09.字符串的扩展改进

学习要点：

1.新增方法

2.模板字符串

本节课我们来开始学习 ES6 新增的字符串方法以及模板字符串功能。

一. 新增方法

1. 对于一些超过两个字符(四字节)的异体字，ES6 新增了 codePointAt()方法；

```
//两个字符的异体字,这里用？代替，文档显示不出来，上面土，下面口
let text = '?';
console.log(text.length);          //2
console.log(text.charAt(0));       //?
console.log(text.charCodeAt(0));   //55362
console.log(text.charCodeAt(1));   //57271
console.log(text.codePointAt(0));  //134071，十进制码点

//一个字符的简体字
let text = '吉';
console.log(text.length);          //1
console.log(text.charAt(0));       //吉
console.log(text.charCodeAt(0));   //21513
console.log(text.charCodeAt(1));   //NaN
console.log(text.codePointAt(0));  //21513
```

2. 对于超过两字符的码点，可以通过 ES6 新增的 String.fromCodePoint()得到；

```
//下面可以得到：上面为土，下面为口的异体字
console.log(String.fromCodePoint(134071));
```

3. ES6 提供 normalize()方法用于有音标的符号组合形式进行统一；

```
console.log('\u01D1');
console.log('\u004F');
console.log('\u030C');
console.log('\u004F\u030C'.normalize() == '\u01D1'.normalize());
```

4. ES6 提供了三种判断字符串的方法：includes()、startsWith()和 endsWith()；

```
let text = 'Hello, Mr.Lee!';
console.log(text.includes('Mr.'));      //true，是否找到'Mr.'
console.log(text.startsWith('ello'));   //false，必须从开头找
console.log(text.endsWith('ee!'));      //true，必须从结尾找
console.log(text.includes('Mr.', 8));   //false，超过位置，第二参数寻找的位置
```

5. repeat()重复字符串，padStart()补全字符串头部，padEnd()补全字符串尾部；

```
console.log('x'.repeat(5));
console.log('xyz'.repeat(3));
console.log('Mr.Lee'.repeat(0));        //空

console.log('x'.padStart(5, 'Mr'));     //MrMrx
console.log('x'.padEnd(5, 'Mr'));       //xMrMr
```

二．模板字符串

1. 在 ES6 之前，字符串内夹杂变量，我们总是通过分离和+号连接解决；

```
let name = 'Mr.Lee',
    age  = 100,
    text =  '我是' + name + '，今年' + age + '岁';

console.log(text);
```

2. 现在可以直接使用(`)反引号配合${var}模版语法格式，直接实现变量解析功能；

```
text = `我是${name}，今年${age}岁`;     // \`转义

//支持多行操作
text = `我是
${name}，今年
${age}岁`;
```

3. 如果我们想在字符串中插入表达式，也可以使用${a + b}模版语法；

```
text = `一加一等于：${1+1}`;
```

4. ${${}}这种模版嵌套的方式，也是支持的；

```
text = `结果：${flag ? `true${1+1}` : 'false'}`;
```

5. 可以使用 String.raw 来得到原生字符串；

```
text = String.raw `我\n 是`;
```