

# XML

```
<?xml version="1.0">
<cours nom_attribut="valeur de cet attribut" nom_attribut2="valeur de cet attribut2" > <!-- balise de début
doit être unique -->
<sous-element>
</sous-element>
</cours> <!-- balise de fin -->
<!--
```

Les attributs peuvent être multiples

Un attribut aura toujours une valeur textuelle ( qui pourra être vue sous forme de numérique, date, ... )

Par contre un élément est "complexe" : il peut contenir des sous-éléments ( une structure )

Un élément qui ne contient rien entre <element> et </element> est simple

Un élément qui ne contient du texte entre <element> et </element> est simple

Un élément qui ne contient des tags entre <element> et </element> est complexe : il indique le contenu de sa hiérarchie.

Un attribut est toujours simple ( toujours un contenu texte )

<element></element> et <element /> sont la même chose ( le second est plus court - préféré - que le premier )

-->

Vous pourrez choisir, devoir respecter, ou vous aligner sur la structure d'un XML que vous écrirez, ou que l'on vous fournira.

Cette structure est basée sur un "**schéma**" : une structure que l'on doit respecter.

Exemple de XML avec un élément racine, des sous éléments, des attributs ( titre, nom ), et une valeur dans l'élément ( plan )

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Date de création : 30/09/07 -->
<cours titre="XML">
  <intervenant nom="alexandre brillant">
  </intervenant>
  <plan>
    Introduction
    XML et la composition de documents
  </plan>
</cours>
```

Element ( cours )

Attribut

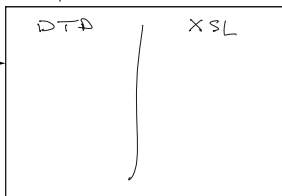
Valeur simple

Valeur complexe

## Document XML

```
<?xml version="1.0" >
<< peut avoir >>
référence vers le schéma
<élément racine>
contenu ....
</élément racine>
```

## Schéma des XML



DTD: Document Type  
Definition

↳ Ancien format

XSD: Xml Schema  
Definition

⇒ Document XML qui  
déclare la structure -

Exemple de DTD :

```
<!DOCTYPE cours [
<!ELEMENT cours ( intervenant, plan )>
<!ELEMENT intervenant EMPTY>
<!ELEMENT plan (#PCDATA)>
<!ATTLIST cours titre CDATA #REQUIRED>
<!ATTLIST intervenant nom CDATA #REQUIRED>
]>
```

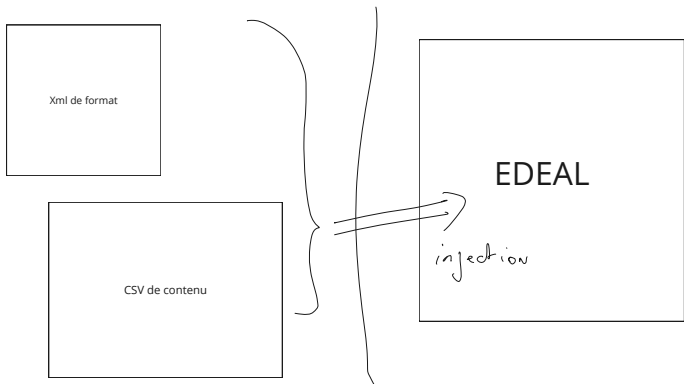
Exemple de schéma XSD :

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:simpleType name="couleurType">
<xs:restriction base="xs:string">
<xs:enumeration value="bleu"></xs:enumeration>
<xs:enumeration value="blanc"></xs:enumeration>
<xs:enumeration value="rouge"></xs:enumeration>
</xs:restriction>
</xs:simpleType>
<xs:element name="crayon">
<xs:complexType>
<xs:attribute name="couleur" type="couleurType"/>
</xs:complexType>
</xs:element>
</xs:schema>
```

Permet de générer par exemple :

```
<?xml version="1.0" ?>
<crayon couleur="rouge">
</crayon>
```

le schéma de ce document



### Exercice de création de document XML

- Créez un document qui stocke une personne, avec un nom et un prénom en attribut ( il n'existe qu'un élément personne )

```
<?xml version="1.0">
```

```
<!-- Créé a l'occasion du stage, le 09/07/2020 -->
```

```
<Personne nom="Titi" prenom="Toto">
```

```
</Personne>
```

- Créez un document XML qui stocke un produit, avec son nom de produit et son prix en attributs.

```
<?xml version="1.0">
```

```
<!--Date de création : 09/07/2020 -->
```

```
<Produit nom="Tartelette" prix="15">
```

```
</Produit>
```

- Modifiez cet élément pour que le nom et le prix soient des sous-élément ( c'est une autre représentation )
- Créez un document qui liste deux personnes ( ! Il ne peut y avoir qu'une seul élément racine )

<element>valeur de l'élément</element>

```
<?xml version='1.0'>
<!--Date de création : 09/07/2020 -->
<Produit>
  <nom>Tartelette</nom>
  < prix>15</prix>
</Produit>
```

Créez un document XML qui affiche une catégorie de produit.

```
<?xml version='1.0'>
<!--Date de création : 09/07/2020 -->
<Categorie>
  <nom>Truchmuche</nom>
  <reference>01</reference>
</Categorie>
```

Créez un document xml qui affiche une commande  
Le numéro de commande est un attribut de l'élément Commande  
La valeur du port est un sous élément  
La date de commande est un sous élément également

```
<?xml version="1.0">
<!--Date de création : 09/07/2020 -->
<Commande numero_commande="0520">
  <port>30</port>
  <date_de_commande>2020</date_de_commande>
</Commande>
```

Créez un document XML qui affiche un client ( avec nom et prénom  
sous forme d'attribut ou d'élément, comme vous préférez )  
Ce client, aura une commande, la commande numéro 0520

```
<?xml version="1.0">
<!--Date de création : 09/07/2020 -->
<Client>
  <nom>Truchmuche</nom>
  <reference>01</reference>
  <Commande numero_commande="0520">
    <port>30</port>
    <date_de_commande>2020</date_de_commande>
  </Commande>
</Client>
```

- Créez un document XML qui stocke une catégorie de produit, elle a un nom sous forme d'attribut ou d'élément ( à votre préférence ). Elle contient 2 produits, constitués d'un nom, et d'un prix unitaire ( sous forme d'attribut ou d'élément - si vous avez utilisé des attribut à la catégorie, utilisez des éléments )

```
<?xml version="1.0">
<!--Date de création : 09/07/2020 -->
<Categorie nom_categorie="Nourriture">
  <Produit>
    <nom>CHOUCROUTE</nom>
    <prix_unitaire>10</prix_unitaire>
  </Produit>
  <Produit>
    <nom>Ficelle picarde</nom>
    <prix_unitaire>15</prix_unitaire>
  </Produit>
</Categorie>
```

- Affichez un document qui contienne un employé et deux commandes qu'il a pu traiter.

```
<?xml version="1.0">
<!--Date de création : 09/07/2020 -->
<Employe nom_empl="DEHON">
  <Commande>
    <ref_commande>001</ref_commande>
    <montant_comm>250</montant_comm>
  </Commande>
  <Commande>
    <ref_commande>002</ref_commande>
    <montant_comm>350</montant_comm>
  </Commande>
</Employe>
```

1. Basé sur cette solution, nous avons maintenant deux employés. Attention, l'élément racine est toujours unique

hint :

```
<?xml version="1.0" ?>
<As>
  <A/>
  <A/>
</As>
```

```

<?xml version="1.0">
<!--Date de création : 09/07/2020 -->
<Employes>
  <Employe>
    <nom>DEHON</nom>
    <Commande>
      <ref_commande>001</ref_commande>
      <montant_comm>250</montant_comm>
    </Commande>
  </Employe>
  <Employe>
    <nom>BOULE</nom>
    <Commande>
      <ref_commande>002</ref_commande>
      <montant_comm>350</montant_comm>
    </Commande>
    <Commande>
      <ref_commande>005</ref_commande>
      <montant_comm>550</montant_comm>
    </Commande>
  </Employe>
</Employes>

```

Corriger un document XML via à vis de son contenu et les caractères spéciaux

```

<?xml version="1.0" ?>
<!-- l'attribut date stocke la date de commande -->
<Commandes>
  <Commande id="1" date="2020">
    <Condition c="il faut que l'id a soit > à b" />
    <Description>Ici j'ai la description de ma commande. attention aux "faux
clients"</Description>
  <Commande>
</Commandes>

```

Correction :

```

<?xml version="1.0" ?>
<!-- l'attribut date stocke la date de commande -->
<Commandes>
  <Commande id="1" date="2020">
    <Condition c="il faut que l'&apostrophe;id a soit &gt; à b">
      </Condition>
    <Description>Ici j'ai la description de ma commande. attention aux &apostrophe;faux
clients&apostrophe;</Description>
  <Commande>
</Commandes>

```

```
<Clients>
  <Client id="1">
    <Nom>L'auberge des cieux</Nom>
    <Commentaires> << Attention ! >> client "important" ! </Commentaires>
  </Client>
</Clients>
```

```
<Clients>
  <Client id="1">
    <Nom>L'auberge des cieux</Nom>
    <Commentaires> &lt;&lt; Attention ! &gt;&gt; client &quot;important&quot; !
  <Commentaires>
</Client>
|</Clients>
```

Document "**Bien formé**" : Document pour lequel il n'y a aucune erreur de syntaxe.

Un document qui n'est pas bien formé sera systématiquement rejeté par les outils ou logiciels qui traiteront ce document.

Document "**Valide**" : Document dont le contenu respecte un schéma fourni, associé.

## Xmlns, namespaces, et schémas

```
<chapitre xmlns="http://www.masociete.com">  
  <paragraphe xmlns="http://www.autresociete.com">  
    ...  
  </paragraphe>  
</chapitre>
```

Un schéma  
Les tags devront  
respecter ce qui est  
défini par ce schéma.  
( ici, chapitre )

On a ici un autre schéma, et donc d'autres autorisations de contenu à respecter

Tout ce qui se trouve en dessous de paragraphe devra respecter ce schéma

Utilisation d'un alias de namespace ( espace de nom )

```
<p:resultat xmlns:p="http://www.masociete.com">  
</p:resultat>
```

1) définition de l'alias  
l'alias peut être n'importe quel  
nom...  
ici : **p**

2) on utilise, on applique l'alias  
aux éléments dont on veut appliquer  
les contraintes du schéma.

Il n'est pas utile d'utiliser des alias si il n'y a qu'un seul schéma.

Avec deux schémas :

```
<p:res xmlns:p="http://www.masociete.com" xmlns:p2="http://www.autresociete.com">  
  <p2:res>  
  </p2:res>  
</p:res>
```

ici, la balise res, est déclarée dans deux schémas différents, avec deux structures autorisées différentes. ( ce qui est un cas "un peu tordu" )



```
<p:element xmlns:p="http://www.masociete.com">
  <autreelement/>
</p:element>
```

Ici on applique un schéma à un alias, et on utilise cet alias pour un élément, mais pas un autre

<autreelement> n'aura pas de schéma, pas de contraintes, car par d'alias appliqué, ni d'attribut xmlns sans alias.

Exercice : associez quelle element ou attribut est associé à quel schéma

```
<p1:livre titre="Mon livre" xmlns:p1="http://www.masociete.com"
  xmlns:p2="http://www.monentreprise.com">
  <p1:auteurs>
    <p1:auteur nom="nom1" prenom="prenom1"/>
    <p1:auteur nom="nom2" prenom="prenom2"/>
  </p1:auteurs>
  <p1:sections>
    <p1:section titre="Section1">
      <p1:chapitre titre="Chapitre1">
        <p1:paragraphe>Premier paragraphe</p1:paragraphe>
        <p1:paragraphe>Deuxième paragraphe</p1:paragraphe>
      </p1:chapitre>
    </p1:section>
    <p2:section titre="Section2">
      <p2:chapitre titre="Chapitre1">
        <p2:paragraphe>Premier paragraphe</p2:paragraphe>
        <p2:paragraphe>Deuxième paragraphe</p2:paragraphe>
      </p2:chapitre>
      <p2:chapitre titre="Chapitre2">
        <p2:paragraphe>Premier paragraphe</p2:paragraphe>
        <paragraphe>Deuxième paragraphe</paragraphe>
      </p2:chapitre>
    </p2:section>
  </p1:sections>
</p1:livre>
```

Pas de schéma