$\begin{array}{c} {\rm Architecture~et~programmation~du~web} \\ {\rm TP~n.1} \\ {\rm Introspection~d'une~page~web~avec~l'API~DOM} \end{array}$

Pierre Pompidor

A un document HTML quelconque vous devez pouvoir lier un code JavaScript qui affichera le contenu de son arbre DOM.

Par exemple, si au code HTML suivant :

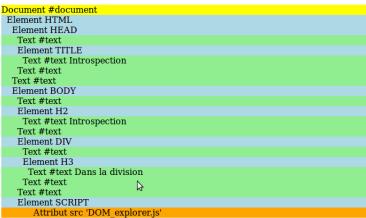
```
<html>
<head> </head>
<body>
<h2> Introspection </h2>
<div>
<h3> Dans la division </h3>
</div>
</body>
</html>
```

nous attachons le code JavaScript $DOM_{explorer.js}$ en ajoutant la balise <script> suivante avant la balise </html> fermante (pourquoi à cet endroit?) :

```
<script src="DOM_explorer.js"></script>
```

vous devez faire apparaître le contenu de l'arbre DOM de la page HTML à la suite de l'affichage des éléments de celle-ci :





Pour ce faire vous devez dans $DOM_{explorer.js}$ créer une fonction récursive qui va parcourir tous les noeuds de l'arbre DOM, dont voici le squelette de code (les ... désignent les parties à compléter mais évidemment, il y a aussi du code à écrire en dehors de cette fonction) :

```
function analyseDOM(noeud, prof) {
   if (noeud == null) return;
   let type = 0;
   switch (noeud.nodeType) {
      case 1 : type = 'Element'; couleur = 'lightblue';
               break;
   }
   if (type) {
      visualiserLeNoeud(couleur, ...);
   }
   if (noeud.attributes) {
     for (let num=0; num < noeud.attributes.length; num++) {</pre>
        visualiserLeNoeud('orange', ...);
     }
  }
   for (let num=0; num < noeud.childNodes.length; num++) {</pre>
       analyseDOM(noeud.childNodes.item(num), prof+1);
   }
}
```

Vous devez trouver également un moyen pour que votre fonction d'exploration récursive ne rencontre pas les nouveaux éléments que vous allez créer dans le DOM...