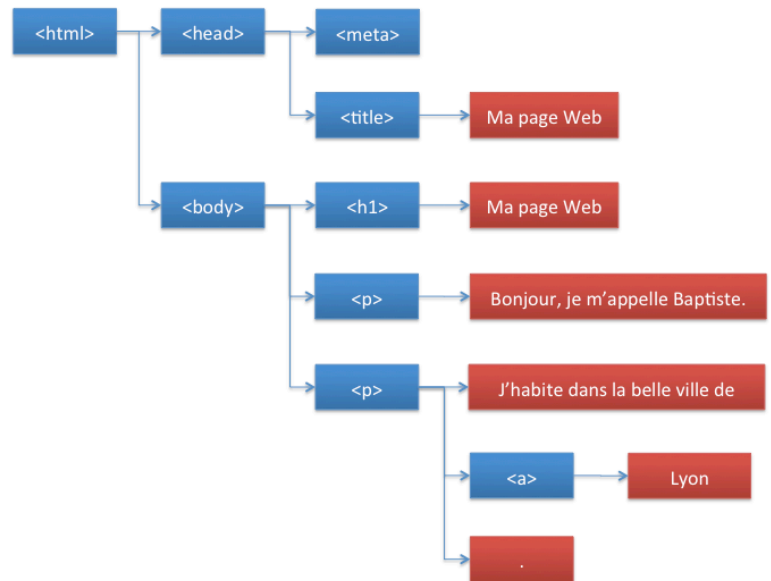


4) Interactions grâce à JavaScript (pour aller plus loin)

Pour afficher un résultat dans un navigateur, ce dernier parcourt le code HTML d'une page afin de construire une représentation de sa structure. Ensuite, il utilise cette représentation pour afficher les différents éléments de la page.

Cette représentation de la structure d'une page web offerte par un navigateur est exploitable via JavaScript : elle est appelée **DOM**, pour **Document Object Model**. Le DOM définit la structure d'une page et le moyen d'interagir avec elle : il s'agit d'une interface de programmation, ou API (Application Programming Interface).

De manière très schématique, le DOM est l'ensemble des balises imbriquées les unes dans les autres de votre site web. On peut la représenter sous une forme d'arborescence qui ressemble à quelque chose comme cela :



En accédant aux éléments noeuds de cette arborescence (pas les feuilles terminales de l'arbre), il est possible de faire des choses bien plus complexes que l'affichage d'un simple pop-up avec JavaScript.

Par exemple, il est possible de modifier le style d'une balise, de modifier la classe (CSS) d'une balise ou encore de modifier le contenu d'une balise.



— À faire vous-même 6 —

Dans le code HTML de l'exercice 3, ajoutez un autre paragraphe :

```
<p id="para2">J'aime le rouge.</p>
```

Modifiez le code JavaScript comme suit :

```
function maFonction() {  
    document.querySelector("#para2").style.color="red";  
}
```

Testez votre page HTML.

En sachant que le style CSS pour changer la couleur s'appelle "color", décrivez le fonctionnement de la ligne de code de la fonction maFonction() :

Modifier le fichier JS afin d'afficher aussi une bordure avec des tirets après le clic.

Dans cet exemple, on modifie le style directement de l'id #para2 directement dans le fichier HTML (**cela n'est pas interdit mais fortement déconseillé!**). Il est possible de travailler plus "proprement" en utilisant les classes CSS :



— À faire vous-même 7 —

Dans le code HTML de l'exercice 3, ajoutez un autre paragraphe et deux boutons :

```
<p id="para3">J'aime encore plus le CSS.</p>
<button onclick="foncRouge()">Rouge</button>
<button>Noir</button>
```

Modifiez le code JavaScript comme suit :

```
function foncRouge() {
    document.querySelector("#para3").classList.remove("vert");
    document.querySelector("#para3").classList.add("rouge");
}
```

Ajoutez au CSS les lignes ci-dessous :

```
.rouge {
    color:red;
    font-size:20px;
}
```

Testez votre page HTML. Que remarquez-vous ?

Explication : La fonction JavaScript "foncVert()" permet de modifier la classe CSS de la balise ayant pour id "para3".

Dans un premier temps, la ligne

```
document.querySelector("#para3").classList.remove("vert")
```

permet de supprimer l'**association** entre la balise d'id "para3" et la classe CSS "vert" (si cette association n'existe pas, cette ligne n'a aucun effet). Dans un deuxième temps, on associe la classe CSS "rouge" avec la balise d'id "para3" avec la ligne

```
document.querySelector("#monPara").classList.add("rouge");
```



— À faire vous-même 8 —

Modifiez votre code HTML, JS et CSS afin de formater en noir et en caractère taille 20 l'id "para3" suite à un click sur le bouton Noir.

Testez votre page HTML.

Il est également possible de modifier le contenu d'une balise HTML !



— À faire vous-même 9 —

Dans le code HTML de l'exercice 3, ajoutez un nouveau paragraphe et un bouton :

```
<p id="para4">J'adore cliquer.</p>
```

```
<button onclick="modifMessage()"> Click! </button>
```

Modifiez le code JavaScript comme suit :

```
var nombreClick=0;
```

```
function modifMessage() {
```

```
    document.querySelector("#para4").innerHTML = "Bravo, vous avez cliqué x  
fois sur le bouton !";
```

```
}
```

Testez votre page HTML.

Que fait la variable globale nombreClick ?

On veut que mon message change en fonction du nombre de clicks et affiche :

"Bravo, vous avez cliqué 1 fois sur le bouton !"

"Bravo, vous avez cliqué 2 fois sur le bouton !"

"Bravo, vous avez cliqué 3 fois sur le bouton !"

etc.

En transposant vos connaissances (concaténation...) en Python au Javascript, faites les modifications nécessaires dans le fichier JS.

Bien sur, il existe énormément d'autres événements HTML que "onclick". Il est possible de détecter le "survol" par le curseur de la souris d'un élément HTML.



— À faire vous-même 10 —

Dans le code HTML de l'exercice 3, rajoutez un dernier paragraphe :

```
<div onmouseover="foncEntre()" onmouseout="foncQuitte()"
id="maDiv"><p>Survolez-moi</p></div>
```

Modifiez le code JavaScript comme suit :

```
function foncEntre(){
    document.querySelector("#maDiv").classList.remove("blanc");
    document.querySelector("#maDiv").classList.add("rouge");
}
function foncQuitte() {
    document.querySelector("#maDiv").classList.remove("rouge");
    document.querySelector("#maDiv").classList.add("blanc");
}
```

Ajoutez les lignes suivantes au code CSS :

```
#maDiv{
    width : 200px;
    height : 100px;
    margin : 0 auto;
    border : 2px solid black;
}
.rouge {
    background-color:red;
}
.blanc {
    background-color : white;
}
```

Après avoir analysé le code ci-dessus, testez cette nouvelle page en cliquant sur le fichier index.html

Conclusion : le code HTML permet de générer des éléments graphiques qui seront affichés par un navigateur web, mais pas seulement : il est aussi possible de mettre en place dans le code HTML des **événements**.

Un événement donné pourra déclencher l'exécution d'instructions JavaScript : onclick, onmouseover, onkeydown, addEventListener sont de ceux-là.

Exo 4 : approfondissement : recherchez comment ajouter une applet Géogebra dans une page web à l'aide de Javascript. Le but est de faire apparaître une telle applet à la suite d'un clic sur un bouton (voir imprim' écran dans l'énoncé dans l'exercice.)