

Les espaces de nommage XML



- Objectifs
- Principes
- Terminologie
- Portée des déclarations
- Espace de nommage par défaut
- Espaces de nommages et attributs
- Surcharge des déclarations
- URIs et préfixes
- Espaces de nommages et DTD
- Espaces de nommages dans le modèle de données

```
<?xml version="1.0"?>
<auteur>
  <nom>Poulard</nom>
  <prénom>Philippe</prénom>
  <titre>Baron</titre>
</auteur>
```

```
<?xml version="1.0"?>
<cours>
  <titre>Fondamentaux XML</titre>
  <contenu>
    .../...
  </contenu>
</cours>
```

Fusion des 2 documents

Si vous avez la maîtrise des DTD que vous utilisez, vous pouvez éviter cette confusion, mais si vous utilisez des DTD écrites par d'autres, vous y serez confronté

```
<?xml version="1.0"?>
<cours>
  <titre>Fondamentaux XML</titre>
  <auteur>
    <nom>Poulard</nom>
    <prénom>Philippe</prénom>
    <titre>Baron</titre>
  </auteur>
  <contenu>
    .../...
  </contenu>
</cours>
```

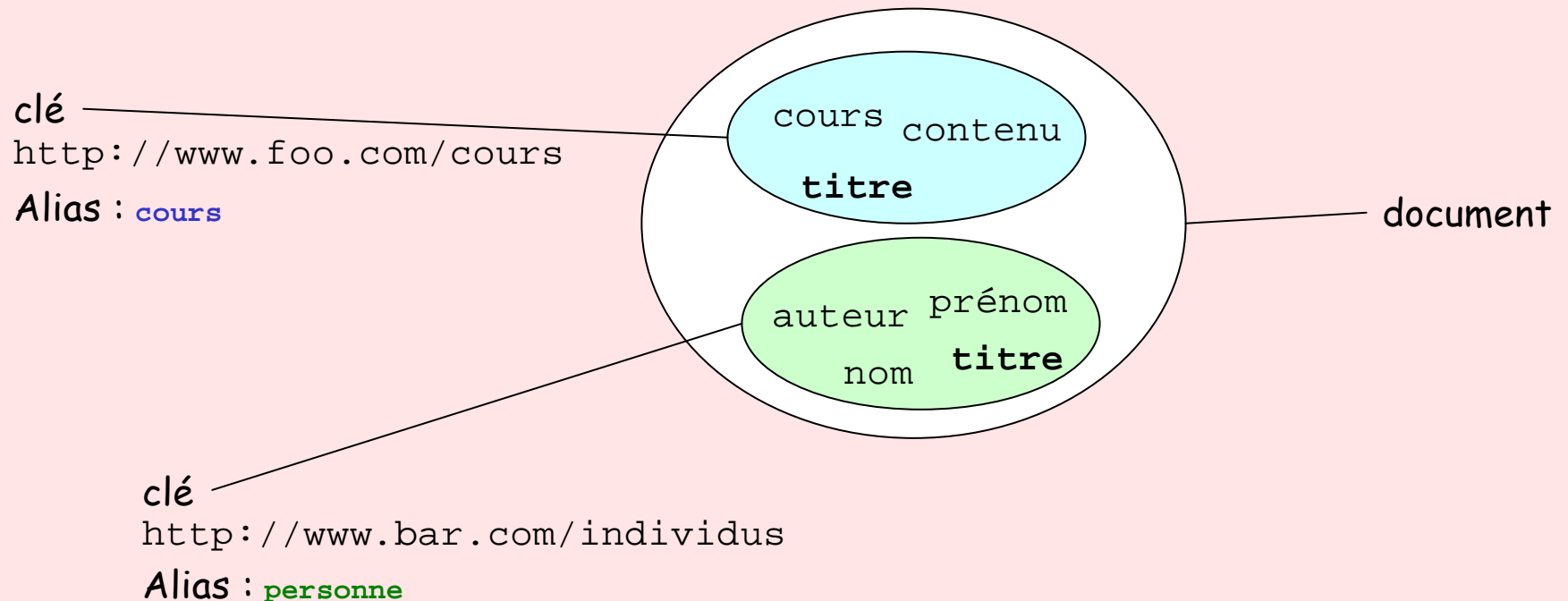
Confusion sur le sens de l'élément titre

Objectifs des espaces de nom :

- **distinguer** les éléments et les attributs de différentes applications XML qui ont le même nom
- **grouper** les éléments et les attributs d'une même application XML pour que les logiciels puissent les reconnaître

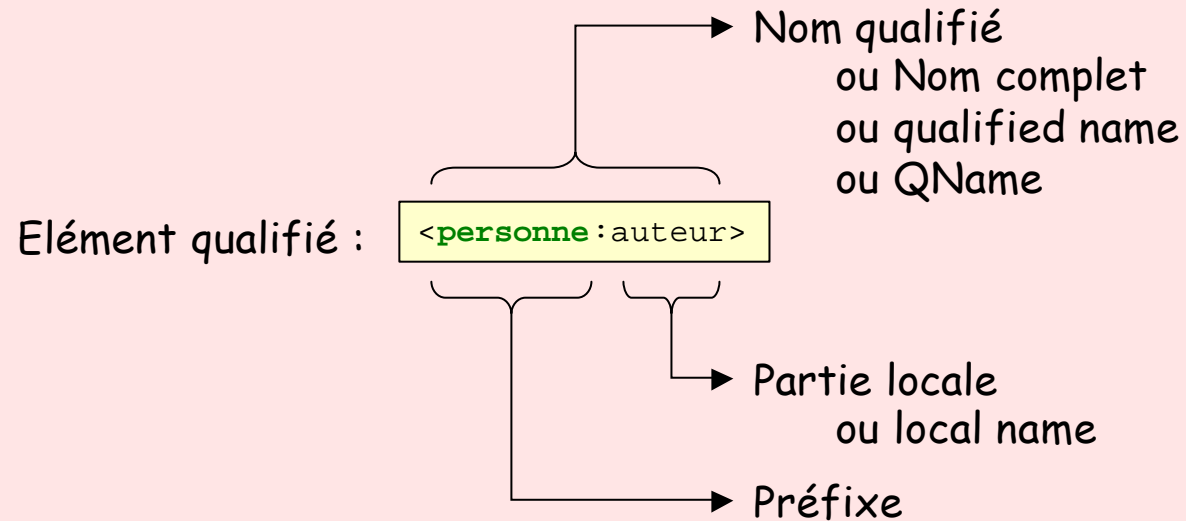
Ambiguïté → assignation des éléments et attributs à une **clé universelle**
(URI=ID universel)

Les URI séparent les éléments et les attributs en 2 ensembles disjoints



Déclaration des espaces de nommage :
Ils se déclarent dans un élément, avec le préfixe spécial `xmlns`

```
<cours:cours  
  xmlns:cours="http://www.foo.com/cours"  
  xmlns:personne="http://www.bar.com/individus">
```



Attribut qualifié : `<a xlink:href="#target">`

URI + local name = **nom universel**

Pour référer un **nom expansé**, la notation `{http://www.bar.com/people}author` est souvent utilisée, mais n'apparaît jamais dans un document XML.

```
<?xml version="1.0"?>
<cours:cours xmlns:cours="http://www.foo.com/cours">
  <cours:titre>Fondamentaux XML</cours:titre>
  <personne:auteur xmlns:personne="http://www.bar.com/individus">
    <personne:nom>Poulard</personne:nom>
    <personne:prénom>Philippe</personne:prénom>
    <personne:titre>Baron</personne:titre>
  </personne:auteur>
  <cours:contenu>
    .../...
  </cours:contenu>
</cours:cours>
```

Portée de http://www.bar.com/individus

Portée de http://www.foo.com/cours

```
<?xml version="1.0"?>
<cours:cours xmlns:cours="http://www.foo.com/cours"
  xmlns:personne="http://www.bar.com/individus">
  <cours:titre>Fondamentaux XML</cours:titre>
  <personne:auteur>
    <personne:nom>Poulard</personne:nom>
    <personne:prénom>Philippe</personne:prénom>
    <personne:titre>Baron</personne:titre>
  </personne:auteur>
  <cours:contenu>
    .../...
  </cours:contenu>
</cours:cours>
```

Il est possible de ne pas spécifier de préfixe dans la déclaration

Définition d'un espace de nommage par défaut

```
<?xml version="1.0"?>
<svg width="18cm" height="6cm" viewBox="0 0 200 120"
  xmlns="http://www.w3.org/2000/svg">
  <rect x="1" y="1" width="198" height="118" fill="#888888" stroke="blue" />
</svg>
```

```
<?xml version="1.0"?>
<svg:svg width="18cm" height="6cm" viewBox="0 0 200 120"
  xmlns:svg="http://www.w3.org/2000/svg">
  <svg:rect x="1" y="1" width="198" height="118" fill="#888888" stroke="blue" />
</svg:svg>
```

Le préfixe `xml` est associé à l'espace de nommage : `http://www.w3.org/XML/1998/namespace`



Dans le modèle logique, tous les éléments disposent de l'espace de nommage `xml`

Le préfixe `xml` sert pour les attributs spéciaux `xml:space`, `xml:lang` et `xml:base`

Le préfixe `xmlns` est utilisé exclusivement pour définir les espaces de nommage

→ URI `http://www.w3.org/2000/xmlns/`

(utilisé dans le DOM)

L'espace de nommage par défaut ne s'applique pas aux attributs sans préfixe

```
<?xml version="1.0"?>
<svg:svg width="18cm" height="6cm" viewBox="0 0 200 120"
  xmlns:svg="http://www.w3.org/2000/svg">
  <svg:rect x="1" y="1" width="198" height="118" fill="#888888" stroke="blue" />
</svg:svg>
```

Ces attributs
ne sont pas dans
un espace de noms

Les applications :

- ne doivent pas tenter d'associer les attributs non préfixés à l'espace de nommage XML par défaut
- doivent associer les attributs non préfixés à leur élément-type (DTD)

Il est donc inutile de préfixer les attributs pour qu'ils soient dans le même espace de nommage de l'élément dont ils dépendent (souvent, les DTD l'interdisent même)

Cependant, il est possible d'associer à un élément des attributs qui font partie d'un autre espace de nommage :

```
<?xml version="1.0"?>
<document xmlns:xlink="http://www.w3.org/1999/xlink">
  <image xlink:type="simple"
    xlink:actuate="onLoad"
    xlink:show="embed"
    xlink:href="http://www.acme.com/foo/bar.jpg"
    width="123"
    height="234" />
</document>
```


Surcharge d'une déclaration d'espace de nommage qui utilise un préfixe :

```
<foo:A xmlns:foo="http://www.foo.org/">
  <foo:B>
    <foo:C xmlns:foo="http://www.bar.org/">
      <foo:D>abcd</foo:D>
    </foo:C>
  </foo:B>
</foo:A>
```

Surcharge d'une déclaration d'espace de nommage par défaut :

```
<A xmlns="http://www.foo.org/">
  <B>
    <C xmlns="http://www.bar.org/">
      <D>abcd</D>
    </C>
  </B>
</A>
```

Surcharge d'une déclaration d'espace de nommage qui utilise un préfixe avec une déclaration vide :

```
<foo:A xmlns:foo="http://www.foo.org/">
  <foo:B>
    <foo:C xmlns:foo="">
      <foo:D>abcd</foo:D>
    </foo:C>
  </foo:B>
</foo:A>
```



Surcharge d'une déclaration d'espace de nommage par défaut avec une déclaration vide :

```
<A xmlns="http://www.foo.org/">
  <B>
    <C xmlns="">
      <D>abcd</D>
    </C>
  </B>
</A>
```

Quelques déclarations célèbres :

RDF

```
<rdf:RDF xmlns:rdf="http://www.w3.org/TR/REC-rdf-syntax#">
```

Dublin-core

```
<dc:title xmlns:dc="http://purl.org/dc/">
```

SVG

```
<svg xmlns="http://www.w3.org/2000/svg">
```

XSLFO

```
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
```

XSLT

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
```

XHTML

```
<xhtml:xhtml xmlns:xhtml="http://www.w3.org/1999/xhtml">
```

XLink

```
<foo xmlns:xlink="http://www.w3.org/1999/xlink">
```

SOAP

```
<SOAP-ENV xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
```

Schema

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
```

Les URI sont purement déclaratives

Il n'y a pas nécessairement quelque chose à l'adresse indiquée

Les parseurs et les applications ne sont pas tenus d'aller y chercher quoi que ce soit

- il n'y a rien qui puisse les intéresser
- la cible n'est pas nécessairement accessible par votre réseau

Les URI des espaces de nommage ne sont que des identificateurs formels

En général, on y trouve un document explicatif de l'utilisation de l'espace de nommage, mais ce n'est pas une obligation.

Les espaces de nommage du W3C font référence aux spécifications appropriées

Les URIs servent seulement à définir des clés universelles :

```
xmlns="urn:guid:88d969c9-f192-11d4-a65f-0040963251e6"  
xmlns="urn:isbn:2-84177-230-6"  
xmlns="urn:schemas-microsoft-com<
```

Cependant, dans un environnement confiné, cela pourrait fonctionner :

```
xmlns:date="*** Processing dates ***"
```

(à manipuler avec précaution !!!)

Les préfixes doivent toujours être associés à une URI par une déclaration `xmlns`

Les préfixes n'ont **aucune** signification particulière

Seule l'URI peut avoir une signification pour une application

Exemple

Un processeur XSLT saura traiter indifféremment les feuilles de style suivantes :

```
<xsl:stylesheet
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
```

```
<extensible-stylesheet-language:stylesheet
  xmlns:extensible-stylesheet-language="http://www.w3.org/1999/XSL/Transform">
```

```
<tiens-voilà-du-boudin:stylesheet
  xmlns:tiens-voilà-du-boudin="http://www.w3.org/1999/XSL/Transform">
```

La recommandation des espaces de nommage ne définit rien d'autre qu'un système de nommage en 2 parties pour les éléments types et les attributs

En particulier, ils ne définissent pas:

- un moyen pour fusionner 2 documents qui utilisent des DTD différentes (les ambiguïtés dans le document fusionné doivent être levées spécifiquement, et les espaces de nommage sont un moyen d'y arriver)
- un moyen pour les associer à des schémas
- un moyen pour valider les documents qui les utilise
- comment associer les éléments types et les déclarations d'attributs des DTD à un espace de nommage

Les espaces de noms sont complètement indépendants des DTD

Il peut y avoir :

- des documents sans déclaration d'espace de nom qui utilisent une DTD
- des documents avec déclaration d'espace de nom sans DTD
- des documents avec déclaration d'espace de nom ET qui utilisent une DTD

Une déclaration `xmlns` peut être omise dans le document et déclarée dans la DTD :

```
<!ATTLIST livre      xmlns:xlink CDATA #FIXED "http://www.w3.org/1999/xlink">
```

```
<!ATTLIST svg xmlns      CDATA #FIXED "http://www.w3.org/2000/svg">
```

```
<?xml version="1.0"?>
<!DOCTYPE svg:svg [
    <!ATTLIST svg:svg xmlns:svg      CDATA #FIXED "http://www.w3.org/2000/svg">
]>
<svg:svg width="18cm" height="6cm" viewBox="0 0 200 120">
    <svg:rect x="1" y="1" width="198" height="118" fill="#888888" stroke="blue" />
</svg:svg>
```

Les DTD peuvent contenir des noms qualifiés, mais les déclarations d'espace de nommage ne s'appliquent pas aux DTD

Les DTD doivent déclarer les éléments avec leur nom qualifié

Il n'y a pas de moyen de déterminer à quel espace de nommage un préfixe utilisé dans une DTD se réfère

Les noms qualifiés dans les DTD ne peuvent pas être associés à des noms universels

Les déclarations d'éléments types et d'attributs sont exprimées en termes de noms qualifiés, pas en terme de nom universel

```
<!ENTITY % dc-prefix "dc">
<!ENTITY % dc-colon ":">
```

```
<!ENTITY % dc-prefix "">
<!ENTITY % dc-colon "">
```

```
<!ENTITY % dc-prefix "dublin-core">
<!ENTITY % dc-colon ":">
```

Au choix...

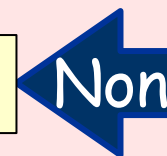
```
<!ENTITY % dc "%dc-prefix;%dc-colon;">

<!ENTITY % dc-title "%dc;title">
<!ENTITY % dc-creator "%dc;creator">
<!ENTITY % dc-description "%dc;description">
<!ENTITY % dc-date "%dc;date">
```

```
<!ELEMENT %dc-title; (#PCDATA)>
<!ELEMENT %dc-creator; (#PCDATA)>
<!ELEMENT %dc-description; (#PCDATA)>
<!ELEMENT %dc-date; (#PCDATA)>
```

```
<!ELEMENT rdf:Description ( (%dc-title; | %dc-creator; |
                             %dc-description; | %dc-date;)* )>
```

```
<!ENTITY % dc "%dc-prefix;%dc-colon;">
<!ELEMENT %dc;title (#PCDATA)>
```



Le parseur peut insérer des blancs

Les éléments héritent des déclarations d'espaces de nommage définis par eux ou leurs ancêtres.

L'élément racine (et ses descendants) est automatiquement doté de la déclaration :

```
xmlns:xml="http://www.w3.org/XML/1998/namespace"
```

Ces 2 documents sont équivalents...

```
<foo:A xmlns:foo="http://www.foo.org/">
  <foo:B>
    <bar:C xmlns:bar="http://www.bar.org/">
      <bar:D>abcd</bar:D>
      <foo:E>efgh</foo:E>
    </bar:C>
  </foo:B>
</foo:A>
```

... mais celui-ci n'est qu'une vue conceptuelle : il est interdit de définir un préfixe avec la chaîne "xml"



```
<foo:A xmlns:foo="http://www.foo.org/"
  xmlns:xml="http://www.w3.org/XML/1998/namespace">
  <foo:B xmlns:foo="http://www.foo.org/"
    xmlns:xml="http://www.w3.org/XML/1998/namespace">
    <bar:C xmlns:bar="http://www.bar.org/"
      xmlns:foo="http://www.foo.org/"
      xmlns:xml="http://www.w3.org/XML/1998/namespace">
      <bar:D xmlns:bar="http://www.bar.org/"
        xmlns:foo="http://www.foo.org/"
        xmlns:xml="http://www.w3.org/XML/1998/namespace">abcd</bar:D>
      <foo:E xmlns:bar="http://www.bar.org/"
        xmlns:foo="http://www.foo.org/"
        xmlns:xml="http://www.w3.org/XML/1998/namespace">efgh</foo:E>
    </bar:C>
  </foo:B>
</foo:A>
```

Non