

# Document Object Model (DOM)

**Jean-Claude Charr**

Maître de conférences

IUT de Belfort – Montbéliard

Université de Franche Comté



# Description générale

- Définit un standard pour accéder aux documents structurés comme HTML ou XML et modifier leur contenu
- C'est une spécification indépendante du langage de programmation
- Définit les propriétés des éléments et les méthodes pour accéder aux éléments

# HTML DOM

- Tous les éléments d'un document HTML sont des noeuds :
  - Tout le document est un "document node"
  - Tout élément html est un "element node"
  - Le texte dans les éléments HTML sont des "text nodes"
  - Les attributs HTML sont des "attribute nodes"

Ex: `<html>`

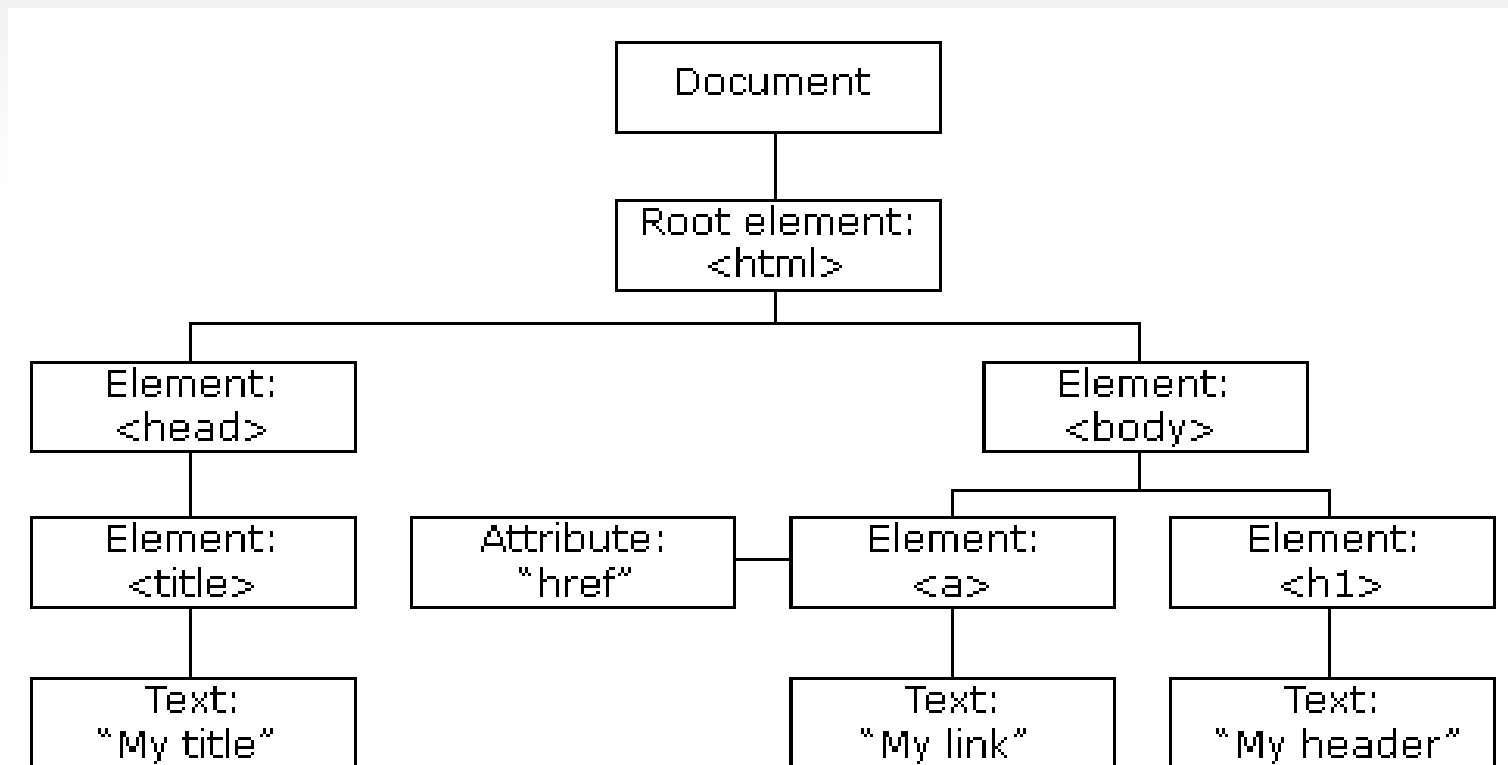
`<head> <title>DOM Tutorial</title> </head>`

`<body> <h1>DOM Lesson one</h1> </body>`

`</html>`

# La structure d'un document HTML en arbre (1/3)

- Un document html peut être structuré en un arbre de noeuds



# La structure d'un document HTML en arbre (2/3)

- En code HTML :

```
<html>
```

```
  <head>
```

```
    <title>My title</title>
```

```
  </head>
```

```
  <body>
```

```
    <h1>My header</h1>
```

```
    <a href="google.com">My link</a>
```

```
  </body>
```

```
</html>
```

# La structure d'un document HTML en arbre (3/3)

- Les noeuds dans l'arbre de noeuds ont une structure hiérarchique :
  - Le premier noeud de l'arbre est appelé la racine
  - Tout noeud, sauf la racine, a un seul noeud père
  - Un noeud peut avoir plusieurs fils
  - Une feuille est un noeud sans fils
  - Les frères sont des noeuds ayant le même père

# Les propriétés des noeuds (1/3)

- Tout élément HTML DOM possède des propriétés auquel on peut accéder et modifier avec un langage de programmation

comme JavaScript, exemples :

- `x.innerHTML` – la valeur du texte dans le noeud `x`
- `x.nodeName` – le nom du noeud `x`
- `x.nodeValue` – la valeur du noeud `x`
- `x.parentNode` – le noeud père du noeud `x`
- `x.childNodes` – les noeuds fils du noeud `x`
- `x.attributes` – les attributs du noeud `x`

# Les propriétés des noeuds (2/3)

- nodeName :
  - N'est pas éditable
  - Pour un élément c'est le mot clé
  - Pour un attribut c'est le nom de l'attribut
  - Pour un noeud texte c'est #text
  - Pour un noeud document c'est #document
- nodeValue :
  - Pour un élément c'est indéfini
  - Pour un attribut c'est la valeur de l'attribut
  - Pour un noeud texte c'est le texte entre les 2 balises



# Les propriétés des noeuds (3/3)

- NodeType :
  - 1 pour les éléments
  - 2 pour les attributs
  - 3 pour les noeuds de texte
  - 8 pour les commentaires
  - 9 pour le document

# Méthodes pour sélectionner des noeuds

- Tout élément HTML DOM possède des méthodes qui permettent de le sélectionner, le modifier ou ajouter un nouveau élément, exemples :
  - `document.getElementById(id)` : retourne l'élément ayant comme identifiant "id"
  - `document.getElementsByTagName(name)` : retourne les éléments ayant le nom clé "name"
  - `document.getElementsByClassName("intro")` : retourne les éléments appartenant à la classe "intro"
  - `document.querySelectorAll("p.intro")` : retourne les éléments de type paragraphe et qui appartiennent à la classe intro

# Exemple sur les propriétés et les méthodes

```
<html>
  <body>
    <p id="intro">Hello World!</p>
    <script type="text/javascript">
      txt=document.getElementById("intro").innerHTML;
      document.write("<p>Du paragraphe intro : " + txt + "</p>");
      txt=document.getElementById("intro").childNodes[0].nodeValue;
      document.write("<p>Du paragraphe intro : " + txt + "</p>")
    </script>
  </body>
</html>
```

# Accès aux noeuds

- `document.getElementById("intro");`
- `x=document.getElementsByTagName("p");`  
`x.length`; taille du tableau  
`x[1]`; deuxième élément du tableau
- `parentNode`, `firstChild`, and `lastChild`  
Ex : `document.getElementById("intro").firstChild;`
- `document.documentElement`; root
- `document.body`; body

# Modifier un noeud

- `document.getElementById("p1").innerHTML="Newtext!";`
- `document.getElementById("p1").style.fontFamily="Arial";`
- `<input type="button" onclick =  
"document.body.style.backgroundColor ='lavender';"  
value="color" />`

# Accéder aux attributs et les modifier

Pour accéder :

```
node.getAttributeNode("nomAttribut").nodeValue
```

ou

```
node.getAttribute("nomAttribut")
```

Pour modifier :

```
node.setAttribute("nomAttribut", "valeur");
```

ou

```
attribut=node.getAttributeNode("nomAttribut");
```

```
attribut.nodeValue="valeur";
```

# Collections

- C'est un groupe de noeuds similaires.
- L'objet **Document** a les propriétés qui contiennent les collections d'images, liens, forms, ... de la page HTML.
- Exemple :

```
var linklist = document.links;  
for ( var i = 0; i <linklist.length ; i++ )  
    alert(linklist[i].innerHTML);
```

# Supprimer des noeuds

- Quand un noeud est supprimé tous ses fils sont supprimés
- Supprimer un noeud fils :  
`node=document.getElementById("idOfNodeToDelete");`  
`node.parentNode.removeChild(node);`
- Supprimer un noeud attribut  
`node.removeAttribute("nomAttribut");`



# Créer des noeuds

- Créer un noeud :

```
newNode=document.createElement("nomNoeud");
```

- Créer un noeud attribut :

```
newAtt=document.createAttribute("nomAttribut");
```

```
newAtt.nodeValue="valeurAttribut";
```

- Créer un noeud texte :

```
newText=document.createTextNode("texte");
```

- Créer un commentaire :

```
newComment=document.createComment("commentaire");
```

# Ajouter des noeuds au document HTML

- Ajouter un noeud fils (élément, texte ou commentaire):  
`node.appendChild(newElement);`
- Ajouter un élément avant un autre élément :  
`node=document.getElementById("idOfNodeToInsertBefore");`  
`document.insertBefore(newNode,node);`
- Ajouter un attribut :  
`node.setAttribute("nomAttribut","valeur");`  
`node.setAttributeNode(newAttribut);`
- Ajouter du texte à un text node : `node.insertData(0,"texte ");`

# Remplacer des noeuds dans le document HTML

- Exemple :

```
newNode1=document.createElement("nomElement1");
```

- newNode2=document.createElement("nomElement2");

```
newText=document.createTextNode("texte");
```

```
newNode2.appendChild(newText);
```

```
newNode1.appendChild(newNode2);
```

```
node=document.getElementById("idOfNodeToReplace");
```

```
node.parentNode.replaceChild(newNode1,node);
```

# Copier un noeud

```
oldNode=document.getElementById('nodeToCopy');
```

```
newNode=oldNode.cloneNode(true);
```

```
Document.getElementById('idOfParentNodeOfNewNode').  
appendChild(newNode);
```