Fiche n°2

Introduction au XPATH appliqué au scraping de pages web.

Le langage XPATH est un langage de requête pour les documents XML. Il permet ainsi d'identifier un ou plusieurs éléments dans ce document. Toutes les applications et langages de programmation ayant besoin de repérer un fragment de document XML peuvent utiliser ce langage.

Différence HTML / XML XML et HTML sont tous les deux des langages de balisage. La principale différence est qu'il existe des dispositions dans XML permettant de définir de nouveaux éléments ou balises, alors que HTML utilise des balises prédéfinies. XML peut donc être utilisé pour créer des langages de marquage alors que HTML lui-même est un langage de marquage

Chemins :

La sélection d'un ou plusieurs éléments peut s'écrire de différentes manières à l'aide d'une expression XPath qui représente le chemin conduisant à (aux) élément(s) recherchés(s).

Un chemin s'écrit généralement comme une suite de sélecteurs depuis la racine du document. Les sélecteurs peuvent être des nœuds ou des attributs. Le chemin peut être relatif ou absolu.

Nom du nœud	Sélectionne ce qui est compris dans le nœud nommé.
/	Sélectionne en partant du nœud racine (chemin absolu)
//	Sélectionne en partant du nœud courant, peu importe le reste de l'emplacement (chemin absolu)
.	Sélectionne à partir du nœud courant (chemin relatif).
..	Sélectionne à partir du parent du nœud courant (chemin relatif)
@	Sélectionne les attributs.
|	Opérateur de sélection multiple.

Exemple : L'expression `/h1 /p/a` sélectionne tous les éléments « a » qui sont sous des éléments « p » qui sont sous des éléments « h1 ».

L'expression `/articles/text()` sélectionne tous les textes qui se trouvent sous un élément article.

L'expression `//a/@href` sélectionne tous les attributs «href »qui sont sous un élément « a », peu importe ce qu'il y a avant.

Filtres:

Pour faciliter l'identification de certains nœuds, notamment avec des balises html redondantes, on peut utiliser des filtres entre crochet pour sélectionner des éléments précis.

Exemples :

L'expression `//div/article [2]` sélectionne tous les deuxièmes articles qui se situent sous un élément div, peu importe ce qu'il y a avant.

L'expression `//h1[@class= "title"] /h2` sélectionne tous les éléments h2 situés sous un élément h1 ayant un attribut de class « title », peu importe ce qu'il y a avant.

L'expression `//a[contains (@href, ''www.utc.fr'')` sélectionne tous les éléments a dont l'attribut href contient la chaîne de caractère www.utc.fr, peu importe ce qu'il y a avant.

Vérifications :

Dans la pratique, les expressions XPath peuvent se révéler très complexes et par conséquent sujettes à des erreurs, c'est-à-dire qu'aucun résultat n'apparaît ou bien que le résultat obtenu n'est pas celui qui était attendu (sur-sélection ou sous-sélection). Il faut donc bien vérifier que seuls les éléments qui nous intéressent sont sélectionnés.

Ils existent plusieurs testeurs de chemins, soit dans l'onglet « Inspect » de votre navigateur, mais aussi sur XPather ou CodeBeautify.