Document et Web Sémantique - TP Schema.org

L'objectif de ce TP est de compléter le TP « Production d'un site Web » de façon à ajouter des métadonnées compatibles avec `Schema.org` au code HTML5 généré. Pour tester le code obtenu, vous utiliserez l'outil de schema.org : `https://validator.schema.org/`

Sélection des métadonnées

En parcourant le site `https://schema.org/`, étudiez les classes `Person`, `CourseInstance`, `Course`.

Utilisation des métadonnées

Modifiez vos feuilles `XSL-T` pour insérer des métadonnées au format JSON-LD, de façon à préciser :
— qu'il s'agit d'un cours ;
— la durée du cours (à partir des crédits) ;
— les informations sur le responsable de ce cours et les intervenants ;
— la langue du cours ;
— l'emplacement du cours ;
— dates de début et de fin ;
— la description ;
— le nom ;
— le public cible ;
— les URL du cours.

Conseils

Je vous conseille de partir du code HTML que génère votre feuille *ec.xsl* que vous copierez dans l'outil de validation. Vous pourrez alors, depuis cet outil, ajouter de manière statique les métadonnées désirées en vérifiant à chaque fois qu'elles sont bien interprétées.

Une fois le cadre de ce que vous voulez ajouter bien défini, vous modifierez votre fichier *ec.xsl* pour que ces métadonnées soient ajoutées dynamiquement.

N'oubliez pas que vous pouvez ajouter des `mode` à vos *template* XSL-T.

Complément d'information sur JSON-LD

La version 1.1 de JSON-LD a été standardisée par le W3C en juillet 2020 [1].

1. `https://www.w3.org/TR/json-ld11/`

JSON-LD se fonde sur JSON, un format de sérialisation de données structurées (issu de Javascript).

JSON

JSON utilise le concept de clés/valeurs (dictionnaire). Un ensemble de couples clés/valeurs est appelé un objet, qui débute par une accolade ouvrante '`{`' et finit par une accolade fermante '`}`'. Les deux points '`:`' séparent la clé de la valeur. Les couples clés/valeurs sont séparés par une virgule '`,`'. Les clés sont représentées par des chaînes de caractères entourées de doubles guillemets '`"`'. Une valeur est soit :

— une chaîne de caractères entourée de doubles guillemets '`"`' ;
— une liste de valeurs entourée de '[' et ']' et séparées par des virgules '`,`' ;
— un objet.

Voici un exemple issu l'article Wikipédia [2] sur ce format :

```
{
    "menu": {
        "id": "file",
        "value": "File",
        "popup": {
            "menuitem": [
                { "value": "New", "onclick": "CreateNewDoc()" },
                { "value": "Open", "onclick": "OpenDoc()" },
                { "value": "Close", "onclick": "CloseDoc()" }
            ]
        }
    }
}
```

Propriétés de JSON-LD

Dans JSON-LD les clés sont les métadonnées.

Indépendemment du schéma de métadonnées utilisé, certaines clés sont proposées par JSON-LD, en voici quelques unes :

`@context` permet de définir l'URI du schéma de métadonnées utilisé, voire à l'aide d'un objet JSON-LD de lister les URI des schémas de métadonnées utilisés en leur associant un préfixe, par exemple :

```
"@context": {
        "foaf": "http://xmlns.com/foaf/0.1/",
        "sch": "http://schema.org/"
    }
```

`@graph` permet de créer un graphe nommée qui regroupe au sein d'une liste un ensemble d'objets JSON-LD ;

`@id` permet d'associer un identifiant (URI) à l'objet JSON-LD courant ou de désigner un objet à l'aide d'un identifiant ;

`@base` permet de définir un préfixe aux identifiants locaux ;

`@type` permet de définir le type de l'objet courant ;

2. `https://fr.wikipedia.org/wiki/JavaScript_Object_Notation`

`@language` permet de définir la langue des valeurs linguistiques de l'objet courant (présent dans le `@context`).