

# Javascript

## Document object model

# Document Object Model

- DOM est un standard permettant de manipuler et consulter un fichier XML.
- Accessible de JavaScript.
  - Possible de modifier l'arbre « à tout moment »

# Ce que permet de DOM

- Naviguer dans le document
- Aller directement sur un nœud
  - balise, texte...
- Changer la valeur d'un nœud
  - balise, attribut, texte...
- Créer ou retirer un nœud
  - balise, attribut...
- Modifier le CSS
  - style d'un élément,
  - entête CSS,
  - CSS actif...

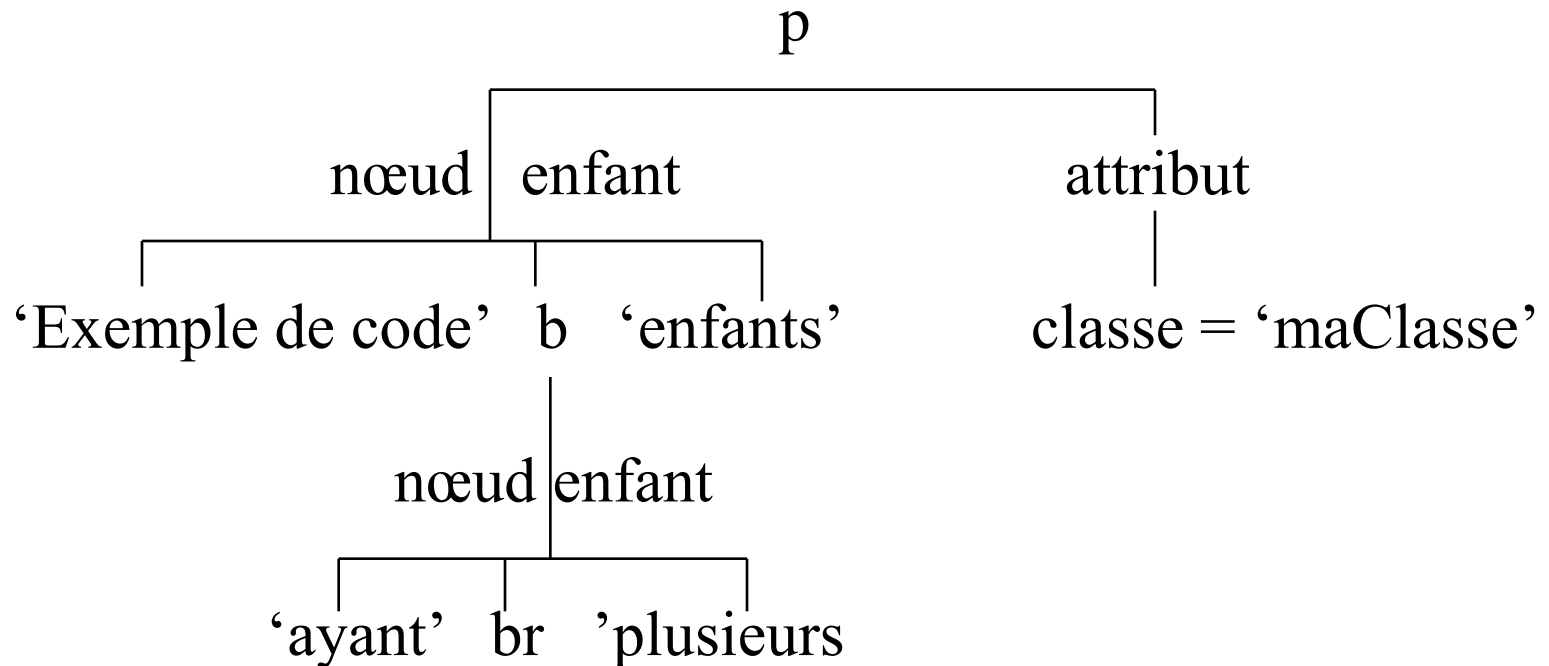
# Arbre avec ses noeuds

- Soit le code suivant:

```
<p class='maClasse'>
```

```
    Exemple de code <b>ayant <br />plusieurs</b>enfants
```

```
</p>
```



# Accéder aux éléments de l'arbre

- En utilisant les fonctions :
  - parent, firstChild, lastChild, ...
    - Pour accéder aux balises
      - `x = document.body.firstChild`
      - `x = document.body.childNodes[0]`
      - `x.parentNode`
      - `document.getElementById('id').firstChild`

# Accès direct au noeud

- `getElementsByTagName()`
  - Permet d'accéder à un ou des nœuds avec leur nom de balise
  - ex.: `x = document.getElementsByTagName('p')`
    - x sera une variable dimensionnée
      - `x.length`
- `getElementById()`
  - mettre un `id="identifiantUnique"` à la balise `<p>`
    - `<p id="unik">`
    - `x = document.getElementById('unik')`

# Modifier le texte d'un noeud

- Pour modifier le texte 'Exemple de code' par 'magie!'
  - `x= document.getElementById('unik');`
  - `x.firstChild.nodeValue = 'magie!';`

`<p id="unik">`

Exemple de code `<b>`ayant `<br/>`

plusieurs`</b>`enfants

`</p>`

# Modifier complètement le code HTML d'un noeud

- Pour modifier le HTML du paragraphe par 'magie!'
  - `x= document.getElementById('unik');`
  - `x.firstChild.innerHTML = 'magie!';`

`<p id="unik">`

Exemple de code `<b>`ayant `<br/>`

plusieurs`</b>`enfants

`</p>`



# Ajouter / modifier un attribut

- `setAttribute`
  - `x = document.getElementsByTagName('img')[0]`
  - `x.setAttribute('alt', 'J. Rabbit')`
    - Écrase l'ancienne valeur si l'attribut existait

# Accéder au texte d'un champ de saisie

- Soit le code:

```
<form ...>
```

```
<input id="unik" type="text" name="nom" value="" />
```

```
</form>
```

- Il est possible d'accéder au champ texte par:

```
var val = document.getElementById('unik').value;
```

# Création de nœuds

- Ici nous voulons ajouter le nœud `<i>` avant `<b>`

## Avant:

```
<p id="un">
```

Exemple de code

```
<b> ayant <br /> plusieurs </b>
```

enfants

```
</p>
```

## Après:

```
<p id="un">
```

Exemple de code

```
<i>
```

```
<b>ayant <br /> plusieurs</b>
```

```
</i>
```

enfants

```
</p>
```

# Création de nœuds – les étapes

- Créer un nœud `<i>`
  - `elementi = document.createElement('i')`
- Repérer le nœud `<b>`
  - `elementb = document.getElementsByTagName('b')[0]`
- Cloner le nœud `<b>` et ses enfants
  - `noeudb = elementb.cloneNode(true)`
- Ajouter au nœud `<i>` le nœud `<b>` et ses enfants
  - `elementi.appendChild(noeudb)`
- Remplacer le nœud `<b>` original par le nœud `<i>` modifié  
`document.getElementsByTagName('p')[0].replaceChild(elementi, elementb)`

# Supression de nœuds

- Repérer le nœud à ôter `<br>`
  - `elementbr = document.getElementsByTagName('br')[0]`
- Repérer le nœud parent `<b>`
  - `elementparent = elementbr.parentNode;`
- Retirer le nœud `<br>`
  - `ancienbr = elementparent.removeChild(elementbr)`
    - `elementbr` est le nœud enfant à retirer du DOM.
    - `elementparent` est le nœud parent de enfant.
    - `ancienbr` conserve une référence au nœud enfant retiré.
- *Le nœud retiré existe toujours en mémoire, mais ne fait plus partie du DOM. Il peut être réutilisé plus tard dans votre code, via la référence à l'objet `ancienbr`.*

# Modification d'un style

- `element.style.nomduStyle = 'style'`
  - ex.:
    - `x = document.body;`
    - `x.style.color = 'blue';`
- La majeure partie des styles sont en mode lecture/écriture
- Si le nom d'un style est séparé par un « - », ôter le « - » et mettre la première lettre du deuxième mot en majuscule
  - Ex.: `background-color`
    - `x.style.backgroundColor = 'red'`