下。

let s = ';

for (let ch of s) {

console.log(ch.codePointAt(0).toString(16));

}

==>20bb7

==>61

三：String.fromCodePoint()

ES6 提供了String.fromCodePoint方法，可以识别大于0xFFFF的字符，弥补了String.fromCharCode方法的不足。在作用上，正好与codePointAt方法相反。

四：字符串的遍历接口

ES6 为字符串添加了遍历器接口，使得字符串可以被for...of循环遍历。

for (let codePoint of 'foo') {

console.log(codePoint)

}

==>f

==>o

==>o

五：normalize()

六：includes(), startsWith(), endsWith()

传统上，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：

****1：includes()****：返回布尔值，表示是否找到了参数字符串。（使用第二个参数n时，针对从第n个位置直到字符串结束。）

****2：startsWith()****：返回布尔值，表示参数字符串是否在原字符串的头部。（使用第二个参数n时，针对从第n个位置直到字符串结束。）

****3: endsWith()****：返回布尔值，表示参数字符串是否在原字符串的尾部。(使用第二个参数n时，针对前n个字符)

var str = 'Hello world!';

console.log(str.startsWith('Hello')); ==》true

console.log(str.endsWith('!')); ==>true

console.log(str.includes('o')); ==>true

▲这三个方法都支持第二个参数，表示开始搜索的位置。

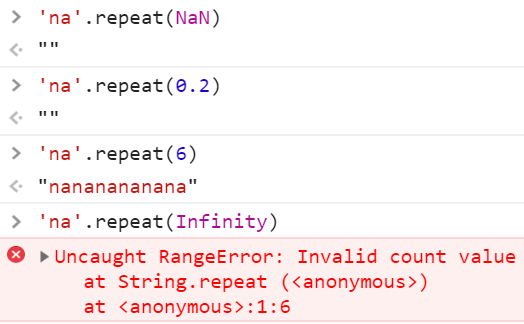
七：repeat()

repeat方法 1:返回一个新字符串，表示将原字符串重复n次；

2:如果是小数，会被取整（四舍五不入）；

3：如果参数是0-1之间的小数或NaN，等同于0，返回空字符串；

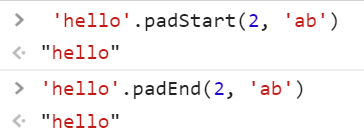
4：如果参数是负数或者Infinity,会报错；

八：ES8

padStart()，padEnd()

接受两个参数，第一个参数是字符串补全生效的最大长度，第二个参数是用来补全的字符串。

1：如果原字符串的长度，等于或大于最大长度，则字符串补全不生效，返回原字符串;



2：如果用来补全的字符串与原字符串，两者的长度之和超过了最大长度，则会截去超出位数的补全字符串;

0679f34a8bf92d88f8b97ff234f37eb

3：如果省略第二个参数，默认使用空格补全长度;



九：matchAll()

返回一个正则表达式在当前字符串的所有匹配

十：**模板字符串**

模板字符串（template string）是增强版的字符串，用反引号（`）标识。它可以当作普通字符串使用，也可以用来定义多行字符串，或者在字符串中嵌入变量。

1:普通字符串：

console.log(`In JavaScript '\n' is a line-feed.`);

==》In JavaScript ‘

‘is a Line-feed.

2:多行字符串

`In JavaScript this is

not legal.`

console.log(`string text line 1

string text line 2`);

===》string text line 1

string text line 2

3:字符串中嵌入变量

let name = "Bob",

time = "today";

console.log(`Hello ${name}, how are you ${time}?`);

==>Hello Bob, how are you today?

## 十二：实例：模板编译

## 十二：标签模板

模板字符串可以紧跟在一个函数名后面，该函数将被调用来处理这个模板字符串。这被称为“标签模板”功能（tagged template）。

**例：alert`123`**

**等同于**

**alert(123)**

**十三：**String.raw()

**十四：模板字符串的限制**

模板字符串默认会将字符串转义，导致无法嵌入其他语言。