URL Encoding HTML Attributes

A tutorial on how to use percent escapes to URL encode the value of an attribute, such as the <a class="nobr" href="../tags/a-tag/#href-attribute">&lt;a href&gt; attribute</a> HTML-5.com is a great guide for web developers. TV Series & Actors and Actresses. Follow TV Series and HTML 5 on Google+.

HTML-5.com ➢ itemscopehttp://data-vocabulary.org/Breadcrumb<span itemprop="title">HTML 5</span> ➢ itemscopehttp://data-vocabulary.org/Breadcrumb<span itemprop="title">HTML Tutorials</span> ➢ itemscopehttp://data-vocabulary.org/Breadcrumb **URL Encoding HTML Attributes**

URL Encoding

(Some links on this page take you to details in the HTML Tag Reference. Bookmark this page in your Favorites so you can come back to it later.) See also JavaScript escape vs. encode for details on `encodeURI`, `escape` and `encodeURIComponent`.

A number of characters are considered <dfn>unsafe</dfn> to use in a URL because they have special meanings in URLs for various reasons<sup>1</sup>.

Any attribute with a value that is a URL must be URL-encoded, including:

- <a href>
- <area href>
- <audio src>
- <base href>
- <blockquote cite>
- <button formaction>
- <command icon>
- <del cite>
- <embed src>
- <form action>
- <iframe src>
- <html manifest>
- <img src> and <img usemap>
- <input src> and <input formaction>
- <ins cite>
- <link href>
- <meta content> when the value contains a URI reference
- <q cite>
- <object data> and <object usemap>
- <script src>
- <source src>

- <video poster> and <video src>

In addition, when a <form> specifies `method="GET"`, the user input returned in the query string will be URL-encoded.

back to top

Percent Escape Codes For Special Characters

	2*x*	3*x*	4*x*	5*x*	6*x*	7*x*
*x*0	*space*	0	@	P	`	p
%20	%40	%60				
*x*1	!	1	A	Q	a	q
*x*2	"	2	B	R	b	r
%22						
*x*3	#	3	C	S	c	s
%23						
*x*4	$	4	D	T	d	t
*x*5	%	5	E	U	e	u
%25						
*x*6	&	6	F	V	f	v
%26						
*x*7	'	7	G	W	g	w
*x*8	(	8	H	X	h	x
*x*9	)	9	I	Y	i	y
*x*A	*	:	J	Z	j	z
%3A						
*x*B	+	;	K	[	k	{
%2B	%3B	%5B	%7B			
*x*C	,	<	L	\	l	|
%3C	%5C	%7C				
*x*D	-	=	M	]	m	}
%3D	%5D	%7D				
*x*E	.	>	N	^	n	~
%3E	%5E	%7E				
*x*F	/	?	O	_	o	#
%2F	%3F	%7F				

Note that, in addition to the special characters enumerated in RFC 1736 and RFC 3986 (below), the following characters are considered <dfn>unsafe</dfn> based on the statement that <q>Other characters are unsafe because gateways and other transport agents are known to sometimes modify such characters</q>:

plus sign ("+")

Many browsers, web servers, application servers and URL decoding functions such as PHP urldecode and methods such as java.net.URLDecoder decode convert a plus sign into a space character. Even when the plus sign is not decoded, they will encode a plus sign as "`%2B`" to avoid ambiguity.

<blockquote id="fn1">
The space character is unsafe because significant spaces may disappear and insignificant spaces may be introduced when URLs are transcribed or typeset or subjected to the treatment of word-processing programs. The characters "<" and ">" are unsafe because they are used as the delimiters around URLs in free text; the quote mark (""") is used to delimit URLs in some systems. The character "#" is unsafe and should always be encoded because it is used in World Wide Web and in other systems to delimit a URL from a fragment/anchor identifier that might follow it. The character "%" is unsafe because it is used for encodings of other characters. Other characters are unsafe because gateways and other transport agents are known to sometimes modify such characters. These characters are "{", "}", "|", "\", "^", "~", "[", "]", and "`".

All unsafe characters must always be encoded within a URL. For example, the character "#" must be encoded within URLs even in systems that do not normally deal with fragment or anchor identifiers, so that if the URL is copied into another system that does use them, it will not be necessary to change the URL encoding.

Many URL schemes reserve certain characters for a special meaning: their appearance in the scheme-specific part of the URL has a designated semantics. If the character corresponding to an octet is reserved in a scheme, the octet must be encoded. The characters ";", "/", "?", ":", "@", "=" and "&" are the characters which may be reserved for special meaning within a scheme. No other characters may be reserved within a scheme.

Usually a URL has the same interpretation when an octet is represented by a character and when it encoded. However, this is not true for reserved characters: encoding a character reserved for a particular scheme may change the semantics of a URL.

Thus, only alphanumerics, the special characters "$-_.+!*'(),", and reserved characters used for their reserved purposes may be used unencoded within a URL.

<cite>RFC 1738</cite> </blockquote> <blockquote>
If two URIs differ only in the case of hexadecimal digits used in percent-encoded octets, they are equivalent. For consistency, URI producers and normalizers should use uppercase hexadecimal digits for all percent- encodings.

For all URIs, the hexadecimal digits within a percent-encoding triplet (e.g., "%3a" versus "%3A") are case-insensitive and therefore should be normalized to use uppercase letters for the digits A-F.

<cite>RFC 3986</cite> </blockquote>

back to top

THE END