

# JavaScript

## DOM 2 Core

Par Richard Vigneux

## Table des matières

- Introduction
- Définitions
- Interfaces génériques
- Interface Node
- Interface Document
- Interface Element
- Autres interfaces
- Exemple
- Exercice du jour

## Introduction

- Le DOM (Document Object Model) est une recommandation du W3C qui décrit une interface indépendante de tout langage de programmation permettant de manipuler un document Web à l'aide de scripts.
- Le DOM2 Core permet d'aborder un document comme un arbre (sous forme de nœuds).
- La majorité des interfaces offertes par le « core » sont des nœuds (ou en sont des spécialisations (héritage)): Node, Document, Element, CharacterData, DocumentType, Notation, ProcessingInstruction, Entity.
- Prenez note que nous n'étudierons pas le DOM 3 dans le cadre du cours, puisque ce dernier n'ajoute que des éléments très spécialisés au DOM 2. Le DOM 3 ne devrait pas vous manquer à moins de devenir un spécialiste de la manipulation de documents avec le DOM.
- Références: <http://www.w3.org/TR/DOM-Level-2-Core/>  
<http://www.w3.org/TR/DOM-Level-3-Core/>

3

## Définitions

- Dans la documentation du W3 vous remarquerez que les définitions des interfaces se retrouvent dans des sections nommées « IDL Definition ». Pour votre culture, sachez que l'acronyme « IDL » signifie « Interface Definition Language ». IDL est un langage voué à la définition de l'interface de composants logiciels, laquelle permet de faire communiquer les modules implémentés dans des langages différents. IDL est défini par l'OMG (Object Management Group) qui est une association américaine à but non lucratif créée en 1989 dont l'objectif est de standardiser et promouvoir le modèle objet sous toutes ses formes. Ce groupe est à la base des standards UML (Unified Modeling Language).
- Dans les définitions des interfaces, vous remarquerez que certaines méthodes sont suffixées de « NS ». Vous remarquerez aussi que le nom de méthode qui précède le « NS » existe aussi sans « NS » dans la même définition. Les méthodes avec le suffixe « NS » ne font rien de plus que celles qui ne l'ont pas. La seule différence est que celles avec « NS » permettent d'indiquer l'espace de nom (NameSpace). L'espace de nom est seulement le cadre de référence pour interpréter l'élément.

4

# Interfaces génériques

Interface	Raison d'être
DOMException	Offre un ensemble de constantes représentant des codes d'erreurs (il y a erreur lorsqu'une opération est impossible à réaliser).
DOMImplementation	Permet principalement de créer un objet Document et de vérifier si un élément du DOM est supporté par le navigateur.
DocumentType	Permet d'accéder aux informations du DOCTYPE (ne fonctionne pas avec IE8).
Notation, ProcessingInstruction, Entity	Interfaces très spécialisées qui sont utilisées principalement avec XML (héritent de « Node »).
DocumentFragment	Objet représentant une partie de l'arbre du document (est un « Node »).

5

# Interface Node

Cette interface est l'interface principale.

- Toutes les propriétés sont en lecture seulement sauf « nodeValue ».
- Cette interface permet principalement de:
  - ◆ Parcourir l'arbre du document;
  - ◆ Obtenir de l'information sur un nœud (type, attributs, position dans l'arbre (parent, enfants));
  - ◆ Ajouter, modifier et détruire des nœuds.
- Référence:
  - ◆ <http://www.w3.org/TR/DOM-Level-2-Core/core.html#ID-1950641247>

6

## Interface Document

Cette interface hérite de l'interface « Node ».

- Toutes les propriétés sont en lecture seulement.
- Cette interface permet principalement de:
  - ◆ Créer les différents types de nœud;
  - ◆ Récupérer un pointeur sur un nœud identifié (ID);
  - ◆ Récupérer une liste de nœuds d'un certain type.
- Référence:
  - ◆ <http://www.w3.org/TR/DOM-Level-2-Core/core.html#i-Document>

7

## Interface Element

Cette interface hérite de l'interface « Node ».

- Sa seule propriété est en lecture seulement.
- Cette interface permet principalement de créer et de manipuler les attributs d'un nœud.
- Référence:
  - ◆ <http://www.w3.org/TR/DOM-Level-2-Core/core.html#ID-745549614>

8

## Autres interfaces

Interface	Raison d’être
NodeList	Permet de manipuler des listes de nœuds ordonnées (par exemple, la liste des nœuds de balises « p » du document).
NamedNodeMap	Permet de manipuler des listes nommées (par exemple, la liste des attributs d’une balise)
Attr	Permet de manipuler un objet attribut d’une balise (Un « Attr » est créé par l’interface Document et modifié à l’aide de l’interface Element).
CharacterData, Comment	Permet de manipuler les chaines de caractères (hérite de « Node »).
Text, CDATASection	Permet de manipuler les nœuds de type « text ». (hérite de « CharacterData »)

9

## Exemple

Voici un exemple pour vous aider à débiter.

- Création et ajout d’une liste au document:

```
var ptr_CorpsDuDocument = document.getElementsByTagName("body")[0];

var obj_ListeNonOrdonnee = document.createElement("ul");
var obj_ElementDeListe   = document.createElement("li");
var obj_TexteElementListe = document.createTextNode("Element_1");

obj_ElementDeListe.appendChild (obj_TexteElementListe);
obj_ListeNonOrdonnee.appendChild(obj_ElementDeListe);
ptr_CorpsDuDocument.appendChild (obj_ListeNonOrdonnee);
```

- Attention: Ce code ne peut être exécuté tant et aussi longtemps que l’ensemble du document n’a pas été chargé. Utilisez « onload » sur la balise body pour vous assurer de la fin du chargement.

10

## Exercice du jour

- Dans l'exercice d'aujourd'hui, vous devez transformer un document XHTML en un document XHTML dont le corps (partie contenue dans la balise « body » est généré à l'aide du DOM Core exclusivement.
- Pour ce faire, je vous ai fourni le document complet en XHTML ainsi que sa feuille de styles CSS. Vous pouvez tester le tout si vous le désirez. De plus, je vous ai fourni un autre document XHTML qui doit vous servir de document de base pour réaliser l'exercice d'aujourd'hui.
- J'insiste pour que vous créiez un document extrêmement lisible (noms) et structuré (fonction et boucle). De plus, je m'attends à ce que toutes valeurs utilisées dans le code qui peut faire l'objet d'une constante (variable globale) soient déclarées au début du programme. Aucune répétition ne sera tolérée lors de la correction.

11

# JavaScript FIN

DOM 2 Core  
Par Richard Vigneux