第 13 章 基本包装类型

学习要点：

1.基本包装类型概述
2.Boolean 类型
3.Number 类型
4.String 类型

类似于java

主讲教师：李炎恢
合作网站：http://www.ibeifeng.com
讲师博客：http://hi.baidu.com/李炎恢

为了便于操作基本类型值，ECMAScript 提供了 3 个特殊的引用类型：Boolean、Number 和 String。这些类型与其他引用类型相似，但同时也具有与各自的基本类型相应的特殊行为。实际上，每当读取一个基本类型值的时候，后台就会创建一个对应的基本包装类型的对象，从而能够调用一些方法来操作这些数据。

一．基本包装类型概述

```
var box = 'Mr. Lee';                    //定义一个字符串
var box2 = box.substring(2);            //截掉字符串前两位
alert(box2);                            //输出新字符串
```

自动装箱和拆箱

变量 box 是一个字符串类型，而 box.substring(2)又说明它是一个对象(PS：只有对象才会调用方法)，最后把处理结果赋值给 box2。'Mr. Lee'是一个字符串类型的值，按道理它不应该是对象，不应该会有自己的方法，比如：

```
alert('Mr. Lee'.substring(2));          //直接通过值来调用方法
```

1.字面量写法：

```
var box = 'Mr. Lee';                    //字面量
box.name = 'Lee';                       //无效属性
box.age = function () {                 //无效方法
    return 100;
};
alert(box);                             //Mr. Lee
alert(box.substring(2));                //. Lee
alert(typeof box);                      //string
alert(box.name);                        //undefined
alert(box.age());                       //错误
```

在js地底层是以字符数组的形式存在的：box[0]为M,其他依次类推

2.new 运算符写法：

```
var box = new String('Mr. Lee');        //new 运算符
box.name = 'Lee';                       //有效属性
box.age = function () {                 //有效方法
    return 100;
```

box对象增加name属性和age方法

```
};
alert(box);                    //Mr. Lee
alert(box.substring(2));       //. Lee
alert(typeof box);             //object
alert(box.name);               //Lee
alert(box.age());              //100
```

属性和方法可以在外部动态添加，python也可以

以上字面量声明和 new 运算符声明很好的展示了他们之间的区别。但有一定还是可以肯定的，那就是不管字面量形式还是 new 运算符形式，都可以使用它的内置方法。并且 Boolean 和 Number 特性与 String 相同，三种类型可以成为基本包装类型。

PS：在使用 new 运算符创建以上三种类型的对象时，可以给自己添加属性和方法，但我们建议不要这样使用，因为这样会导致根本分不清到底是基本类型值还是引用类型值。

二．Boolean 类型

Boolean 类型没有特定的属性或者方法。

三．Number 类型

Number 类型有一些静态属性(直接通过 Number 调用的属性，而无须 new 运算符)和方法。

Number 静态属性

属　性	描述
MAX_VALUE	表示最大数
MIN_VALUE	表示最小值
NaN	非数值
NEGATIVE_INFINITY	负无穷大，溢出返回该值
POSITIVE_INFINITY	无穷大，溢出返回该值
prototype	原型，用于增加新属性和方法

Number 对象的方法

方　法	描述
toString()	将数值转化为字符串，并且可以转换进制
toLocaleString()	根据本地数字格式转换为字符串
toFixed()	将数字保留小数点后指定位数并转化为字符串
toExponential()	将数字以指数形式表示，保留小数点后指定位数并转化为字符串
toPrecision()	指数形式或点形式表述数，保留小数点后面指定位数并转化为字符串

```
var box = 1000.789;
alert(box.toString());                  //转换为字符串，传参可以转换进制
alert(box.toLocaleString());            //本地形式，1,000.789
alert(box.toFixed(2));                  //小数点保留，1000.78
alert(box.toExponential());             //指数形式，传参会保留小数点
alert(box.toPrecision(3));              //指数或点形式，传参保留小数点
```

四．String 类型

String 类型包含了三个属性和大量的可用内置方法。

构造函数

String 对象属性

属　性	描述
length	返回字符串的字符长度
constructor	返回创建 String 对象的函数
prototype	通过添加属性和方法扩展字符串定义

String 也包含对象的通用方法，比如 valueOf()、toLocaleString()和 toString()方法，但这些方法都返回字符串的基本值。

等价于[]访问

字符方法

方　法	描述
charAt(n)	返回指定索引位置的字符
charCodeAt(n)	以 Unicode 编码形式返回指定索引位置的字符

```
var box = 'Mr.Lee';
alert(box.charAt(1));                   //r
alert(box.charCodeAt(1));               //114
alert(box[1]);                          //r，通过数组方式截取
```

PS：box[1]在 IE 浏览器会显示 undefined，所以使用时要慎重。

切片

字符串操作方法

方　法	描述
concat(str1...str2)	将字符串参数串联到调用该方法的字符串
slice(n,m)	返回字符串 n 到 m 之间位置的字符串
substring(n,m)	同上
substr(n,m)	返回字符串 n 开始的 m 个字符串

包含头不包含尾部

```
var box = 'Mr.Lee';
alert(box.concat(' is ', ' Teacher ', '!'));          //Mr.Lee is Teacher !
alert(box.slice(3));                                  //Lee
alert(box.slice(3,5));                                //Le
alert(box.substring(3));                              //Lee
alert(box.substring(3,5));                            //Le
alert(box.substr(3));                                 //Lee
alert(box.substr(3,5));                               //Lee

var box = 'Mr.Lee';
alert(box.slice(-3));                                 //Lee，6+(-3)=3 位开始
alert(box.substring(-3));                             //Mr.Lee  负数返回全部
alert(box.substr(-3));                                //Lee，6+(-3)=3 位开始

var box = 'Mr.Lee';
alert(box.slice(3, -1));                              //Le 6+(-1)=5, (3,5)
alert(box.substring(3, -1));                          //Mr.  第二参为负，直接转 0，
                                                      //并且方法会把较小的数字提前，(0,3)
alert(box.substr(3, -1));                             //''  第二参数为负，直接转 0 ，(3,0)
```

substring不识别负数的参数

PS：IE 的 JavaScript 实现在处理向 substr()方法传递负值的情况下存在问题，它会返回原始字符串，使用时要切记。

字符串位置方法

方　法	描述
indexOf(str, n)	从 n 开始搜索的第一个 str，并将搜索的索引值返回
lastIndexOf(str, n)	从 n 开始搜索的最后一个 str，并将搜索的索引值返回

从指定的位置向前搜索而不是向后搜索

```
var box = 'Mr.Lee is Lee';
alert(box.indexOf('L'));                              //3
alert(box.indexOf('L', 5));                           //10
alert(box.lastIndexOf('L'));                          //10
alert(box.lastIndexOf('L', 5));                       //3，从指定的位置向前搜索
```

PS：如果没有找到想要的字符串，则返回-1。

示例：找出全部的 L

```
var box = 'Mr.Lee is Lee';                            //包含两个 L 的字符串
var boxarr = [];                                      //存放 L 位置的数组
var pos = box.indexOf('L');                           //先获取第一个 L 的位置
while (pos > -1) {                                    //如果位置大于-1，说明还存在 L
    boxarr.push(pos);                                 //添加到数组
    pos = box.indexOf('L', pos + 1);                  //从新赋值 pos 目前的位置
```

```
}
alert(boxarr);                          //输出
```

大小写转换方法

方　法	描述
toLowerCase(str)	将字符串全部转换为小写
toUpperCase(str)	将字符串全部转换为大写
toLocaleLowerCase(str)	将字符串全部转换为小写，并且本地化
toLocaleupperCase(str)	将字符串全部转换为大写，并且本地化

```
var box = 'Mr.Lee is Lee';
alert(box.toLowerCase());               //全部小写
alert(box.toUpperCase());               //全部大写
alert(box.toLocaleLowerCase());         //
alert(box.toLocaleUpperCase());         //
```

PS：只有几种语言（如土耳其语）具有地方特有的大小写本地性，一般来说，是否本地化效果都是一致的。

有点像indexOf，只是在这里可以使用正则表达式

字符串的模式匹配方法

方　法	描述
match(pattern)	返回 pattern 中的子串或 null
replace(pattern, replacement)	用 replacement 替换 pattern
search(pattern)	返回字符串中 pattern 开始位置
split(pattern)	返回字符串按指定 pattern 拆分的数组

正则表达式在字符串中的应用，在前面的章节已经详细探讨过，这里就不再赘述了。以上中 match()、replace()、serach()、split()在普通字符串中也可以使用。

```
var box = 'Mr.Lee is Lee';
alert(box.match('L'));                  //找到 L，返回 L 否则返回 null
alert(box.search('L'));                 //找到 L 的位置，和 indexOf 类型
alert(box.replace('L', 'Q'));           //把 L 替换成 Q
alert(box.split(' '));                  //以空格分割成字符串
```

其他方法

方　法	描述
fromCharCode(ascii)	静态方法，输出 Ascii 码对应值
localeCompare(str1,str2)	比较两个字符串，并返回相应的值

说明charCode小

```
alert(String.fromCharCode(76));                    //L，输出 Ascii 码对应值
```

localeCompare(str1,str2)方法详解：比较两个字符串并返回以下值中的一个；
1.如果字符串在字母表中应该排在字符串参数之前，则返回一个负数。(多数-1)
2.如果字符串等于字符串参数，则返回 0。
3.如果字符串在自附表中应该排在字符串参数之后，则返回一个正数。(多数 1)

```
var box = 'Lee';
alert(box.localeCompare('apple'));                 //1
alert(box.localeCompare('Lee'));                   //0
alert(box.localeCompare('zoo'));                   //-1
```

HTML 方法

方　法	描述
anchor(name)	str
big()	<big>str</big>
blink()	<blink>str</blink>
bold()	Str
fixed()	<tt>Str</tt>
fontcolor(color)	str
fontsize(size)	str
link(URL)	str
small()	str
strike()	<strike>str</strike>
italics()	italics
sub()	_{str}
sup()	^{str}

以上是通过 JS 生成一个 html 标签，根据经验，没什么太大用处，做个了解。

```
var box = 'Lee';                                   //
alert(box.link('http://www.yc60.com'));            //超链接
```

感谢收看本次教程！

本课程是由北风网(ibeifeng.com)

瓢城 Web 俱乐部(yc60.com)联合提供：

本次主讲老师：李炎恢

我的博客：hi.baidu.com/李炎恢/

我的邮件：yc60.com@gmail.com