

原来浏览器原生支持JS Base64编码解码

这篇文章发布于 2018年08月4日, 星期六, 01:02, 归类于 [JS API](#)。阅读 16424 次, 今日 37 次 [35 条评论](#)

by zhangxinxu from <https://www.zhangxinxu.com/wordpress/?p=7899>

本文可全文转载, 但需得到原作者书面许可, 同时保留原作者和出处, 摘要引流则随意。

一、基础不牢, 选型糟糕

糟糕的技术选型往往源自自身技术广度不足。

就在数月前, 一个前端HTML字符信息转Base64的需求, 我是毫不犹豫去找了个开源的base64.js。

使用很简单, 浏览器引入该JS文件, 然后Base64编码这样:

```
Base64.encode('zhangxinxu');  
// 返回: 'emhhbmd4aw54dQ=='
```

解码就调用 `decode` 方法, 如下:

```
Base64.decode('emhhbmd4aw54dQ==');  
// 返回: 'zhangxinxu'
```

数据准确, 功能良好。当时弄完还洋洋得意, 以为是个完美的解决。

结果, 今天发现, 尼玛原来浏览器很早就支持了JS Base64加密解密, 而上面这种洋洋得意的做法完全就是瞎子点灯——白费蜡! 根本就不是一个好的技术选型。

由于技术广度掌握不足, 或者说JS基础掌握不牢, 导致浪费了大量时间去找Base64的JS语言库, 学习其API用法, 换来一个冗余完全不需要加载的JS, 现在来看, 真是个糟糕的技术选型。😓

二、原生atob和btoa方法

实际上, 从IE10+浏览器开始, 所有浏览器就原生提供了Base64编码解码方法, 不仅可以用于浏览器环境, Service Worker环境也可以使用。

方法名就是 `atob` 和 `btoa`, 具体语法如下:

1. Base64解码

语法为 (浏览器中):

```
var decodedData = window.atob(encodedData);
```

或者 (浏览器或js Worker线程中):

```
var decodedData = self.atob(encodedData);
```

例如:



```
window.atob('emhhbmd4aw54dQ==');  
// 返回: 'zhangxinxu'
```

眼见为实，您可以狠狠地点击这里：[window.atob解码Base64字符demo](#)

结果如下图：



记住**atob**

atob 这个方法名称乍一看，很奇怪，不知道这个单词什么意思。我们可以理解为 A to B，也就是从A到B。这里的B指的就是Base64吗？哈哈，恭喜你！猜错了！A指的才是Base64，反的，B才是普通字符，普通意思就是low，俗称low B。所以我们这么记忆，Low B，Low B，B表示很Low的普通字符，A才是Base64，和首字母对应关系是反的。

因此，atob表示Base64字符to普通字符，也就是Base64解码。

2. Base64编码

语法为（浏览器中）：

```
var encodedData = window.btoa(stringToEncode);
```

或者（浏览器或js Worker线程中）：

```
var encodedData = self.btoa(stringToEncode);
```

例如：

```
window.btoa('zhangxinxu');  
// 返回: 'emhhbmd4aw54dQ=='
```

眼见为实，您可以狠狠地点击这里：[window.btoa编码字符为Base64 demo](#)

结果如下图：



记住**btoa**方法

btoa 这个方法名称乍一看，很奇怪，不知道这个单词什么意思。我们可以理解为 B to A，也就是从B到A。那B指什么，A指什么呢？和 **atob** 方法一样，B指的是low B普通字符串，A指的是Base64字符。

因此，**btoa** 方法表示low B普通字符to Base64字符，也就是Base64编码。

补充于2018-08-09

有人提到中文Base64数据转换会有报错问题。

错误如下：

```
The string to be encoded contains characters outside of the Latin1 range
```

雾霜月同学提出了解决方法，就是中文先encode转码和decode编码：

```
window.btoa(window.encodeURIComponent('嘻嘻哈哈啦啦啦啦'));  
window.decodeURIComponent(window.atob('JUU1JTk4JUJCJU01JTk4JUJCJU01JTkzJTg4JU01JTkzJTg4JU01JTkzJTg4JU01JTkzJTg4JU01JTk1JUE2JU01JTk1JUE2JU01JTk1JUE2'));
```

我自己搜索了下，可以试试下面这样：

```
btoa(unescape(encodeURIComponent(str)))
```

我用SVG代码测试是OK的。

三、IE8/IE9的polyfill

当下，仍有不少PC项目还需要兼容IE9，所以，我们可以专门针对这些浏览器再引入一段polyfill脚本或者一个JS文件即可。

polyfill JS脚本[戳这里](#)，或者直接[右键这里](#)下载源文件！

实际使用，我们可以借助IE条件注释无缝对接。

也就是HTML中嵌入下面一段代码：

```
<!--[if IE]>  
<script src="./base64-polyfill.js"></script>  
<![endif]-->
```

`[if IE]` 表示所有IE浏览器，由于IE10+浏览器已经放弃了著名的IE条件注释的支持，Chrome等浏览器本身就不支持这个IE私有语法，因此，很天然的，上面一段script引入只在IE9-浏览器下有效。而我们本来就希望只IE8，IE9浏览器引入polyfill，于是正好完美衔接上。

也就是原生支持atob和btoa方法的浏览器认为就是一段无需关心的HTML注释，不支持atob和btoa的IE9及其以下浏览器则会加载我们的base64-polyfill.js，使浏览器也支持 `window.btoa` 和 `window.atob` 这个语法。

眼见为实，您可以狠狠地[点击这里](#)：[btoa/atob Base64编码解码polyfill demo](#)

在Chrome浏览器下，可以看到没有polyfill相关JS的加载：



而IE8原生不支持atob和btoa方法的浏览器也正常编码解码Base64字符了，如下截图：



四、任意文件Base64编码

借助 `FileReader` 对象和 `readAsDataURL` 方法，我们可以把任意的文件转换为Base64 Data-URI。假

设我们的文件对象是 `file`，则转换的JavaScript代码如下：

```
var reader = new FileReader();
reader.onload = function(e) {
    // e.target.result就是该文件的完整Base64 Data-URI
};
reader.readAsDataURL(file);
```

基于此原理，我特意做了个简易小工具，可以得到任意文件的Base64 Data-URI，工具戳这里：[base64.html](#)

把对应的文件拖到工具页面任意位置即可。

例如一个只有'zhangxinxu'字符内容的txt文本拖进去可以得到：

可以看到文件的MIME type也自动识别出来了。然后 `base64,` 后面的关键内容数据会发现和执行 `btoa('zhangxinxu')` 结果一模一样，都是 `'emhhbmd4aW54dQ=='`，肯定必须要一样，不然我反而会怀疑自己哪里是不是弄错了。

此工具我用得很频繁，SVG转Base64，小尺寸图片转Base64，Woff字体转Base64，都是文件往里面一拖就搞定了，简单快捷。

此工具也是IE10+支持。

五、结束语

这两天看到一个新闻，说今年北半球很热：

近日，持续极端高温天气袭击整个北半球。瑞典北部北极圈内的温度一度达到30°C，10余起由高温导致的森林大火烧入北极圈；西伯利亚北部地区，气温也在本月初一度达到32°C，而往年同期，这一地区的平均温度只有10°C；日本继暴雨袭击后又迎热浪考验，高温致死案例频发；欧洲航天局近日公布卫星图片，丹麦西兰岛一处农田去年7月时还是遍地绿色，如今已呈“烤焦”的棕黄色……

加上看到不少人说今年东北很热，东北水库都干了，我就意识到，美的和格力的股票可以买起来了，2季度，3季度财报肯定会很好，因为天一热，买空调的人就会多。

最后，再分享个有技术含量的关于Low B的冷笑话——

一只小蜜蜂啊，飞到花丛中啊，飞呀飞呀，越来越高呀，然后对着下面小蜜蜂说了句：你个low bee！【捂脸】



《CSS世界》签名版独家发售，包邮，可指定寄语，点击显示购买码

（本篇完） // 想要打赏？点击[这里](#)。有话要说？点击[这里](#)。



« 万岁，浏览器原生支持ES6 export和import模块啦！

看，for..in和for..of在那里吵架！»

猜你喜欢

- 了解JS中的全局对象window.self和全局作用域self
- 学习了，CSS中内联SVG图片有比Base64更好的形式
- 小tip: base64:URL背景图片与web页面性能优化
- 基于HTML5的可预览多图片Ajax上传
- 小tips: 纯前端JS读取与解析本地文本类文件
- 了解HTML/HTML5中的download属性
- JS字符串补全方法padStart()和padEnd()简介
- 纯前端实现可传图可字幕台词定制的GIF表情生成器
- js页面滚动时层智能浮动定位实现(jQuery/MooTools)
- 小tip: iframe高度动态自适应
- 直接剪切板粘贴上传图片的前端JS实现

分享到:        { 1 }

标签: atob, base64, btoa, decode, encode, FileReader, polyfill, readAsDataURL, self, window

发表评论（目前35条评论）

名称 (必须)

邮件地址(不会被公开) (必须)

网站

提交评论

1. 匆匆过客说道:

2018年10月4日 22:43

周六凌晨1点多多了，还在撸帖子，蛮拼的嘛！

不过再看到你不知道原生有base64支持，瞬间有一丁点优越感，哈哈！?

atob 意思: ascii to binary atob(“emhhbmd4aW54dQ==”) // “zhangxinxu”

btoa 意思: binary to ascii btoa(“zhangxinxu”) // “emhhbmd4aW54dQ==”

所以直接用中文转base64会报错

另外: 建议用escape处理中文，encodeURIComponent返回会比escape长

[回复](#)



xiaopingbuxiao 说道:

2019年01月1日 14:55

ECMAScript v3 已从标准中删除了 unescape() 函数，并反对使用它，因此应该用 decodeURI() 和 decodeURIComponent() 取而代之。

[回复](#)



2. 布谷说道:

2018年09月26日 03:44

php脚本

```
$mycode=StripStr("你好我是 'www.baidu.com'");
```

```
$base64show=base64_encode($mycode);
```

结果是 5L2g5aW95oiR5pivICYjMDM5O3d3dy5iYWlkS5jb20mlzAzOTs=

转码还原单引号变成转义成 #039;

怎么得到不转义单引号的 base64字符串

[回复](#)



3. 黑客说道:

2018年09月4日 15:16

不错！

[回复](#)



黑客说道:

2018年09月4日 15:16

哈哈

[回复](#)



黑客说道:

2018年09月4日 15:17

循环嵌套

[回复](#)



黑客说道:

2018年09月4日 15:17

在来



4. WangNianyi2001说道:

2018年08月24日 02:11

还有一个思路:

“的原型方法`toDataURL`支持将 canvas 里的内容转成 base64，可以将字符串用`fromCharCode`和位运算截断转换成 image data 格式的`Uint8ClampedArray`，放进 canvas 里再搞成 base64。

不过要注意浏览器会把带透明度的图片有损压缩，所以注意写 image data 的时候不要写到 alpha 值（即`下标&11===0`的那些数据）。

[回复](#)



5. weapon张说道:

2018年08月21日 05:27

我一直纳闷Base64.js是怎么想出来的轮子

[回复](#)



6.

gang说道:

2018年08月20日 11:56

大佬, 向您请教个问题, 在有 contenteditable="true" 的元素上, 或者表单元素上, focus时, 文字下方会有红色波浪号, 如何取消这个波浪号?

回复

gang说道:

2018年08月20日 12:01

在我输入这段文字的时候就会出现这种情况.
This is a piece of wroooooooooooooongly written character.
评论里无法发图, 这段文字复制到评论框就能看到我说的现象,

回复

张鑫旭说道:

2018年08月21日 13:49

关闭拼写检查?

回复

gang说道:

2018年08月22日 08:13

嗯嗯,是的,这个这么实现呢?

回复

gang说道:

2018年08月22日 08:19

试了试 spellcheck = "false" 不管用...

回复



7.

丹丹说道:

2018年08月17日 17:09

大神, 我可以转载你的文章吗? 然后表明出处

回复

张鑫旭说道:

2018年08月17日 23:05

可以。

回复



8.

徐大大说道:

2018年08月6日 21:33

很喜欢大神的博客

回复



9.

姚大宝说道:

2018年08月6日 20:59

老哥, 如何修炼才能像你这样为人风骚代码风骚

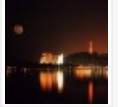
回复



神秘博士说道：
2018年08月8日 18:11
文笔也骚
回复



10. 前端码农说道：
2018年08月6日 12:21



关于svg转base64，个人不建议转base64，采用svg+xml方式，代码比base64小，而且可以直观的改大小、颜色等。

原svg代码：

转换后（可用于img的src或作为背景图 background-image: url()使用）：

```
"data:image/svg+xml,%3csvg xmlns='http://www.w3.org/2000/svg' width='64' height='64'%3e %3cpath fill='none' stroke='%23999' stroke-width='4' stroke-linecap='round' d='M11 32h42M33 52l20-20M33 13l20 19'/%3e %3c/svg%3e"
```

回复

前端码农说道：
2018年08月6日 12:24



原svg代码：

```
<svg xmlns="http://www.w3.org/2000/svg" width="64" height="64">  
<path fill="none" stroke="#999" stroke-width="4" stroke-linecap="round" d="M11 32h42M33 52l20-20M33 13l20 19"/>  
</svg>
```

回复

11. distu说道：
2018年08月6日 10:40



没看懂冷笑话？

回复

12. falseLuffy说道：
2018年08月6日 10:20



抱紧大佬的鱼竿，学习前端技能

回复

13. nakeu说道：
2018年08月6日 08:52



666~ 学习了

回复

14. 电单车说道：
2018年08月5日 14:14



这几天美的格力大跌 没被套吧

回复

15. kongkx说道：
2018年08月5日 08:54



atob，btoa的解释好奇怪。专门去mdn看了下文档，这里做个记录。a是指ASCII，b是指binary。

Base64 is a group of similar binary-to-text encoding schemes that represent binary data in an ASCII string format by translating it into a radix-64 representation

回复

2018年08月4日 23:14



at :1:9

回复

2018年08月9日 09:28



```
window.decodeURIComponent(window.atob('JUUIJTk4JUCJUUIJTk4JUCJUUIJTkzJg4JUUIJTkzJg4JUUIJTkzJg4JUUIJTkzJg4JUUIJTk1JUE2JUUIJTk1JUE2JUUIJTk1JUE2JUUIJTk1JUE2JUUIJTk1JUE2'));
```

回复

2018年08月14日 18:16



```
function b64_to_utf8( str ) {
return decodeURIComponent(escape(window.atob( str )));
}
```

回复

2018年08月4日 21:38

[回复](#)

2018年08月5日 13:56




回复

2018年08月5日 22:36

[回复](#)

2018年08月6日 09:32



Tony说道:

2018年08月7日 18:52

Chrome 67内核, 转中文直接报错

VM340:1 Uncaught DOMException: Failed to execute 'btoa' on 'Window': The string to be encoded contains characters outside of the Latin1 range.

[回复](#)



18. **DeathGhost**说道:

2018年08月4日 10:30

去年看到过这两个, 的确挺方便
以前还专门引入个js工具进行转.

[回复](#)



最新文章

- » [常见的CSS图形绘制合集](#)
- » [粉丝群第1期CSS小测点评与答疑](#)
- » [分享三个纯CSS实现26个英文字母的案例](#)
- » [小tips: 纯CSS实现打字动画效果](#)
- » [CSS/CSS3 box-decoration-break属性简介](#)
- » [CSS :placeholder-shown伪类实现Material Design占位符交互效果](#)
- » [从天猫某活动视频不必要的3次请求说起](#)
- » [CSS vector-effect与SVG stroke描边缩放](#)
- » [CSS ::backdrop伪元素是干嘛用的?](#)
- » [周知: CSS -webkit-伪元素选择器不再导致整行无效](#)

今日热门

- » [常见的CSS图形绘制合集](#) (179)
- » [粉丝群第1期CSS小测点评与答疑](#) (113)
- » [未来必热: SVG Sprite技术介绍](#) (111)
- » [HTML5终极备忘大全 \(图片版+文字版\)](#) (85)
- » [让所有浏览器支持HTML5 video视频标签](#) (83)
- » [Selectivizr-让IE6~8支持CSS3伪类和属性选择器](#) (80)
- » [CSS3下的147个颜色名称及对应颜色值](#) (78)
- » [小tips: 纯CSS实现打字动画效果](#) (73)
- » [写给自己看的display: flex布局教程](#) (70)
- » [分享三个纯CSS实现26个英文字母的案例](#) (70)

今年热议

- » [《CSS世界》女主角诚寻靠谱一起奋斗之人](#) (76)
- » [不借助Echarts等图形框架原生JS快速实现折线图效果](#) (64)
- » [看, for..in和for..of在那里吵架!](#) (60)
- » [是时候好好安利下LuLu UI框架了!](#) (47)
- » [原来浏览器原生支持JS Base64编码解码](#) (35)
- » [妙法攻略: 渐变虚框及边框滚动动画的纯CSS实现](#) (33)
- » [炫酷H5中序列图片视频化播放的高性能实现](#) (31)
- » [CSS scroll-behavior和JS scrollIntoView让页面滚动平滑](#) (30)
- » [windows系统下批量删除OS X系统.DS_Store文件](#) (26)
- » [写给自己看的display: flex布局教程](#) (26)

猜你喜欢

- [了解JS中的全局对象window.self和全局作用域self](#)
- [学习了, CSS中内联SVG图片有比Base64更好的形式](#)
- [小tip: base64:URL背景图片与web页面性能优化](#)
- [基于HTML5的可预览多图片Ajax上传](#)
- [小tips: 纯前端JS读取与解析本地文本类文件](#)
- [了解HTML/HTML5中的download属性](#)
- [JS字符串补全方法padStart\(\)和padEnd\(\)简介](#)

- 纯前端实现可传图可字幕台词定制的GIF表情生成器
- js页面滚动时层智能浮动定位实现(jQuery/MooTools)
- 小tip: iframe高度动态自适应
- 直接剪切板粘贴上传图片的前端JS实现