Scholity 2 - Stable  acceptations	<ul> <li>querystring (查询字符串)</li> <li>querystring.escape(str)</li> <li>querystring.parse(str[, sep[</li> <li>querystring.stringify(obj[, s</li> <li>querystring.unescape(str)</li> </ul>		
werpstring 保険提供了一些使用工具,用于影影与情報化化以及 查询学特点、它可以通过以下方式的用:	看英文版 / 参与翻译	(查询字符串)	#
はいていています。  「		用工具,用于解析与格式化 URL 查询字符串。它可以通过以下方	式被
### ### ### ### ### ### ### ### ### ##			#
(場外开放、是为了在電影时可以通过的 querystring,escape 就值一个函数未更有端的效果。  Querystring,parse(str[, sep[, eq[, options]]]) 角 表示。	看英文版 / 参与翻译 描于: v0.1.25 ● str 〈String〉 querystring.escape() 方法对约	合定的 str 执行 URL 百分号编码。	
* sep cstrings 用于界定直询字符串中的映值对的子字符串,就认为 1**。 ** eq cstrings 用于界定直询字符串中的映图对字符串,就认为 1**。 ** eq cstrings 用于界定直询字符串中的映图如字符中目分号编码的字符对使用的函数。就 认为 querystrings unescape()。 ** c maskings crumbners 指定要解析的缺nactx数量。就认为 1888。指定为 a 则将稀键数的映 别。  ** particles () 方法维用一个 URL 直询字符中(str.) 解析成一个维值对的集合。  ** P. 有物字符串 "foo-burkable-syzkable-123" 核解析成:  ( foo: 'bur', abit [foo-burkable-syzkable-123" 核解析成:  ( foo-burkable-syzkable-123" foo-burkable-123" foo-burkable-syzkable-123" foo-burkable-syzkable-123" foo-burkable-123" foo	J对外开放 , 是为了在需要时可I	以通过给 querystring.escape 赋值一个函数来重写编码的实现。	
。 maxikeys (number) 指定要條析的健的最大数量。就认为 1990,指定为 6 则修称键数的限 制。  如 如 如 如 如 如 如 如 如 如 如 如 如 如 如 如 如 如 如	<ul><li>sep <string> 用于界定查询</string></li><li>eq <string> 用于界定查询</string></li><li>options <object></object></li></ul>	旬字符串中的键值对的子字符串。默认为 '&'。  字符串中的键与值的子字符串。默认为 '='。	默
### (fost 'bar', sabs: ['vyz', '133']	○ maxKeys <number> 指定制。</number>	E要解析的键的最大数量。默认为 1000 。指定为 o 则移除键数的	限
East : querystring.parse() 方法返回的对象不继承自 JavaScript 的 Object. 这意味着典型的 Object 方法如 Obj.toString()、 obj.hasOunProperty() 等没有被定义且无法使用。  (从信况下,查询字符串中的百分号编码的字符会被认为使用了 UTF-8 编码。如果使用的是另一种字编码。则 decodeURIComponent 选项需要被指定,如以下例子:  // 但是 gMoDrendeURIComponent 选项后作。	<pre>{   foo: 'bar',   abc: ['xyz', '123']</pre>	c=xyz&abc=123' 被解析成:	
// 世段 gbKoecodeURIComponent 函数で存在。 querystring.parse('w-MoKDBKCEKCASfoo-bar', null, null, { decodeURIComponent: gbkDecodeURIComponent })  querystring.stringify(obj[, sep[, eq[, pptions]]])  和	注意: querystring.parse() 方 Object 方法如 obj.toString()	、obj.hasOwnProperty() 等没有被定义且无法使用。	
querystring.stringify(obj[, sep[, eq[, #Dptions]]])  (AUXID / Separate  * obj (objecto 要序列化成一个 URL 查询字符串的对象。  * obj (objecto 要序列化成一个 URL 查询字符串的对象。  * sep (string) 用于界定查询字符串中的键与值的子字符串。默认为 'a'。  * options  * oncodeURIComponent (Function) 当把对 URL 不安全的字符转换成查询字符串中的百分号编码时使用的函数。默认为 querystring,escape()。  uerystring.stringify() 方法通过遍历对象的自有属性,从一个给定的 obj 产生一个 URL 查询字符串。  (以信用 'foo-bankhaz-quxkhaz-quxkkaz-quxxkcorge='  querystring.stringify({foo: 'ban', baz: 'qux', 'quux'], corge: ''})  // 发创 'foo-bankhaz-quxkkaz-quxxkcorge='  querystring.stringify({foo: 'ban', baz: 'qux', ';', ':')  // 发创 'foo-bankhaz-quxkkaz-quxxkcorge='  querystring.stringify({foo: 'ban', baz: 'qux'}, ';', ':')  // 发创 'foo-bankhaz-quxkkaz-quxxkcorge='  querystring.stringify({foo: 'ban', baz: 'qux'}, ';', ':')  // 发创 gbkEncodeURIComponent 所表已存在。  querystring.stringify({w: '中文', foo: 'ban'}, null, null, (encodeURIComponent: gbkEncodeURIComponent })  querystring.unescape() 方法对给定的 str 上的 URL 百分号编码的字符执行解码。  uerystring.unescape() 方法是供 querystring.parse() 使用的,且通常不被直接使用。它之所以分外开放,是为了在需要时可以通过给 querystring.unescape 账值一个函数来重写解码的实现。	// 假设 gbkDecodeURIComponen querystring.parse('w=%D6%D0%	nt 函数已存在。 %CE%C4&foo=bar', null, null,	
***  ** ** ** ** ** ** ** ** ** ** ** *	querystring.str options]]])		#
。 encodeURIComponent 〈Function〉当把对 URL 不安全的字符转换成查询字符串中的百分号编码的使用的函数。默认为 querystring.escape()。 puerystring.stringify() 方法通过遍历对象的自有属性,从一个给定的 obj 产生一个 URL 查询字语。  可子:  querystring.stringify({ foo: 'bar', baz: ['qux', 'quux'], corge: '' })  // 返回 'foo-bar&baz=qux&baz=quux&corge='  querystring.stringify({foo: 'bar', baz: 'qux'}, ';', ':')  // 返回 'foo:bar;baz:qux'    沈认情况下,查询字符串中需要百分号编码的字符会作为 UTF-8 被编码。如果需要的是另一种编码,则encodeURIComponent 选项需要被指定,如以下例子:  // 赘议 gbkEncodeURIComponent 所数已存在。 querystring.stringify({ w: '中文', foo: 'bar' }, null, null, { encodeURIComponent: gbkEncodeURIComponent }) <b>Querystring.unescape(</b> **  **  **  **  **  **  **  **  **	• sep <string> 用于界定查询</string>	旬字符串中的键值对的子字符串。默认为 '&'。	
querystring.stringify({ foo: 'bar', baz: ['qux', 'quux'], corge: '' })  // 返回 'foo=bar&baz=qux&baz=qux&corge='  querystring.stringify({foo: 'bar', baz: 'qux'}, ';', ':')  // 返回 'foo:bar;baz:qux'	<ul><li>options</li><li>encodeURIComponent &lt;</li><li>码时使用的函数。默认:</li><li>uerystring.stringify() 方法</li></ul>	Function> 当把对 URL 不安全的字符转换成查询字符串中的百分 为 querystring.escape()。	
// 返回 'foo:bar;baz:qux'			
querystring.stringify({ w: '中文', foo: 'bar' }, null, null, { encodeURIComponent: gbkEncodeURIComponent }}  querystring.unescape(str)  着英文版 / 参与翻译  r增于: v0.1.25  • str <string> querystring.unescape() 方法对给定的 str 上的 URL 百分号编码的字符执行解码。 querystring.unescape() 方法是供 querystring.parse() 使用的,且通常不被直接使用。 它之所以对外开放,是为了在需要时可以通过给 querystring.unescape 赋值一个函数来重写解码的实现。</string>	// 返回 'foo:bar;baz:qux' <b>状认情况下,查询字符串中需要</b> [	百分号编码的字符会作为 UTF-8 被编码。 如果需要的是另一种编	码,
中では、「神子: v0.1.25  * str 〈String〉 「地子: v0.1.25  * str 〈String〉 「uerystring.unescape() 方法对给定的 str 上的 URL 百分号编码的字符执行解码。 「uerystring.unescape() 方法是供 querystring.parse() 使用的,且通常不被直接使用。它之所以对外开放,是为了在需要时可以通过给 querystring.unescape 赋值一个函数来重写解码的实现。	// 假设 gbkEncodeURIComponen querystring.stringify({ w: '	nt 函数已存在。 '中文', foo: 'bar' }, null, null,	
• str <string> uerystring.unescape() 方法对给定的 str 上的 URL 百分号编码的字符执行解码。 uerystring.unescape() 方法是供 querystring.parse() 使用的,且通常不被直接使用。它之所以 为开放,是为了在需要时可以通过给 querystring.unescape 赋值一个函数来重写解码的实现。</string>	querystring.un		#
uerystring.unescape() 方法是供 querystring.parse() 使用的 , 且通常不被直接使用。 它之所以 付外开放 , 是为了在需要时可以通过给 querystring.unescape 赋值一个函数来重写解码的实现。	• str <string></string>	付给定的 str 上的 URL 百分号编码的字符执行解码。	