

Web (Persistence)



Andrea G. B. Tettamanzi

Université de Nice Sophia Antipolis

Département Informatique

andrea.tettamanzi@unice.fr

CM - Séance 10

Programmation côté client en JavaScript

Plan

- Objets et leur utilisation
 - N.B. : on abordera plus en profondeur le sujet de la POO en JavaScript lors de la dernière séance
- Objets standard
- Gestion des erreurs
- Document Object Model

Objets

- Un objet JavaScript peut être pensé comme un ensemble de couples (propriété, valeur) que l'on peut représenter avec la notation intuitive
 - propriété : valeur
- qui est aussi utilisé par la syntaxe du langage pour définir des objets littéraux
- Propriétés
 - Propres : apparaissent explicitement dans un objet
 - Héritées : ne font pas explicitement partie de l'objet, mais sont dérivées implicitement d'autres objets (dits prototypes)
- Une propriété peut prendre des valeurs de n'importe quel type
- Les objets JavaScript sont des tables de hachage

Méthodes

- Une propriété peut avoir une fonction comme valeur
- On parle alors d'une méthode
- Lorsqu'une méthode d'un objet est appelée, la variable spéciale « this » passe à référencer l'objet
- De ce fait, le code contenu dans la méthode peut accéder aux propriétés de l'objet à travers de la variable « this »

Création d'objets

- Un objet peut être créé de plusieurs façons, parmi lesquelles
 - Syntaxiquement, à l'aide d'une notation littérale :
`obj = { x : 5, y : 10 } ;`
 - Par des constructeurs
 - Un constructeur, en JavaScript, est une fonction qui est appelée avec l'opérateur « new »
`new f(...)` ;
 - Renvoie un nouvel objet moulé sur l'objet `f.prototype`
 - `Object.create(proto)`.

Utilisation d'un objet

- Soit « obj » une référence à un objet
 - Par exemple, `obj = new f()`
- Pour accéder à une propriété d'un objet, on utilise l'opérateur « . »
 - `obj.prop`
- Pour appeler une méthode d'un objet, on utilise l'opérateur « . »
 - `obj.m(<arguments>)`

Objets intégrés

- Un certain nombre d'objets intégrés sont disponibles lors qu'un programme JavaScript est exécuté.
- L'objet global fait directement partie de l'environnement lexical du programme
 - Il contient les variables et les fonctions « globales »
- Les autres objets intégrés sont accessibles comme propriétés initiales de l'objet global
- Beaucoup des objets intégrés sont des fonctions
 - Ils peuvent être appelés avec des arguments
 - Certains sont, en outre, des constructeurs
- Les noms de ces objets s'inspirent lourdement aux noms des classes de la plate-forme Java et de leurs méthodes et attributs

L'objet « Object »

- Permet de créer un objet de conversion pour une valeur donnée
- Par défaut, le prototype des objets créés est « null »
- Parmi ses méthodes on trouve
 - `create` : crée un nouvel objet avec l'objet prototype et les propriétés spécifiées
 - `keys` : renvoie un tableau contenant les noms de toutes les propriétés énumérables de l'objet donné
 - `getPrototypeOf(o)` : renvoie le prototype de l'objet `o` passé comme argument

L'objet « Function »

- Toute fonction en JavaScript est en réalité un objet qui hérite le prototype de Function
- Comme constructeur Function crée des fonctions à run-time
- Parmi les méthodes des objets créés par Function on peut citer :
 - `apply(o [, args])` : applique l'objet fonction comme s'il était une méthode de l'objet `o` passé comme argument ; les arguments de la fonction doivent être passés comme un objet Array
 - `bind` : crée une nouvelle fonction qui, lorsqu'elle est appelée, invoque cette fonction comme si c'était une méthode de la valeur fournie
 - `call` : la même chose que `apply`, mais avec les arguments passés un par un au lieu que comme un objet Array

L'objet « Array »

- Constructeur de tableaux
- Conteneurs de haut niveau, dont le comportement et les caractéristiques les rapprochent à des listes
- Un tableau peut être créé soit par une expression littérale,
 - [élément 0 , élément 1, . . . , élément n-1]
- soit par une invocation du constructeur Array :
 - new Array(élément 0 , élément 1 , . . . , élément n-1)
 - new Array(n)
- La propriété length d'un tableau contient toujours sa taille

L'objet « String »

- Constructeur de chaînes de caractères
- La propriété length d'une chaîne de caractères contient sa taille
- Des chaînes de caractères peuvent être comparées en utilisant les opérateurs de comparaison <, <=, ==, >=, >, et !=
- L'opérateur de concaténation est +
- De nombreuses méthodes utilitaires font partie de cet objet :
 - charAt, concat, contains, endsWith, indexOf, match, replace, search, slice, split, substr, trim, toLowerCase, toUpperCase, ...

L'objet « Math »

- Un objet qui possède des propriétés et des méthodes qui mettent à disposition du programmeur des constantes et des fonctions mathématiques.
- Math n'est pas un constructeur
- Toutes les propriétés et méthodes de Math sont statiques

L'objet « Date »

- Crée des objets qui représentent des dates et des temps
- Permet de manipuler des dates et des temps
 - `new Date()`
 - `new Date(u)` : avec l'heure POSIX
 - `new Date(chaîne)`
 - `new Date(A, M , J [, h, m, s, ms])`
- L'objet Date fournit trois méthode d'utilité
 - `now` : renvoie l'heure POSIX actuelle
 - `parse` : analyse une date en format chaîne de caractères
 - `UTC` : prend les mêmes arguments de le forme la plus longue du constructeur et renvoie l'objet Date correspondant.

Gestion des erreurs

- Le mécanisme de gestion repose sur trois ingrédients :
 - un ou plusieurs routines de traitement d'exceptions (les handlers) ;
 - un mécanisme de signalement d'exceptions ;
 - un mécanisme qui permet d'associer les exceptions à leurs handlers.
- En JavaScript, le mécanisme d'association des exceptions à leurs handlers est fourni par la structure `try ... catch ... finally`
- Les routines de traitement d'exception sont contenues dans les clauses « `catch` » de cette construction
- Le signalement des exception se fait à l'aide de l'opérateur « `throw` » et d'objets créés par le constructeur `Error`.

Document Object Model (DOM)

- C'est-à-dire le modèle orienté objet d'une page Web
- Standard du W3C
 - Interface d'accès à la structure et au style de documents XML et HTML
 - indépendant du langage de programmation
 - Indépendant de la plate-forme
- Permet de lire ou mettre à jour le contenu de la page
- Interaction étroite avec le navigateur

Niveaux du DOM

- DOM 0. Version de Netscape Navigator 2.0
- DOM 1 (1998). Internet Explorer 5 / Netscape 6
 - Représentation d'un document sous forme d'un arbre.
 - Chaque élément correspond à un nœud
 - Méthodes permettant de manipuler cet arbre
- DOM 2 (2000). Constitué de six parties :
 - Core, HTML, Events, Style, View, Traversal and Range
- DOM 3 (2004). Version actuelle et définitive

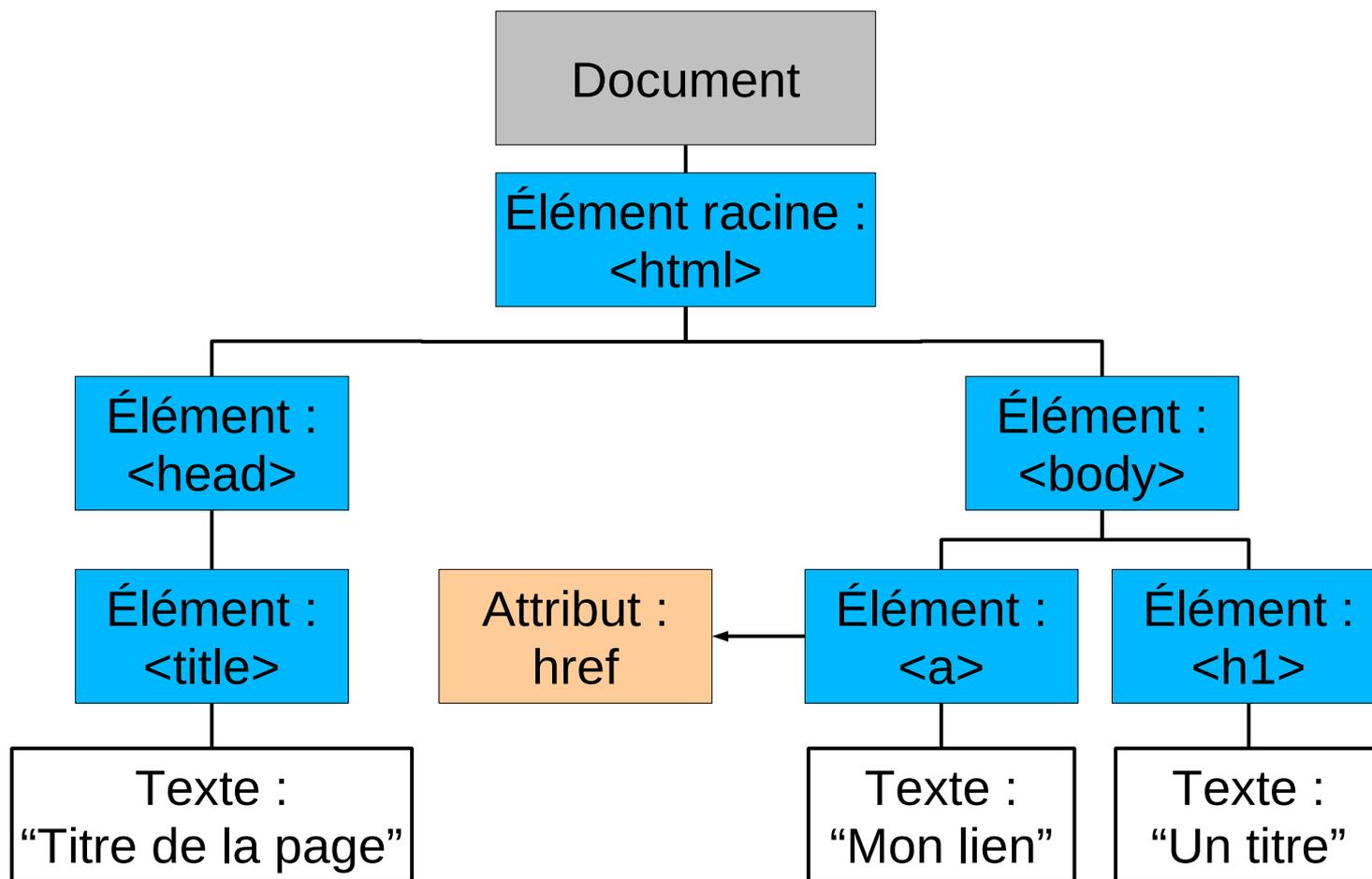
Structure du standard DOM

- Noyau (Core DOM)
 - Modèle standard pour n'importe quel document structuré
- Modèle XML (XML DOM)
 - Objets et propriétés de tous les éléments XML
 - Méthodes pour y accéder et les manipuler
- Modèle HTML (HTML DOM)
 - API standard pour manipuler des pages HTML
 - Objets et propriétés de tous les éléments HTML
 - Méthodes pour y accéder et les manipuler

Nœuds DOM

- Tout est un nœud dans un document HTML :
 - le document dans son complexe est un **nœud document**
 - chaque élément HTML est un **nœud élément**
 - le texte dans les éléments HTML est constitué par des **nœuds texte**
 - chaque attribut HTML est un **nœud attribut**
 - même les commentaires sont des **nœuds commentaire**.
- Le document lui-même est un arbre, dont la racine est le nœud document
- On peut accéder à tous les nœuds de l'arbre avec JavaScript

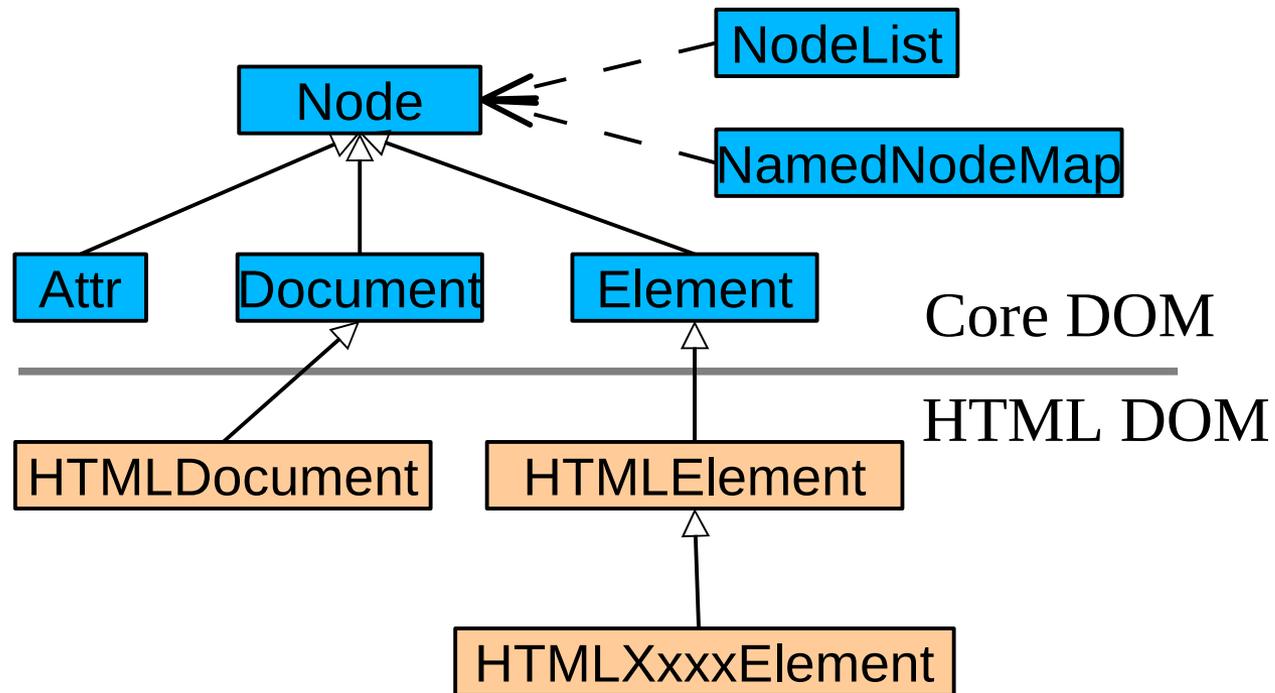
Exemple d'arbre DOM



Interface de programmation

- L'objet « document » encapsule le document HTML
 - Méthode getElementById(id)
 - getElementsByTagName(n) et getElementsByClassName(n)
- Objets nœuds éléments
 - Méthode appendChild(nœud) : ajoute un nouveau nœud fils
 - Méthode removeChild(nœud) : élimine un nœud fils
 - Propriétés :
 - innerHTML : le texte HTML contenu dans un nœud
 - parentNode : le nœud parent d'un nœud
 - childNodes : la collection des nœuds fils d'un nœud
 - attributes : les nœuds attributs d'un nœud

Les objets DOM et leurs relations



Objets du noyau

- Node, représente un nœud d'un document HTML
- NodeList, une collection ordonnée de nœuds, dont les éléments peuvent être récupérés par leur indice à l'aide de la méthode `item(i)`, où $i = 0, 1, \dots, \text{length}$
- NamedNodeMap, une collection non ordonnée de nœuds, dont les éléments peuvent être récupérés par leur nom
- Document, qui représente un document abstrait et contient des méthodes pour créer, modifier et récupérer des nœuds
- Element, un élément HTML (donc un cas particulier d'un Node)
- Attr, un attribut d'un élément HTML (cas particulier d'un Node)

Objets HTML

- HTMLDocument (hérite de Document)
- HTMLInputElement (hérite de Element)
- Pour chaque balise HTML, l'objet correspondant s'appelle HTMLXxxxElement, par exemple :
 - <a> → HTMLAnchorElement
 - <p> → HTMLParagraphElement
 - <input> → HTMLInputElement
 - → HTMLUListElement
 - Etc.

Objets « navigateur »

- Window
 - Navigator
 - Screen
 - History
 - Location
-
- Ces objets permettent d'accéder à des propriétés et à des méthodes qui ne sont pas liées au document, mais à l' « agent utilisateur » qui le visualise

Merci de votre attention

