

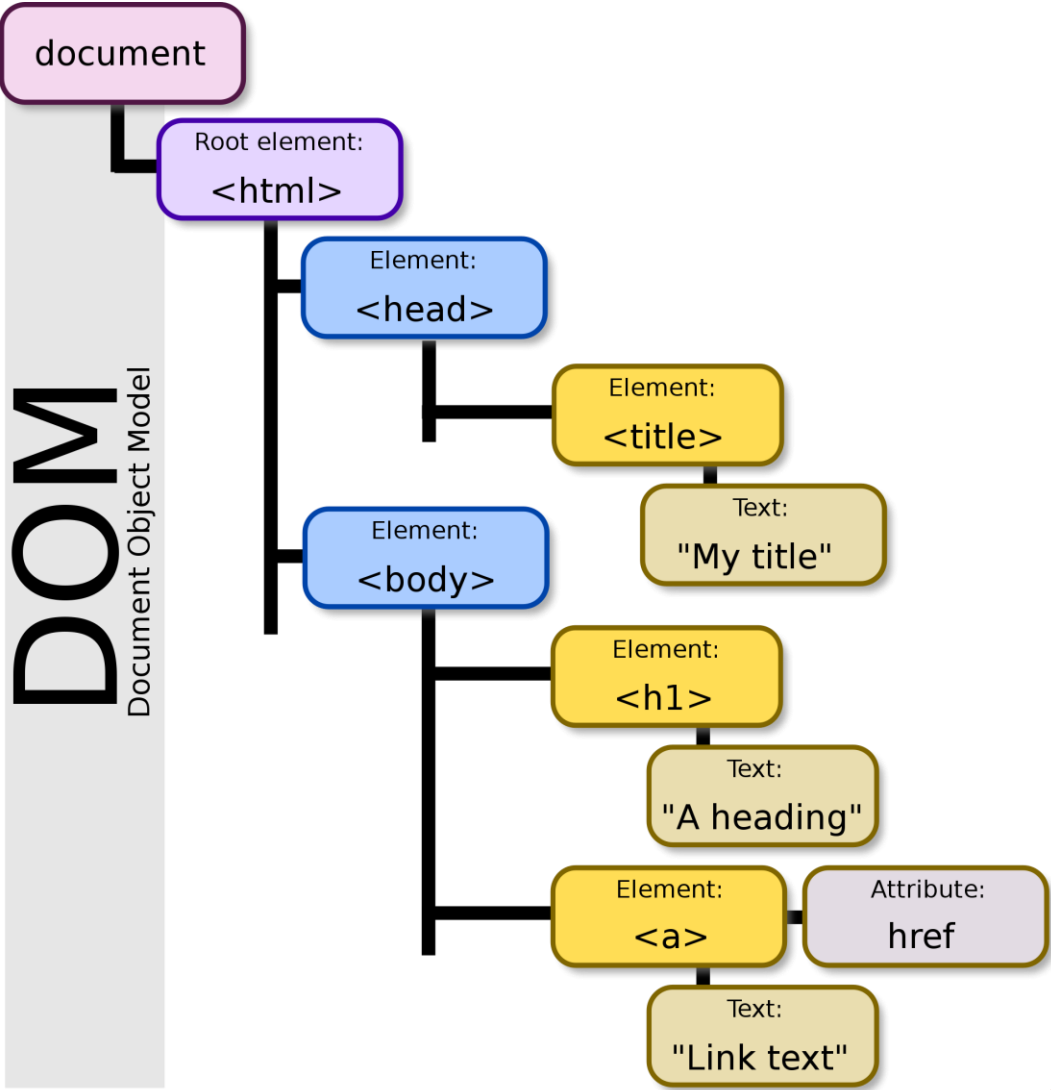
420-1WE-BB

Web dynamique

DOM

Tirées des notes de Jean-François Brodeur

DOM



Le Modèle Objet de Document (DOM: Document object Model)

- ▶ Le DOM est la représentation de donnée d'objets qui comprend la structure et le contenu d'un document web
- ▶ Le DOM représente un document HTML en mémoire et comment vous utilisez les API - Application Programming Interface
- ▶ Il fournit une page dont les programmes peuvent modifier la structure, son style et son contenu. Cette représentation du document permet de le voir comme un groupe structuré de nœuds et d'objets possédant différentes propriétés et méthodes. Pour créer du contenu web et des applications.

DOM

- ▶ Le DOM est une représentation entièrement orientée objet de la page Web, et peut être manipulé à l'aide d'un langage de script comme JavaScript.
- ▶ Chacune des propriétés et des méthodes et chacun des évènements disponibles permettant la manipulation et la création de pages Web sont organisés dans des objets (par exemple, l'objet document qui représente le document lui-même, l'objet table qui implémente l'interface DOM particulière HTMLTableElement permettant d'accéder aux tables HTML, etc.).

DOM & JS

- ▶ Le DOM n'est pas un langage de programmation, mais sans lui le langage JavaScript n'aurait aucun modèle ni aucune notion des pages Web,
- ▶ À l'origine, JavaScript et le DOM étaient fortement liés, mais ils ont fini par évoluer en deux entités distinctes.
- ▶ Le contenu de la page est stocké dans le DOM et on peut y accéder et le manipuler via JavaScript, de la sorte qu'on pourrait écrire cette équation approximative :
- ▶ $\text{API}(\text{page Web}) = \text{DOM} + \text{JS}(\text{langage de script})$

Les éléments racines du DOM

- ▶ En accédant aux éléments document ou window, vous pouvez manipuler le document lui-même ou parcourir les enfants de ces éléments, qui sont les divers éléments de la page Web.

Les objets Windows et Document

- ▶ L'objet window représente une fenêtre contenant un document DOM; la propriété document pointe vers le document DOM chargé dans cette fenêtre. Une fenêtre pour un document donné peut être obtenue en utilisant la propriété document.defaultView.
- ▶ Exemples de propriétés et méthodes:
(<https://developer.mozilla.org/fr/docs/Web/API/Window>)
 - ▶ Window.document: (Lecture seule) Renvoie une référence au document que la fenêtre contient.
 - ▶ Window.fullScreen: Cette propriété indique si la fenêtre est affichée en plein écran ou non.
 - ▶ Window.alert(): Affiche une boîte de message d'alerte.
 - ▶ Window.close(): Ferme la fenêtre en cours.

Document

- ▶ L'interface Document représente n'importe quelle page Web chargée dans le navigateur et sert de point d'entrée dans le contenu de la page Web, qui est l'arborescence DOM.
- ▶ L'arborescence DOM inclut des éléments tels que <body> (corps) et <table> (tableau), parmi beaucoup d'autres. Il fournit des fonctionnalités globales au document, comme le moyen d'obtenir l'URL de la page et de créer de nouveaux éléments dans le document.
- ▶ L'interface Document décrit les propriétés et méthodes communes à tout type de document.

Exemples de propriétés et méthodes

- ▶ `Document.documentElement` (Lecture seule)
 - ▶ Renvoie l'Element qui est l'enfant direct du document. Pour les documents HTML, il s'agit normalement de l'élément `<html>`.
- ▶ `Document.lastStyleSheetSet` (Lecture seule)
- ▶ retourne le nom de l'ensemble de feuilles de style qui a été activé en dernier. A la valeur null jusqu'à ce que la feuille de style soit modifiée en définissant la valeur de `selectedStyleSheetSet`.

HTMLDocument

- ▶ L'interface Document, pour les documents HTML, hérite de l'interface HTMLDocument ou depuis HTML5, est étendue pour eux.
- ▶ Document.body
 - ▶ body renvoie l'élément <body> du document en cours.
- ▶ Document.cookie
 - ▶ Renvoie une liste des cookies du document, séparés par des points virgules, ou définit un cookie.

Méthode `document.getElementById`

- ▶ La méthode `getElementById()` de `Document` renvoie un objet `Element` représentant l'élément dont la propriété `id` correspond à la chaîne de caractères spécifiée.
- ▶ Étant donné que les ID d'élément doivent être uniques, s'ils sont spécifiés, ils constituent un moyen utile d'accéder rapidement à un élément spécifique.
- ▶ Si vous avez besoin d'accéder à un élément qui n'a pas d'ID, vous pouvez utiliser `querySelector()` pour trouver l'élément en utilisant un sélecteur.

Sélecteurs

- ▶ La méthode `querySelector()` de l'interface `Document` retourne le premier `Element` dans le document correspondant au sélecteur - ou groupe de sélecteurs - spécifié(s), ou `null` si aucune correspondance n'est trouvée.

Les Sélecteurs CSS

- ▶ Les sélecteurs CSS : définition
- ▶ Pour pouvoir appliquer un style à un contenu, il va déjà falloir le cibler, c'est-à-dire trouver un moyen d'indiquer qu'on souhaite appliquer tel style à un contenu en particulier.
- ▶ Pour cela, nous allons utiliser des sélecteurs. Les sélecteurs sont l'un des éléments fondamentaux du CSS.
- ▶ De manière très schématique et très simplifiée, nous allons utiliser les sélecteurs en CSS pour cibler des contenus HTML et leur appliquer des styles.
- ▶ Il existe différents types de sélecteurs en CSS : certains sélecteurs vont s'appuyer sur le nom des éléments, comme le sélecteur CSS `p` par exemple qui va servir à cibler tous les éléments `p` d'une page.
- ▶ Ce type de sélecteurs est appelé « sélecteur d'éléments » tout simplement car ils vont être identiques aux éléments HTML sélectionnés ou encore « sélecteurs simples »

Les sélecteurs complexes

- ▶ en utilisant uniquement les sélecteurs éléments, nous allons être obligés d'appliquer les mêmes styles à tous les éléments d'un même type ce qui n'est pas très flexible.
- ▶ permettent de sélectionner un élément HTML en particulier ou un jeu d'éléments HTML en fonction de leurs attributs ou de leur état.
 - ▶ appliquer des styles à un élément uniquement lorsque la souris de l'utilisateur passe dessus par exemple.

```
p{  
  color: blue;  
  border: 2px solid orange;  
  padding: 5px;  
}
```