Review C: Body elements

Table of contents

1 Body elements

The body elements support the most common types of content for topics: paragraphs, lists, phrases, figures, and other common document components.

1.1 <alt>

Alternate text is a textual description of an image. Systems often render the alternate text when the reader is using assistive technology or the image cannot be rendered.

Attributes

The following attributes are available on this element: universal attributes.

Example

The following code sample shows how alternate text is associated with an image of a marketing banner:

```
<image href="newCampaign.jpg">
  <alt>Marketing banner for new product campaign</alt>
</image>
```

1.2 <cite>

A citation is the name or the title of a bibliographic resource, for example, a document, online article, or instructional video.

Attributes

The following attributes are available on this element: universal attributes and @keyref.

Example

The following code sample shows how the <cite> element can be used to mark up the title of an article:

```
<p>The online article <cite>Specialization in the Darwin Information Typing
Architecture</cite> provides a detailed explanation of how to define new
topic types.</p>
```

1.3 <dd>

The definition description is the definition for an item in a definition list entry.

Attributes

The following attributes are available on this element: universal attributes.

Example

See <dl> (5).

1.4 <ddhd>

A definition heading is an optional heading or title for descriptions or definitions in a definition list.

Attributes

The following attributes are available on this element: universal attributes.

Example

See `<dlhead>` (6).

1.5 <desc>

A description is a statement that describes or contains additional information about an object.

Usage information

The following list outlines common uses of the `<desc>` element:

<table> and <fig>
    Provides more information than can be contained in the title

<xref> and <link>
    Provides a description of the target

<object>
    Provides alternate content to use when the context does not permit displaying the object

Rendering expectations

001 (32)   When used in conjunction with `<fig>` or `<table>` elements, processors **SHOULD** consider the content of `<desc>` elements to be part of the content flow.

When used in conjunction with `<xref>` or `<link>` elements, processors often render the content of `<desc>` elements as hover help or other form of link preview.

Attributes

The following attributes are available on this element: universal attributes.

Examples

This section contains examples of how the `<desc>` element can be used.

Figure 1: Description of a figure

In the following code sample, the `<figure>` element contains a reference to an image of a famous painting by Leonardo da Vinci. The `<title>` element provides the name of the painting, while the `<desc>` element contains information about when the portrait is thought to have been painted.

```
<fig>
<title>Mona Lisa</title>
<desc>Circa 1503-06, perhaps continuing until 1517</desc>
<image href="mona-lisa.jpg">
 <alt>Photograph of Mona Lisa painting</alt>
```

```
</image>
</fig>
```

Figure 2: Description of a cross reference

In the following code sample, the `<link>` element contains a `<desc>` element. Some processors might render the content of the `<desc>` element as hover help.

```
<link keyref="dita-13-02">
   <linktext>DITA 1.3 Errata 02</linktext>
   <desc>Final errata version of DITA 1.3, published 19 June 2018</desc>
</link>
```

1.6 <div>

A division is a grouping of contiguous content within a topic. There is no additional semantic meaning.

Usage information

The `<div>` element is useful primarily for reuse and as a specialization base.

Attributes

The following attributes are available on this element: universal attributes.

Example

In the following code sample, a `<div>` element is used to organize several elements together so that they can be referenced by `@conref` or `@conkeyref`:

```
...
<div id="div-01">
 <p>The first paragraph</p>
 <p>The second paragraph</p>
 <note>This is a note</note>
</div>
...
```

Without using a `<div>` element, the content could not be grouped for content referencing since the start and end elements are of different types.

1.7 <dl>

A definition list is a list of items and their corresponding definitions.

Attributes

The following attributes are available on this element: universal attributes, `@compact`, and `@spectitle`.

Example

The following code sample shows how a definition list can be used to describe the message levels that are generated by a monitoring application. The `@compact` attribute instructs processors to tighten the vertical spacing.

```
<dl compact="yes">
   <dlentry>
     <dt>Warning</dt>
     <dd>Problems were detected, but the software will continue to monitor activity.</dd>
```

```
    </dlentry>
    <dlentry>
      <dt>Error</dt>
      <dd>Problems were detected, and the software is in danger of shutting down.</dd>
    </dlentry>
    <dlentry>
      <dt>Severe</dt>
      <dd>Monitoring activity has ceased.</dd>
    </dlentry>
 </dl>
```

1.8 <dlentry>

A definition list entry is a group within a definition list. It contains an item and its definitions.

Attributes

The following attributes are available on this element: universal attributes.

Example

See <dl> (5).

1.9 <dlhead>

A definition list heading is a group that contains a heading for items and a heading for definitions within the list.

Attributes

The following attributes are available on this element: universal attributes.

Example

The following code sample shows a definition list with a header:

```
<dl>
 <dlhead>
  <dthd>Image selection</dthd>
  <ddhd>Resulting information</ddhd>
 </dlhead>
 <dlentry>
  <dt>File Type</dt>
  <dd>The file extension of the image</dd>
 </dlentry>
 <dlentry>
  <dt>Image class</dt>
  <dd>Whether the image is raster, vector, or 3D</dd>
 </dlentry>
 <dlentry>
  <dt>Fonts</dt>
  <dd>Names of the fonts contained within a vector image</dd>
 </dlentry>
</dl>
```

Rendering of definition lists will vary by application and by display format.

1.10 <draft-comment>

A draft comment is content that is intended for review and discussion, such as questions, comments, and notes to reviewers. This content is not intended to be included in production output.

Rendering expectations

002 (32) | By default, processors **SHOULD NOT** render `<draft-comment>` elements. Processors **SHOULD** provide a mechanism that causes the content of the `<draft-comment>` element to be rendered in draft output only.

Attributes

The following attributes are available on this element: universal attributes and the attributes defined below.

For this element, the `@translate` attribute has a default value of "no".

@author
Designates the originator of the draft comment.

@disposition
Specifies the status of the draft comment.

@time
Specifies when the draft comment was created.

Example

The following code samples shows how a content developer can use a `<draft-comment>` element to pose a question to reviewers:

```
<draft-comment author="EBP" time="23 May 2017">
    <p>Where's the usage information for this section?</p>
</draft-comment>
```

Processors might render the information from the highlighted attributes at viewing or publishing time. Authors might use the value of the `@disposition` attribute to track the work that remains to be done on a content collection.

1.11 <dt>

A definition term is the item that is defined in a definition list entry.

Attributes

The following attributes are available on this element: universal attributes.

Example
See `<dl>` (5).

1.12 <dthd>

A definition term heading is an optional heading or title for the items in a definition list.

Attributes

The following attributes are available on this element: universal attributes.

Example

See `<dlhead>` (6).

1.13 <example>

An example illustrates the subject of the topic or a portion of the topic.

Usage information

> **Comment by Kristen J Eberlein on 05 December 2021**
>
> Zoe has asked that we revise the content of "Usage information" to be less programming related. Her comment:
>
>> This usage is super programming related. Isn't part of the reason we expanded where example can go is to be usable wherever for whatever? If you say for the usage I should have a code sample, I'm going to think I can't use this for other types of examples.
>
> I think we also need to be clear that example can be part of a section, not just an element that is peer to a section.T This is a change from DITA 1.3, and while we don't want to call it out as such, the "Usage information" should make it clear where example can go in a topic.

Use `<example>` to contain both sample code (or similar artifacts) and the discussion that illustrates the sample. For example, a topic about programming code could use the `<example>` element to contain both the sample code and the text that describes the code.

Attributes

The following attributes are available on this element: universal attributes and `@spectitle`.

Example

The following code sample shows an `<example>` element that contains a code block and a textual explanation of it:

```
<section id="AddingRecord">
   <title>ADD</title>
   <p>New database records are created using the <cmdname>ADD</cmdname> command.</p>
   <example>
      <p>The following example illustrates the creation of a new record.
All parameter settings are strictly optional.</p>
      <codeblock>01 OPTIONS ABC,ADD,DEF,HIJK,LMNO,AOW=25000,HF=2</codeblock>
   </example>
</section>
```

1.14 <fallback>

Fallback content is content to be presented when multimedia objects or other referenced content cannot be rendered.

Processing expectations

The contents of this element are displayed only when the media that is referenced by the containing element cannot be displayed or viewed.

Attributes

The following attributes are available on this element: universal attributes.

Example

See `<audio>` and `<video>`.

1.15 <fig>

A figure is a container for a variety of objects, including artwork, images, code samples, equations, and tables.

Usage information

A `<fig>` element enables associating other elements, such as a title or description, with the contents of the `<fig>` element.

Attributes

The following attributes are available on this element: universal attributes, display attributes, and `@spectitle`.

Example

The following code sample shows how a `<fig>` element can associate a title and a description with an image:

```
<fig>
    <title>The handshake</title>
    <desc>This image shows two hands clasped in a formal, business-like handshake.</desc>
    <image href="59j0p66.jpg">
        <alt>A handshake</alt>
    </image>
</fig>
```

1.16 <figgroup>

A figure group is a grouping of segments within a figure.

Usage information

The `<figgroup>` element is useful primarily as a base for complex specializations, such as nestable groups of syntax within a syntax diagram. The `<figgroup>` element can nest. It can also contain multiple cross-references, footnotes, and keywords.

Attributes

The following attributes are available on this element: universal attributes.

Example

For the most part, `<figgroup>` is intended to be used as a base for specialization. This example uses it directly for purposes of illustration.

The following code sample shows how the `<figgroup>` can group content with associated metadata:

```
<fig>
  <title>Sample complex figure</title>
  <figgroup>
    <data name="MetaItem" value="13"/>
    <data name="MetaThing" value="31"/>
    <ph>These elements are grouped with associated metadata</ph>
  </figgroup>
</fig>
```

1.17 <fn>

A footnote is ancillary information that typically is rendered in the footer of a page or at the end of an online article. Such content is usually inappropriate for inline inclusion.

Usage information

There are two types of footnotes: *single-use footnote* and *use-by-reference footnote*.

Single-use footnote
This is produced by a `<fn>` element that does not specify a value for the `@id` attribute.

Use-by-reference footnote
This is produced by a `<fn>` element that specifies a value for the `@id` attribute. It must be used in conjunction with an `<xref>` element with `@type` set to "fn".

To reference a footnote that is located in another topic, the conref or conkeyref mechanism is used.

Rendering expectations

The two footnote types typically produce different types of output:

Single-use footnote
When rendered, a superscript symbol (numeral or character) is produced at the location of the `<fn>` element. The superscript symbol is hyperlinked to the content of the footnote, which is placed at the bottom of a PDF page or the end of an online article. The superscript symbol can be specified by the value of the `@callout` attribute. When no `@callout` value is specified, footnotes are typically numbered.

Use-by-reference footnote
Nothing is rendered at the location of the `<fn>` element. The content of a use-by-reference footnote is only rendered when it is referenced by an `<xref>` with the `@type` attribute set to "fn". If an `<xref>` with the `@type` attribute set to "fn" is present, a superscript symbol is rendered at the location of the `<xref>` element. Unless conref or conkeyref is used, the `<fn>` and `<xref>` must be located in the same topic.

However, the details of footnote processing and formatting are implementation dependent. For example, a tool that renders DITA as PDF might lack support for the `@callout` attribute, or footnotes might be collected as end notes for certain types of publications.

Attributes

The following attributes are available on this element: universal attributes and the attribute defined below.

@callout
Specifies the character or character string that is used for the footnote link

Examples

> **Comment by Kristen J Eberlein on 29 November 2021**
>
> In this section, we need to replace each of "examples of rendered output" with screen captures. This effects the following examples:
>
> - A single-use footnote
> - A single-use footnote with a `@callout` attribute
> - A use-by-reference footnote

This section contains examples of how the `<fn>` element can be used.

Figure 3: A single-use footnote

The following code sample shows a single-use footnote. It contains a simple `<fn>` element, with no `@id` or `@callout` attribute.

```
The memory storage capacity of the computer is
2 GB<fn>A GB (gigabyte) is equal to
1000 million bytes</fn> with error correcting support.
```

When rendered, typically a superscript symbol is placed at the location of the `<fn>` element; this superscript symbol is hyperlinked to the content of the `<fn>`, which is typically placed at the bottom of a PDF page or the end of an online article. The type of symbol used is implementation specific.

The above code sample might produce output similar to the following:

> The memory storage capacity of the computer is 2 GB[1] with error correcting support.
>
> . . . . . .
>
> [1] A GB (gigabyte) is equal to 1000 million bytes
>
> ----- [bottom of page] ---

Figure 4: A single-use footnote with a @callout attribute

The following code sample shows a single-use footnote that uses a `@callout` attribute:

```
The memory storage capacity of the computer is
2 GB<fn callout="#">A GB (gigabyte) is equal to
1000 million bytes</fn> with error correcting support.
```

The rendered output is similar to that of the previous example, although processors that support it will render the footnote symbol as # (hashtag).

Figure 5: A use-by-reference footnote

The following code sample shows use-by-reference footnotes. The `<fn>` elements have `@id` attributes, and inline `<xref>` elements reference those `<fn>` elements:

```
<fn id="dog-name">Fido</fn>
<fn id="cat-name">Puss</fn>
<fn id="llama-name">My llama</fn>
```

```
...
<p>I like pets. At my house, I have a dog<xref href="#topic/dog-name" type="fn"/>, a
cat<xref href="#topic/cat-name" type="fn"/>, and a
llama<xref href="#topic/llama-name" type="fn"/>.</p>
```

The code sample might produce output similar to the following:

      . . . . . .

I like pets. At my house, I have a dog[1], a cat[2], and a llama[3].

      . . . . . .

[1]Fido

[2]Puss

[3]My llama

----- [bottom of page] ---

Figure 6: A single-use footnote that uses conref

The following code sample shows footnotes stored in a shared topic (`footnotes.dita`):

```
<!-- Content from footnotes.dita -->
<topic id="footnotes">
...
  <fn id="strunk">Elements of Style</fn>
  <fn id="DQTI">Developing Quality Technical Information, 2nd edition</fn>
...
</topic>
```

To use those footnotes, authors conref them into the relevant topics:

```
<p>See the online resource<fn conref="footnotes.dita#footnotes/DQTI"/> for more
   information about how to assess the quality of technical documentation ...</p>
```

Figure 7: A use-by-reference footnote that uses conref

The following code sample shows a use-by-reference footnote that uses conref:

```
<topic id="evaluating-quality">
  <title>Evaluating documentation quality</title>
  <body>
  ...
  <fn conref="footnotes.dita#footnotes/DQTI" id="dqti"/>
  ...
  <p>See the online resource<xref href="./evaluating-quality/dqti" type="fn"/> for more
     information about how to assess the quality of technical documentation ...</p>
  ...
  </body>
<topic>
```

1.18 <image>

An image is a reference to artwork that is stored outside of the content.

Rendering expectations

The referenced image typically is rendered in the main flow of the content.

003 (32)   Processors **SHOULD** scale the object when values are provided for the `@height` and `@width` attributes. The following expectations apply:

- If a height value is specified and no width value is specified, processors **SHOULD** scale the width by the same factor as the height.
- If a width value is specified and no height value is specified, processors **SHOULD** scale the height by the same factor as the width.
- If both a height value and width value are specified, implementations **MAY** ignore one of the two values when they are unable to scale to each direction using different factors.

Attributes

The following attributes are available on this element: universal attributes, href (link-relationship attributes), format (link-relationship attributes), scope (link-relationship attributes), `@keyref`, and the attributes defined below.

@align
Controls the horizontal alignment of an image when `@placement` is specified as "break". Common values include "left", "right", and "center".

@height
Specifies the vertical dimension for the resulting display. The value of this attribute is a real number expressed in decimal notation, optionally followed by a unit of measure. The following units of measurement are supported: cm, em, in, mm, pc, pt, and px (centimeters, ems, inches, millimeters, picas, points, and pixels, respectively). The default unit is px (pixels). Possible values include:"5", "5in", and "10.5cm".

@placement
Indicates whether an image is displayed inline or on a separate line. The default value is inline. Allowable values are "inline", "break", and "-dita-use-conref-target".

@scale
Specifies a percentage as an unsigned integer by which to scale the image in the absence of any specified image height or width; a value of 100 implies that the image should be presented at its intrinsic size. If a value has been specified for the `@height` or `@width` attribute (or both), the `@scale` attribute is ignored.

@scalefit

Specifies whether an image is scaled up or down to fit within available space. The allowable values are "yes", "no", and "-dita-use-conref-target". If `@height`, `@width`, or `@scale` is specified, those attributes determine the graphic size, and the `@scalefit` attribute is ignored. If none of those attributes are specified and `scalefit="yes"`, then the image is scaled by the same factor in both dimensions, so that the graphic will just fit within the available height or width, whichever is more constraining.

The available width would be that of the prevailing column or table cell, that is, the width a paragraph of text would have if the graphic were a paragraph instead od text. The available height is implementation dependent, but if feasible, it is suggested to be the page or table cell height or some other reasonable value.

@width
Specifies the horizontal dimension for the resulting display. The value of this attribute is a real number expressed in decimal notation, optionally followed by a unit of measure. The following units of measurement are supported: cm, em, in, mm, pc, pt, and px (centimeters, ems, inches, millimeters, picas, points, and pixels, respectively). The default unit is px (pixels). Possible values include:"5", "5in", and "10.5cm".

Example

The following code sample shows how an image is referenced. The `@placement` attribute is set to "break"; this ensures that the image is not rendered inline.

```
<image href="bike.gif" placement="break">
  <alt>Two-wheeled bicycle</alt>
</image>
```

1.19 <include>

Included content is a reference to non-DITA content outside the current document that will be rendered at the location of the reference. The resource is specified using either a URI or a key reference. Processing expectations for the referenced data can also be specified.

Usage information

The `<include>` element is intended as a base for specialization and for the following use cases:

- The transclusion of non-DITA XML within `<foreign>` element using `parse="xml"`
- The transclusion of preformatted textual content within `<pre>` element using `parse="text"`
- The transclusion of plain-text prose within DITA elements using `parse="text"`

In addition, processors can support additional values for the `@parse` attribute.

For example, the `<include>` element can be specialized to an element such as `<coderef>` as a way to include preformatted sample programming code.

The `<include>` element is not intended to reference DITA content. Use `@conref` or `@conkeyref` to reuse DITA content.

Processing expectations

The `<include>` element instructs processors to insert the contents of the referenced resource at the location of the `<include>` element. If the content is unavailable to the processor or cannot be processed using the specified `@parse` value, the contents of the `<fallback>` element, if any, are presented instead.

004 (32)	Processors **SHOULD** support the `@parse` values "text" and "xml".
005 (32)	Processors **SHOULD** detect the encoding of the referenced document based on the rules described for the `@encoding` attribute.

Attributes

The following attributes are available on this element: universal attributes, inclusion attributes, link-relationship attributes, and `@keyref`.

Examples

For the most part, `<include>` is intended to be used as a base for specialization. The following examples use it directly for purposes of illustration.

Figure 8: Inclusion of XML markup other than SVG or MathML

In the following code sample, the `<include>` element references a tag library descriptor file:

```
<fig>
  <title>JSP Tag Library Elements and Attributes</title>
  <foreign outputclass="tld">
    <include href="../src/main/webapp/WEB-INF/jsp-tag-library.tld"
             parse="xml" format="tld"/>
  </foreign>
</fig>
```

Figure 9: Inclusion of README text into a DITA topic, with fallback

In the following code sample, a README text file is referenced in order to reuse a list of changes to a set of source code:

```
<topic id="readme">
  <title>Summary of changes</title>
  <shortdesc>This topic describes changes in the project source code.</shortdesc>
  <body>
    <include href="../src/README.txt" parse="text" encoding="UTF-8">
      <fallback>See README.txt in the source package for a list of changes.</fallback>
    </include>
  </body>
</topic>
```

Figure 10: Inclusion of preformatted text

In the following code sample, the `<include>` element references a JSON file:

```
<pre>
  <include href="../src/config.json" format="json" parse="text" encoding="UTF-8"/>
</pre>
```

Figure 11: Proprietary vendor handling for CSV tables

In the following code sample, the `<include>` element specifies a proprietary `@parse` value that instructs a processor how to render a comma-separated data set within the figure:

```
<fig>
  <title>Data Table</title>
  <include href="data.csv"  encoding="UTF-8"
    parse="http://www.example.com/dita/includeParsers/csv-to-simpletable"/>
</fig>
```

1.20 <keyword>

A keyword is text or a token that has a unique or key-like value, such as a product name or unit of reusable text.

Processing expectations

When used within the `<keywords>` element, the content of a `<keyword>` element is considered to be metadata and should be processed as appropriate for the given output medium.

Elements that are specialized from the `<keyword>` element might have extended processing, such as specific formatting or automatic indexing.

Attributes

The following attributes are available on this element: universal attributes and `@keyref`.

Examples

This section contains examples of how the `<keyword>` element can be used.

Figure 12: <keyword> element used to store a product name

In the following code sample, the `<keyword>` element holds a product name that can be referenced using content reference (conref) or content key reference (conkeyref):

```
<keyword id="acme-bird-feeder">ACME Bird Feeder</keyword>
```

To enable referencing variable text using `@keyref`, store the product name in a `<keytext>` element.

Figure 13: <keyword> element referencing a product name

In the following example, the `<keyword>` element references a product name using `@conkeyref`:

```
<p>To fill the <keyword conkeyref="productnames/acme-bird-feeder"/>, unscrew the top ...</p>
```

Figure 14: <keyword> element as metadata

In the following code sample, "Big data" is specified as metadata that applies to the topic:

```
<prolog>
  <metadata>
    <keywords>
      <keyword>Big data</keyword>
    </keywords>
  </metadata>
</prolog>
```

1.21

A list item is an item in either an ordered or unordered list.

Attributes

The following attributes are available on this element: universal attributes.

Example

See `<ol>` (23) or `<ul>` (30).

1.22 <lines>

Lines are lines of text where white space is significant. The `<lines>` element can be used to represent dialogs, poetry, or other text fragments where line breaks are significant.

Rendering expectations

006 (32)   Processors **SHOULD** preserve or otherwise indicate white space within the `<lines>` element.

Attributes

The following attributes are available on this element: universal attributes, display attributes, `@xml:space`, and `@spectitle`.

Example

In the following code sample, a `<lines>` element contains the text of *[Buffalo Bill 's]*, a poem by e. e. cummings:

```
Buffalo Bill 's
defunct
            who used to
            ride a watersmooth-silver
                                            stallion
and break onetwothreefourfive pigeonsjustlikethat
                                                            Jesus

he was a handsome man
                                    and what i want to know is
how do you like your blue-eyed boy
Mister Death
```

1.23 <longdescref>

A long description reference is a reference to a textual description of a graphic or object. This is typically used to provide an extended description when the graphic or object is too complicated to describe with alternate text.

Attributes

The following attributes are available on this element: universal attributes, link-relationship attributes, and `@keyref`.

Examples

This section contains examples of how the `<longdescref>` element can be used.

Figure 15: <longdescref> which references a local DITA description

In the following code sample, the `<longdescref>` references a detailed image description that is stored in a DITA topic:

```
<image href="llama.jpg">
  <alt>Llama picture</alt>
```

```
    <longdescref href="my-pet-llama.dita"/>
  </image>
```

Figure 16: <longdescref> which references an external description

In this code sample, the long description is stored remotely, on a external Web site:

```
<image href="puffin.jpg">
  <alt>Puffin pigure</alt>
  <longdescref href="http://www.example.org/birds/puffin.html"
               scope="external"
               format="html"/>
</image>
```

1.24 <lq>

A long quotation is a quotation that contains one or more paragraphs. The title and source of the document that is being quoted can be specified.

Attributes

The following attributes are available on this element: universal attributes and `@keyref`.

Example

The following code sample contains a quotation. The `@href` attribute indicates a Web site where the full text of the address can be accessed, and the `<cite>` attribute specifies the title of the document that is quoted

```
<p>This is the first line of the address that Abraham Lincoln delivered
on November 19, 1863 for the dedication of the cemetery at Gettysburg, Pennsylvania.</p>
<lq reftitle="Gettysburg address"
href="https://en.wikisource.org/wiki/Gettysburg_Address_(Nicolay_draft)" format="html"
scope="external">Four score and seven years ago our fathers brought forth on this continent
a new nation, conceived in liberty, and dedicated to the proposition that all men
are created equal. <cite>Gettysburg address</cite></lq>
```

1.25 <note>

A note is information that expands on or calls attention to a particular point.

Usage information

The nature of a note (for example, caution, danger, or warning) is indicated through the values selected for the `@type` attribute.

The values "danger", "notice", and "warning" have meanings that are based on ANSI Z535 and ISO 3864 regulations.

If `@type` is set to "other", the value of the `@othertype` attribute can be used as a label for the note. Many processors will require additional information on how to process the value.

Attributes

The following attributes are available on this element: universal attributes, `@spectitle`, and the attributes defined below.

@othertype
> Specifies an alternate note type. This value is used as the user-provided note label when the `@type` attribute value is set to "other".

@type
> Specifies the type of a note. This differs from the `@type` attribute on many other DITA elements. The following are the allowable values:

> **Comment by Kristen J Eberlein on 23 November 2021**
>
> The following explanation needs work! Most of this appeared in DITA 1.0 and has not been edited since then (with the exception of where there is association with the hazard statement domain). TC members, please help.

> **attention**
> > Please pay extra attention to this note.

> **caution**
> > Care is required when proceeding.

> **danger**
> > When used with the `<hazardstatement>` element, this indicates an imminently hazardous situation which, if not avoided, **will** result in death or serious injury.

> **important**
> > This note is important.

> **note**
> > This is just a note.

> **notice**
> > When used with the `<hazardstatement>` element, this indicates a potential situation which, if not avoided, might result in an **undesirable result or state**.

> **other**
> > This is something other than a normal note.

> **remember**
> > Don't forget to do what this note says.

> **restriction**
> > You can't do what this note says.

> **tip**
> > This is a fine little tip.

> **trouble**
> > Provides information about how to remedy a trouble situation.

> **warning**
> > Indicates a potentially hazardous situation. When used with the `<hazardstatement>` element, this indicates a situation which, if not avoided, could result in death or serious injury.

> **-dita-use-conref-target**
> > See for more information.

Example

The following code sample shows a `<note>` with `@type` set to "tip":

```
<note type="tip">Thinking of a seashore, green meadow, or cool
mountain overlook can help you to relax and be more
patient.</note>
```

1.26 <object>

The DITA `<object>` element corresponds to the HTML `<object>` element, and the attribute semantics derive from the HTML definitions. For example, the `@type` attribute differs from the `@type` attribute on many other DITA elements.

Usage information

The `<object>` element enables authors to include animated images, applets, plug-ins, video clips, and other multimedia objects in a topic.

Rendering expectations

007 (32)
> Processors **SHOULD** scale the object when values are provided for the `@height` and `@width` attributes. The following expectations apply:
>
> - If a height value is specified and no width value is specified, processors **SHOULD** scale the width by the same factor as the height.
> - If a width value is specified and no height value is specified, processors **SHOULD** scale the height by the same factor as the width.
> - If both a height value and width value are specified, implementations **MAY** ignore one of the two values when they are unable to scale to each direction using different factors.

008 (32)
> When an object cannot be rendered in a meaningful way, processors **SHOULD** present the contents of the `<fallback>` element, if it is present.

Attributes

The following attributes are available on this element: universal attributes and the attributes defined below.

@archive
Specifies a space-separated list of URIs indicating resources needed by the object. These resources might include those URIs specified by the `@classid` and `@data` attributes. Preloading these resources usually results in faster load times for objects. The URIs in the list should be relative to the URI specified in the `@codebase` attribute.

@archivekeyrefs
Specifies key references to one or more archives, as for `@archive`. The value is a space-separated list of key names. Each resolvable key reference is treated as a URI as though it had been specified on the `@archive` attribute. When specified, and at least one key name is resolvable, the key-provided archive list is used. If `@archive` is specified, it is used as a fallback when no key names can be resolved to a URI.

@classid
Contains a URI that specifies the location of an object's implementation. It can be used together with the `@data` attribute which is specified relative to the value of the `@codebase` attribute.

@classidkeyref
   Specifies a key reference to the URI that specifies the location of an object's implementation, as for `@classid`. When specified, and the key is resolvable, the key-provided class ID URI is used. If `@classid` is specified, it is used as a fallback when the key cannot be resolved to a URI.

@codebase
   Specifies the base URI used for resolving the relative URI values given for `@classid`, `@data`, and `@archive` attributes. If `@codebase` is not set, the default is the base URI of the current element.

@codebasekeyref
   Specifies a key reference to the base URI used for resolving other attributes, as for `@codebase`. When specified, and the key is resolvable, the key-provided code base URI is used. If `@codebase` is specified, it is used as a fallback if the key cannot be resolved to a URI. If no URI results from processing `@codebasekeyref` and `@codebase` is not specified, the default is the base URL of the current element.

@data
   Contains a reference to the location of an object's data. If this attribute is a relative URL, it is specified relative to the value of the `@codebase` attribute. If this attribute is set, the `@type` attribute should also be set.

@datakeyref
   Provides a key reference to the object. When specified and the key is resolvable, the key-provided URI is used. A key that has no associated resource, only link text, is considered to be unresolved. If `@data` is specified, it is used as a fallback when the key cannot be resolved to a resource.

@declare
   Specifies whether the current object definition is only a declaration. When this attribute is set to "declare", the current object definition is a declaration only. The object must be instantiated by a later nested object definition referring to this declaration. The only allowable value is "declare".

@height
   Specifies the vertical dimension for the resulting display. The value of this attribute is a real number expressed in decimal notation, optionally followed by a unit of measure. The following units of measurement are supported: cm, em, in, mm, pc, pt, and px (centimeters, ems, inches, millimeters, picas, points, and pixels, respectively). The default unit is px (pixels). Possible values include:"5", "5in", and "10.5cm".

@name
   Defines a unique name for the object.

@standby
   Contains a message to be displayed while an object is loading.

@tabindex
   Specifies the position of the object in tabbing order.

@type
   Indicates the content type (MIME type) for the data specified by the `@data` or `@datakeyref` attribute. This attribute should be set when the `@data` attribute is set to avoid loading unsupported content types. Note that this differs from the `@type` attribute on many other DITA elements (it specifies a MIME type rather than a content type). If `@type` is not specified, the effective type value for the key named by the `@datakeyref` attribute is used as the this attribute's value.

@usemap
   Indicates that a client-side image map is to be used. An image map specifies active geometric regions of an included object and assigns a link to each region. When a link is selected, a document might be retrieved or a program might run on the server.

@width

    Specifies the horizontal dimension for the resulting display. The value of this attribute is a real number expressed in decimal notation, optionally followed by a unit of measure. The following units of measurement are supported: cm, em, in, mm, pc, pt, and px (centimeters, ems, inches, millimeters, picas, points, and pixels, respectively). The default unit is px (pixels). Possible values include:"5", "5in", and "10.5cm".

Example

This section contains examples of how the `<object>` element can be used.

Figure 17: Referencing a PDF

The following code sample shows how an `<object>` element can be used to render a PDF as an inline object:

```
<object type="application/pdf"
        data="resources/pdf/terms.pdf"
        width="300"
        height="200"
        id="terms-and-conditions">
   <desc>Terms and conditions for using the site.</desc>
   <fallback>PDF cannot be displayed; please download the terms and
             conditions from <xref keyref="site-terms"/>.</fallback>
</object>
```

Figure 18: Object with reference to video using key reference on the <param> elements

The following code sample shows how key definitions can be used to reference supporting resources for an `<object>`:

```
<object id="E5123_026.mp4"
      width="300" height="300">
  <fallback>Media not available.</fallback>
  <param name="poster" keyref="E5123_026_poster" />
  <param name="source" keyref="E5123_026_video" />
</object>
```

In this scenario, the keys could be defined as follows:

```
<map>
  <!-- ... -->
  <keydef keys="E5123_026_poster"
      href="../images/E5123_026_poster.png"
      type="video/mp4"
  />
  <keydef keys="E5123_026_video"
      href="../media/E5123_026_poster.mp4"
      type="video/mp4"
  />
  <!-- ... -->
</map>
```

Figure 19: Object with indirect reference to a flash file

The following code sample shows how key definitions can be used to reference the main content for an `<object>`:

```
<object id="cutkey370"
    classidkeyref="video_classid"
    codebasekeyref="video_codebase"
    datakeyref="cutkey370"
    height="280"
```

```
    width="370">
  <desc>Video illustration of how to cut a key</desc>
  <fallback>Media not available.</fallback>
  <param name="movie" keyref="cutkey370"/>
  <param name="quality" value="high"/>
  <param name="bgcolor" value="#FFFFFF"/>
</object>
```

In this scenario, the keys could be defined as follows:

```
<map>
  <!-- ... -->
  <!-- NOTE: Using @scope="external" because the class ID is a URI
       that is not intended to be directly resolved. -->
  <keydef keys="video_classid"
    href="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
    scope="external" />

  <!-- NOTE: Using @scope="external" to avoid systems trying to
            download this file when they don't need to. -->
  <keydef keys="video_codebase"
    href="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=6,0,0,0"
    format="shockwave"
    scope="external" />

  <!-- Using @scope="external" here because the referenced URL is not intended
       to be resolved in isolation but relative to the codebase URI. -->
  <keydef keys="cutkey370"
    href="cutkey370.swf"
    type="application/x-shockwave-flash"
    scope="external" />
  <!-- ... -->
</map>
```

1.27

An ordered list is a list of items that are sorted by sequence or order of importance.

Attributes

The following attributes are available on this element: universal attributes, @compact, and @spectitle.

Example

The following code sample shows the use of an ordered list:

```
<p>Here is a list of the five longest-living people who were born in the 19th century:</p>
<ol>
 <li>Jeanne Calment (1875-1997)</li>
 <li>Sarah Knauss (1880-1999)</li>
 <li>Marie-Louise Meilleur (1880-1998)</li>
 <li>Emma Morano (1899-2017)</li>
 <li>Misao Okawa (1898-2015)</li>
</ol>
<p>Note that systematic verification has only been practised in recent years and only
 in certain parts of the world.</p>
```

1.28 <p>

A paragraph is a group of related sentences that support a central idea.

Attributes

The following attributes are available on this element: universal attributes.

Example

The following code sample contains a paragraph:

```
<p>A paragraph is a group of related sentences that support a central
idea. Paragraphs typically consist of three parts: a topic sentence, body sentences,
and a concluding or bridging sentence.</p>
```

1.29 <param>

The `<param>` (parameter) element specifies a set of values that might be required by an `<object>` at runtime.

Usage information

Any number of `<param>` elements might appear in the content of an `<object>` in any order, but must be placed at the start of the content of the enclosing object. This element is comparable to the HMTL `<param>` element, and the attribute semantics derive from their HTML definitions. For example, the `@type` attribute differs from the `@type` attribute on many other DITA elements.

Processing expectations

The `@keyref` and `@valuetype` attributes on `<param>` have the following expectations:

- If `@valuetype` is specified but is not set to "ref", the `@keyref` attribute is ignored.
- When `@valuetype` is not specified and `@keyref` is specified, it implies a setting of `valuetype="ref"`.
- When `@keyref` is specified and the effective value of `@valuetype` is "ref":
    1. When the key specified by `@keyref` is resolvable and has an associated URI, that URI is used as the value of this element (overriding `@value`, if that is specified).
    2. When the key specified by `@keyref` is resolvable and has no associated resource (only link text), the `@keyref` attribute is considered to be unresolvable for this element. If `@value` is specified, it is used as a fallback.
    3. When the key specified by `@keyref` is not resolvable, the value of the `@value` attribute is used as a fallback target for the `<param>` element.

In addition, the following expectations apply to the `@type` attribute:

1. When `@valuetype` is set to "ref", the `@type` attribute directly specifies the content type of the resource designated by `@value`.
2. Otherwise, if `@type` is specified and `@keyref` is specified and resolvable, this attribute specifies the content type of the resource designated by `@keyref`.
3. Otherwise, if `@type` is not specified and `@keyref` is specified and is resolvable, the effective type value specified for the key that is named by the `@keyref` attribute is used as the value of the `@type` attribute.

Attributes

The following attributes are available on this element: universal attributes and the attributes defined below.

@keyref
    Specifies a key reference to the thing the parameter references.

@name (REQUIRED)
> Specifies the name of the parameter.

@type
> Specifies for a user agent the type of values that will be found at the URI designated by `@value`. Note that this differs from the `@type` attribute on many other DITA elements.

@value
> Specifies the value of a run-time parameter that is described by the `@name` attribute.

@valuetype
> Specifies the type of the `@value` attribute. Allowed values are:
>
> **data**
> > A value of "data" means that the value will be evaluated and passed to the object's implementation as a string.
>
> **ref**
> > A value of "ref" indicates that the value of the `@value` attribute is a URL that designates a resource where run-time values are stored. This allows support tools to identify URLs that are given as parameters.
>
> **object**
> > A value of "object" indicates that the value of the `@value` attribute is an identifier that refers to an object declaration in the document. The identifier must be the value of the `@id` attribute set for the declared `<object>` element.
>
> **-dita-use-conref-target**
> > See for more information.

Example

See `<object>` (22).

1.30 <ph>

A phrase is a small group of words that stand together as a unit, typically forming a component of a clause.

Usage information

The `<ph>` element often is used to enclose a phrase for reuse or conditional processing.

The `<ph>` element frequently is used as a specialization base, to create phrase-level markup that can provide additional semantic meaning or trigger specific processing or formatting. For example, all highlighting domain elements are specializations of `<ph>`.

Attributes

The following attributes are available on this element: universal attributes and `@keyref`.

Example

The following code sample shows `<ph>` elements that are used for conditional processing:

```
<p>The Style menu is the <ph product="Software1000"/>third item</ph>
<ph product="Software9000"/>fourth item</ph> from the left on the menu bar.</p>
```

1.31 <pre>

Preformatted text is text that contains line breaks and spaces that are intended to be preserved at publication time.

Usage information

The <pre> element is often used for ASCII diagrams and code samples. It is the specialization base for the @codeblock element in the Technical Content edition.

Rendering expectations

009 (32)   Processors **SHOULD** preserve line the breaks and spaces that are present in the content of a <pre> element.

Attributes

The following attributes are available on this element: universal attributes, display attributes, @xml:space, and @spectitle.

Example

The following code sample shows preformatted text that contains white space and line breaks. When the following code sample is published, the white space and line breaks are preserved.

```
<pre>
          MEMO: programming team fun day
Remember to bring a kite, softball glove, or other favorite
outdoor accessory to tomorrow's fun day outing at Zilker Park.
Volunteers needed for the dunking booth.
</pre>
```

1.32 <q>

A quotation is a small group of words that is taken from a text or speech and repeated by someone other than the original author or speaker.

Rendering expectations

Processors add appropriate styling, such as locale-specific quotation marks, around the contents of the <q> element and render it inline.

Attributes

The following attributes are available on this element: universal attributes.

Example

In the following code sample, the <q> element contains a quotation. Note that no quotation marks are included; locale-specific quotation marks will be generated during processing.

```
<p>
George said, <q>Disengage the power supply before servicing the unit.</q>
</p>
```

1.33 <section>

A section is an organizational division in a topic. Sections are used to organize subsets of information that are directly related to the topic.

Usage information

Multiple sections within a single topic do not represent a hierarchy, but rather peer divisions of that topic. Sections cannot be nested. Sections can have titles.

Note	For maximum flexibility in creating specialization, sections allow plain text as well as phrase and block level elements. Because of the way XML grammars are defined within a DTD, any element that allows plain text cannot restrict the order or frequency of other elements. As a result, the `<section>` element allows `<title>` to appear anywhere as a child of `<section>`. However, the intent of the specification is that `<title>` only be used once in any `<section>`, and when used, that it precede any other text or element content.

Rendering expectations

010 (32)

Processors **SHOULD** treat the presence of more than one `<title>` element in a `<section>` element as an error.

Attributes

The following attributes are available on this element: universal attributes and `@spectitle`.

Example

The following code sample shows how element-reference topics in the DITA specification use titled sections to provide a consistent structure for grouping information:

```
<reference id="pb" xml:lang="en-us">
  <title><xmlelement>p</xmlelement></title>
  <shortdesc conkeyref="library-short-descriptions/p"/>
  <refbody>
    <section><title>Usage information</title>
      <p>...</p>
    </section>
    <section><title>Rendering expectations</title>
      <p>...</p>
    </section>
    <section><title>Processing expectations</title>
      <p>...</p>
    </section>
    <section><title>Specialization hierarchy</title>
      <p>...</p>
    </section>
    <section><title>Attributes</title>
      <p>...</p>
    </section>
    <example><title>Example</title>
      <p>...</p>
    </example>
  </refbody>
</reference>
```

1.34 <sl>

A simple list is a list that contains a few items of short, phrase-like content.

Attributes

The following attributes are available on this element: universal attributes, `@compact`, and `@spectitle`.

Example

The following code sample shows how a simple list could be used in a topic that discusses related modules:

```
<section>
  <title>Messages</title>
  <p>Messages from the ags_open module are identical with messages from:</p>
  <sl>
    <sli>ags_read</sli>
    <sli>ags_write</sli>
    <sli>ags_close</sli>
  </sl>
</section>
```

1.35 <sli>

A simple list item is a component of a simple list. A simple list item contains a brief phrase or text content, adequate for describing package contents, for example.

Attributes

The following attributes are available on this element: universal attributes.

Example

See `<sl>` (28).

1.36 <term>

A term is a word or phrase that might have or require extended definitions or explanations.

Usage information

The `@keyref` attribute can be used to associate a term with a resource, typically a definition of the term. The `@keyref` attribute can also be used to supply the text content for `<term>` using standard `@keyref` processing for variable text.

Attributes

The following attributes are available on this element: universal attributes and `@keyref`.

Example

The following code sample shows how the `<term>` element can be used:

```
<p>A <term>reference implementation</term> of DITA implements the standard,
fallback behaviors intended for DITA elements.</p>
```

1.37 \<text\>

Text is a container for text with no associated semantics.

Usage information

The `<text>` element is primarily used as a base for specialization or to enable reuse. The `<text>` element can contain only text or nested `<text>` elements.

Attributes

The following attributes are available on this element: universal attributes.

Example

In the following code sample, the `<text>` element is used to contain text that is intended to be reused:

```
<p>This an example of <text id="reuse">Text
    that is reusable</text>, with no extra
    semantics attached to the text.</p>
```

1.38 \<tm\>

A trademark is a term or phrase that is trademarked. Trademarks include registered trademarks, service marks, slogans, and logos.

Usage information

The business rules for indicating and displaying trademarks differ from company to company. These business rules can be enforced by either authoring policy or processing.

Attributes

The following attributes are available on this element: universal attributes and the attributes defined below.

@tmtype (REQUIRED)
> Specifies the trademark type. Allowable values are:

> **tm**
>> Trademark

> **reg**
>> Registered trademark

> **service**
>> Service mark

> **-dita-use-conref-target**
>> See for more information.

@trademark
> Specifies the trademarked term.

@tmowner
> Specifies the trademark owner, for example "OASIS".

@tmclass
> Specifies the classification of the trademark. This can be used to differentiate different groupings of trademarks.

Example

The following code sample shows how a company might use the `<tm>` element:

```
<p>The advantages of using the
  <tm trademark="Acme" tmtype="reg">Acme</tm>
  <tm trademark="SuperFancyWidget" tmtype="tm">SuperFancyWidget</tm>
  are well known to Bugs Bunny fans.</p>
```

1.39

An unordered list is a list in which the order of items is not significant.

Attributes

The following attributes are available on this element: universal attributes, `@compact`, and `@spectitle`.

Example

The following code sample shows a list in which the order of items is unimportant:

```
<p>Here are the countries that I have visited:</p>
<ul>
 <li>Germany</li>
 <li>France</li>
 <li>Japan</li>
 <li>Mexico</li>
</ul>
```

1.40 <xref>

A cross reference is an inline link. A cross reference can link to a different location within the current topic, another topic, a specific location in another topic, or an external resource such as a PDF or web page.

Attributes

The following attributes are available on this element: universal attributes, link-relationship attributes, and `@keyref`.

Examples

This section contains examples of how the `<xref>` element can be used.

Figure 20: Cross reference to another topic, without link text

The following code sample shows a cross reference to another topic. Link text is not provided. Processor typically use the topic title as the link text.

```
<p>Background information about DITA is provided in
<xref href="overview-of-dita.dita"/>.</p>
```

The same cross reference could be created using `@keyref` instead of `@href`. Using `@keyref` allows the link to be redirected to different resources when the topic is used in different contexts.

Figure 21: Cross references with link text specified

The following code sample shows a cross reference that specifies link text:

```
<p>While this set of tutorials gives several simple examples of
<xref keyref="markup-examples">common DITA features</xref>, a comprehensive
list of DITA features is available in the DITA specification
<xref keyref="dita-conformance">conformance clause</xref>.</p>
```

Figure 22: Cross reference to an external ressource

The following code sample shows a cross reference to a web page:

```
<xref href="https://www.example.com/docview.wss?rs=757"
scope="external" format="html">Part number SSVNX5</xref>
```

Related concepts
DITA addressing

Related reference
link

A Aggregated RFC-2119 statements

This appendix contains all the normative statements from the DITA 2.0 specification. They are aggregated here for convenience in this non-normative appendix.

Item	Conformance statement
001 (4)	When used in conjunction with `<fig>` or `<table>` elements, processors **SHOULD** consider the content of `<desc>` elements to be part of the content flow.
002 (7)	By default, processors **SHOULD NOT** render `<draft-comment>` elements. Processors **SHOULD** provide a mechanism that causes the content of the `<draft-comment>` element to be rendered in draft output only.
003 (12)	Processors **SHOULD** scale the object when values are provided for the `@height` and `@width` attributes. The following expectations apply: • If a height value is specified and no width value is specified, processors **SHOULD** scale the width by the same factor as the height. • If a width value is specified and no height value is specified, processors **SHOULD** scale the height by the same factor as the width. • If both a height value and width value are specified, implementations **MAY** ignore one of the two values when they are unable to scale to each direction using different factors.
004 (14)	Processors **SHOULD** support the `@parse` values "text" and "xml".
005 (14)	Processors **SHOULD** detect the encoding of the referenced document based on the rules described for the `@encoding` attribute.
006 (17)	Processors **SHOULD** preserve or otherwise indicate white space within the `<lines>` element.
007 (20)	Processors **SHOULD** scale the object when values are provided for the `@height` and `@width` attributes. The following expectations apply: • If a height value is specified and no width value is specified, processors **SHOULD** scale the width by the same factor as the height. • If a width value is specified and no height value is specified, processors **SHOULD** scale the height by the same factor as the width. • If both a height value and width value are specified, implementations **MAY** ignore one of the two values when they are unable to scale to each direction using different factors.
008 (20)	When an object cannot be rendered in a meaningful way, processors **SHOULD** present the contents of the `<fallback>` element, if it is present.
009 (26)	Processors **SHOULD** preserve line the breaks and spaces that are present in the content of a `<pre>` element.
010 (27)	Processors **SHOULD** treat the presence of more than one `<title>` element in a `<section>` element as an error.

Index

A

accessibility
    images 3, 12
alternate text 3

B

body elements 3

C

citations 3
cross-references 30

D

definition lists
    description 3
    entries 6
    headings 6
        definition columns 4
        term columns 8
    overview 5
    terms 7
descriptions 4
divisions 5
draft comments 7

E

element groups
    body 3
elements
    body
        `<alt>` 3
        `<cite>` 3
        `<dd>` 3
        `<ddhd>` 4
        `<desc>` 4
        `<div>` 5
        `<dl>` 5
        `<dlentry>` 6
        `<dlhead>` 6
        `<draft-comment>` 7
        `<dt>` 7
        `<dthd>` 8
        `<example>` 8
        `<fallback>` 9
        `<fig>` 9
        `<figgroup>` 9

elements *(continued)*
    body *(continued)*
        `<fn>` 10
        `<image>` 12
        `<include>` 14
        `<keyword>` 15
        `<li>` 16
        `<lines>` 17
        `<longdescref>` 17
        `<lq>` 18
        `<note>` 18
        `<object>` 20
        `<ol>` 23
        `<p>` 23
        `<param>` 24
        `<ph>` 25
        `<pre>` 26
        `<q>` 26
        `<section>` 27
        `<sl>` 28
        `<sli>` 28
        `<term>` 28
        `<text>` 29
        `<tm>` 29
        `<ul>` 30
        `<xref>` 30

F

figures 9
footnotes 10

G

grouping elements
    `<div>` 5
    `<figgroup>` 9

I

images
    accessibility 3
    alternate text 12
    long descriptions 17
    overview 12
    placement 12
    size 12

L

links