

# 深入理解包装类及其所以常用方法和属性



jCodeLife



0.129

2020.08.13 12:30:04 字数 2,514 阅读 36

编辑文章

你是否想了解什么是包装类？

或者想了解包装类的原型上定义了哪些方法和属性？

这篇文章能帮助你梳理清楚，并且会手动封装常见的包装类方法

我们先来看一个现象：

```
1 var str = "hello" ;
2 console.log(str.length) ;//5
```

定义了一个字符串变量str，字符串调用了`length`属性，打印出字符串的长度！

你是否会奇怪？字符串属于原始值类型，原始值里头不应该有属性才对？

确实，对象身上才有属性和方法！

那上面是怎么回事？

你应该听说过这么一句话：“JavaScript语言一切皆是对象”。其实不仅仅数组和函数是对象，我们知道的原始值（数值、字符串、布尔值），在一定条件下，也会转成对象，这种对象被称为“**包装类对象**”。

所以，上面代码能实现的原因在于：当调用原始值的属性和方法时，js发现原始值根本没有属性和方法，你怎么调用呢？为了不然程序报错，于是它偷偷地将原始类型的变成了包装类对象，在调用这个新创建好的包装类对象的属性和方法执行，执行完后偷偷的delete此对象。

看个例子：

```
1 var num = 3;
2 num.len = 2;
3 //new Number(3).len =2; delete
4 console.log(num.len);//undefined 这里会重新new Number(3).len，跟之前的不一样
5
6 var str = "abcd";
7 str.length = 2;
8 //new String('abcd').length = 2; delete;
9 console.log(str);//abcd
10 console.log(str.length);//重新new String('abcd')：注意：length是string系统自带的属性。所以返回4
```

上面例子可以看出，每次调用都会偷偷重新生成一个全新的对象，全新的对象跟之前的没有关系，因为它每次执行完就delete掉了。所以包装类对象的生命周期非常短暂，导致我们给包装类添加自定义属性和方法没有任何意义！

其实说白了包装类就是函数，当原始值在调用属性或方法时，会触发一个机制，这个机制会在你不知情的情况下，将原始类型的值传入对应的函数，然后构造出对应的对象！这个过程有包装、打包的意思，即你给我个原始值，我给你一个对象。所以大家把这类函数称为包装类。反正就是一个名称，大家记住就好了。

这下明白了吧！

那我们来看看都有哪些包装类？

其实也就三个，就是三个原始值对应的包装类，包括Number、String、Boolean

先来讲前面两个简单的，`Number()`和`Boolean()`

## 推荐阅读

Vue干货小技巧——学会再也不做加班狗

阅读 1,314

超级基础却又超级容易出错的前端面试题（1）

阅读 89

学会这6个强大的CSS选择器，将真正帮你写出干净的CSS代码！

阅读 258

面试了几个前端，给爷整哭了！

阅读 29,568

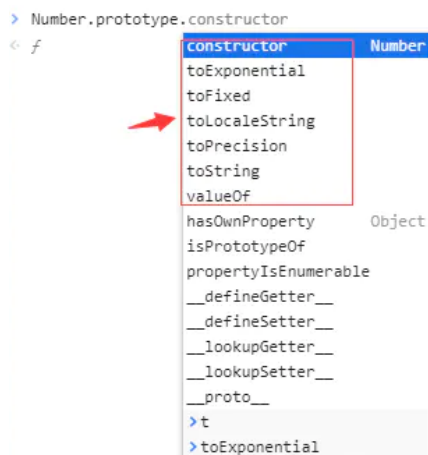
vue优化页面

阅读 818

```
1 console.log(Number(true))//1
2 console.log(Number("hello"))//NaN
```

通过new Number()产生的包装对象能继承Number原型链上的属性和方法，包括Number.prototype上的和Object.prototype上的  
这里主要是讲直接原型Number.prototype上的

看看都有哪些：



1. constructor属性: 返回Number原型的构造函数，即返回Number(){}
2. toExponential方法: 返回一个数的科学计数法的形式
3. toFixed方法: 将一个数转为指定位数的小数，返回这个小数对应的 **字符串** 如下：

```
> var num = 1
< undefined
> num.toFixed(2)
< "1.00"
```

4. toLocaleString方法: 返回一个数转换为本地字符串的形式
5. toPrecision方法: 用于将一个数转为指定位数的有效数字，返回字符串形式

```
> var num1 = 12
< undefined
> num1.toPrecision(4)
< "12.00" 4位有效数字的字符串形式
```

6. toString方法: 返回一个数字的字符串形式；注意这里的toString不是Object身上的哦

```
> Number.prototype.toString === Object.prototype.toString
< false
```

7. valueOf方法: 用于返回一个 Number 对象的原始数字值

```
> new Number(12).valueOf()
< 12
```

另外再来看Number这个构造函数本身的静态属性（通过Number.直接调用的）有哪些：

1. MAX\_VALUE: 可表示的最大的数
2. MIN\_VALUE: 可表示的最小的数
3. NEGATIVE\_INFINITY: 负无穷大，溢出时返回该值
4. NaN: 非数

## 推荐阅读

Vue干货小技巧——学会再也不做加班狗

阅读 1,314

超级基础却又超级容易出错的前端面试题（1）

阅读 89

学会这6个强大的CSS选择器，将真正帮你写出干净的CSS代码！

阅读 258

面试了几个前端，给爷整哭了！

阅读 29,568

vue优化页面

阅读 818

下面再来讲另外一个包装类：Boolean

Boolean函数本身调用的话：是将一个数据变成对应的布尔值，返回true或者false

```
Boolean(undefined) // false
Boolean(null) // false
Boolean(0) // false
Boolean('') // false
Boolean(NaN) // false

Boolean(1) // true
Boolean('false') // true
Boolean([]) // true
Boolean({}) // true
Boolean(function () {}) // true
Boolean(/foo/) // true
```

Boolean的原型上也有一个属性constructor和两个方法：

```
> Boolean.prototype.constructor
Boolean
  toString
  valueOf
  hasOwnProperty
  isPrototypeOf
  propertyIsEnumerable
  toLocaleString
  __defineGetter__
  __defineSetter__
  __lookupGetter__
  __lookupSetter__
  __proto__
  > constructor
```

1. toString方法： 返回一个布尔值的字符串形式
2. valueOf： 返回一个Boolean对象的原始布尔值

Boolean和Number都比较少东西。下面讲包装类中的老大哥：String()

我们常常说的字符串的方法，都是源于它String.prototype

首先来看String()本身的正常用法是：将任意类型数据转换为字符串

1. 数值变成数字字符串
2. 字符串任是字符串
3. 布尔值true为“true”,false为“false”
4. undefined、null变成对应的字符串“undefined”和“null”

再来看看，new String产生的对象继承自原型上的属性和方法有多少：

## 推荐阅读

Vue干货小技巧——学会再也不做加班狗

阅读 1,314

超级基础却又超级容易出错的前端面试题（1）

阅读 89

学会这6个强大的CSS选择器，将真正帮你写出干净的CSS代码！

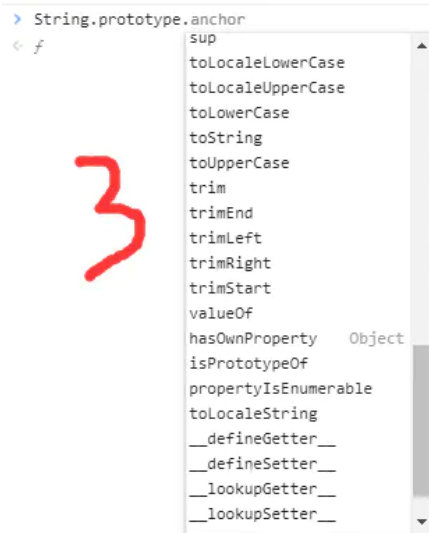
阅读 258

面试了几个前端，给爷整哭了！

阅读 29,568

vue优化页面

阅读 818



不要怀疑，不要惊讶，就是有这么多！不然怎么叫它老大哥呢 233...



大哥！社会哥！

不要慌，下面常用的方法一个个都会讲到：

1. length属性：返回字符串长度
2. charAt方法：返回指定位置的字符；位置从0开始
3. charCodeAt方法：返回指定位置的字符的Unicode编码
4. concat方法：跟数组中的concat类似，用于连接两个字符串，返回新的字符串，不改变原字符串
5. slice方法：跟数组中的slice也类似，用于返回从原字符串中截取的字字符串，不改变原字符串

## 推荐阅读

Vue干货小技巧——学会再也不做加班狗

阅读 1,314

超级基础却又超级容易出错的前端面试题（1）

阅读 89

学会这6个强大的CSS选择器，将真正帮你写出干净的CSS代码！

阅读 258

面试了几个前端，给爷整哭了！

阅读 29,568

vue优化页面

阅读 818



只有一个参数时：则表示从该位开始，截取到最后并返回

不传参时：相当于是截取了整个字符串，可以理解为复制了一个一样的值

如果参数是负值：表示从倒数位置

如果第一个参数大于第二个参数：如slice(3,1)，方法返回一个空字符串

6. substring方法：也使用从原字符串中截取子字符串并返回，但有一些奇怪的规则，因此很多人不建议使用这个方法，优先使用slice。

7. substr方法：也是从原字符串取出子字符串并返回；规则跟数组中的splice有点相似：

如果两个参数：第一个参数表示开始位，第二个参数表示截取的子字符串的长度

当只有一个参数时：则表示从开始位开始，截取到原字符串最后

如果第一个参数是负数：表示倒数计算的字符位置

如果第二个参数是负数，将被自动转为0即截0位，因此会返回空字符串

8. indexOf()和lastIndexOf()：跟ES5中数组的indexOf和lastIndexOf类似，用于查找指定字符串在原字符串出现的位置，如果没有找到返回-1

indexOf从字符串头部开始匹配，lastIndexOf从尾部开始匹配

接受第二个参数时，indexOf第二个参数表示从该位置开始向后匹配；lastIndexOf第二个参数表示从该位置起向前匹配

9. trim()：用于去除字符串两端的空格，返回去除空格后的新字符串，不改变原字符串

10. toLowerCase()：用于将字符串全部转为小写形式的字符串返回，不改变原字符串

11. toUpperCase()：用于将字符串全部转为大写形式的字符串返回，不改变原字符串

12. localeCompare()：比较两个字符串Unicode编码大小，第一个大于第二个返回1，第一个小于第二个返回-1，相等返回0

13. match()：通过正则匹配原字符串是否有某个子字符串，返回一个数组，成员为匹配的第一个字符串。如果没有找到匹配，则返回null

14. search()：查找字符串中是否有某个字符串。返回值为匹配的第一个位置。如果没有找到匹配，则返回-1。

15. replace()：用于替换匹配的子字符串，一般情况下只替换第一个匹配（除非使用带有g修饰符的正则表达式）

16. split()：将字符串通过指定分隔符拆分成数组。按照什么拆分，什么就没有了

```
var arr = [1, 2, 3, 4, 5];
var str = arr.join("-");//1-2-3-4-5
console.log(str.split("-"));//["1", "2", "3", "4", "5"]
```

17. includes：判断字符串中是否包含什么，返回true或false

18. startsWith：判断是不是以某字符串开头，返回true或false

19. endsWith：判断是不是以某字符串结尾，返回true或false

20. repeat()：返回重复连接指定次数的字符串

21. trimLeft()和trimStart()：去除左边空格

22. trimRight()和trimEnd()：去除右边空格

## 推荐阅读

Vue干货小技巧——学会再也不做加班狗

阅读 1,314

超级基础却又超级容易出错的前端面试题（1）

阅读 89

学会这6个强大的CSS选择器，将真正帮你写出干净的CSS代码！

阅读 258

面试了几个前端，给爷整哭了！

阅读 29,568

vue优化页面

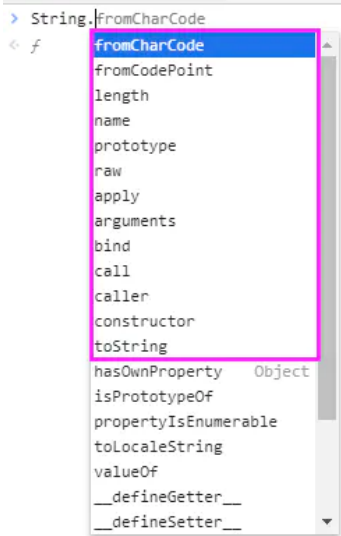
阅读 818

23. 有很多方法用于将字符串包含在相对应的 HTML 标签中返回。注意：可能在某些浏览器下不支持

方法	描述
<code>anchor()</code>	创建 HTML 锚。
<code>big()</code>	用大号字体显示字符串。
<code>blink()</code>	显示闪烁字符串。
<code>bold()</code>	使用粗体显示字符串。
<code>fixed()</code>	以打字机文本显示字符串。
<code>fontcolor()</code>	使用指定的颜色来显示字符串。
<code>fontsize()</code>	使用指定的尺寸来显示字符串。
<code>italics()</code>	使用斜体显示字符串。
<code>link()</code>	将字符串显示为链接。
<code>small()</code>	使用小字号来显示字符串。
<code>strike()</code>	用于显示加删除线的字符串。
<code>sub()</code>	把字符串显示为下标。
<code>sup()</code>	把字符串显示为上标。

```
> str.fixed(111)
< "<tt> xxxx xx </tt>"
> str.link()
< "<a href='undefined'> xxxx xx </a>"
> str.fontsize()
< "<font size='undefined'> xxxx xx </font>"
> str.anchor()
< "<a name='undefined'> xxxx xx </a>"
> str.italics()
< "<i> xxxx xx </i>"
> str.defaultMessage()
< " xxxx xx "
> str.bold()
< "<b> xxxx xx </b>"
```

最后再来看一眼，String身上的静态属性和方法。有兴趣的可以自己试试：



以上就是包装类的总结内容！

参考资料

推荐阅读

Vue干货小技巧——学会再也不做加班狗

阅读 1,314

超级基础却又超级容易出错的前端面试题（1）

阅读 89

学会这6个强大的CSS选择器，将真正帮你写出干净的CSS代码！

阅读 258

面试了几个前端，给爷整哭了！

阅读 29,568

vue优化页面

阅读 818

写下你的评论...

评论0 赞2 ...



JavaScript toLocaleString() 方法—W3SCHOOL

JavaScript Number 对象

深度剖析JS类型转换规律（显示类型转换和隐式类型转换）



"小礼物走一走，来简书关注我"

赞赏支持

还没有人赞赏，支持一下



jCodeLife 书山有路勤为径，学海无涯苦作舟  
总资产14 (约0.99元) 共写了12.3W字 获得231个赞 共23个粉丝



写下你的评论...

全部评论 只看作者 关闭评论

按时间倒序 按时间正序

被以下专题收入，发现更多相似内容

投稿管理

+ 收入我的专题



领跑计划笔记

