

DOM (DOCUMENT OBJECT MODEL)

OUALI MOHAMED

JS



STRUCTURE MINIMALE D'UNE PAGE EN HTML5

- Doctype : **<!DOCTYPE html>**
 - Pour indiquer le type du document
- Éléments :
 - **html** : va contenir toute la page web
 - **head** : va contenir des informations sur la page
 - **Title** : le titre de la page
 - **meta** : pour les meta données (charset)
 - **body** : tout le contenu html de la page

<!doctype html>

<html>

<head>

<meta charset="utf-8">

<title>Document sans nom</title>

</head>

<body>

</body>

</html>

html

<html>

Entête

<head>

Corps

<body>

titre

<title>

meta

<meta />

```
<!doctype html>
```

```
<html>
```

```
  <head>
```

```
    <meta charset="utf-8">
```

```
    <title>Document sans nom</title>
```

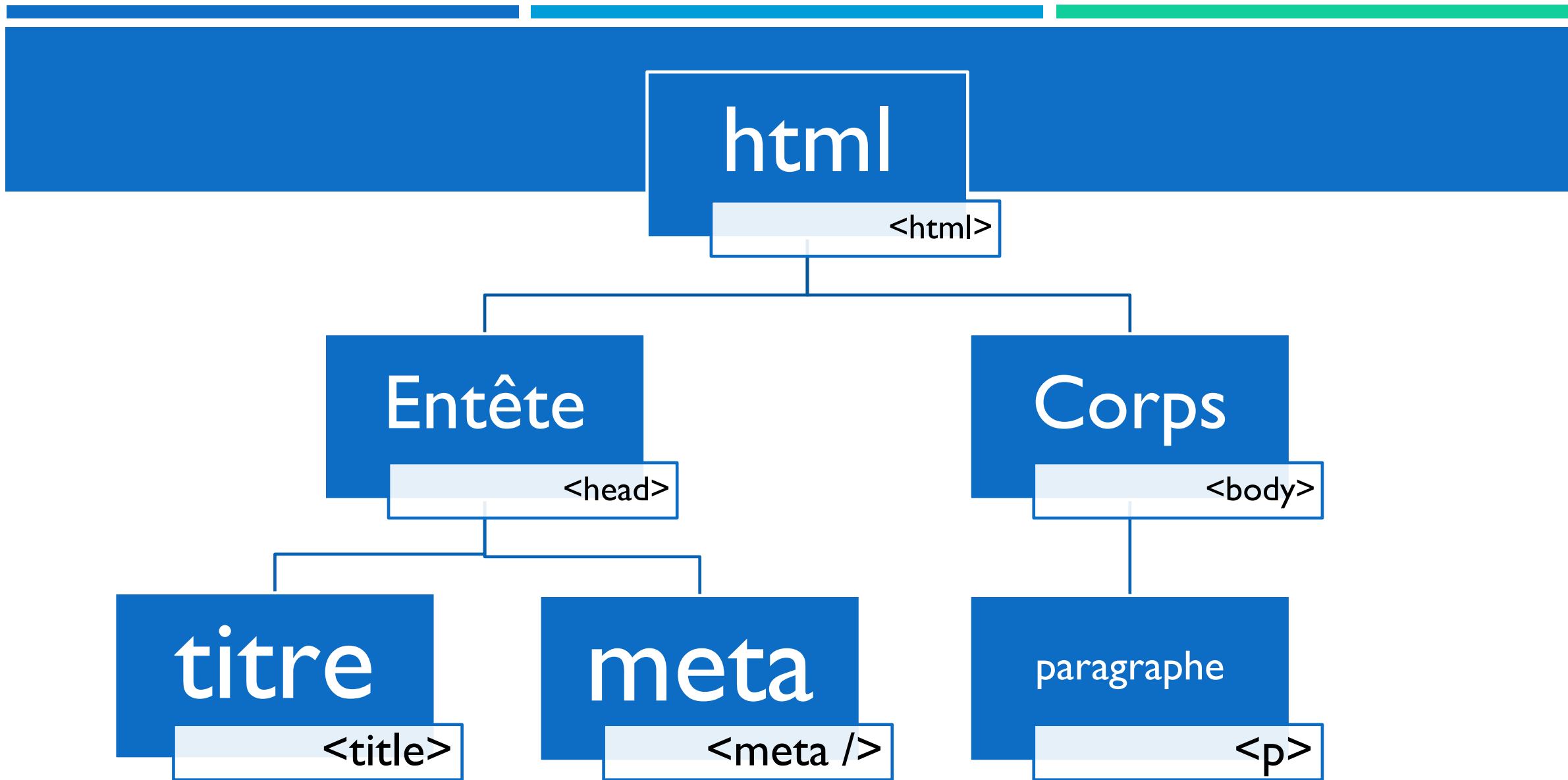
```
  </head>
```

```
  <body>
```

```
    <p>un paragraphe</p>
```

```
  </body>
```

```
</html>
```



object [HTMLHtmlElement]

<html>

object HTMLHeadElement

<head>

object HTMLTitleElement

<title>

object HTMLMetaElement

<meta />

object HTMLBodyElement

<body>

object HTMLParagraphElement

<p>

OBJECT HTMLPARAGRAPHELEMENT

□ **Attributs :**

- childNodes, children, className, currentStyle, dir, document, filters, id, innerHTML, innerText, lang, offsetHeight, offsetLeft, offsetParent, offsetTop, offsetWidth, style, title, uniqueID, tagName ...

□ **Méthodes :**

- getAttribute, getElementsByTagName, removeAttribute, setAttribute, appendChild, cloneNode, hasChildNodes, insertBefore, removeChild, removeNode, replaceChild

DÉFINITION

- Le **Document Object Model** (ou **DOM**) est un standard du W3C
 - qui décrit une interface de programmation ou API (Application Programming Interface) permettant à des programmes informatiques et à des scripts d'accéder ou de mettre à jour le contenu, la structure ou le style de documents HTML.
 - Le document peut ensuite être traité et les résultats de ces traitements peuvent être ré-incorporés dans le document tel qu'il sera présenté.

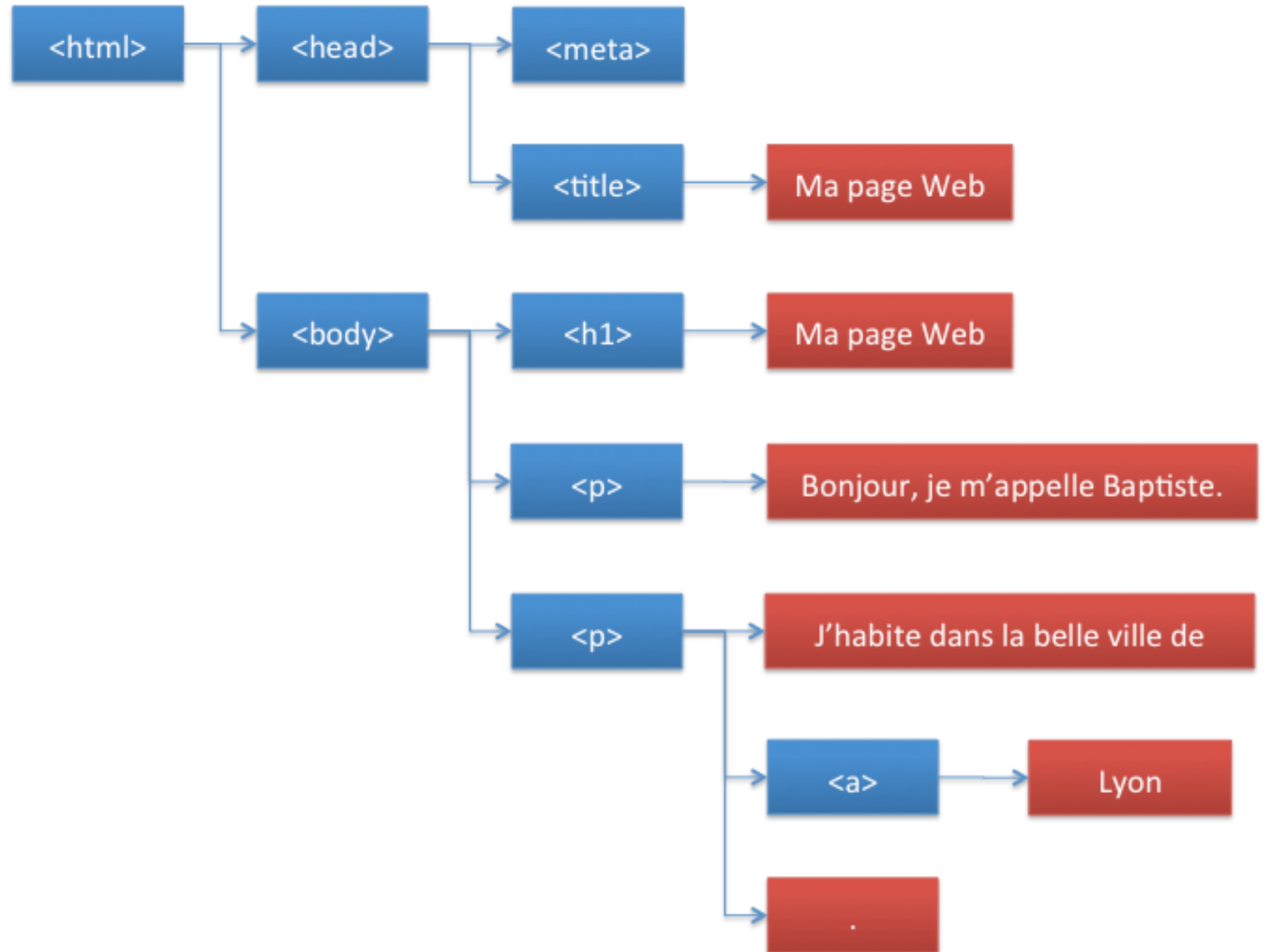
LES NŒUDS

- Conformément à la norme **W3C HTML DOM**, le tout dans un document HTML est un nœud :
- L'ensemble du document est un nœud de document
- Chaque élément HTML est un nœud d'élément
- Le texte dans des éléments HTML sont des nœuds de texte
- Chaque attribut HTML est un nœud d'attribut
- Les commentaires sont les nœuds de commentaires

ARBORESCENCE

- Le DOM HTML considère que les documents HTML sont des structures arborescentes.
- La structure est appelée arborescence de nœuds

ARBORESCENCE

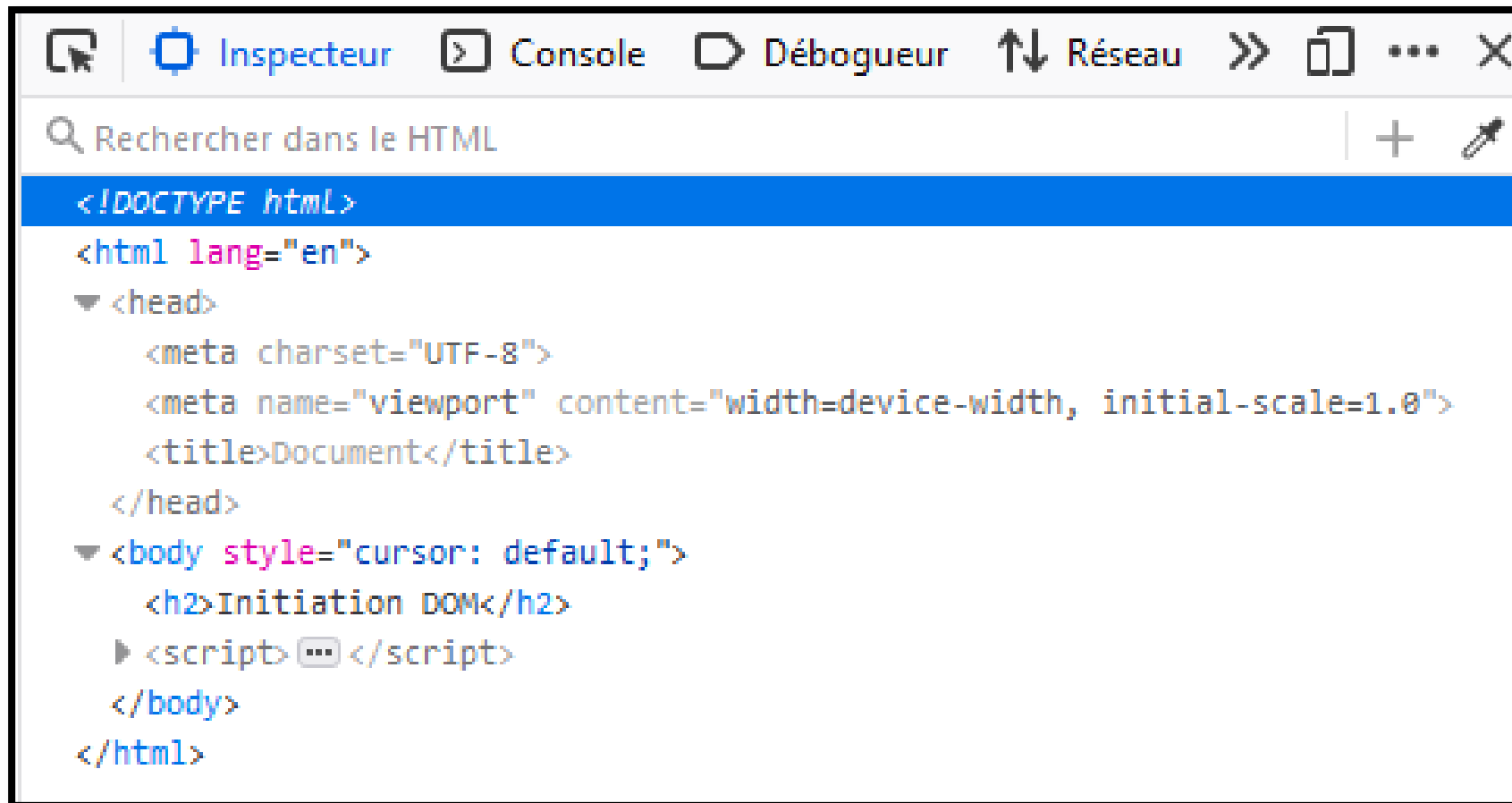


ARBORESCENCE

- On distingue deux types de nœuds :
 - Ceux (en bleu) qui correspondent à des éléments **HTML**, comme `<body>` ou `<p>`. Ces nœuds peuvent avoir des sous-nœuds, appelés fils ou enfants (children).
 - Ceux (en rouge) qui correspondent au **contenu textuel** de la page. Ces nœuds ne peuvent pas avoir de fils.

ARBORESCENCE

- Le navigateur web permet de visualiser la hiérarchie des éléments d'une page web.



LES PARENTS NŒUD, LES ENFANTS, FRÈRES ET SŒURS



- Les nœuds dans l'arborescence de nœuds ont une relation hiérarchique entre eux.
 - Les termes **parent**, **enfant**, **frère** ou **sœur** sont utilisés pour décrire les relations.
- Les enfants d'un même niveau sont appelés frères et sœurs.
 - Dans une arborescence de nœuds, le nœud supérieur est appelé la racine
 - Chaque nœud a exactement un parent, à l'exception de la racine (qui n'a pas de parent)
 - Un nœud peut avoir n'importe quel nombre d'enfants
 - Les frères et sœurs sont des nœuds ayant le même parent

LES PARENTS NŒUD, LES ENFANTS, FRÈRES ET SŒURS

- Regardez le fragment HTML suivant :

```
<html>
  <head>
    <title> Initiation DOM </title>
  </head>
  <body>
    <h1>DOM Leçon une</h1>
    <p> Bonjour tout le monde!</p>
  </body>
</html>
```

- Le nœud **<html>** n'a pas de nœud parent, c'est le nœud racine
- Le nœud parent des nœuds **<head>** et **<body>** est le nœud **<html>**
- Le nœud parent de "**Bonjour tout le monde!**" nœud de texte est le nœud **<p>**

- 
- 
- ▶ Le nœud `<html>` a deux nœuds enfants: `<head>` et `<body>`
 - ▶ Le nœud `<head>` a un nœud enfant: le nœud `<title>`
 - ▶ Le nœud `<title>` dispose également d'un nœud enfant: le nœud de texte "DOM Tutoriel"
 - ▶ Le `<h1>` et nœuds `<p>` sont frères et sœurs et des nœuds enfants de `<body>`

et:

- ▶ L'élément `<head>` est le premier enfant de l'élément `<html>`
- ▶ L'élément `<body>` est le dernier enfant de l'élément `<html>`
- ▶ L'élément `<h1>` est le premier enfant de l'élément `<body>`
- ▶ L'élément `<p>` est le dernier enfant de l'élément `<body>`

MÉTHODES ET PROPRIÉTÉS

- Le DOM HTML peut être consulté avec JavaScript (et autres langages de programmation).
- **getElementById (id)** - obtenir le nœud (élément) avec un id spécifié
- **getElementsByTagName (balise)** - Retourne une liste de nœuds (collection / tableau de nœuds) contenant tous les éléments avec un nom de tag spécifié.
- **getElementsByTagName (class)** - Retourne une liste de nœuds contenant tous les éléments d'une classe spécifiée.
- **appendChild (nœud)** – Permet d'insérer un nouveau nœud enfant (élément)
- **removeChild (nœud)** – Permet de supprimer un nœud enfant (élément)

MÉTHODES ET PROPRIÉTÉS

- **innerHTML** - la valeur du texte d'un nœud (élément)
- **parentNode** - le nœud parent d'un nœud (élément)
- **childNodes** - les nœuds enfants d'un nœud (élément)
- **nodeType** – Le type du nœud
- ...

ACCÉDER AU DOM AVEC LA VARIABLE DOCUMENT

- Lorsqu'un programme JavaScript s'exécute dans le contexte d'un navigateur web, il peut accéder à la racine du DOM en utilisant la variable ***document***.

La variable ***document*** correspond à l'élément **<html>**.

- Cette variable est un objet et dispose des propriétés ***head*** et ***body*** qui permettent d'accéder respectivement aux éléments **<head>** et **<body>** de la page.

```
var h = document.head; // La variable h contient l'objet head du DOM
console.log(h);
var b = document.body; // La variable b contient l'objet body du DOM
console.log(b);
```

DÉCOUVRIR LE TYPE D'UN NŒUD

- Chaque objet du DOM a une propriété ***nodeType*** qui indique son type.
- La valeur de cette propriété est *document.ELEMENT_NODE* pour un nœud "élément" (balise HTML) et *document.TEXT_NODE* pour un nœud textuel.

```
if (document.body.nodeType === document.ELEMENT_NODE) {  
    console.log("Body est un nœud élément");  
} else {  
    console.log("Body est un nœud textuel");  
}
```

<i>Element type</i>	<i>NodeType</i>
<i>Element</i>	<i>1</i>
<i>Attribute</i>	<i>2</i>
<i>Text</i>	<i>3</i>
<i>Comment</i>	<i>8</i>
<i>Document</i>	<i>9</i>

ACCÉDER AUX ENFANTS D'UN NŒUD ÉLÉMENT

- Chaque objet du DOM de type élément possède une propriété ***childNodes***.
- Il s'agit d'une collection ordonnée regroupant tous ses nœuds enfants sous la forme d'objets DOM.
- On peut utiliser cette collection un peu comme un tableau pour accéder aux différents enfants d'un nœud.

ACCÉDER AUX ENFANTS D'UN NŒUD ÉLÉMENT

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Document</title>
7  </head>
8  <body>
9      <h2>Initiation DOM</h2>
10     <script>
11
12     </script>
13 </body>
14 </html>
```

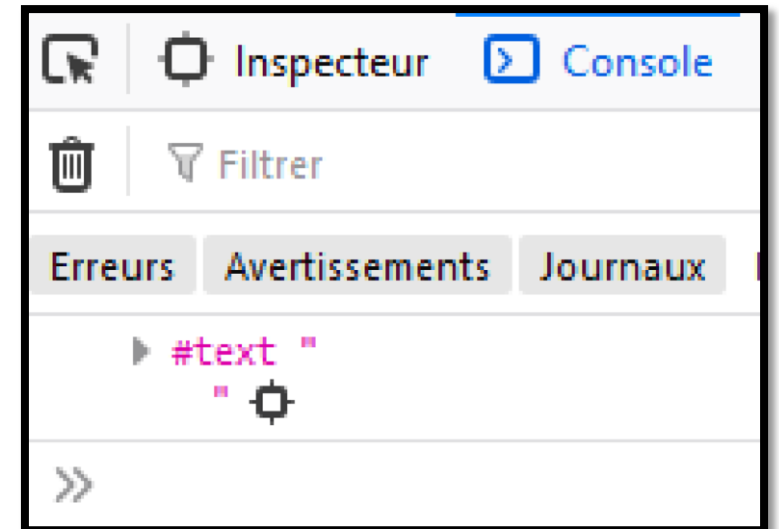
ACCÉDER AUX ENFANTS D'UN NŒUD ÉLÉMENT

```
// Accès au premier enfant du nœud body  
console.log(document.body.childNodes[0]);
```

Comme vous pouvez le constater, le tout premier enfant du nœud **body** n'est pas le nœud **h1** mais un nœud textuel !

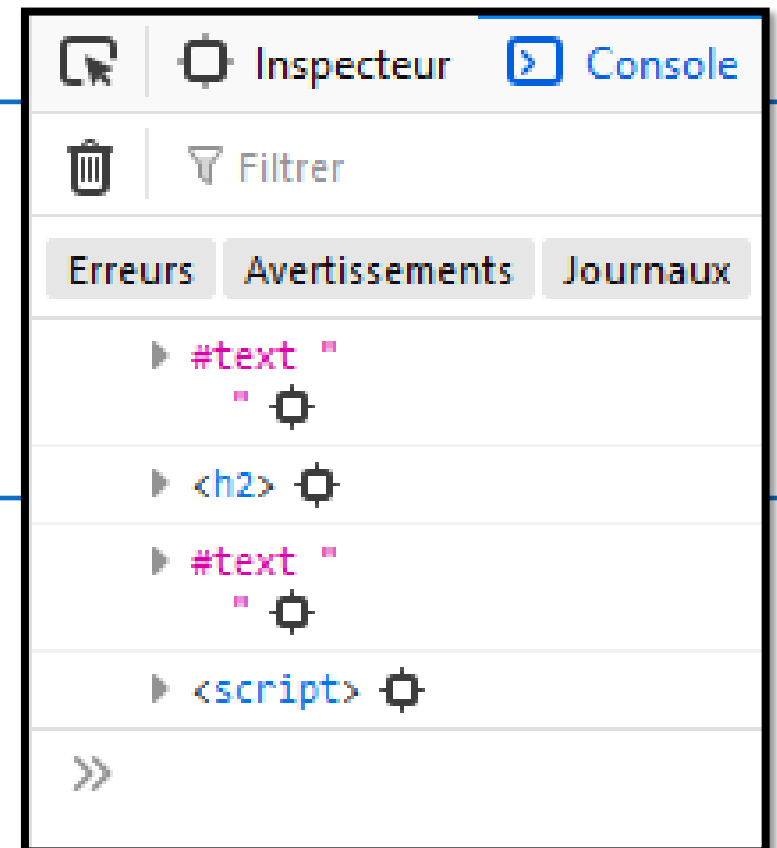
Les espaces entre les balises ainsi que les retours à la ligne dans le code HTML sont considérés par le navigateur comme des nœuds textuels.

Ici, le nœud **h1** n'est donc que le deuxième enfant du nœud **body**.



PARCOURIR LA LISTE DES NŒUDS ENFANTS

```
// Affiche les nœuds enfant du nœud body
for (let i = 0; i < document.body.childNodes.length; i++) {
  console.log(document.body.childNodes[i]);
}
```



ACCÉDER AU PARENT D'UN NŒUD

- Chaque objet du DOM possède une propriété ***parentNode*** qui renvoie son nœud parent sous la forme d'un objet DOM.

```
var h1 = document.body.childNodes[1];  
console.log(h1.parentNode); // Affiche le nœud body  
console.log(document.parentNode); // Affiche null : document n'a aucun nœud parent
```