

Université de Jijel
Faculté des Sciences Exactes et d'Informatique
Département d'Informatique
L3 – Systèmes Informatiques

Données Semi-Structurées

Chapitre 04

Galaxie XML

Tarek Boutefara
t_boutefara@univ-jijel.dz
2020/2021

Plan

- Exemple : SVG
- DOM
- XPATH
- XSL
- SAX
- XLINK
- XPOINTER

Plan

- **Exemple : SVG**
- DOM
- XPATH
- XSL
- SAX
- XLINK
- XPOINTER

SVG

- SVG
 - Scalable Vector Graphics,
 - Permet de définir des graphes basés sur des vecteurs en utilisant une syntaxe XML,
 - Notre objectif :
 - Exemple sur XML,
 - On ne va pas le détailler.

SVG

- Exemple :

```
<html>
<body>

  <h1>Exemple SVG</h1>

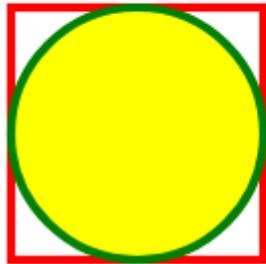
  <svg width="100" height="100">
    <rect x="20" y="20"
      Height="60" width="60"
      stroke="red" stroke-width="2"
      fill="white" />
    <circle cx="50" cy="50"
      r="30"
      stroke="green" stroke-width="2"
      fill="yellow" />
  </svg>

</body>
</html>
```

SVG

- Exemple :

Exemple SVG



SVG

- SVG
 - Liste des formes (les plus simples) :
 - `<rect>`
 - `<circle>`
 - `<ellipse>`
 - `<line>`
 - `<polyline>` : ensemble de lignes droites
 - `<polygon>` : polyline fermé (le dernier point est lié au premier)

SVG

- SVG
 - `<rect>`
 - x, y : coordonnées (point haut gauche),
 - width, height : largeur et hauteur,
 - rx, ry : arrondissement des coins.

SVG

- SVG
 - `<circle>`
 - cx, cy : coordonnées (centre),
 - r : rayon

SVG

- SVG
 - `<ellipse>`
 - cx, cy : coordonnées (centre),
 - rx, ry : rayons (horizontal, vertical).

SVG

- SVG
 - `<line>`
 - x_1, y_1 : coordonnées du premier point,
 - x_2, y_2 : coordonnées du deuxième point.

SVG

- SVG
 - `<polyline>`
 - points : liste des points nécessaires pour dessiner les lignes, ensemble de paires x,y :
 - `points="100,100 150,25 150,75 200,0"`

SVG

- SVG
 - <polygon>
 - points : liste des points nécessaires pour dessiner les lignes, ensemble de paires x,y :
 - points="100,100 150,25 150,75 200,0"

SVG

- SVG
 - Bordure et remplissage :
 - Bordure :
 - stroke : couleur bordure, « none » pour ne pas avoir de bordure,
 - stroke-width : largeur de la bordure,
 - stroke-dasharray : pour rendre la bordure pointillée :
 - stroke-dasharray = « 4 »
 - stroke-dasharray = « 4 1 »
 - stroke-linecap : définir la forme de la fin de la ligne
 - butt, round, square

SVG

- SVG
 - Bordure et remplissage :
 - Remplissage :
 - fill : couleur de remplissage, « none » pour ne pas avoir de remplissage,
 - fill-opacity : transparence du remplissage,
 - fill-rule : l'algorithme de définition de l'intérieur
 - nonzero, evenodd

Plan

- Exemple : SVG
- **DOM**
- XPATH
- XSL
- SAX
- XLINK
- XPOINTER

DOM

- Définition
 - Interface abstraite de programmation qui permet de représenter un document XML (ou HTML) sous forme d'un arbre de nœuds,

DOM

- Définition
 - Tout est un nœud
 - Elements,
 - Attributs,
 - DocumentType,
 - Commentaire,
 - Text,
 - ...

DOM

- Déplacement :
 - Un document représente la racine du DOM,
 - On peut récupérer les fils d'un éléments,
 - On peut accéder aux voisins (siblings),
 - On peut récupérer le premier fils,
 - On peut récupérer le dernier fils,
 - ...

Plan

- Exemple : SVG
- DOM
- **XPATH**
- XSL
- SAX
- XLINK
- XPOINTER

XPath

- Définition
 - XPath est un langage abstrait pour l'interrogation des documents XML,
 - Cette interrogation donne en retour un sous-ensemble des nœuds (ou partie du document),
 - Il a été proposé par la W3C.

XPath

- Principe
 - XPath construit l'arbre logique du document XML,
 - L'expression passée est évaluée par rapport à cette arborescence,
 - L'expression passée est sous la forme d'un « chemin »
 - Inspiré des chemins des fichiers utilisés sur le disque dur

XPath

- Principe
 - L'expression peut pointer tous les types de nœuds :
 - Elements,
 - Attributs,
 - Entités,
 - Etc.

XPath

- Principe
 - L'évaluation se fait dans l'ordre d'apparition dans le document,
 - L'ordre des nœuds est important,
 - contrairement au modèle relationnel des bases de données où l'ordre n'est pas important.

XPath

- Les expressions :
 - Noeud :
 - `<exemple>Bonjour</exemple>`
 - Élément,
 - `Lang = "fr"`
 - Noeud attribut
 - Valeur atomique (des nœuds sans parent et sans enfants) :
 - Bonjour
 - "fr"

XPath

- Les expressions :
 - Liens :
 - Parent : chaque élément ou attribut possède un seul parent,
 - Fils : chaque élément peut posséder un ou plusieurs fils comme il peut n'avoir aucun,
 - Frères : des neuds peuvent avoir le même parent,
 - Ancêtres : le parent, le parent du parent, etc,
 - Descendants : les fils, les fils des fils, etc.

XPath

- Les expressions :
 - /step/step/step/...|/step/step/...|...

XPath

- Les expressions :
 - `nom_noeud` : sélectionner les nœuds avec le nom « `nom_noeud` »,
 - `//` : sélection à partir du nœud en cours quelque soit leur position
 - `/` : sélection à partir de la racine,
 - `.` : sélectionner le nœud en cours,
 - `..` : sélectionner le nœud parent,
 - `@` : sélectionner attribut.

XPath

- Exemple
 - Recette.xml

XPath

- Exemple
 - Recette.xml
 - Sélectionner tout :
 - recette

XPath

XPath online real-time tester, evaluator and generator for XML & HTML — Mozilla Firefox

XPath online real-time test

xpather.com

recette

Elements found: 1

XML mode Format Save

```
<?xml version="1.0" ?>
<recette id="1336">
  <titre>French Meringues
</titre>
  <ingredients>
    <ingredient qte="4">egg whites
  </ingredient>
    <ingredient qte="2.25" unite="cup">confectioners' sugar
  </ingredient>
  </ingredients>
  <directions>
    <step n="1">
      Preheat the oven to 200 degrees F (95 degrees C). Butter and flour a baking sheet.
    </step>
    <step n="2">
      In a glass or metal bowl, whip egg whites until foamy using an electric mixer. Sprinkle in
    </step>
    <step n="3">
      Place the meringues in the oven and place a wooden spoon handle in the door to keep it from
    </step>
  </directions>
</recette>
```

Copy Text Node

1. French Meringues egg whites confectioners' sugar Preheat the oven to 200 degrees F (95 degrees C). Butter and flour a baking sheet. In a glass or metal bowl, whip egg whites until foamy using an electric mixer. Sprinkle in sugar a little at a time, while continuing to whip at medium speed. Place the meringues in the oven and place a wooden spoon handle in the door to keep it from closing all the way. Bake for 3 hours, or until the meringues are dry, and can easily be removed from the pan.

XPath

- Exemple
 - Recette.xml
 - Sélectionner le titre de la recette :
 - /recette/titre

XPath

XPath online real-time tester, evaluator and generator for XML & HTML — Mozilla Firefox

XPath online real-time test

xpather.com

/recette/titre

Elements found: 1

XML mode Format Save

```
<?xml version="1.0" ?>
<recette id="1336">
  <titre>French Meringues
</titre>
  <ingredients>
    <ingredient qte="4">egg whites
  </ingredient>
    <ingredient qte="2.25" unite="cup">confectioners' sugar
  </ingredient>
  </ingredients>
  <directions>
    <step n="1">
      Preheat the oven to 200 degrees F (95 degrees C). Butter and flour a baking sheet.
    </step>
    <step n="2">
      In a glass or metal bowl, whip egg whites until foamy using an electric mixer. Sprinkle in
    </step>
    <step n="3">
      Place the meringues in the oven and place a wooden spoon handle in the door to keep it from
    </step>
  </directions>
</recette>
```

Copy Text Node

1. French Meringues

XPath

- Exemple
 - Recette.xml
 - Sélectionner tous les ingrédients :
 - /recette/ingredients/ingredient
 - /recette//ingredient

XPath

XPath online real-time tester, evaluator and generator for XML & HTML — Mozilla Firefox

XPath online real-time test

xpather.com

/recette/ingredients/ingredient

Elements found: 2

XML mode Format Save

```
<?xml version="1.0" ?>
<recette id="1336">
  <titre>French Meringues
</titre>
  <ingredients>
    <ingredient qte="4">egg whites
  </ingredient>
    <ingredient qte="2.25" unite="cup">confectioners' sugar
  </ingredient>
  </ingredients>
  <directions>
    <step n="1">
      Preheat the oven to 200 degrees F (95 degrees C). Butter and flour a baking sheet.
    </step>
    <step n="2">
      In a glass or metal bowl, whip egg whites until foamy using an electric mixer. Sprinkle in
    </step>
    <step n="3">
      Place the meringues in the oven and place a wooden spoon handle in the door to keep it fir
    </step>
  </directions>
</recette>
```

Copy Text Node

1. egg whites
2. confectioners' sugar

XPath

XPath online real-time tester, evaluator and generator for XML & HTML — Mozilla Firefox

XPath online real-time test

xpather.com

/recette//ingredient

Elements found: 2

XML mode Format Save

```
<?xml version="1.0" ?>
<recette id="1336">
  <titre>French Meringues
</titre>
  <ingredients>
    <ingredient qte="4">egg whites
  </ingredient>
    <ingredient qte="2.25" unite="cup">confectioners' sugar
  </ingredient>
  </ingredients>
  <directions>
    <step n="1">
      Preheat the oven to 200 degrees F (95 degrees C). Butter and flour a baking sheet.
    </step>
    <step n="2">
      In a glass or metal bowl, whip egg whites until foamy using an electric mixer. Sprinkle in
    </step>
    <step n="3">
      Place the meringues in the oven and place a wooden spoon handle in the door to keep it from
    </step>
  </directions>
</recette>
```

Copy Text Node

1. egg whites
2. confectioners' sugar

XPath

- Exemple
 - Recette.xml
 - Sélectionner le id de la recette :
 - /recette/@id

XPath

XPath online real-time tester, evaluator and generator for XML & HTML — Mozilla Firefox

XPath online real-time test

xpather.com

/recette/@id

Elements found: 1

XML mode Format Save

```
<?xml version="1.0" ?>
<recette id="1336">
  <titre>French Meringues
</titre>
  <ingredients>
    <ingredient qte="4">egg whites
  </ingredient>
    <ingredient qte="2.25" unite="cup">confectioners' sugar
  </ingredient>
</ingredients>
  <directions>
    <step n="1">
      Preheat the oven to 200 degrees F (95 degrees C). Butter and flour a baking sheet.
    </step>
    <step n="2">
      In a glass or metal bowl, whip egg whites until foamy using an electric mixer. Sprinkle in
    </step>
    <step n="3">
      Place the meringues in the oven and place a wooden spoon handle in the door to keep it fir
    </step>
  </directions>
</recette>
```

1. 1336

Copy Text Node

XPath

- Exemple
 - Prédicats
 - Pour sélectionner des nœuds avec des valeurs précises,
 - Toujours entre crochets []

XPath

- Exemple
 - Prédicats
 - *position* :
 - Sélectionner un élément avec la position donnée,
 - Les éléments sont comptés à partir de 1,
 - Exemple : le deuxième ingrédient
 - `//ingredient[2]`

XPath

- Exemple
 - Prédicats
 - last() :
 - Sélectionner le dernier élément
 - Exemple : sélectionner le dernier ingrédient
 - //ingrédient[last()]
 - position() :
 - Retourne la position de l'élément
 - Exemple : sélectionner les deux première étapes :
 - //step[position() < 3]

XPath

- Exemple
 - Prédicats
 - @attribut :
 - Sélectionner les éléments qui possèdent l'attribut « attribut »,
 - @attribut=valeur
 - Sélectionner les éléments pour lesquels l'attribut « attribut » possède « valeur » comme valeur,
 - Exemple : sélectionner « step » numéro 2
 - //step[@n= "2"]

XPath

- Exemple
 - Prédicats
 - * : TOUT
 - /* : tous les éléments
 - /@* : tous les attributs

XPath

- Exemple
 - Sélectionner plusieurs chemins :
 - |
 - Exemple : sélectionner les étapes et les ingrédients :
 - `//ingredient|//step`

XPath

- Opérateurs :
 - Nous avons déjà utilisé deux opérateurs :
 - =
 - <

XPath

- Opérateurs :
 - Liste :

+	<
-	<=
*	>
div	>=
mod	and
=	or
!=	

Plan

- Exemple : SVG
- DOM
- XPATH
- **XSL**
- SAX
- XLINK
- XPOINTER

Transformation XSL (XSLT)

- Définition
 - eXtensible Stylesheet Language
 - XSLT est un langage de programmation (basé sur XML) qui permet de transformer les documents XML à d'autres formats,
 - Il peut aussi être utilisé pour transformer un document XML en un autre document XML (changement de schéma).

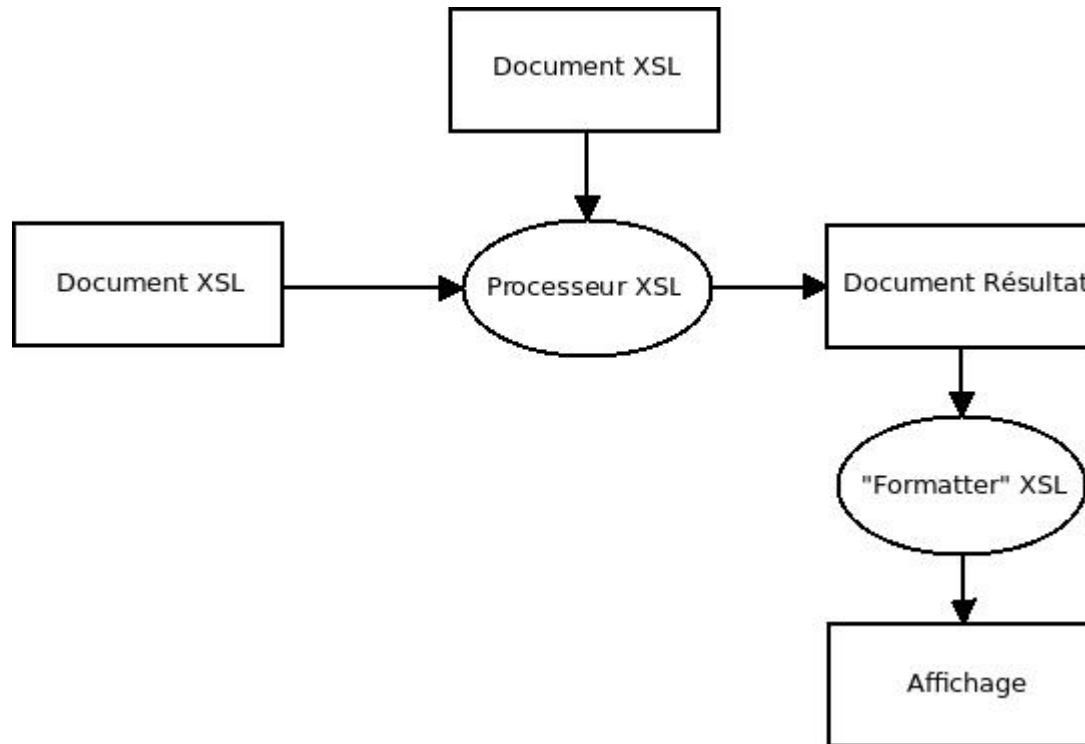
Transformation XSL (XSLT)

- Définition
 - XSLT est un langage de programmation (basé sur XML) qui permet de transformer les documents XML à d'autres formats,
 - Il peut aussi être utilisé pour transformer un document XML en un autre document XML (changement de schéma).

Transformation XSL (XSLT)

- Comment elles fonctionnent :

–



Transformation XSL (XSLT)

- Lier document XML à son feuille de transformation XSL :
 - `<?xml version = "1.0"?>`
 - `<?xml-stylesheet type = "text/xsl" href = "feuille.xsl"?>`

Transformation XSL (XSLT)

- Un fichier XSLT de base (avec le domaine nominal – namespace) :
 - `<?xml version = "1.0" ?>`
 - `<xsl:stylesheet`
 - `version = "1.0"`
 - `xmlns:xsl = "`
`http://www.w3.org/1999/XSL/Transform"`
 - `>`
 - `</xsl:stylesheet>`

Transformation XSL (XSLT)

- Éléments de base :
 - « template »
 - XSLT permet de définir une template, c'est à dire un modèle (ou « moule ») qui sera rempli à partir du fichier XML,
 - Une template s'applique sur un pattern (XPath) précisé dans sa définition.

Transformation XSL (XSLT)

- Éléments de base :
 - « template »
 - <xsl:template
 - name = QName
 - match = Pattern
 - priority = number
 - mode = QName
 - >
 - </xsl:template>

Transformation XSL (XSLT)

- Éléments de base
 - « value-of »
 - Récupérer la valeur d'un élément du fichier XML,
 - `<xsl:value-of`
 - `select = Expression`
 - `>`
 - `</xsl:value-of>`

Transformation XSL (XSLT)

- Éléments de base :
 - « for-each »
 - Itérer (boucle) pour parcourir une liste de nœuds,
 - `<xsl:for-each`
 - `select = Expression >`
 - `</xsl:for-each>`

Transformation XSL (XSLT)

- Éléments de base :
 - « sort »
 - Trier les nœuds (à l'intérieur d'une boucle, par exemple) suivant un critère donné (valeur d'un élément, d'un attribut, ...)
 - `<xsl:sort`
 - `select` = Expression
 - `data-type` = { "text" | "number" | QName }
 - `order` = { "ascending" | "descending" }
 - `</xsl:sort>`

Transformation XSL (XSLT)

- Éléments de base :
 - « if »
 - La structure conditionnelle, seuls les nœuds qui vérifient une condition donnée seront traités,
 - <xsl:if
 - test = « Expression booléenne » >
 - </xsl:if>
 - Exemple d'expression booléenne :
 - Valeur d'un élément ou d'un attribut comparée en utilisant les opérateurs de comparaison.

Transformation XSL (XSLT)

- Éléments de base :
 - « choose » avec « when » et « otherwise »
 - La structure conditionnelle multiple, (équivalente de switch, case et default en langages comme C et Java).

Transformation XSL (XSLT)

- Éléments de base :
 - « message »
 - Afficher un message,
 - Peut être utilisé pour exprimer une erreur (arrêt d'exécution)
 - `<xsl:message`
 - `terminate = "yes" | "no" >`
 - `</xsl:message>`
 - Si on met « terminate » à « yes », l'exécution s'arrête en affichant le message.

Transformation XSL (XSLT)

- Éléments de base :
 - « apply-template »
 - Il est possible de définir plusieurs, chacune traite un pattern (chemin) particulier,
 - La template principale aura pour rôle d'appliquer les bonnes templates au bon endroit suivant la structure du document XML,
 - L'appel se fait à travers « apply-template »

Transformation XSL (XSLT)

- Éléments de base :
 - « apply-template »
 - `<xsl:apply-template`
 - `select = Expression`
 - `>`
 - `</xsl:apply-template>`

Plan

- Exemple : SVG
- DOM
- XPATH
- XSL
- **SAX**
- XLINK
- XPOINTER

SAX

- SAX
 - Simple API for XML,
 - Permet de parser un fichier XML sans créer l'arbre DOM,
 - Repose sur le principe des évènements :
 - Parcour le fichier de son début à sa fin,
 - A chaque fois, un élément XML est rencontré, un évènement est signalé.

SAX

- Mode de développement sous SAX :
 - On ne manipule pas des neouds,
 - On écrit des méthodes pour recevoir et traiter les évènements,
 - Très similaire au principe des Listener utilisé par Java Swing.
 - Chaque méthode est spécialisée (appelée) en un type donné d'évènements,
 - A son arrivée, l'évènement (les éléments XML) est passé à la méthode adéquate.

SAX

- Mode de développement sous SAX :
 - Liste des évènements :
 - Varie selon la plateforme et le parseur utilisé,
 - Sous Java, la classe à étendre est la classe DefaultHandler :
 - <https://docs.oracle.com/javase/7/docs/api/org/xml/sax/helpers/DefaultHandler.html>

SAX

- SAX vs. DOM
 - SAX consomme moins d'espace et il est plus rapide,
 - Néanmoins,
 - Il ne permet pas une navigation dans la structure comme DOM,
 - Il est inefficace dans la création et l'interrogation.
 - SAX est utilisé pour parcourir les grands fichiers avec une petite profondeur.

Plan

- Exemple : SVG
- DOM
- XPATH
- XSL
- SAX
- **XLINK**
- XPOINTER

XLink

- Objectif
 - Définir un lien (hypertext),
 - L'équivalent de la balise « a » sous HTML,
 - Peut être attaché à n'importe quelle balise (pas de balise spécifique pour définir les liens)

XLink

- Utilité
 - Définition des structures Hypertext (et Hypermédia) en liant des éléments du documents à d'autres ressources sur Internet,
 - Annotation des documents et des ressources,
 - Indexation des documents et des ressources.

XLink

- Inconvéient
 - Pas de balise précise :
 - Non prise en charge par la majorité des navigateurs.

XLink

- Utilisation
 - Ajout du name space (domaine nominal) :
 - Préfixe : xlink
 - URI : "<http://www.w3.org/1999/xlink>"
 - Ajout des attributs :
 - type (xlink:type)
 - Pour spécifier le type du lien.
 - Suivant ce type, d'autres attributs sont ajoutés.

XLink

- Types des liens
 - Simple
 - Pour spécifier qu'il s'agit d'un lien simple (similaire à la balise « a » en HTML)
 - type = « simple »
 - Attributs :
 - href : lien vers la cible,
 - role : rôle du lien,
 - title : titre du lien,
 - ...

XLink

- Types des liens
 - Extended
 - Pour pointer vers plusieurs ressources,
 - Type = « extended »
 - Attributs :
 - role : rôle du lien,
 - title : titre du lien.

XLink

- Types des liens
 - Locator
 - Pour pointer vers une ressource externe,
 - Type = « locator »
 - Attributs :
 - href: le lien vers la ressource,
 - role : rôle du lien,
 - title : titre du lien.

XLink

- Types des liens
 - Arc
 - Pour designer un passage (au sens de traverser) d'une ressource à une autre,
 - Type = « arc »
 - Attributs :
 - title : titre du lien,
 - from : ressource du départ,
 - to : ressource d'arrivée.

XLink

- Types des liens
 - Resource
 - Pour designer une ressource interne,
 - Type = « resource »
 - Attributs :
 - title : titre du lien,
 - role : du lien,
 - label : étiquette du lien.

XLink

- Prise en charge,
 - XLink n'est pas pris en charge par la majorité des navigateurs,
 - Seule l'extension SVG inclut les liens de type simple,
 - Il est toujours possible de les utiliser si un parseur dédié (programme de traitement) est envisagé.

Plan

- Exemple : SVG
- DOM
- XPATH
- XSL
- SAX
- XLINK
- **XPOINTER**

XPointer

- Objectif :
 - Pointer une section d'un document XML,
 - Un « lien » plus « précis » :
 - Vers un point, un nœud ou une région.
- Principe :
 - Combinaison entre :
 - XLink : pour pointer le document,
 - XPath : pour pointer la section.

XPointer

- Syntax :
 - Un lien (xlink:href),
 - Une précision (expression Xpointer) après le symbole : #
 - Exemple :
 - xlink:href= « http://univ-jijel.dz/doc.xml#xpointer(id('cours_01')) »

XPointer

- Expressions XPointer
 - Accès par Id
 - Fonction : id()
 - id(« valeur »), ou plus simplement :
 - Valeur
 - http://univ-jijel.dz/doc.xml#cours_01

XPointer

- Expressions XPointer
 - Séquence de fils (child sequence)
 - En utilisant la position des nœuds.
 - Fonction : `element()`
 - Exemple : le deuxième fils, du troisième fils, de l'élément avec le id `cours_01` :
 - `element(cours_01/3/2)`

XPointer

- Expressions XPointer
 - Autres fonctions :
 - self()
 - origin()
 - here()

XPointer

- Expressions XPointer
 - Construction d'expression à base de blocs :
 - child Locates : noeuds fils du noeud sélectionné,
 - descendant : noeud à l'interieur du noeud sélectionné,
 - descendant-or-self : comme "descendant" mais le noeud sélectionné est inclu,
 - parent : le noeud parent (niveau supérieur direct) du noeud sélectionné,
 - ancestor : tous les noeuds supérieurs du noeud sélectionné,
 - ancestor-or-self : comme "ancestor" mais le noeud sélectionné est inclu,

XPointer

- Expressions XPointer
 - Construction d'expression à base de blocs :
 - preceding-sibling : "frère" précédent,
 - following-sibling : "frère" suivant,
 - preceding : les noeuds précédents,
 - following : les noeuds suivants,
 - attribute : accès aux attributs du noeud sélectionné.

Université de Jijel
Faculté des Sciences Exactes et d'Informatique
Département d'Informatique
L3 – Systèmes Informatiques

Données Semi-Structurées

Chapitre 04

Galaxie XML

Tarek Boutefara
t_boutefara@univ-jijel.dz
2020/2021