dom-annexe.zip sur site

1 Problématique

Le *DOM (Document Object Model)* est une interface de programmation qui permet à des scripts (Javascript mais pas seulement) d'examiner et de modifier le contenu du navigateur web. Le DOM est normalisé par la *W3C (World Wide Web Consortium)* et est incorporé dans la norme HTML5.

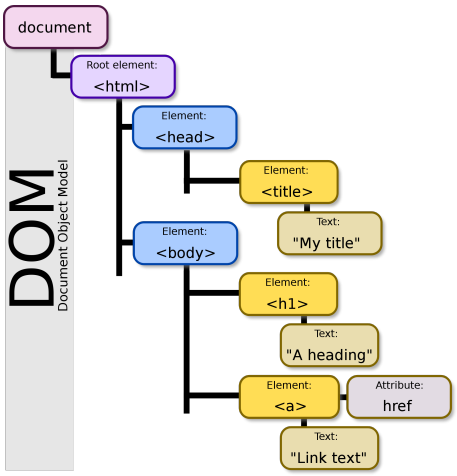

FIGURE 1 – Document Object Model source

Comment implémenter la structure du DOM en Python ?

2 Structure arborescente

2.1 Élément HTML

Un élément HTML est constitué d'une paire de *balises* (ouvrante et fermante), d'un *contenu*. De plus la balise peut contenir des attributs.

```
1 <a href="https://cviroulaud.github.io">Un super site</a>
```

Code 1 – Un lien hypertexte

Nous pouvons représenter la balise *a* sous forme d'une structure arborescente (figure 2) et par extension représenter l'ensemble d'une page web.

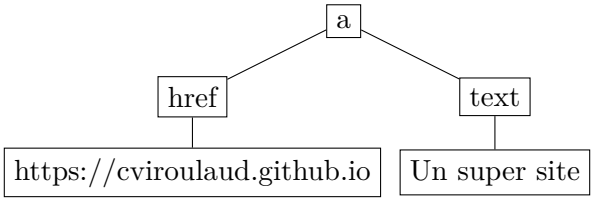

FIGURE 2 – Structure arborescente d'un lien hypertexte

Christophe Viroulaud

2.2 Une page web

Une page web est une imbrication de balises HTML. La structure arborescente du DOM est évidente.

```
<html>
    <head>
        <meta charset="utf-8">
        <title>Présentation NSI seconde</title>
    </head>
    <body>
        <p>La NSI est trop cool!! Le programme est très riche:</p>
        <ul>
            <li>Programmation</li>
            <li>Algorithmie</li>
            <li>Architecture machine</li>
            <li>Langages du web</li>
        </ul>
        <a href="https://cviroulaud.github.io">Les cours de NSI</a>
        <img src="images/nsi.png">
    </body>
</html>
```

Code 2 – Une page web

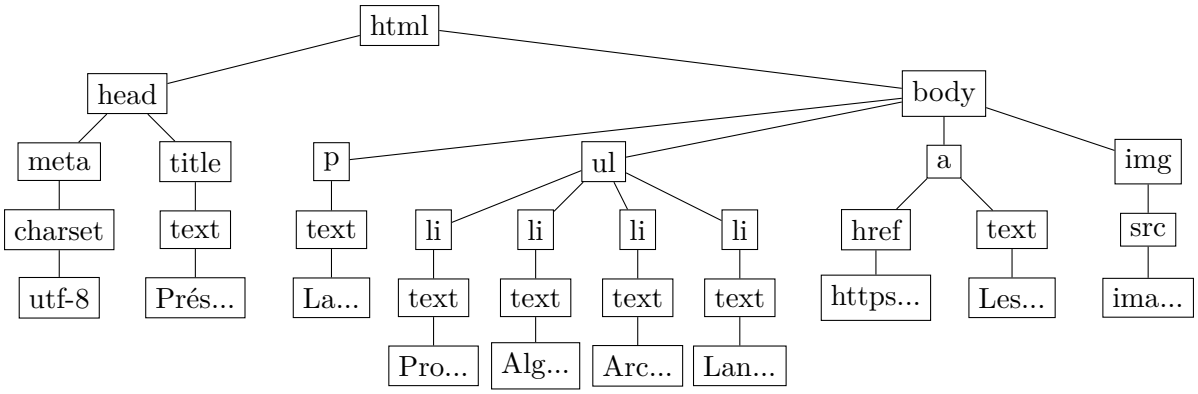

FIGURE 3 – Structure arborescente d'une page web

3 Implémentation en Python

3.1 Un nœud

Pour représenter une structure arborescente il est commun d'utiliser la programmation orientée objet.

```
class Noeud:
    def __init__(self, v: str, f: list)->None:
        self.valeur = v
        self.fils = f
```

Code 3 – Nœud d'une arborescence

Christophe Viroulaud

Dans le code 3 l'attribut *valeur* représente le nom de la balise, l'attribut, le contenu. L'attribut *fils* est la liste des nœuds enfants.
Nous pouvons alors construire une représentation du DOM.

```
dom = Noeud("html", [Noeud("head", []),
                     Noeud("body", []) ])
```

Code 4 – Représentations des premières balises du DOM

3.2 Manipulation du DOM

Le *JavaScript* permet d'interagir avec une page web en manipulant le *DOM*. La méthode *Element.getElementsByTagName()* retourne une liste des éléments portant le nom de balise donné. Le DOM étant ici implémenté en Python, il est possible d'imiter le comportement de cette méthode.

Activité 1 :

1. Télécharger et décompresser *dom-annexe.zip* .
2. Écrire la fonction *récursive* **taille(dom : Noeud)** → **int** qui renvoie le nombre d'éléments du DOM.
3. Écrire la fonction *récursive*

   get_elements_by_tagname(arbre : Noeud, tag : str, res : list) → **list**

   qui renvoie la liste *res* des nœuds fils de *tag*.

Christophe Viroulaud