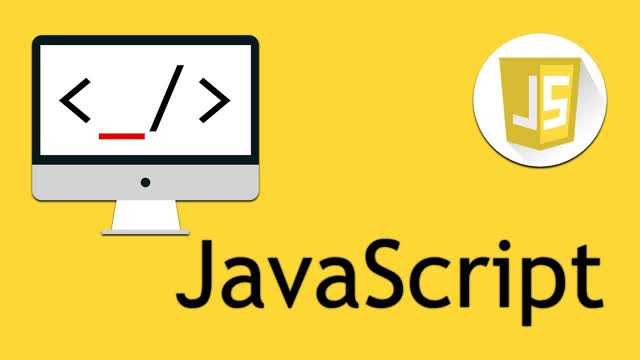
啦啦啦 - JS 基础系列之字符串



图片来源 <https://www.beansmile.com/blog/posts/javascript-prototype-inherit>

* [一 引子](" \l "一 引子)
* [二 正文](" \l "二 正文)
  1. [2.1 关于 String](" \l "2.1 关于 String)
  2. [2.2 关于字面量](" \l "2.2 关于字面量)
  3. [2.3 与 String 相关的知识点](" \l "2.3 与 String 相关的知识点)
     + [2.3.1 binary string](" \l "2.3.1 binary string)
     + [2.3.2 base64](" \l "2.3.2 base64)
     + [2.3.3 DataURLs](" \l "2.3.3 DataURLs)
     + [2.3.4 encodeURIComponent](" \l "2.3.4 encodeURIComponent)
* [三 总结](" \l "三 总结)

## 一 引子

对于该知识点的探讨初衷源自对 DOMString的了解。

对于 DOMString，很多同学可能也没有真正的想过与它相关的定义：

* 为什么是 DOMString 而不是 String？
* 与 String 有什么不同吗？

这些知识点可能在实际的业务中并不会碰到，但是了解它可以使我们更深刻的理解这门语言。   
就像知道 “JavaScript的命名由来” 一样，或者说，就如 JavaScript 中的 null 和 undefined 的渊源一般。

基于以上，我所散发出来思考的内容是：

* JS 数据类型中的 String
* 与 String 相关的知识点

## 二 正文

上面的第一个问题网上搜索一番就会得到答案，因为基础性的知识也比较简单。下面一一解答：

* 在早期，DOM标准刚出的时候，String 是标准的一部分。
* 等同于 String

需要区分注意的是：我们在开发时文件的编码是 utf8，而在 js 运行过程中，字符串的编码是 utf16，也就是说在 js 中字符的编码是 utf16，因为 js 的字符集是 Unicode。

而 DOMString 也就是 String 是 DOM标准中对于 utf16编码字符的接口实现。

### 2.1 关于 String

我有时候会有这样的疑问，创建字符串有很多方式，那这些方式的区别是什么？

字面量的方式

* 1. var str = 'ym'

* 1. // 得到的是字符串值

构造函数

* 1. String ( 1 )

* 1. // 得到的是字符串值

类实例化

* 1. new String ( 123 )

* 1. // 得到的是字符串实例化对象

在 js 中，存在 **基本字符串**和 **字符串对象**，同理，布尔类型和数字类型也同样存在。

字符串字面量 和 直接调用 String 方法的字符串都是基本字符串。

JavaScript会自动将基本字符串转换为字符串对象，只有将基本字符串转化为字符串对象之后才可以使用字符串对象的方法。

当基本字符串需要调用一个字符串对象才有的方法或者查询值的时候(基本字符串是没有这些方法的)，JavaScript 会自动将基本字符串转化为字符串对象并且调用相应的方法或者执行查询。

### 2.2 关于字面量

如果你没有认真思考过为什么会叫“字面量”的话，你应该对 js 的基础还缺乏系统性的认识。

我们会看到很多字面量表示法：

* 字符串字面量
* 布尔类型字面量
* 数字类型字面量
* 数组字面量
* 对象字面量
* 正则表达式字面量

字面量的英文表示： literal，直接翻译过来：“字面意义的、完全按照原文的”，是个 adj形容词。

那也就是说：“你写出来的就是最终的值”，譬如

1. 'ym' // 就是表示一个字符串

1. 123 // 就是表示一个整数

1. true

1. []

1. {}

1. // 等等，都是字面量表示的数据类型

网上搜到的很多关于它的解释是我觉得都是错误的。

### 2.3 与 String 相关的知识点

首先能想到的就是 binary string，其次是 base64。

#### 2.3.1 binary string

上文中我们提到 js 是使用的 unicode 字符集，我们都知道最开始使用的是 ascii 字符集，因为计算机普及字符肯定不够用才统一了字符集。

binary string 也是一种字符集，规范定义是代表 255 以内的字符（这里的255是指ascii编码的值），它的设计之初是用来代表 js 中的二进制数据。

我们知道 ascii 字符集是一个字节长度，想当然，binary string 需要 2 个字节才能表示。

#### 2.3.2 base64

MDN 上的解释是

是一组相似的二进制到文本（binary-to-text）的编码规则

所谓文本其实是指 ascii 字符中的某些，base64 是由 64 个 ascii 字符组成：

* a-z
* A-Z
* 0-9
* + 和 /

所以 base64 应该就是指将 二进制 用这 64 个字符表示出来的一种编码规则。

utf16 中一个字节是 8 位，而在 base64 中一个字节是 6 位，所以用 base64 将二进制转化后的文本的大小是增大的。

在 window 对象中我们会常用到 btoa()和 atob()，其实就是 base64 的互相转化的方法。

便于记忆，方法其实对应 binary to ascii和 ascii to binary。

很显然，当输入的 utf16 字符码位超过了 ascii 字符集的范围，就应该会提示 Range Error了。

需要补充说明的是，由于 base64 的编码规则（长度）必然会导致在转化为 ascii 字符集的时候会有多余的字节位，多余的字节位会使用 =填充。

#### 2.3.3 DataURLs

我们接触到的最常用的场景是，本地预览图片，我们会使用 FileReader 对象，将本地图片转化为 DataURLs 来实现预览。

DataURLs 的构成:

1. data :[< mediatype >][; base64 ],< data >

我们可以反问一下：为什么 DataURLs 能够实现图片预览？

因为 base64 可以用来描述二进制啊，通过 input选择的 File 对象实际是二进制数据的封装。这样一想就对了。

mediatype 就是 mimetype

我们从 FileReader 对象中可以看到常见的几种数据格式：

* FileReader.readAsArrayBuffer()
* FileReader.readAsBinaryString()
* FileReader.readAsDataURL()
* FileReader.readAsText()

关于 ArrayBuffer 就不在本节的讲述范围了，与它跟相关的应该是： **js与二进制数据类型的交互”**。

#### 2.3.4 encodeURIComponent

仔细思考一下，它的作用是用来编码 URI，在 js 中，等同于编码 URL，那么名称上为什么会有 Component呢？

应该是指对 URL 的组成部分单独进行编码，那么是将什么编码转换成什么编码呢？

应该是根据当前 html 的 charset 来定，譬如我们都是使用的 utf8，所以 encodeURIComponent 是将不在 URI 规范中的字符编码成 utf8 直接序列来表示。

注意：此处未深入了解，如有错误请指出。

encodeURIComponent 常见的使用场景：

* url 跳转
* xhr 请求参数

区别在 [mdn](https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Global_Objects/encodeURIComponent)上都有提到，可以了解一下。

## 三 总结

其实，与 String 相关的知识点大多与字符编码有关，不同的数据类型是为了给解决某些问题带来更好的处理办法。当然，这些知识点是零碎的，需要自己花时间系统性的了解一遍，如果说有那么一本书介绍了 js 的发展历程中各个特性或数据类型的由来，我觉得也是非常值得一看的。

基础还是非常重要的。

@2020 讯盟文档

Design by 讯盟 FE