Table des matières

2

Introduction

- Comme nous l'avons vue au dernier cours, lorsqu'on veut manipuler un document sous sa forme objet, on utilise le DOM 2 Core. Par contre, lorsque vient le temps de manipuler les feuilles de styles de ce document, le DOM 2 Core n'est d'aucune utilité.
- Pour manipuler les styles CSS appliqués à un document, nous avons besoin d'autres interfaces qui nous permettront de manipuler les styles sous leur forme objet. Pour ce faire, le W3C a développé la norme DOM 2 Style.
- La norme DOM 2 Style offre toutes les interfaces (classe) nécessaires pour manipuler l'ensemble des styles CSS appliqué à un document.
- La norme DOM 2 Style offre deux façons d'aborder les styles appliqués:
  - La première permet de travailler avec les différentes feuilles de styles appliqués à un document.
  - La deuxième permet de travailler avec les styles appliqués à un élément particulier.
- Références: http://www.w3.org/TR/2000/REC-DOM-Level-2-Style-20001113/

3

Récupérer un pointeur sur une feuille de style

Il y a deux façons de récupérer un pointeur sur les feuilles de style CSS d'un document.

- La première consiste à utiliser l'interface « LinkStyle » et sa propriété « sheet ». Pour pouvoir utiliser cette propriété, on doit partir d'un objet « HTMLStyleElement » qui peut être obtenu à l'aide de la méthode « getElementByTagName » du DOM 2 Core.
  - document.getElementByTagName ("style")[0].sheet
  - document.getElementByTagName ("link")[0].sheet

Attention, vous devez indiquer l'indice de la feuille que vous désirez manipuler.

- La deuxième approche, qui permet plus facilement de manipuler tous les types de feuilles liées, consiste à utiliser l'interface « DocumentStyle » et sa propriété « styleSheets ». Cette propriété retourne la liste des feuilles de styles attachées au document. Il faudra alors utiliser l'interface « StyleSheetList » pour récupérer une feuille en particulier ou connaitre le nombre de feuilles de styles liées.
  - obj_FeuilleStyles = document.styleSheets[0];
  - obj_FeuilleStyles = document.styleSheets.item(0);

Attention, vous devez indiquer l'indice de la feuille que vous désirez manipuler.

4

Gestion d'une feuille de style

À partir du moment où vous avez un pointeur sur une feuille de style, vous pouvez manipuler ses règles.

- L'interface « StyleSheet » représente une feuille de style et offre plusieurs propriétés permettant d'obtenir de l'information sur la feuille. De plus, une de ses propriété, « disabled », permet d'activer ou de désactiver une feuille (c'est la seule propriété de cette interface qui est en lecture/écriture).
- L'interface la plus intéressante à ce niveau est « CSSStyleSheet » qui hérite de l'interface « StyleSheet ». Cette interface offre une méthode pour ajouter une règle et une autre pour détruire une règle. De plus, cette interface offre la propriété « cssRules » qui permet de récupérer la liste de toutes les règles de la feuille de styles CSS. Pour connaître le nombre de règles d'une feuille de styles ou pour accéder à une règle en particulier nous utiliserons l'interface « CSSRuleList ».
  - obj_Regle = obj_FeuilleStyles.cssRules[0];
  - obj_Regle = obj_FeuilleStyles.cssRules.item(0);

Attention, vous devez indiquer l'indice de la règle que vous désirez manipuler.

5

Gestion des règles

À partir du moment où vous avez un objet règle, vous pouvez le manipuler.

- L'interface « CSSRule » représente une règle. Cette interface offre deux méthodes et deux propriétés.
  - La première propriété, « type », permet de savoir le type de règle et donc de pouvoir déterminer la façon dont on pourra la manipuler.
  - La deuxième propriété, « cssText », permet de récupérer ou de modifier l'ensemble de la règle sous sa forme textuelle.
- Il existe trois interfaces principales qui héritent de l'interface « CSSRule ».
  - La première qui est « CSSCharsetRule » représente une règle « @charset » et permet de lire ou modifier l'encodage de caractère lié au document.
  - La seconde, « CSSImportRule », représente une règle « @import » et permet de lire ou modifier les propriétés de ce type de règle.
  - Enfin, la dernière que je vous présente, et qui est la plus importante est « CSSStyleRule » qui permet de manipuler les règles de styles ordinaires. Cette dernière offre deux propriétés.
    - La première, « selectorText », permet de récupérer le sélecteur de la règle.
    - La deuxième, « style », permet d'accéder à la déclaration de la règle (c'est ce qui se retrouve entre les accolades { } lorsque vous écrivez une règle CSS).
  - Obj_Declaration = obj_Regle.style;

6

Gestion de la déclaration d'une règle

À partir du moment où vous avez un objet déclaration, vous pouvez le manipuler.

- L'interface « CSSStyleDeclaration » représente une déclaration. Cette interface offre plusieurs propriétés dont « cssText » qui permet de lire ou d'écrire une propriété avec ses valeurs en format texte et « length » qui permet de savoir le nombre de couples propriété/valeur présents dans la déclaration. De plus, cette interface offre une méthode pour parcourir la liste des couples propriété/valeur et des méthodes pour récupérer, créer, modifier ou détruire des propriétés ou leurs valeurs.
  - var_ContenuDeclarationEnTexte = obj_Declaration.cssText;
  - var_ValeurProprieteFontSize = obj_Declaration.getPropertyValue("font-size");
  - obj_Declaration.setProperty ("font-size", "14px", "");
- Une autre interface est « CSS2Properties » qui hérite de l'interface « CSSStyleDeclaration ». Cette interface offre l'ensemble des propriétés CSS sous la forme d'objets.
  - obj_Declaration.fontSize = "14px";

7

Manipulation de la déclaration CSS d'un élément

Il existe deux façons d'accéder aux déclarations de styles d'un élément (balise).

- La première, qui utilise l'interface « ElementCSSInlineStyle », consiste à récupérer les styles qui ont été définis directement sur une balise à l'aide de l'attribut « style ». Cette interface utilise un objet « HTMLElement » obtenu par l'entremise du DOM 2 Core.
  - obj_Declaration = document.getElementById("attributID").style;
- La deuxième, qui utilise l'interface « ViewCSS », permet de récupérer les styles qui sont appliqués réellement à un élément quelle que soit la source de ce style (local, interne, externe et même les styles par défaut appliqués par le navigateur). L'avantage de cette interface est qu'on obtient vraiment les valeurs qui sont réellement appliquées sur l'élément.
  - obj_Element = document.getElementById("attributID");
  - obj_Declaration = window.getComputedStyle(obj_Element, null);
- Enfin, sachez pour votre culture que le W3C définit l'interface « DocumentCSS » qui n'a pas été implantée par les navigateurs. Cette interface aurait permis de créer des styles qui auraient préséance sur toutes les autres définitions de styles.

8

Exercice du jour

- Dans l'exercice d'aujourd'hui, vous devez, à l'aide du DOM 2 Core et principalement à l'aide du DOM 2 Style, créer la feuille de styles CSS du document XHTML de l'exercice de la semaine dernière qui vous avait été fourni.
- Pour ce faire, je vous demande de créer un tableau à trois dimensions qui contiendra l'ensemble des règles de la feuille de styles fourni au dernier cours. Utiliser une déclaration explicite avec des crochets [ ] pour rendre le tout plus lisible (voir référence JSON).
- Suite à la création du tableau, vous devez créer minimalement une fonction pour générer la feuille de style du document. Vous devez créer une feuille de styles interne globale.
- Enfin, je vous demande de créer une procédure pour afficher différentes informations concernant la feuille de styles du document. L'objectif ici est de parcourir la feuille de styles pour récupérer les différentes informations demandées.
- Bien entendu, je m'attends à un code hautement lisible et structuré.

9

JavaScript FIN

DOM 2 Style

Par Richard Vigneux