**[JS处理URL中的中文](https://www.cnblogs.com/Rock-Lee/p/7146154.html)或者特殊字符**

 var parame=encodeURIComponent ("我是 隔壁 滴 老王");

 window.location.href = decodeURIComponent（url）;

区别：encodeURI()不会对本身属于URI的特殊字符进行编码，例如冒号、正斜杠、问号和井字号；而encodeURIComponent()则会对它发现的任何非标准字符进行编码。

适用场景：URL编码

详细场景：

encodeURI：适用于url跳转时。

encodeURIComponent：适用于url作为参数传递时。

注意：当url作为参数传递时如果没有用encodeURIComponent进行编码，往往会造成传递时url中的特殊字符丢失。

<script language="javascript">   
var a = encodeURI("电影");        
alert(a);  
var b = decodeURI(a);  
alert(b)  
</script>

<?php  
$a = urlencode(iconv("gb2312", "UTF-8", "电影")); //等同于javascript encodeURI("电影");  
echo $a;  
$b = iconv("utf-8","gb2312",urldecode("%E7%94%B5%E5%BD%B1")); //等同于javascript decodeURI("%E7%94%B5%E5%BD%B1");  
echo $b;  
?>

如果编码是UTF-8的话就可以直接用urlencode 或 urldecode 转换

btoa和atob是window对象的两个函数，其中btoa是binary to ascii，用于将binary的数据用ascii码表示，即Base64的编码过程，而atob则是ascii to binary，用于将ascii码解析成binary数据，看一个例子：

// Define the string

var string = 'Hello World!';

// Encode the String

var encodedString = btoa(string);

console.log(encodedString); // Outputs: "SGVsbG8gV29ybGQh"

// Decode the String

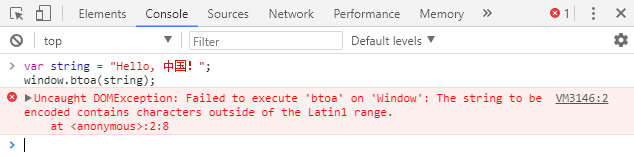
var decodedString = atob(encodedString);

console.log(decodedString); // Outputs: "Hello World!"

可以看到，字符串“Hello World!”已被Base64编码和解码。但是，atob和btoa不能编码Unicode字符：

var string = "Hello, 中国！";

window.btoa(string);



那如何用这种方式支持汉字呢？这里我们可以先将带有非Latin1字符的串先用encodeURIComponent编码：

var string = "Hello, 中国！";

//"SGVsbG8lMkMlMjAlRTQlQjglQUQlRTUlOUIlQkQlRUYlQkMlODE="

var encodedString = btoa(encodeURIComponent(string));

var decodedString = decodeURIComponent(atob(encodedString));

console.log(decodedString); //"Hello, 中国！"

注意，编码的过程是先encodeURI，再编码，那么解码的过程应该是先解码，再decodeURI。

在PHP中有urlencode()、urldecode()、rawurlencode()、rawurldecode(). urlsafe\_b64decode()这些函数来解决网页URL编码解码问题。

$("#mainConfiguration").load(this.url+"/"+this.activeName+"/"+***app\_uuid***+"/"+btoa(encodeURIComponent(***app\_name***)));//js编码

$app\_name = urldecode(urlsafe\_b64decode($app\_name));php解码