HTML Namespaces and Namespace Declarations (xmlns=)

HTML-5.com is an HTML User's Guide and quick reference of HTML elements and attributes for web developers who code HTML web pages, not only for HTML 5 but for HTML coding in general, with demos and examples of HTML code plus a cheat sheet 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 **HTML Namespaces and Namespace Declarations (xmlns=)**

HTML Namespace Definitions

Definition: HTML Namespace(s)

(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.)

A <dfn>namespace</dfn> is a property associated with the name of some object, such as a node in an HTML document, that distinguishes the type of object represented by the name from other types of objects with the same name in other namespaces. For example, in

```
<head><title>One Of My Web Pages</title></head>
```

`title` is the name of an element (the **title** element) in HTML while in

```
<a href="..." title="One of my links" ...>
```

`title` is the name of a title attribute on an HTML <a> tag.

The <dfn>HTML namespaces</dfn> (plural) in general are the collection of various namespaces in HTML code. The <dfn>HTML namespace</dfn> (singular) itself is the one associated with the namespace URI `http://www.w3.org/1999/xhtml`.

On this site, the notation `<a href>` refers to the `href` attribute in the `html:a` element namespace, where `html:` is the namespace prefix for the HTML namespace.

Why are there namespaces in HTML?

Names that are spelled the same and have the same namespace cannot be distinguished from one another without knowing the context in which they appear. For example, in the RSS version 2.0 and OPML version 2.0 code below, the three `<title>` elements with the same name in the same namespace partition (the element partition without a namespace URI in this case) all identify the same type of object:

```
<?xml version="1.0" encoding="UTF-8"?>
<rss version="2.0">
   <channel>
      <title>Feed Title</title>
      ...
      <item>
         <title>Item Title</title>
         ...
      </item>
   </channel>
```

```
</rss>

<?xml version="1.0" encoding="UTF-8"?>
<opml version="1.0">
   <head>
      <title>My OPML</title>
   </head>
   <body>
      ...
   </body>
</opml>
```

In order to differentiate the `title` elements above you would have to know that the first one was a descendant of an `rss` element and `channel` element; the second was a descendant of an `rss` element, `channel` element and `item` element; and the third was a descendant of an `opml` element and a `head` element.

In addition, without namespaces there might be elements with the same element name that have different meanings or content depending on where they are used. Their different models would have to be combined to create a single definition (XML Schema definition or DTD) of the element. This might work if all meanings were determined by the same entity, but it would likely be a major issue if different standards bodies or industry groups were involved.

This type of confusion in XML content without namespaces makes it very difficult to aggregate that content into other types of content, even if the other content formats do support namespaces. For example, if portions of the content above was included in an HTML document (without overriding the namespace with `xmlns=""`), the `title` elements would be treated as a valid HTML titles, the `head` element and `body` element would be treated as duplicates of their HTML counterparts and most of the other elements would be treated as invalid HTML code.

Note that the HTML `title` element would not be confused with the `title` elements shown above, since its name is in a different namespace partition associated with the HTML namespace URI.

back to top

Namespace Partitions

For each namespace, including the one with no namespace URI, there are a number of namespace partitions:

- an <dfn>element name</dfn> partition for the names of element types
- for each element type, a separate <dfn>element attribute name</dfn> partition for the names of local attributes specific to that element type
- a <dfn>global attribute name</dfn> partition for the names of global attributes that can be used on any type of element, regardless of which namespace contains the element name

Since those partitions exist for each namespace, the collection of namespace partitions can be viewed as a two-dimensional table:

Namespace URI	Prefix	Element Names	Element Attribute Names	Global Attribute Names

<span style="font-weight: normal">(in `xmlns` attribute)</span>				
none (`xmlns=""`)	*none*	`title` (in RSS, OPML, etc.)	`<rss version>`	HTML global attributes such as `id` and `style`
			`<opml version>`	
http://www.w3.org/1999/xhtml	*default <span class="nobr">(no prefix)</span>*	`<a>` `<link>` `<form>`	`<a href>` `<a rel>`	
			`<link href>` `<link rel>`	
			`<form action>` `<form method>`	
http://www.w3.org/XML/1998/namespace	`xml:` *`reserved`*			`xml:lang` `xml:space`
http://www.w3.org/2000/xmlns/	`xmlns:` *`reserved`*			default namespace (`xmlns=`) and namespace prefix declarations (`xmlns:`*`xxx`*`=`)
http://www.w3.org/2001/XMLSchema	xsd: *for the <span class="nobr">HTML examples</span> on this site*	`<element>` `<attribute>`	`<element name>` `<element ref>`	
			`<attribute name>` `<attribute ref>`	
http://www.w3.org/2001/XMLSchema-instance	xsi: *for the <span class="nobr">HTML examples</span> on this site*			`xsi:type`

http://www.w3.org/1998/Math/MathML	math: *for the <span class="nobr">HTML examples</span> on this site*	`<math>` `<mo>`	`<math mode>` `<mo form>`	
http://www.w3.org/1999/02/22-rdf-syntax-ns#	rdf: *for the <span class="nobr">HTML examples</span> on this site*	`rdf:RDF` `rdf:Description` `rdf:List`		`rdf:about`
http://www.w3.org/2000/01/rdf-schema#	rdfs: *for the <span class="nobr">HTML examples</span> on this site*	`rdfs:Datatype`		
http://www.w3.org/2000/svg	svg: *for the <span class="nobr">HTML examples</span> on this site*	`<svg>` `<rect>`	`<svg viewBox>` `<rect width>` `<rect height>`	

When designing schema for one or more sets of objects, you should ensure that no namespace partition (as shown in the table above) contains the same name for two different object types. Otherwise, it will be difficult to create a definition (XML Schema definition or DTD) that properly describes the schema. For example, in the table above the `title` element in the "no namespace URL" partition violates this requirement because it is used for RSS channel titles, RSS item titles and OPMN titles.

back to top

`http://www.w3.org/1999/xhtml` and other HTML Namespaces

The namespace URIs in HTML documents

Commonly used namespaces in HTML include:

http://www.w3.org/XML/1998/namespace

the URI for the XML base namespace; associated with the implicitly declared reserved prefix `xml:`

http://www.w3.org/2000/xmlns/

the namespace URI for XML namespace declarations; associated with the implicitly declared reserved prefix `xmlns:`

http://www.w3.org/1999/xhtml

the URI for the HTML namespace, the same namespace URI defined by the 2000-2010 Recommendations from the W3C HTML Working Group; implicitly declared as the default namespace for unprefixed tag names when document is being parsed as HTML, but explicitly coding it is recommended for cases when document is being parsed as xHTML or XML

http://www.w3.org/1998/Math/MathML

the MathML namespace for MathML tags

http://www.w3.org/1999/02/22-rdf-syntax-ns#

the RDF namespace for RDF tags

http://www.w3.org/2000/01/rdf-schema#

the RDF schema namespace

http://www.w3.org/2000/svg

the SVG namespace for SVG tags

http://www.w3.org/1999/xlink

the XLink namespace

http://www.w3.org/2001/XMLSchema

the namespace for <dfn>XML Schema Definitions</dfn>, which are XML documents that replace non-XML DTDs

http://www.w3.org/2001/XMLSchema-instance

the namespace for XML Schema instance documents, which can be used to specify whether the data for a field is binary (possibly encrypted) or plain text:

```
 <span id="masked-credit-card-number" xsi:type="xsd:string">4321 **** ****
➥ 8765</span>
 <span id="encrypted-credit-card-number" xsi:type="xsd:base64Binary">BAM0NCo
➥ mFzC2TOsmRzW0NTueQU==</span>
```

http://www.w3.org/1999/XSL/Transform

the namespace for XSL style sheets, which is not used in HTML documents themselves, but is used in the style sheet documents that are referenced by the xml-stylesheet instruction and contain XSLT elements intermingled with HTML code.

back to top

`xmlns` namespace declarations in the <html> tag

How to declare namespaces in an HTML document

The easiest way to declare namespaces is by putting the `xmlns` attributes in the top element of the XML document, which in this case is the <html> tag:

```
 <html xmlns="http://www.w3.org/1999/xhtml"
    xmlns:mathml="http://www.w3.org/1998/Math/MathML"
    xmlns:svg="http://www.w3.org/2000/svg"
```

In HTML 5, all elements (HTML tags) are automatically considered to be qualified with the HTML 5 namespace, which makes the declaration of the HTML namespace optional, but this only works with HTML browsers *with HTML 5 support* that are parsing the web page as an HTML <span class="underline">version 5</span> document. It's best to continue coding the `xmlns="http://www.w3.org/1999/xhtml"` `xmlns` attribute explicitly to provide backward compatibility with browsers lacking full <span class="nobr">HTML 5</span> support and other types of programs that may be processing HTML documents, such as RSS feed readers and generic XML parsers - otherwise all of the HTML tags will appear to be in the partition with no namespace URI.

back to top

THE END