

Introduction à DITA

Contents

Chapter 1. Qu'est-ce que DITA?	1
Langages de balises.....	2
DITA est de la création structurée.....	3
Qu'est-ce qu'un topic DITA ?.....	3
Évaluation de Qu'est-ce que DITA ?.....	4
.....	4
.....	4
.....	4
.....	5
.....	5
.....	5
.....	5
.....	6
Chapter 2. Les Topics DITA	7
Le type de thème générique.....	7
Les topics Concept.....	8
Les topics Task.....	9
Les topics Reference.....	10
Les topics Glossary Entry.....	10
La spécialisation.....	10
Évaluation sur les Topics DITA.....	11
.....	11
.....	11
.....	12
.....	12
.....	12
.....	12
.....	13
.....	13
.....	13

.....	14
Chapter 3. Métadonnées.....	15
Que sont les métadonnées ?.....	15
Comment utiliser les métadonnées ?.....	16
Évaluation sur les Métadonnées.....	18
.....	18
.....	19
.....	19
.....	19
.....	19
.....	20
.....	20
.....	20
.....	20
.....	21
Chapter 4. Création de contenu DITA.....	22
Créer un topic dans un éditeur de texte.....	22
Créer un topic dans oXygen.....	23
Notes, mises en garde, et alertes.....	24
Listes à puces et numérotées.....	25
Éléments de bloc contre intégrés.....	26
Évaluation sur la création de contenu DITA.....	27
Contenu du thème.....	27
Création de mise en garde.....	28
Listes numérotées.....	28
Bloc et intégré.....	28
Chapter 5. Tableaux.....	29
L'élément simpletable.....	30
L'élément table.....	31
Les meilleures pratiques pour les tableaux.....	33
Évaluation sur les Tableaux.....	34
.....	34
.....	34

.....	34
.....	35
Images dans les tables.....	35
Chapter 6. Création de relations entre topics.....	36
Les fichiers carte.....	36
Restitution par transformation des fichiers cartes.....	37
Références croisées.....	38
Liens connexes.....	39
Les tables de relations.....	40
Réutiliser du contenu avec les conrefs.....	41
Évaluation sur la création de relations entre les topics.....	42
.....	42
.....	42
.....	43
.....	43
.....	43
.....	44
.....	44
.....	44
.....	45
.....	45
.....	45

Chapter 1. Qu'est-ce que DITA?

- Présenter le contexte et l'origine de DITA.
- Identifier le balisage DITA.
- Décrire ce qui rend un topic réutilisable.
- Montrer un topic minimal.

DITA est l'abréviation de Darwin Information Typing Architecture. C'est un standard ouvert créé, à l'origine par IBM. IBM donna DITA à [OASIS](#) (l'Organisation pour l'Avancement des Normes d'Information Structurée - Organization for the Advancement of Structured Information Standards) en 2005.

DITA est une méthode basée sur XML et indépendante des outils pour créer, organiser et gérer du contenu. DITA repose sur :

- La création thématique
- La séparation du contenu du formatage
- Le minimalisme
- les concepts de création structurée

Le fait que DITA soit un standard ouvert basé sur XML signifie qu'il existe un ensemble d'outils divers pour créer, éditer, formater et stocker des fichiers DITA. Au fur et à mesure que les besoins de votre entreprise évoluent, vous pouvez utiliser l'outil le mieux adapté à vos besoins.

Related information

- [Histoire de DITA](#)
- [Gestion draconienne des erreurs](#)
- [Les dix commandements de DITA, blog Scriptorium](#)
- [Guide de style DITA, Chapitre 3, Syntaxe et Balisage](#)
- [Exemple de contenu DITA : un projet de documentation basé sur un Manuel du Propriétaire d'une Mini de 1960, rédigé par Felicity Brand](#)

Langages de balises

Un langage de balisage est un moyen de marquer le contenu dans un fichier texte brut (un fichier que vous pourriez éditer avec le Bloc-notes Windows). Le langage de balisage le plus connu est le HTML, qui est le langage de balisage fondamental pour Internet.

DITA est construit sur le langage de balisage XML. XML ressemble beaucoup à HTML. Ils utilisent tous les deux des chevrons (< et >) pour identifier les balises de marquage (par exemple, <title>). Dans les deux langues, une barre oblique identifie une balise de fermeture (</title>)

```
<title>Ceci est un titre</title>
```

En HTML et en XML, Les balises peuvent aussi avoir des attributs (de la forme attribute="value") qui fournissent plus d'information à propos de la balise :


```
<note type="warning">Garder vos bras et vos jambes à l'intérieur du
véhicule en tout temps.</note>
```

Il y a deux principales différences entre HTML et XML :

- HTML peut être assez tolérant lorsque vous oubliez de fermer des balises ou de mettre des guillemets autour des valeurs d'attributs ; XML est strict en les exigeant.
- HTML utilise un ensemble prédéfini de balises (<body>, <p>, , etc.). En XML, les balises sont définies dans un fichier séparé et peuvent être modifiées et ajoutées par un architecte d'informations.

DITA utilise cette fonction de nommage des balises en XML pour définir son propre ensemble de noms de balises ou « éléments ». Les éléments DITA vous permettent de baliser le contenu en utilisant des noms pour des éléments qui ont du sens, tels que <note> pour les notes, <section> pour les sections, <image> pour les images, etc. Toutefois, étant donné que de nombreux noms de balises HTML ont du sens, leurs noms sont également utilisés pour les éléments DITA, tels que <p> pour paragraphe, et et pour les listes non ordonnées et ordonnées.

Une autre différence de dénomination : en HTML, la balise la plus externe ou « racine » est <html>. Dans DITA, le nom de la balise racine dépend du type de topics que vous créez, par exemple, <concept>, <task> ou <reference> (entre autres).

 **Note:** Il est utile de savoir à quoi ressemble le balisage dans XML et DITA. Cependant, la plupart du temps, vous créez du contenu DITA à l'aide d'outils de création intelligents qui gèrent les balises et attributs individuels pour vous.

Vidéo : [Aperçu du balisage HTML par rapport au balisage DITA](#)

DITA est de la création structurée

La *création structurée* est un processus de publication qui vous permet de définir et de mettre en application une organisation cohérente des informations dans les documents.

Les éléments définis dans DITA ont une hiérarchie et des relations très spécifiques. Autrement dit, chaque élément de DITA possède un ensemble spécifique d'éléments pouvant le contenir, ainsi qu'un ensemble spécifique d'éléments pouvant être contenus par celui-ci. DITA spécifie également quels éléments peuvent (ou doivent) être placés avant ou après un élément et combien d'éléments d'un élément particulier sont autorisés.

Par exemple, un élément DITA simpletable peut contenir une entête facultative, qui doit être suivie d'une ou plusieurs lignes simpletables. Les règles empêchent un rédacteur de créer une table simpletable avec une entête au milieu des lignes ou de créer une table ne contenant aucune ligne.

Les règles définissant cette structure sont définies par le langage de balise DITA. Les programmes informatiques peuvent utiliser le même langage pour s'assurer que les documents DITA sont correctement structurés. Si un ensemble particulier de documents nécessite différentes règles ou différents noms d'éléments, un architecte de l'information peut définir ces noms et ces règles.

Bien que DITA soit une forme de création structurée largement utilisée, il existe d'autres normes de création structurée, notamment DocBook, S1000D, NLM, etc.

Qu'est-ce qu'un topic DITA ?

Un topic DITA est l'unité de base de la création et de la réutilisation.

Dans la création thématique, vous créez un certain nombre de topics individuels, chacun abordant une idée ou répondant à une question. Ces topics peuvent ensuite être utilisés et réutilisés dans n'importe quel ordre, dans différents documents. Dans DITA, les rubriques sont organisées en *cartes* (*map*), qui ressemblent beaucoup à une table des matières ; la carte vous permet de spécifier l'ordre et la hiérarchie des topics.

Pour rendre vos topics réutilisables :

- Un topic doit aborder une seule idée ou répondre à une seule question.
- Un topic doit contenir suffisamment d'informations pour être autonome.
- Un topic ne devrait prendre aucun contexte. Vous ne devriez pas émettre d'hypothèses sur ce qui vient avant ou après le topic.
- Un seul fichier doit contenir un seul topic.

Tous les topics DITA doivent avoir au moins un élément title et un attribut id à la racine du thème. En d'autres termes, voici un topic DITA valide :

```
<topic id="sample">  
  <title>Le titre du topic vient ici</title>  
</topic>
```

Cependant, un topic *utile* aura un contenu supplémentaire.

Évaluation

Répondre aux questions suivantes pour vérifier votre compréhension des concepts de cette leçon.

Les noms de tous les éléments DITA sont prédéfinis et ne peuvent pas être modifiés

- Vrai
- Faux

En HTML, les noms de balises sont prédéfinis, mais dans DITA, ils peuvent être modifiés ou de nouveaux noms créés.

Les éléments définis dans DITA ont une hiérarchie et des relations très spécifiques

- Vrai
- Faux

La création structurée nécessite que la relation de tous les éléments soit clairement définie.

La seule façon de réaliser une création structurée consiste à utiliser le standard DITA

- Vrai
- Faux

Il existe d'autres normes, telles que DocBook, S1000D, NLM, etc. Vous pouvez également créer votre propre structure personnalisée à partir de zéro.

DITA est un standard ouvert, ce qui signifie (choisissez une réponse) :

- Tout le monde peut l'utiliser sans payer.
- Lorsque les mises à jour sont publiées, vous devez effectuer une mise à niveau et payer ou perdre l'accès à vos informations.
- Les mises à jour sont contrôlées par l'Organisation Internationale de Normalisation (ISO - International Standards Organization).

DITA est contrôlé par l'Organisation pour l'Avancement des Normes d'Information Structurée (OASIS - Organization for the Advancement of Structured Information Standards). Les standards ouverts sont gratuits.

Pour être valide, un topic doit avoir (choisissez une réponse) :

- un élément title
- un élément title et un attribut id
- un attribut href

Un élément est (choisissez tout ce qui s'applique) :

- une balise pour du contenu.
- inséré lorsque vous générez une sortie pour contrôler le formatage.
- identifié avec des chevrons (tel que <p>).
- un terme DITA pour l'indexation du contenu.

Dans DITA, l'unité de base de création et de réutilisation est (choisissez une réponse) :

- un conteneur d'attributs spécialisés
- un topic
- une liste à puces
- un élément

Parce que DITA est un format ouvert (choisissez tout ce qui s'applique) :

- Personne ne peut facturer pour les outils.
- Aucune de ces réponses n'est correcte.
- Vous pouvez seulement utiliser un éditeur de texte.
- Tout le monde peut utiliser le standard.

Chapter 2. Les Topics DITA

- Identifier les éléments communs à tous les topics.
- Citer les types de topics standards définis dans DITA.
- Identifier les éléments généralement présents dans chaque type de topic.
- Définir la spécialisation et les contraintes.

Un topic DITA est un élément constitutif du contenu.

L'un des principes fondamentaux de DITA est que différents types de contenu nécessitent différents conteneurs (types de topics). De cette manière, les éléments du topic peuvent être spécifiques aux informations décrites.

Tous les types de topics DITA sont liés à un type de topic commun ou générique. Le type de topic générique est alors « spécialisé » dans un certain nombre de types de topics spécifiques, chacun d'entre eux contenant un type d'informations spécifique. Les types de topics DITA « standard » incluent :

- Concept - contient les informations conceptuelles ; répond à la question « Pourquoi ? »
- Tâche (Task) — contient l'information procédurale étape-par-étape ; répond à la question « Comment ? »
- Référence (Reference) — contient les informations de référence ; répond à la question « Quoi ? »
- Entrée de Glossaire (Glossary entry) - définit un simple terme ; répond à la question « Qu'est-ce que ça signifie ? »

Cette leçon présente le type de topic générique, puis décrit chacun des types de topics standards et leur utilisation.

Related information

- [Guide de Style DITA, Chapitre 1](#)
- [Rédaction Technique 101, Chapitre 7](#)


Le type de thème générique

Tous les types de topics DITA sont basés sur un seul type de topic générique. Le type de topic générique est utilisé comme base pour créer les types de thèmes topics. De nombreux éléments et organisations du topic générique sont également utilisés par les types de topics spécifiques.

Tous les topics DITA ont cette structure générale :

```
<topic id="sample">
  <title>Le titre du topic vient ici</title>
  <shortdesc>Une courte description</shortdesc>
  <body>
    (La plupart des éléments viennent ici.)
  </body>
</topic>
```

- L'élément `<shortdesc>` (courte description) fournit un résumé du contenu du topic en 2-3 phrases.
- L'élément `<title>` est le seul élément requis dans un topic.
- L'élément `<body>` contient la majeure partie des informations du topic. Dans les types de topics spécifiques, l'élément `body` a un nom associé, tel que `<conbody>` pour les concepts et `<taskbody>` pour les tâches.

 **Note:** Bien que vous puissiez utiliser le type de topic générique pour la création de contenu, il est préférable d'utiliser l'un des types de topics spécifiques, tels que concept, tâche et référence.

Il existe d'autres éléments dans le type de topic générique, mais ce sont les plus courants.

Les topics Concept

Un topic concept répond à la question « Pourquoi ? ». Il fournit des informations générales sur un sujet que le lecteur doit connaître.

Les concepts contiennent généralement du texte et des listes, mais inclut également des notes, des tables, et des graphiques nécessaires à la compréhension des idées derrière un sujet particulier.

Les éléments courants utilisés dans un topic de type concept comprennent :

- `<conbody>` (le corps du topic concept)
- `<p>` (un paragraphe)
- `` (une liste à puce ou non ordonnée)
- `` (une liste numérotée ou ordonnée)
- `` (élément de liste à l'intérieur d'un `` ou ``)
- `<fig>` (une figure, comprenant un titre optionnel)
- `<image>` (un graphique à l'intérieur d'une figure ou dans le texte)
- `<section>` (une subdivision dans le thème, avec un titre optionnel)

Les topics Task

- Comprendre l'objectif du type de tâche DITA
- Identifier les composants clés d'une tâche
- Créer une tâche DITA

Un topic tâche répond à la question « Comment fais-je ? ». Cela inclut des instructions pas-à-pas pour compléter une procédure. DITA autorise aussi des résultats d'étapes, des graphiques, de notes et un niveau de sous-étapes.

Le contenu technique contient souvent énormément de tâches.

DITA fournit deux types de tâches :

- La tâche stricte
- La tâche générale

La tâche stricte exige que tous les éléments apparaissent dans un ordre spécifique, autorise seulement un élément `<example>`, et comporte deux éléments très formels pour les étapes (`<steps>` ou `<steps-unordered>`). La « tâche générale », également appelée « tâche laxiste » permet plus de flexibilité dans l'ordre des éléments, autorise plusieurs éléments `<example>` et autorise un élément `<steps-informal>` pour les étapes, qui peut contenir du contenu bien plus varié. Les tâches strictes sont appropriées pour des points qui exigent des instructions détaillées ; les tâches générales sont utiles pour les aperçus de processus.

Les éléments courants utilisés dans un topic tâche stricte comprennent :

- `<taskbody>` (le corps du topic tâche)
- `<steps>` (la séquence d'actions)
- `<step>` (chaque action individuelle)
 - `<cmd>` (l'action à faire ; c'est un élément requis dans un `<step>`)
 - `<info>` (information additionnelle à propos de l'étape)
 - `<stepresult>` (ce qui arrive après avoir réalisé l'action)
 - `<stepxmp>` (un exemple de comment réaliser l'étape)
- `<example>` (un exemple de comment réaliser la tâche entière)

Vidéo: [Aperçu du code d'un topic tâche DITA](#)

Les topics Reference

Un topic reference répond à la question « Qu'est-ce ? ». Un topic reference contient généralement des faits descriptifs, tels que la syntaxe d'une commande ou de l'appel à une fonction API, un tableau répertoriant les caractéristiques opérationnelles et les tolérances d'un périphérique, ou un tableau identifiant les éléments d'un écran du logiciel.

Les topics Reference n'incluent pas d'étapes ou d'informations de fond. Les topics Reference sont similaires à des entrées du dictionnaire ou fournissent uniquement des faits.

Les éléments courants utilisés dans un topic Reference comprennent :

- <refbody> (le corps du topic reference)
- <section> (une subdivision dans le thème reference, avec un titre optionnel)
- <table> (une table)
- <fig> (une figure, y compris un titre optionnel)
- <properties> (une liste de propriétés)
- <refsyn> (un diagramme de syntaxe)

Vidéo : [Aperçu du code d'une thème reference DITA reference](#)

Les topics Glossary Entry

Un topic d'entrée de glossaire répond à la question « Que signifie ce mot ou cette phrase ? » Les topics glossaire contiennent généralement un terme et une ou plusieurs définitions.

Les éléments courants utilisés dans le topic glossaire sont :

- <glossentry> (le type de thème d'entrée de glossaire)
- <glossterm> (le mot ou la phrase)
- <glossdef> (la définition du terme glossaire)

Video: [Aperçu de code du topic glossaire DITA](#)


La spécialisation

La spécialisation permet à l'architecte d'information de créer des éléments et des attributs qui conviennent mieux à votre organisation que ceux fournis par défaut dans DITA. Les éléments créés avec la spécialisation sont basés sur des éléments existants dans DITA.

Les organisations ont souvent recours à la spécialisation pour prendre en compte leurs exigences uniques, tel que :

- Création de nouveaux éléments pour contenir des informations spécifiques dans un ordre spécifique
- Création de nouveaux attributs pour identifier une information spécifique
- Création de nouveaux noms pour des éléments qui sont plus pertinent pour les auteurs.

Par exemple, vous pourriez exiger un ensemble d'éléments `<codeblock>` spécifique à un langage, tel que `<codeblocks-java>` ou `<codeblock-php>` pour faire la distinction entre des exemples de code Java et PHP. Cela pourrait aussi être réalisé en créant un nouvel attribut `language` pour l'élément `<codeblock>` : `<codeblock language="java">` ou `<codeblock language=php">`.

 **Important:** Attention à la spécialisation. Lorsque vous spécialisez, vous définissez des balises plus spécifiques, mais la spécialisation augmente le coût de la mise en œuvre. Vous devez équilibrer la valeur de la spécialisation par rapport aux coûts de mise en œuvre et de maintenance des spécialisations.

En plus de la spécialisation, les *contraintes* permettent aux architectes d'information d'éliminer les éléments inutiles ou indésirables. Par exemple, si votre organisation ne documente pas le code logiciel, votre architecte d'information peut éliminer les balises `<codeblock>` et `<codeph>` (expression de code). La contrainte réduit le nombre d'éléments présentés aux auteurs, ce qui peut leur faciliter la vie.

Évaluation

Répondre aux questions suivantes pour vérifier votre compréhension des concepts de cette leçon.

Faites glisser le type de thème le plus étroitement associé à chacun des éléments à côté de cet élément :

glossdef	topic glossaire
steps	topic tâche
conbody	topic concept
properties	topic référence

La spécialisation signifie l'ajout de nouveaux éléments basés sur des éléments existants.

- Vrai
- Faux

La spécialisation permet à un architecte d'information de créer de nouveaux éléments à partir d'éléments existants.

Les contraintes permettent à un architecte d'informations de supprimer des éléments inutiles.

- Vrai
- Faux

Les contraintes permettent à un architecte d'informations de supprimer des éléments.

Un topic est un élément de contenu qui aborde une question ou une idée

- Vrai
- Faux

Un topic est une unité de contenu. Les topics concept décrivent une idée ; les topics référence fournissent des informations sur un élément et les topics tâche fournissent des instructions pas à pas pour une procédure.

Pour chaque type de topic, identifiez la question à laquelle il répond :

Qu'est-ce?	topic référence
Que signifie ce terme ou cette phrase ?	topic glossaire
Comment fais-je ?	topic tâche
Pourquoi ?	topic concept

Lors de la création d'un topic concept, vous aurez souvent à (choisissez une réponse) :

- Éviter d'introduire de nouveaux termes ou expressions
- Décrire en détail comment accomplir une tâche

- Fournir les paragraphes de texte nécessaires pour comprendre les idées derrière un sujet particulier.

Les concepts sont fréquemment utilisés pour introduire de nouveaux mots ou expressions. Les instructions de tâches appartiennent aux topics tâches.

Parmi les catégories suivantes, lesquelles sont des types de topic dans la norme DITA (cochez toutes les réponses qui s'appliquent) ?

- Référence
- Tâche
- Procédure
- Référence croisée
- Concept
- Entrée de glossaire

DITA prend en charge les procédures, mais le type de topic est appelé tâche. Les références croisées DITA sont des éléments contenus dans les topics.

Quelle sorte de topic utilise un élément glossterm ?

- Entrée de glossaire
- Tâche
- Concept

les topics tâche incluent les composants suivants (choisissez toutes les réponses qui s'appliquent) :

- Sous-étapes facultatives
- Aperçu du contenu
- Résultats d'étape facultatifs
- Information étape par étape

Vous pouvez fournir des informations contextuelles avec l'élément context, mais les informations d'aperçu détaillé doivent figurer dans un topic concept.

Associez les éléments du glossaire à leur signification :

<code><glossterm></code>	Mot ou phrase
<code><glossdef></code>	Définition du mot ou de la phrase
<code><glossentry></code>	Conteneur pour une entrée de glossaire

Chapter 3. Métadonnées

- Définir les métadonnées.
- Décrire l'utilisation des métadonnées.
- Montrer comment ajouter des métadonnées aux thèmes.

Métadonnées signifie « au-delà des données » ou données sur les données. Dans le contenu XML, les métadonnées vous permettent de classifier et de manipuler l'information. Dans DITA, vous pouvez assigner des métadonnées aux thèmes, à des éléments individuels, etc.

Related information

- [Guide de Style DITA, Chapitre 8, Métadonnées, traitement conditionnel et indexation](#)
- [FAQ sur l'identification de langue de opentag.com \(adresse l'utilisation de xml:lang pour l'identification de langue\)](#)

Que sont les métadonnées ?

Les *métadonnées* fournissent de l'information sur l'information. Par exemple, les applications de traitement de texte ont souvent des propriétés de document qui indiquent qui a créé le fichier et à quelle date il a été modifié pour la dernière fois. L'auteur et la date de modification dans les propriétés du document ne font pas partie du texte affiché. Les propriétés du document sont des métadonnées sur le document lui-même.

Les métadonnées fournissent différents moyens de classifier l'information. Par exemple, vous pouvez :

- Étiqueter l'information avec un type de document, comme « livre blanc » ou « proposition »
- Associer l'information avec un numéro de produit spécifique
- Indiquer un niveau d'utilisateur, comme « débutant », « intermédiaire » ou « expert »
- Indiquer l'audience du contenu, comme « responsable » ou « employé »
- Fournir une information sur le statut d'une rubrique, comme « brouillon », « révision » ou « final »
- Fournir une date d'expiration lorsque l'information n'est plus valide

Après avoir assigner des métadonnées à votre contenu, vous (ou votre lecteur) pouvez les utiliser pour :

- Trouver l'information efficacement

- Soutenir les efforts de gouvernance (contrôle lorsque des informations sont rendues publiques ou supprimées d'un site internet)
- Personnaliser l'information (votre lecteur ne peut demander que des informations pour débutants)
- Filtrer les informations pour différentes variantes de livraison (telles que Windows ou Macintosh)
- Réutiliser les informations efficacement
- Indiquer la langue utilisée dans le contenu
- Gérer le statut du projet

Dans DITA ou autres formats XML, les métadonnées sont encodées à l'intérieur de vos topics, soit au niveau du topic, soit sur des éléments spécifiques. Vous pouvez manipuler le contenu DITA en utilisant ces valeurs de métadonnées..

DITA nécessite une métadonnée ; chaque topic doit avoir un attribut id. Ces identifiants sont souvent attribués automatiquement par des outils de création ou des systèmes de gestion de contenu. Voici un exemple de topic avec un attribut id :

```
<topic id="xyz">
  <title>Votre titre ici</title>
  <body>
    ....
  </body>
</topic>
```

D'autres éléments peuvent utiliser l'attribut id. Par exemple, l'attribut id sur une table vous permet de créer une référence croisée vers celle-ci. La plupart des outils de création attribuent automatiquement des ID à des éléments pouvant faire l'objet de références croisées.

Votre organisation ou votre configuration DITA peut nécessiter des métadonnées supplémentaires. Il est courant d'inclure la langue du thème avec l'attribut xml:lang.

```
<topic id="xyz" xml:lang="en-us">
<topic id="abc" xml:lang="de-de">
```

En règle générale, vos lecteurs ne voient pas les métadonnées. ils voient les résultats du filtrage ou utilisent les métadonnées lorsqu'ils recherchent le contenu.

Comment utiliser les métadonnées ?

Vous pouvez assigner des métadonnées au contenu DITA en plusieurs endroits différents :

- Au niveau du topic
- Au niveau de l'élément

- Au niveau du fichier carte (un fichier carte vous permet de collecter plusieurs rubriques pour créer un document, un système d'aide, etc ; un module ultérieur explique plus en détail les fichiers carte)

Au niveau des topics, DITA fournit un élément `<prolog>` dans lequel vous pouvez stocker des métadonnées pour l'ensemble du sujet. Voici un exemple de métadonnées des thèmes de base :

```
<topic id="xyz">
  <title>Exemple de Métadonnées</title>
  <prolog>
    <author>Sarah O'Keefe, Scriptorium</author>
    <critdates>
      <created date="2015-05-01"/>
    </critdates>
  </prolog>
  <body>
    <p>Le contenu du corps va ici</p>
  </body>
</topic>
```

L'auteur est spécifié dans l'élément `<author>`, et `<critdates>` fournit un emplacement pour les éléments date `<created>` et `<revised>`.

Certains éléments utiles du prologue sont :

- `<author>` (l'auteur du contenu)
- `<critdates>` (dates critiques telles que `<created>` et `<revised>`)
- `<copyright>` (année du droit d'auteur `<copyryear>`, et détenteur du droit d'auteur `<copyrholder>`)
- `<vrn>` (version du produit, release, et information de modification)

⚠ Important: Utilisez les métadonnées `<prolog>` uniquement pour les informations système, telles que l'auteur et les dates de création / révision. DITA n'utilise pas les métadonnées `<prolog>` pour filtrer les topics.

Au niveau élément, vous utilisez généralement des attributs dans les éléments pour spécifier les métadonnées. Voici un exemple :

```
<step>
  <cmd>Localisez la boîte de purée pour canard.</cmd>
  <info audience="novice">Consultez le côté de la boîte de purée de
    canard pour déterminer
      la quantité de purée dont vos canards ont besoin.</info>
</step>
<step>
  <cmd>Mesurer la purée pour vos canards. </cmd>
</step>
<step>
  <cmd>Verser dans le mixeur.</cmd>
</step>
<step>
  <cmd>Mettre dans la mangeoire.</cmd>
```

```
</step>
```

Seuls les novices doivent se rappeler que la boîte fournit des mesures. Lorsque vous générez votre sortie, vous pouvez supprimer l'élément `<info>` avec les métadonnées `audience="novice"` pour un public de niveau expert.

Au niveau de la carte, vous pouvez spécifier des métadonnées dans les éléments `<topicref>` faisant référence à des thèmes. Cela vous permet de supprimer des thèmes entiers lorsque vous générez une sortie. Voici un exemple :

```
<topicref href="abc.dita">
<topicref href="def.dita" audience="novice">
```

Par défaut, DITA vous fournit trois attributs prenant en charge le filtrage ou le traitement conditionnel. Ce sont :

- audience
- product
- platform

Si vous avez besoin d'attributs de filtrage supplémentaires ou différents, votre architecte d'informations devra recourir à la spécialisation pour définir des métadonnées supplémentaires. Certaines exigences communes sont :

- customer, pour une information spécifique client
- region, pour des informations qui ne s'appliquent qu'à des zones géographiques spécifiques
- product-family, pour les informations qui s'appliquent à un groupe de produits

Évaluation

Répondre aux questions suivantes pour vérifier votre compréhension des concepts de cette leçon.

Où voudriez-vous trouver un élément `<author>` ?

- DITA ne permet pas d'utiliser un élément auteur ; vous devez utiliser un attribut auteur.
- À la fin du corps du topic.
- Dans l'élément `<prolog>`.

DITA ne fournit pas d'attribut auteur. Le corps du topic est suivi par `<related-links>`.

Les métadonnées sont des informations sur le contenu.

- Vrai
- Faux

Les métadonnées sont souvent décrites comme étant "de l'information sur l'information". Elles fournissent un contexte supplémentaire à votre contenu.

Avec les métadonnées et le filtrage de sortie, vous pouvez créer une expérience de contenu personnalisé au lecteur.

- Vrai
- Faux

Si votre contenu inclut des attributs de métadonnées, tel que platform ou product, vous pouvez filtrer à partir de la valeur des métadonnées pour créer différents résultats.

Durant le process de publication, vous utilisez les éléments de métadonnées pour inclure ou exclure du contenu.

- Vrai
- Faux

Le filtrage est l'un des objectifs premiers des métadonnées.

Faites attention lorsque vous créer des métadonnées - vos lecteurs voient toujours ces tags.

- Vrai
- Faux

Il est possible d'exposer les métadonnées à vos lecteurs, mais par défaut, ils ne les voient pas.

Associez la métadonnée à sa signification :

L'élément <vrml> dans le prologue	La version du contenu
L'attribut xml:lang à la racine (par exemple, <topic> ou <concept>)	La langue du contenu
L'attribut audience d'un élément	L'audience cible du contenu
L'attribut platform d'un élément	La plateforme cible du contenu
L'élément <author> dans le prologue	La personne qui a créé le contenu

Les métadonnées stockées dans l'élément prologue sont des informations sur le topic entier.

- Vrai
- Faux

Le prologue est au niveau du topic. Pour une métadonnée de niveau élément, utilisez les attributs de l'élément. Souvenez-vous que vous ne pouvez pas filtrer efficacement sur une métadonnée du prologue.

Si vous assignez une métadonnée au prologue, vous ne pouvez assigner de métadonnée supplémentaire aux éléments de contenu dans le topic.

- Vrai
- Faux

Le prologue du topic contient les métadonnées de niveau topic. Vous pouvez aussi assigner une métadonnée au niveau élément en utilisant les attributs.

Les métadonnées sont utiles car elles vous permettent de (choisissez ce qui s'applique) :

- Aucune des propositions.
- Créer des résultats conditionnés
- Écrire du contenu
- Trouver du contenu
- Créer des entrées pour le glossaire

Vous écrivez le contenu dans les éléments ; les métadonnées sont des informations sur le contenu. Vous utilisez des topics GlossEntry pour créer les entrées du glossaire.

La métadonnée suivante est requise pour un topic (sélectionnez toutes celles qui s'appliquent) :

- `<title id="xxx">`
- `<topic author="xxx">`
- `<topic id="xxx">`

L'attribut id est requis pour le topic, mais pas pour son titre. L'information sur l'auteur n'est pas requis. Quand vous le fournissez, utilisez l'élément `<author>` dans l'élément `<prolog>`.

Chapter 4. Création de contenu DITA

- Décrire comment créer un topic DITA dans un éditeur de texte.
- Décrire comment créer un topic DITA dans l'éditeur XML <oXygen/>
- Décrire comment créer des notes, des précautions et autres avertissements.
- Décrire comment créer des listes à puces et numérotées.
- Distinguer les éléments bloc et intégrés

Related information

- [Éléments bloc contre intégré, webdesignfromscratch.com](http://webdesignfromscratch.com)
- [Guide de style DITA, Chapitre 3, Syntaxe et balisage](#)
- [Darwin Information Typing Architecture \(DITA\) Version 1.2, OASIS Standard](#)

Créer un topic dans un éditeur de texte

Les fichiers DITA sont au format XML et XML est du texte clair. Par conséquent, vous pouvez créer un fichier XML dans n'importe quel éditeur de texte.

Un topic DITA de base nécessite une déclaration XML, une déclaration DOCTYPE, un topic, un attribut id sur le topic et un titre. C'est tout.

Voici un topic minimal valide :

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE topic PUBLIC "-//OASIS//DTD DITA Topic//EN" "topic.dtd">
<topic id="monpremiertopic">
  <title>Bonjour le monde</title>
</topic>
```

Ce topic est valide, mais il n'est pas particulièrement utile car il n'a aucun corps de contenu. Pour rendre un topic utile, vous avez besoin de quelque chose de plus semblable à ceci :

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE topic PUBLIC "-//OASIS//DTD DITA Topic//EN" "topic.dtd">
<topic id="monpremiertopic">
  <title>Bonjour le monde</title>
  <body>
    <p>Voici un paragraphe</p>
    <ul>
      <li>les listes à puces sont sympas</li>
      <li>surtout si vous avez au moins deux points</li>
    </ul>
  </body>
</topic>
```


```

</ul>
<note>Et n'oubliez pas les notes.</note>
</body>
</topic>


```

Créer un topic dans oXygen

Lire un fichier DITA dans un éditeur de texte peut être un peu difficile. Les informations importantes ne ressortent pas et la hiérarchie des éléments peut ne pas être apparente (car les espaces et les retours à la ligne n'ont aucune signification pour XML). La plupart des gens trouvent beaucoup plus facile d'utiliser un éditeur visuel (tel que SyncROSoft <oXygen/>) pour éditer des fichiers DITA, où le contenu du fichier DITA est présenté sous une forme modérément formatée.

 **Note:** Un éditeur visuel tel que <oXygen/> peut ressembler à un traitement de texte, mais le formatage que vous voyez à l'écran n'est qu'une possibilité de la manière dont la sortie finale sera présentée. Ne vous fiez pas à l'apparence du texte (fins de ligne, polices, styles de caractère, etc.) pour savoir à quoi ressemblera votre contenu une fois publié.

Pour créer un topic dans oXygen, suivez ces étapes :

1. Cliquer sur l'icône Nouveau (.
2. Naviguez vers Modèles du framework/DITA/Topics, cliquez sur l'entrée Topic, puis sur Créer. oXygen crée un nouveau thème avec du contenu réservé

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE topic PUBLIC "-//OASIS//DTD DITA Topic//EN" "topic.dtd">
<topic id="topic_xks_flr_bs">
  <title></title>
  <body>
    <p></p>
  </body>
</topic>

```

oXygen crée automatiquement un identifiant de topic unique. Vous pouvez le changer si vous le souhaitez, mais souvenez-vous que les identifiants doivent être uniques et doivent respecter des règles strictes. Dans la vue Auteur, avec les balises activées, vous voyez ceci :



Vous pouvez maintenant cliquer à l'intérieur du p (qui est un espace réservé pour une balise <p>) et commencer à taper votre contenu. Pour créer un nouveau paragraphe, appuyez deux fois sur Retour. Si vous souhaitez passer à un autre élément, appuyez une fois sur la touche Retour et choisissez dans le menu déroulant qui apparaît à l'emplacement du curseur.

Notes, mises en garde, et alertes

L'élément <note> vous permet de créer des notes, des mises en garde, des alertes et autres avertissements.

```
<note>
La plupart des canards qui s'alimentent en surface sont agiles et
s'envoleront facilement. Approchez ces canards lentement.
</note>
```

L'attribut type de l'élément <note> vous permettent d'indiquer les différentes sortes de notes. DITA défini un nombre de valeurs pour l'attribut type :

- note (par défaut)
- attention

- danger
- caution
- important
- tip

```
<note type="warning">
Ne donnez pas de pain aux canards. Ce n'est pas un aliment naturel pour eux
et peut être malsain.
C'est bien mieux de leur donner du raisin ou du maïs concassé.
</note>
```

Ce sont les valeurs d'attributs les plus courantes. Des valeurs d'attribut supplémentaires sont disponibles, ainsi qu'une valeur d'attribut fourre-tout « other », que vous utilisez avec l'attribut `othertype` pour spécifier un autre type de note.

Listes à puces et numérotées

Les listes les plus communément utilisées dans DITA sont les listes non-ordonnées (« à puces » - ``) et ordonnées (« numérotées » - ``).

Les listes ordonnées et non ordonnées contiennent un ou plusieurs éléments d'élément de liste (``). Les éléments `` montrés dans ces exemples sont assez simples. Cependant, un élément `` peut contenir des données de texte et les éléments bloc et en-ligne les plus courants.

```
<p>Le cycle de vie d'un canard est similaire à celui de la plupart des
oiseaux :</p>
<ol>
  <li>Oeuf</li>
  <li>Éclosion</li>
  <li>Caneton</li>
  <li>Jeune</li>
  <li>Adulte</li>
</ol>
```

On notera que l'auteur n'ajoute pas de numéros à cette liste ordonnée ; comme en HTML, les numéros sont insérés lors de la génération de la sortie. Pour vous aider à visualiser les choses, la plupart des éditeurs visuels vous montreront les numéros des points dans une liste numérotée.

Une liste ordonnée est utile pour montrer des choses qui ont une séquence, cependant, si vous avez besoin de décrire un processus, vous devriez plutôt utiliser l'élément `<steps>` dans un topic tâche.

Vous pouvez imbriquer une liste en commençant un nouvel élément `` ou `` dans un élément ``.

```
<p>Les canards ont ces caractéristiques :</p>
<ul>
  <li>Pieds palmés pour nager.</li>
```

```

<li>Pattes plus courtes que les autres oiseaux aquatiques.</li>
<li>Bec particulier.
  <ul>
    <li>Large et plat pour filtrer la nourriture.</li>
    <li>Long et étroit pour la pêche (Grand Harle)</li>
  </ul>
</li>
</ul>

```

Lorsque la liste est finalement sortie, le caractère de puce ou le numéro utilisé dans la liste est déterminé par la feuille de style utilisée pour la sortie et la position dans la liste dans la hiérarchie. Si une liste est imbriquée, la feuille de style déterminera le niveau d'indentation de la liste.

En plus de la liste non ordonnée (), DITA définit un élément simple liste (<sl>) que vous pouvez utiliser pour les listes ne contenant que quelques mots dans chaque élément. L'élément dans une liste simple utilise l'élément <sli>.

```

<p>Il y a plusieurs types de canards :</p>
<sl>
  <sli>Les canards forestiers</sli>
  <sli>Les canards de surface</sli>
  <sli>Les canards plongeurs</sli>
  <sli>Les canards piscivores</sli>
</sl>

```

Éléments de bloc contre intégrés

La séparation du contenu et la mise en forme furent probablement la première chose qui vous ayez appris au sujet d'XML.

Il existe toutefois quelques cas où la mise en forme est intégrée au contenu DITA/XML/HTML lui-même. La distinction entre les éléments de bloc et les éléments intégrés est l'un de ces cas :

Un *élément de bloc* est un élément de niveau paragraphe, tel que la balise <p> (paragraphe), (point d'une liste), ou <codeblock> (code logiciel). En règle générale, un élément de bloc est séparé verticalement sur la page des informations qui le précèdent et le suivent. Ainsi, si vous avez deux balises <p> à la suite, elles sont séparées par un saut de ligne lors du rendu des informations. Notez que tout caractère de rupture de ligne dans XML lui-même n'est pas pertinent.

```

<p>Ceci est un paragraphe. </p><p>Un second paragraphe. </p>

```

Les résultat dans la vue Auteur d'oXygen :

Ceci est un paragraphe.

Un second paragraphe.

Un *élément intégré* est un élément qui fournit un balisage à l'intérieur d'un paragraphe, tel que `<tm>` (marque déposée), `<i>` (italique), `<xref>` (référence croisée), ou `<term>` (terme). Les éléments intégrés, par défaut, n'entraînent pas de sauts de ligne.

```
<p>Un élément intégré <i>n'entraîne pas</i> de saut de ligne. </p>
<p>Les sauts de ligne dans le fichier XML
ne sont pas <i>pertinents</i>
pour le formatage de la sortie.</p>
```

oXygen ignore les sauts de ligne dans le code lorsque vous consultez la vue Auteur :

Un élément intégré *n'entraîne pas* de saut de ligne.

Les sauts de ligne dans le fichier XML ne sont pas *pertinents* pour le formatage de la sortie.

XML lui-même ne fait pas la distinction entre les éléments en bloc et les éléments intégrés. Ils utilisent le même balisage, vous devez donc savoir quels éléments sont des blocs et lesquels sont intégrés pour déterminer le mode de présentation des informations. Les éléments intégrés sont similaires aux styles de caractère des outils de traitement de texte. Cependant, il existe d'autres éléments intégrés, tels que `<fn>` (note de bas de page) et `<indexterm>` (entrée d'index) qui sont traités différemment. Par exemple, les entrées d'`<indexterm>` sont traitées pour créer un index en fin d'ouvrage. Le contenu textuel de `<indexterm>` n'apparaît généralement pas à l'emplacement où l'élément intégré se produit.

```
<p>Un élément intégré<indexterm>élément intégré</indexterm> <i>n'entraîne
pas</i> de saut de ligne.</p>
```

Évaluation

Répondre aux questions suivantes pour vérifier votre compréhension des concepts de cette leçon.

Contenu du thème

L'élément `<body>` contient la majeure partie du contenu utile d'un topic.

- Vrai
- Faux

Les éléments `<shortdesc>` et `<related-links>` contiennent du contenu, mais le contenu le plus utile est généralement dans l'élément `<body>`.

Création de mise en garde

Pour créer une mise en garde, vous devez utiliser l'élément `<warning>`

- Vrai
- Faux

Pour créer une mise en garde, vous utilisez `<note type="warning">`.

Listes numérotées

La séquence d'étiquettes utilisée dans une liste ordonnée (``) appartient à l'auteur du topic.

- Vrai
- Faux

La séquence d'étiquettes est déterminée au moment de la sortie par la feuille de style du générateur de sortie.

Bloc et intégré

XML lui-même ne fait pas la distinction entre les éléments de bloc et les éléments intégrés.

- Vrai
- Faux

La distinction entre les éléments de bloc et les éléments intégrés dépend de la manière dont les informations sont présentées en sortie.

Chapter 5. Tableaux

- Distinguer les tableaux simples et les tableaux.
- Décrire les éléments d'une simpletable.
- Décrire les éléments d'un tableau.
- Présenter les meilleures pratiques pour les tableaux DITA.

Les tableaux occupent une position intermédiaire difficile entre la mise en forme et la structure. Cependant, les tableaux constituent souvent le conteneur approprié pour le contenu.

DITA comprend deux types de tableaux :

Les tableaux simples

L'élément `<simpletable>` prend en charge les tableaux de base avec une personnalisation minimale

Les tableaux

L'élément `<table>` peut prendre en charge des tableaux complexes avec des lignes et colonnes étendues (straddles) et des propriétés d'affichage précises

La plupart des langages de balisage utilisent le modèle de table CALS, créé dans le cadre d'une initiative du département de la Défense des États-Unis. Les tables HTML utilisent également le modèle CALS. Par conséquent, si vous connaissez le balisage de table HTML, vous reconnaîtrez les similitudes dans les tableaux DITA.

Les tableaux dans DITA sont toujours décrits ligne par ligne.

Tableaux simples

Les tableaux simples sont idéaux pour la représentation de base du contenu dans les colonnes et les lignes. Vous pouvez utiliser des lignes d'en-tête, mais vous ne pouvez pas utiliser un titre de tableau, des colonnes ou des lignes fractionnées ni définir la largeur des colonnes.

Tableaux

L'élément `<table>` permet des arrangements plus complexes du contenu tabulaire. Il autorise les titres de tableau, les colonnes ou lignes étendues, plusieurs groupes de lignes et les largeurs de colonne spécifiées par l'utilisateur.

Autres tableaux

Il existe deux autres types de tableau très spécifiques (basés sur l'élément `<simpletable>`) :

- `<properties>`, que vous utilisez dans les topics référence pour définir une liste de propriétés.
- `<choicetable>`, que vous utilisez dans les topics tâches pour présenter les différences entre divers choix.


Related information

- [Guide de Style DITA, Chapitre 3, section Tableaux](#)
- [Informations contextuelles sur CALS](#)

L'élément simpletable

les tableaux simples (<simpletable>) utilisent les éléments suivants pour représenter et organiser les données tabulaires :

Élément	Description	Quantité
<sthead>	Une ligne d'en-tête	Il peut y avoir au plus une ligne d'en-tête.
<strow>	Une ligne	Il doit y avoir au moins une ligne, mais le tableau peut contenir plusieurs lignes.
<stentry>	Une entrée de tableau simple	Chaque ligne peut contenir un ou plusieurs éléments <stentry>, mais chaque ligne (y compris la ligne d'en-tête) doit inclure le même nombre d'éléments <stentry>.

 **Note:** Ce tableau est construit en utilisant un élément <simpletable>.

L'élément <stentry> peut contenir du texte ou d'autres éléments de bloc ou en-ligne.

Voici un exemple d'un tableau avec une ligne d'en-tête, deux lignes et deux colonnes.

```
<simpletable>
  <sthead>
    <stentry>En-tête de colonne 1</stentry>
    <stentry>En-tête de colonne 2</stentry>
  </sthead>
  <strow>
    <stentry>ligne 1, colonne 1</stentry>
    <stentry>ligne 1, colonne 2</stentry>
  </strow>
  <strow>
    <stentry>ligne 2, colonne 1</stentry>
    <stentry>ligne 2, colonne 2</stentry>
  </strow>
</simpletable>
```

Visuellement, le tableau ressemble à ceci :

En-tête de colonne 1	En-tête de colonne 2
ligne 1, colonne 1	ligne 1, colonne 2
ligne 2, colonne 1	ligne 2, colonne 2

L'élément table

Les tableaux DITA (<table>) utilisent les éléments suivants pour représenter et organiser les données tabulaires :

Élément	Description	Quantité
<title>	Contient le titre du tableau.	Zéro ou un.
<tgroup>	Contient les spécifications de colonnes, les lignes d'en-tête et les lignes de corps.	Chaque tableau contient au moins un élément <tgroup>.
<colspec>	Définit la largeur des colonnes et les informations d'identification.	Il devrait y avoir un élément <colspec> par colonne dans votre tableau.
<thead>	Contient les lignes d'en-tête du tableau.	Un <tgroup> peut contenir, au plus, un élément <thead>.
<tbody>	Contient les lignes du corps du tableau.	un <tgroup> doit avoir un et seulement un élément <tbody>.
<row>	Contient une seule ligne du tableau.	Un élément <thead> ou <tbody> peuvent contenir un nombre quelconque d'éléments <row>.
<entry>	Contient les données d'une cellule du tableau. Peut contenir du texte ou d'autres éléments de bloc ou en-ligne.	Un élément <row> peut avoir un ou plusieurs éléments <entry>.

Voici un exemple d'une structure minimale d'un tableau DITA (une ligne d'en-tête, une ligne de corps, et trois colonnes). Notez que dans ce cas, le <tgroup> spécifie uniquement l'attribut cols, qui est requis :

```
<table>
  <tgroup cols="3">
    <thead>
      <row>
        <entry>colonne 1, ligne d'en-tête</entry>
        <entry>colonne 2, ligne d'en-tête</entry>
        <entry>colonne 3, ligne d'en-tête</entry>
      </row>
    </thead>
    <tbody>
      <row>
        <entry>colonne 1, ligne de corps</entry>
        <entry>colonne 2, ligne de corps</entry>
```

```

        <entry>colonne 3, ligne de corps</entry>
    </row>
</tbody>
</tgroup>
</table>

```

Le résultat ressemble à ceci :

colonne 1, ligne d'en-tête	colonne 2, ligne d'en-tête	colonne 3, ligne d'en-tête
colonne 1, ligne de corps	colonne 1, ligne de corps	colonne 1, ligne de corps

Voici un tableau plus complexe :

```

<table>
  <title>Mon premier tableau</title>
  <tgroup cols="2">
    <colspec colname="c1" colnum="1" colwidth="1*" />
    <colspec colname="c2" colnum="2" colwidth="4*" />
    <thead>
      <row>
        <entry>ligne d'en-tête, colonne 1</entry>
        <entry>ligne d'en-tête, colonne 2</entry>
      </row>
    </thead>
    <tbody>
      <row>
        <entry>ligne 1, colonne 1</entry>
        <entry>ligne 1, colonne 2</entry>
      </row>
      <row>
        <entry namest="c1" nameend="c2">Cette cellule s'étend sur
deux colonnes.</entry>
      </row>
      <row>
        <entry morerows="1">Cette cellule s'étend sur deux lignes.</
entry>
        <entry>ligne 3, colonne 2</entry>
      </row>
      <row>
        <entry>ligne 4, colonne 2</entry>
      </row>
    </tbody>
  </tgroup>
</table>

```

ce qui donne :

ligne d'en-tête, colonne 1	ligne d'en-tête, colonne 2
ligne 1, colonne 1	ligne 1, colonne 2
Cette cellule s'étend sur deux colonnes.	
Cette cellule s'étend sur deux lignes.	ligne 3, colonne 2
	ligne 4, colonne 2

Quelques notes sur le tableau :

- L'élément `<colspec>` utilise des attributs pour spécifier les noms de colonne (`colname`), les numéros (`colnum`) et les largeurs (`colwidth`). Dans cet exemple, les colonnes sont les largeurs sont « 1* » et « 4* ». L'astérisque indique que les paramètres sont proportionnels. La première colonne s'étend donc sur 20% de la largeur disponible et la deuxième colonne, 80%.
- Chaque ligne a un élément `<row>` avec des éléments `<entry>` pour chaque cellule.
- Notez que la ligne 2 a seulement un seul `<entry>`. C'est parce que l'entrée s'étend sur les deux colonnes, ce qui est spécifié avec les attributs `name-start` (`namestart`) et `name-end` (`nameend`).
- À la ligne 3, la première entrée couvre les lignes 3 et 4, vous avez donc un attribut `morerows`.

Le moyen le plus simple et le moins stressant de configurer correctement le code du tableau consiste à utiliser un éditeur qui gère ces paramètres pour vous. Coder manuellement les tableaux n'est pas pour les âmes sensibles.

Les meilleures pratiques pour les tableaux

- Toujours encadrer le contenu de `<entry>` du tableau dans des balises `<p>`. Si vous oubliez cela, le contenu du tableau peut être formaté de manière inattendue.
- Bien que la spécification DITA vous permet d'imbriquer des tableaux, c'est une Mauvaise Idée™.
- Dans la mesure du possible, envisagez d'organiser le contenu de votre tableau de manière à inclure davantage de lignes au lieu de plusieurs colonnes. Lorsque vous envoyez le contenu vers une sortie, des problèmes de pagination peuvent faire apparaître de manière inattendue de nombreuses colonnes.
- Dans de nombreux outils de traitement de texte, les utilisateurs utilisent des tableaux pour mettre en forme le contenu, par exemple les listes indentées. Dans DITA, cela n'est pas fait et peut entraîner des résultats inattendus.
- Considérez l'expansion du texte de localisation dans les tableaux. Si possible, concevez le contenu pour une expansion du texte pouvant atteindre 40%.
- Le standard DITA autorise les graphiques dans les cellules du tableau. Cependant, dans la plupart des cas, il est préférable de limiter ces graphiques à de petites icônes dans les cellules du tableau.
- Considérez les sorties possibles lorsque vous créez des tableaux. Le rendu du tableau peut être très différent (et difficile à lire) sur un petit écran, tel qu'un appareil mobile.
- Dans certains cas, vous pouvez envisager d'utiliser d'autres éléments, tels que des listes de définitions, qui peuvent être restitués sous forme de tableau ou de liste de style glossaire, en fonction de la sortie cible.

Évaluation

Répondre aux questions suivantes pour vérifier votre compréhension des concepts de cette leçon.

Associez les éléments table CALS suivants avec leur fonction.

<tgroup>	Définit les colonnes, les lignes, les entêtes et les bas
<colspec>	Spécifie les largeurs et les intervalles
<table>	Le conteneur principal pour un tableau
<tbody>	Contient les colonnes et les lignes
<entry>	Une simple cellule du tableau

L'élément <simpletable> autorise les éléments suivants (choisissez tout ceux qui s'appliquent) :

- <thead> (ligne d'entête)
- <tbody> (corps du tableau)
- <strow> (ligne du tableau)
- <title> (titre du tableau)
- <stentry> (cellule du tableau)

Seules les tables CALS définissent le corps du tableau. Seules les tables CALS peuvent avoir des titres.

Quels éléments table DITA supporte-t-il (Choisissez une réponse) ?

- <complextable> et <simpletable>
- <table> et <simpletable>
- <ditatable> et <simpletable>
- <ditatable> et <calstable>

Les deux éléments table dans DITA sont <table> et <simpletable>.

Les tables et simpletables supportent toutes deux les lignes d'en-tête.

- Vrai
- Faux

Les deux types de tableaux dans DITA supportent les lignes d'en-tête, mais les simpletables ne supportent pas les titres.

Images dans les tables

DITA vous permet d'inclure des images dans les cellules d'un tableau.

- Vrai
- Faux

Il est possible d'inclure des images dans les cellules d'un tableau, mais en général, elles devraient être limitées à de petites icônes.

Chapter 6. Création de relations entre topics

- Décrire l'objectif et l'utilisation d'un fichier carte (map).
- Résumer comment le résultat est généré.
- Décrire l'objectif et l'utilisation des références croisées (xref) et des liens connexes.
- Décrire l'objectif et l'utilisation d'un tableau de relations (reltable).
- Décrire l'objectif et l'utilisation d'un contenu de référence (conref).

Les topics sont indépendants, mais le contenu d'un topic a souvent des liens avec d'autres topics. Un fichier carte, par exemple, décrit la séquence et la hiérarchie des topics d'un livrable.

Vous pouvez créer des relations entre topics de plusieurs façons différentes :

- Fichiers carte
- Conrefs
- Références croisées
- Liens connexes
- Tables de relations

Related information

- [Le Guide de Style DITA, Chapitre 2 \(fichiers carte et tables de relations\)](#)
- [Guide de Style DITA, Chapitre 6 \(références croisées\)](#)
- [Guide de Style DITA, Chapitre 7 \(réutilisation du contenu\)](#)

Les fichiers carte

Les fichiers carte sont la façon dont vous organisez le contenu pour la livraison. Ils ressemblent à une table des matières : ils créent une séquence et une hiérarchie entre les topics. Lorsque vous générez un fichier PDF ou un système d'aide à partir d'un fichier carte, votre lecteur voit les rubriques dans l'ordre et la hiérarchie établis par le fichier carte.

En général, vous n'ajoutez pas toutes les runiques disponibles à un fichier carte, mais uniquement celles que vous souhaitez inclure dans un livrable. En outre, vous pouvez inclure la même rubrique dans plusieurs fichiers carte, ce qui est un autre exemple de réutilisation dans DITA.

Vidéo: [Exemple de fichier carte DITA](#)

Les fichiers cartes sont principalement constitués des composants suivants :

- les éléments `topicref`, qui fournissent un lien vers un topic spécifique
- les éléments `mapref`, qui fournissent un lien vers une autre carte

Dans un fichier carte, vous mettez les `topicrefs` dans l'ordre, de haut en bas, pour indiquer une séquence. Pour indiquer une hiérarchie, vous imbriquez les topics. Considérons l'exemple suivant :

```
<map>
  <title>Ma première carte</title>
  <topicref href="canards.dita">
    <topicref href="variete.dita"/>
    <topicref href="taille.dita"/>
    <topicref href="nids.dita"/>
  </topicref>
</map>
```

Pour plus de commodité, le code est indenté. Mais ce qui compte, c'est que le premier `topicref` (canards) englobe les trois autres fichiers. Les sujets `variete.dita`, `taille.dita` et `nids.dita` sont tous subordonnés au sujet `canards.dita`. Le résultat est une table des matières structurée comme ceci :

- Canards
 - Variété
 - Taille
 - Nids

En plus des liens vers des topics, vous pouvez référencer des fichiers carte dans des fichiers carte. Dans cette approche, les cartes subordonnées (sous-cartes) sont généralement une collection de contenu associé. Par exemple, vous pouvez créer un fichier carte de niveau chapitre pour chaque chapitre d'un livre, puis référencer ces fichiers carte de niveau chapitre dans votre fichier carte principal de niveau livre.

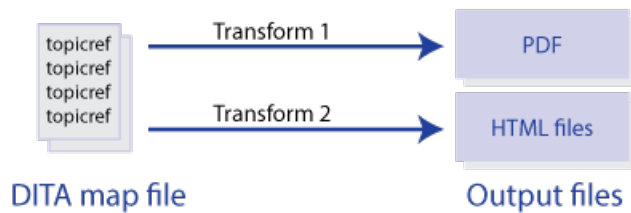
À la place d'un `topicref`, une référence vers une carte utilise un `mapref`. Si vous souhaitez utiliser le contenu de la carte canard en tant que composant d'une autre carte, vous devez l'insérer dans la carte parente, comme indiqué ci-dessous:

```
<topicref href="poissons.dita">
<topicref href="oiseauxmarins.dita">
<bmapref href="canard.ditamap" format="ditamap"/>
```

Vidéo: [Réutilisation d'un fichier carte dans un fichier carte](#)

Restitution par transformation des fichiers cartes

La plupart des livraisons de contenus n'utilisent pas de fichiers DITA. À la place, vous appliquez des feuilles de styles aux fichiers DITA pour créer des formats HTML, PDF ou autres. Ce processus est appelée *transformation*.



Le DITA Open Toolkit est une collection de feuilles de style, principalement écrites en Extensible Stylesheet Language (XSL), fournissant un point de départ à la création d'un résultat. Le résultat par défaut d'une transformation DITA Open Toolkit non modifiée est notoirement peu attrayante. Généralement, les transformations sont améliorées avant que le résultat ne soit utilisé dans un environnement de production.

Note: L'Open Toolkit n'est pas l'unique moyen de générer un résultat depuis DITA. Vous pouvez utiliser des outils commerciaux pour créer un résultat. Certains outils fournissent seulement une aide à la publication ; d'autres fournissent aussi une aide à la création pour le contenu DITA.

Les transformations sont indépendantes des fichiers carte, et vous pouvez appliquer de multiple transformations à un seul fichier carte. Par exemple, vous pouvez exécuter les transformations pour un fichier PDF et un système d'aide en ligne sur la même carte pour obtenir deux résultats.

Related information


- [DITA Open Toolkit](#)
- [Démarrer avec le webinaire DITA Open Toolkit, Suite Solutions \(vidéo, 53 minutes\)](#)

Références croisées

Les références croisées vous permettent de créer des liens d'un texte vers d'autres emplacements textuels, vers des figures ou des tableaux, ou vers des sites Web.

Toutes les références croisées utilisent l'élément `<xref>`. L'attribut `href` spécifie la cible de la référence croisée.


L'élément `<xref>` peut éventuellement contenir du texte. Si du texte est fourni, il est utilisé comme la zone de texte réactive pour le lien. Si l'élément `<xref>` ne contient pas de texte, le générateur utilise le titre de la cible comme zone sensible pour le lien.

 **Note:** bien que cela semble la bonne chose à faire, lors de la création d'une référence croisée vers une section, une figure ou un tableau, l'attribut href doit spécifier la section, la figure ou le tableau et non l'élément <title> de la section, de la figure ou du tableau.

Vidéo : [Exemples de code de références croisées DITA](#)

Vous pouvez également utiliser l'élément <xref> pour créer des liens vers des ressources extérieures à vos rubriques DITA. Pour les références externes, vous fournissez un attribut scope avec une valeur external et un attribut format, comme présenté ici :

```
<xref href="http://www.scriptorium.com" scope="external" format="html" />
```

 **Note:** Pour les liens vers un fichier PDF, utilisez format="pdf".

Techniquement, il est possible d'utiliser <xref> pour relier un topic à un autre. Cette liaison en ligne est une Mauvaise Idée™ car vous devez créer et gérer le lien manuellement. Vous devez spécifier le lien vers lequel vous vous connectez. Lorsque vous configurez la référence croisée, vous avez les topics source et cible dans votre fichier de carte. Mais si vous réutilisez le topic source dans une autre map n'incluant pas le topic cible, vous obtiendrez un lien rompu lors de la génération du résultat à partir de cette carte et vous ne pourrez pas être averti du problème. Au lieu d'utiliser des liens de topic à topic, il est préférable d'utiliser des tables de relations.

Liens connexes


À la fin d'un topic, vous pouvez insérer un élément related-links. Utilisez related-links pour pointer une information additionnelle que le lecteur pourrait vouloir.

Les liens connexes contiennent un lien et un texte de lien. Voici un exemple :

```
<topic id="exemple">
  <title>Exemple de titre</title>
  <body>
    ...
  </body>
  <related-links>
    <link href="http://www.exemple.com" format="html" scope="external">
      <linktext>Exemple de lien</linktext>
    </link>
  </related-links>
</topic>
```

Dans l'élément lien, vous spécifiez la cible (en utilisant l'attribut href), le format et la portée. Pour les liens internet et tout autre lien en dehors du fichier de carte courante, vous devez spécifier scope="external".





L'élément linktext contient le texte qui deviendra la zone sensible du lien.

 **Note:** Évitez d'utiliser des liens connexes pour les liens entre les topics de votre fichier de carte. Ils souffrent des mêmes limitations que les références croisées en ligne — vous pouvez vous retrouver avec des liens rompus. Utilisez plutôt des tables de relations.

Les tables de relations

Les tables de relations, ou reltables, vous permettent de décrire les relations d'un topic qui ne sont ni séquentielles ni hiérarchiques. Un reltable est une partie d'un fichier carte ; il peut apparaître n'importe où dans la carte, mais la convention est d'ajouter la reltable à la fin de la carte principale. Chaque ligne du tableau contient des éléments `topicref` liés à des sujets connexes.


Voici un exemple de ligne reltable (et la ligne d'en-tête reltable) dans lequel les colonnes contiennent des `topicref` pour des informations sur les concepts, références et tâches. La ligne affichée ici contient des `topicrefs` vers des topics connexes.

concept	reference	task
 <code>c_about_ducks.dita</code>	 <code>r_breedsofducks.dita</code>  <code>r_goodbreedsforpets.dita</code>	 <code>t_feeding.dita</code>

Lorsque vous générez un résultat via DITA Open Toolkit, les relations décrites dans la reltable sont utilisées pour créer une liste de liens vers les topics connexes. Dans la sortie HTML par défaut, les entrées reltable permettent de créer une section *Rubriques Liées* à la fin de chaque topic.

Une ligne reltable est montré ici :

- Le topic concept `c_about_ducks.dita` aurait des liens vers `r_breedsofducks.dita`, `r_goodbreedsforpets.dita` et `t_feeding.dita`.
- Le topic référence `r_breedsofducks.dita` aurait des liens vers `c_about_ducks.dita`, `r_goodbreedsforpets.dita` et `t_feeding.dita`.
- Le topic référence `r_goodbreedsforpets.dita` aurait des liens vers `c_about_ducks.dita`, `r_breedsofducks.dita` et `t_feeding.dita`.
- Le topic tâche `t_feeding.dita` aurait des liens vers `c_about_ducks`, `r_breedsofducks.dita` et `r_goodbreedsforpets.dita`.

 **Note:** Pour illustrer un concept plus large, cet exemple contient une simplification : pour que les liens `r_breedsofducks.dita` et `r_goodbreedsforpets.dita` soient liés, la cellule contenant les deux liens doit les identifier comme une « famille ». Les détails des attributs de cellule de la table de relation seront décrits dans un cours ultérieur.

Les relations que vous capturez dans la reltable ne sont généralement *pas* affichées lorsque vous créez des topics.

Vidéo: [Aperçu d'une table de relation de DITA \(reltable\)](#)

Commencez simple avec des reltables. Elles peuvent devenir très complexes.

Les reltables sont préférées aux liens associés ou aux xrefs pour les liens dans un fichier carte en raison des facteurs suivants :

- Les topicrefs de la reltable sont évaluées par rapport au fichier carte courant. Si votre reltable contient un lien vers un fichier qui n'est pas inclus dans le fichier carte, ce lien n'est pas généré dans le résultat. Cela évite le problème de lien brisé qui peut survenir avec les liens connexes et les xrefs.
- Les reltables sont plus faciles à maintenir que les liens connexes incorporés. Chaque ligne dans une reltable peut contenir plusieurs topics et capturer leurs interactions. Ainsi, si vous avez huit topics liés, il est beaucoup plus facile de créer une seule ligne dans une reltable qui répertorie ces huit topics que de créer huit listes de liens connexes légèrement différentes (consistant en 8×7 ou 56 liens) dans vos huit topics. Si vous devez supprimer un topic de la liste, la modification de la reltable est effectuée une fois au lieu de sept fois dans les fichiers.

Réutiliser du contenu avec les conrefs

Dans DITA, vous utilisez un conref pour réutiliser des éléments de contenu. Ces éléments de contenu peuvent être des avertissements (notes, mises en garde et alertes), du texte passe-partout (tel que l'adresse de votre entreprise), etc. Par exemple, la description d'un produit peut être configurée en tant que référence, de sorte qu'elle soit identique pour tous vos documents.

Si vous connaissez d'autres outils de création, les conrefs sont à peu près équivalents à :

- Les snippets de Flare
- Les topics embarqués de RoboHelp
- Les encarts de texte et variables FrameMaker
- Les inclusions côté serveur HTML


Supposons que vous ayez la note suivante dans un topic appelée location.dita :

```
<topic id="topicaid">
  ...
  <note id="whatduckslike">Les canards préfèrent les lacs aux déserts.</
note>
  ...
</topic>
```


Pour réutiliser cette note dans un autre topic, le code conref est le suivant :

```
<note conref="location.dita#topicaid/whatduckslike"/>
```

La plupart des outils de création DITA rationalisent le processus d'insertion de code conref. Par conséquent, vous ne taperez probablement pas le chemin du fichier, l'id du topic, etc., comme indiqué dans l'exemple.

 **Note:** L'élément cible doit être du même type que l'élément contenant l'attribut conref. Ainsi, vous ne pouvez pas utiliser l'attribut conref d'un élément <note> qu'à partir d'un autre élément <note> ; vous ne pouvez pas utiliser l'attribut conref d'un élément qu'à partir d'un autre élément , etc.

Vidéo : [exemple de code conref DITA](#)

 **Tip:** Il est recommandé de créer des topics séparés contenant des éléments réutilisables, au lieu de simplement choisir un topic au hasard et de faire pointer une référence vers ce topic. Cela vous permet de mieux suivre le contenu réutilisé et de contrôler son évolution.

Évaluation

Répondre aux questions suivantes pour vérifier votre compréhension des concepts de cette leçon.

Pour créer un lien vers une figure ou un tableau à l'intérieur du topic courant, vous devriez utiliser l'élément <xref>.

- Vrai
- Faux

Les éléments <xref> sont dangereux pour les liens entre topics à cause de la possibilité de liens cassés. À l'intérieur d'un topic, cependant, vous pouvez les utiliser sans problème.

La création de relation entre topics dans DITA exige que vous localisiez manuellement quels topics sont inclus dans le résultat.

- Vrai
- Faux

Vous devez conserver une trace des références croisées et des liens connexes, mais si vous utilisez une table de relation (reltable), DITA supprimera les liens pointant hors de portée du fichier carte.

Vous pouvez créer un lien vers une page web. La meilleure façon de le faire en DITA est d'utiliser un conref.

- Vrai
- Faux

Les conrefs vous laissent inclure, à l'emplacement courant, du contenu provenant d'un autre. Pour les liens internet, utilisez un élément `<xref>` avec les attributs `scope="external"` et `format="html"`.

Les conrefs vous permettent de :

- créer des références croisées
- réutiliser du contenu
- créer de texte conditionnel
- Indiquer que le contenu peut être réutilisé.

Vous utilisez l'élément `<xref>` pour créer des références croisées. Vous utilisez les attributs de filtre (audience, platform et product) pour créer du texte conditionnel. La présence d'un attribut `id` indique que le contenu peut être réutiliser (par un conref d'un autre élément).

Lorsqu'on souhaite faire une référence croisée dans un autre topic, la meilleure approche est de créer un lien avec l'élément `<xref>`.

- Vrai
- Faux

Les meilleures pratiques DITA décourage l'utilisation de liens à cause de problèmes de maintenance. Une meilleure approche consiste en une table de relations.

Lisez le code suivant. Que montre-t-il ?

```
<ol>
  <li conref="shared/DuckPelletSteps.dita#pellet_steps/openbox"/>
  <li>Scoop out 1 cup of pellets.</li>
  <li>Pour in the blender.</li>
  <li>Heat the Duckling mash.</li>
  <li>Pour in the blender.</li>
  <li>Put in feeding pan.</li>
</ol>
```

- Une liste numérotée incluant un corner vers une étape fréquemment réutilisée.
- Un topic de type concept sur l'alimentation des canards.
- Une liste à puces nommé DuckPelletSteps.dita.

Ce fragment montre seulement des listes ol et li, vous ne pouvez donc pas déterminer le type de son topic. Le nom de fichier référence est une partie d'un conref, non une étiquette pour cette liste et indique une liste ordonnée (numérotée).

Pour créer un lien vers une page internet, utilisez la syntaxe suivante :

- <xref href="http://www.scriptorium.com" scope="external" format="html"/>
- <xref href="http://www.scriptorium.com"/>
- <xref link="http://www.scriptorium.com" scope="external" format="html"/>

Vous avez besoin des attributs scope et format. Il n'y a pas d'attribut link dans un xref ; vous utilisez l'attribut href.

Les tables de relation décrivent des relations hiérarchique entre topics.

- Faux
- Vrai

Les fichiers carte décrivent des relations hiérarchiques. Les tables de relations décrivent les liens entre topics homologues liés.

Pour créer des liens entre topics liés, la meilleure pratique est de créer une liste à la fin de chaque topic.

- Faux
- Vrai

Les tables de relations sont bien plus efficaces et faciles à maintenir que de créer des liens dans chaque topic.

Supposons que le fragment de code suivant est dans la table de relation du fichier carte courant et que tous les topics référencés sont également dans le fichier carte. Quel sera le résultat dans la section des rubriques associées (choisissez-en une) ?

 c_wild_ducks.dita	 r_wild_breeds.dita	 t_feeding_wild.dita
---	--	---

- Chaque topic aura des liens avec les deux autres topics listés dans la ligne.
- Chaque topic aura des liens avec les trois topics listés dans la ligne.
- Il n'y aura aucun lien ; ils ne sont pas associés.
- Les liens seront toujours listés sur une page par eux-mêmes.

Quand un topic est listé dans une ligne d'une table de relation, ce topic n'aura pas de lien avec lui-même. Les topicrefs ont été placés dans cette ligne parce que leur contenu est similaire. Les liens vers les autres topics sont habituellement placés dans une section à la fin du topic (selon le format de sortie).

Un fichier DITA map fait ce qui suit (choisir toutes les réponses qui s'appliquent) :

- Décrire la séquence des topics référencés
- Décrire la hiérarchie des topics référencés
- Créer un lien vers le contenu web
- Fournir un moyen de créer des liens internes dans les topics

Utilisez des xrefs pour créer des liens externes. Comme règle, vous devez éviter les liens internes. Les fichiers carte listent une collection de topics, mais pas le contenu à l'intérieur des topics.