

百分号编码与 encodeURIComponent

`encodeURIComponent``UTF-8``URI``URL``GB2312`

百分号编码、`encodeURIComponent`、URL encode 其实都是指在 URL 参数中转义任意字符。在百分号编码中，每个字符被编码成3个字符，包括第一个起始的 `%`，以及接着的两个字符表示 16 进制的一个字节。比如空格字符（`00100000`）编码后的结果为 `%20`。

TL;DR

- URL 是 URI 的一种形式。
- URI 的合法字符包括18个保留字符和66个非保留字符
- 在 `application/x-www-form-urlencoded` 中空格编码为 `+`
- `decodeURIComponent()` 对非 UTF-8 的源会抛出 `URIError: URI malformed`

URI 与 URL

URI（统一资源标识）定义在 [RFC 3986](#) 中，[URL](#)（统一资源地址）是 URI 的一种特殊形式，提供了资源的网络位置。URL 最初的标准 [RFC 1738](#) 已经废弃，最新的 [URL Living Standard](#) 目前由 WHATWG 维护。其中定义了 URL, 域名, IP 地址, `application/x-www-form-urlencoded` 格式，以及相应的 API。

A URI can be further classified as a locator, a name, or both. The term “Uniform Resource Locator” (URL) refers to the subset of URIs that, in addition to identifying a resource, provide a means of locating the resource by describing its primary access mechanism (e.g., its network “location”). [1.1.3. URI, URL, and URN](#), RFC 3986

字符集

URI 中合法的字符包括两种：保留字（18个）和非保留字（66个）。其中保留字又分为两部分：组件定界符（gen-delims）和子组件定界符（sub-delims）。

```
reserved      = gen-delims / sub-delims
gen-delims    = ":" / "/" / "?" / "#" / "[" / "]" / "@"
sub-delims    = "!" / "$" / "&" / "'" / "(" / ")" / "*" / "+" / "," / ";" / "="
unreserved    = ALPHA / DIGIT / "-" / "." / "_" / "~"
```

URI 每一部分的生成式可参考：<https://tools.ietf.org/html/rfc3986#appendix-A> 本文关心的是URL 的参数部分，其生成式为：

```
query = *( unreserved / "%" HEXDIG HEXDIG / sub-delims / ":" / "@" / "/" / "?" )
```

可见 URL 参数部分除了非保留字、子组件定界符、百分号编码之外，只允许`:`，`@`，`/`，`?`。

编解码

百分号编码在如下几种情况都会发生：

- 在浏览器地址栏键入了需要编码的字符并回车
- 通过 `pushState()` 等 API 操作 URL
- 调用 `encodeURIComponent()` JavaScript API
- 提交默认编码（`application/x-www-form-urlencoded`）的表单

上述几种情况都会触发 URL 编码，但算法各有不同。其中 `application/x-www-form-urlencoded` 表单编码定义在 [URL 标准](#) 中：

编码时字母、数字、`*`、`-`、`.`、`_` 不进行编码，空格则编码成 `+`。这就是为什么 Google 搜索时空格在 URL 上会变成 `+` 而不是 `%20`。

此外，URI 标准推荐不要对不必编码的字符进行编码，在更容易阅读的同时也更加一致。

For consistency, percent-encoded octets in the ranges of ALPHA (%41-%5A and %61-%7A), DIGIT (%30-%39), hyphen (%2D), period (%2E), underscore (%5F), or tilde (%7E) should not be created by URI producers and, when found in a URI, should be decoded to their corresponding unreserved characters by URI normalizers. – [2. Characters, RFC 3986](#)

encodeURIComponent 编码异常

`encodeURIComponent()` 对不合法的百分号编码会抛出 `"URI malformed"` 异常。这通常是因为百分号编码前的字节流不是 UTF-8 编码。根据 [URI 标准](#) GB2312 直接进行百分号编码得到的 URL 是不合法的：

When a new URI scheme defines a component that represents textual data consisting of characters from the Universal Character Set [UCS](#), the data should first be encoded as octets according to the UTF-8 character encoding [STD63](#); then only those octets that do not correspond to characters in the unreserved set should be percent- encoded. – 2.5. Identifying Data, [Uniform Resource Identifier \(URI\): Generic Syntax](#)

下面是一个简单的复现过程：

分别从 utf-8 和 gbk 编码得到 Hex：

```
$ echo -n 你好 | xxd -p
e4bda0e5a5bd
$ echo -n 你好 | iconv -f utf8 -t gb2312 | xxd -p
c4e3bac3
```

解码对应的百分号编码：

```
$ node
> decodeURIComponent('%e4%bd%a0%e5%a5%bd')
'你好'
> decodeURIComponent('%c4%e3%ba%c3')
Uncaught URIError: URI malformed
```

转载请注明来源：<http://harttle.land/2017/05/23/percentage-encoding.html> 欢迎对文章中的引用来源进行考证，欢迎指出任何有错误或不够清晰的表达。可以在下面评论区评论（可能需要在能访问 disqus 服务的网络），也可以邮件至 harttle@harttle.com。

上一篇：[内联和外链脚本的性能实验](#)

下一篇：[为 JS 代码生成 Markdown 文档](#)