



(Cours n° 06)

2^{ème} année Licence Informatique
Université Abou-Bekr Belkaïd - Tlemcen
2018 / 2019

Dr. Mahfoud Houari
mahfoud.houari@gmail.com

Plan du chapitre

- 1 Introduction
- 2 Généralités
- 3 Fonctions
- 4 Tableaux et chaînes de caractères
- 5 Le DOM HTML

JavaScript



Les tableaux indexés

Déclaration :

```
var Tab1 = []; //Tableau vide  
var Tab2 = [10, 20.34, true, "phrase"]; //Tableau avec valeurs primitives  
var Tab3 = [[1, 2], [3, 4]]; //Matrice
```

Afficher un tableau :

```
alert(Tab2); //Affiche: 10, 20.34, true, phrase
```

Taille d'un tableau :

```
alert(Tab3.length); //Affiche: 2
```

Accéder aux items :

```
alert(Tab3[1]); //Affiche: 3, 4  
alert(Tab3[1][0]); //Affiche: 3
```

Les tableaux indexés

Opérations sur les tableaux:

```
var Tab = [2, 1, 5];
```

Ajouter des éléments à la fin :

```
Tab.push(4);
```

//Résultat: 2, 1, 5, 4

Ajouter des éléments au début :

```
Tab.unshift(-13);
```

//Résultat: -13, 2, 1, 5, 4

Supprimer le premier élément :

```
Tab.shift();
```

//Résultat: 2, 1, 5, 4

Supprimer le dernier élément :

```
Tab.pop();
```

//Résultat: 2, 1, 5

Les tableaux indexés

Opérations sur les tableaux:

Inverser le tableau :

Tab = *Tab.reverse()*; //Résultat: 5, 1, 2

Sélectionner une partie :

Tab = *Tab.slice(0, 2)*; //Résultat: 5, 1

Trier un tableau :

Tab = *Tab.sort()*; //Résultat: 1, 5

Les chaînes de caractères

Déclaration :

```
var chaine = "ma chaine de caractères";
```

Taille :

```
alert(chaine.length); //Affiche: 23
```

Caractère à un indice donné :

```
chaine.charAt(0); //Retourne: ' m'
```

Sélectionner une partie :

```
chaine.slice(3, 9); //Retourne: "chaine"
```

Découper une chaîne :

```
chaine.split(" "); //Retourne: ["ma", "chaine", "de", "caractères"]
```

Écriture en minuscule : `chaine = chaine.toLowerCase();`

Écriture en majuscule : `chaine = chaine.toUpperCase();`

Plan du chapitre

- 1 Introduction
- 2 Généralités
- 3 Fonctions
- 4 Tableaux et chaînes de caractères
- 5 Le DOM HTML

JavaScript



Le DOM HTML

Qu'est-ce que c'est ?

- **Document Object Model**: Représentation du document HTML à base d'objets.
- C'est une **API** de programmation (c.-à-d. un ensemble de classes, de méthodes, ou d'objets prédéfinies à utiliser).
- Permet de manipuler le contenu d'un document HTML :
 - ✓ **Accéder** aux éléments HTML (à leurs propriétés, leurs contenus,...).
 - ✓ **Modifier** les différentes parties d'un document HTML (supprimer des éléments existants, ajouter de nouveaux éléments,...).
- L'objet prédéfini **document** représente tout le document HTML.
- Autres objets prédéfinis: **window**, **event**.

Le DOM HTML

document.getElementById()

Exemple 1:



Tapez le texte

v4XBG

Étape suivante

← id="captcha" + oninput="verif()"

← id="btn"

Le DOM HTML

document.**getElementById()**

Exemple 1:

```
<script>
function verif(){
    if(document.getElementById("captcha").value == "v4XBG")
        document.getElementById("btn").disabled = false;
    else{
        document.getElementById("btn").disabled = true;
        document.getElementById("captcha").style.borderColor = "red";
    }
}
</script>
```

Le DOM HTML

Accéder au contenu textuel :

- **value** : La valeur de l'attribut **value** (éléments **input**, **progress**, **option**)
- **textContent** : La valeur textuelle des éléments **label**, **p**, **textarea**, **td**,...

Accéder aux propriétés CSS :

- **style.color** : La couleur du texte.
- **style.width** : La largeur de l'élément.
- **style.height** : La hauteur de l'élément.
- **style.borderColor** : La couleur de la bordure.
- **style.borderStyle** : Le style de la bordure.
- **style.backgroundColor** : La couleur du fond.
- **style.fontFamily** : La police du texte.
- **style.textAlign** : L'alignement du texte.

Le DOM HTML

document.**getElementsByName()**

Exemple 2:

```
<div>
```

Avez-vous un compte ?

```
<input type="radio" name="existe" onchange="choix();" checked>Oui
```

```
<input type="radio" name="existe" onchange="choix();" >Non
```

```
</div>
```

```
<form>
```

```
<fieldset id="field_1">
```

```
<legend id="legend_1">Authentication:</legend>
```

```
</fieldset>
```

```
</form>
```

Avez-vous un compte ? ☒ Oui ☐ Non

Authentication:

Le DOM HTML

document.getElementsByName()

Exemple 2:

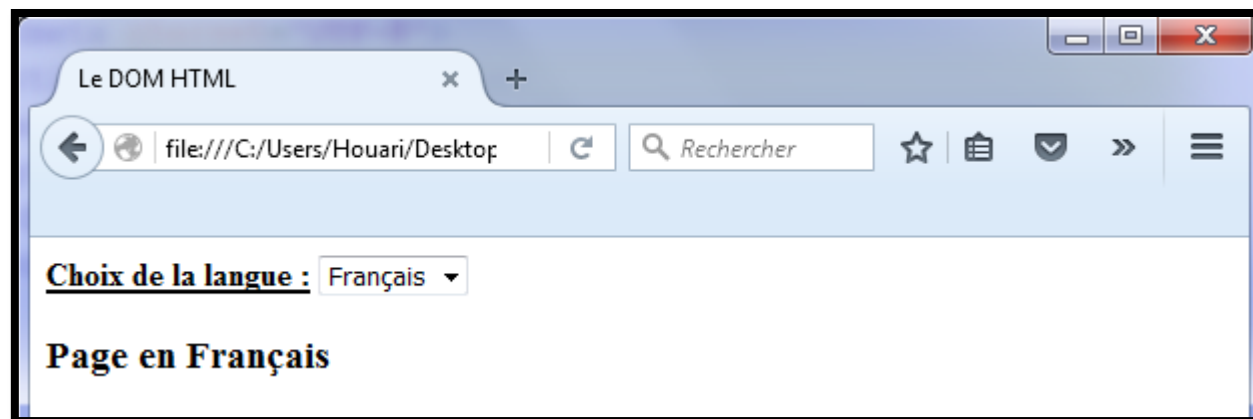
```
<script>
  function choix(){
    var choix = document.getElementsByName("existe");
    if(choix[0].checked){
      document.getElementById("legend_1").textContent = "Authentification: ";
      //Ajouter des éléments d'authentification
    }
    else if(choix[1].checked) {
      document.getElementById("legend_1").textContent = "Inscription: ";
      //Ajouter des éléments d'inscription
    }
  }
</script>
```

Le DOM HTML

Récupérer le choix dans une liste déroulante

Exemple 3:

```
<body>  
  <b><u>Choix de la langue :</u><b>  
  <select id="langue" onchange="choix();">  
    <option value="Ar">Arabe</option>  
    <option value="Fr" selected>Français</option>  
    <option value="An">Anglais</option>  
  </select>  
  <h3>Page en Français</h3>  
</body>
```




Le DOM HTML

Récupérer le choix dans une liste déroulante

Exemple 3:

```
<script>
function choix(){
    var liste = document.getElementById("langue");
    var langue_choisie = liste.options[liste.selectedIndex].textContent;
    if(langue_choisie == "Fr"){
        document.location = "Francais.HTML";
    } else if(langue_choisie == "An"){
        document.location = "Anglais.HTML";
    } if(langue_choisie == "Fr"){
        document.location = "Francais.HTML";
    } else {
        document.location = "Arabe.HTML";
    }
}
</script>
```

 **document.location** permet de faire une redirection vers une autre page

Le DOM HTML

La propriété ***innerHTML***: retourne le code HTML du contenu de l'élément

Exemple 4:

```
<body>
  <div id="div1">
    <div>
      <p>Paragraphe 1</p><p>Paragraphe 2</p>
    </div>
  </div>
  <script>
    alert(document.getElementById("div1").innerHTML);
  </script>
</body>
```


Le DOM HTML

La propriété ***innerHTML***: retourne le code HTML du contenu de l'élément

Exemple 4:

```
<body>
```

```
  <div id="div1">
```

```
    <div>
```

```
      <p>
```

```
    </div>
```

```
  </div>
```

```
  <script type="text/javascript">
```

```
    alert(document.getElementById("div1").innerHTML);
```

```
  </script>
```

```
</body>
```

Paragraphe 1

Paragraphe 2

```
<div>
```

```
  <p>Paragraphe 1</p><p>Paragraphe 2</p>
```

```
</div>
```

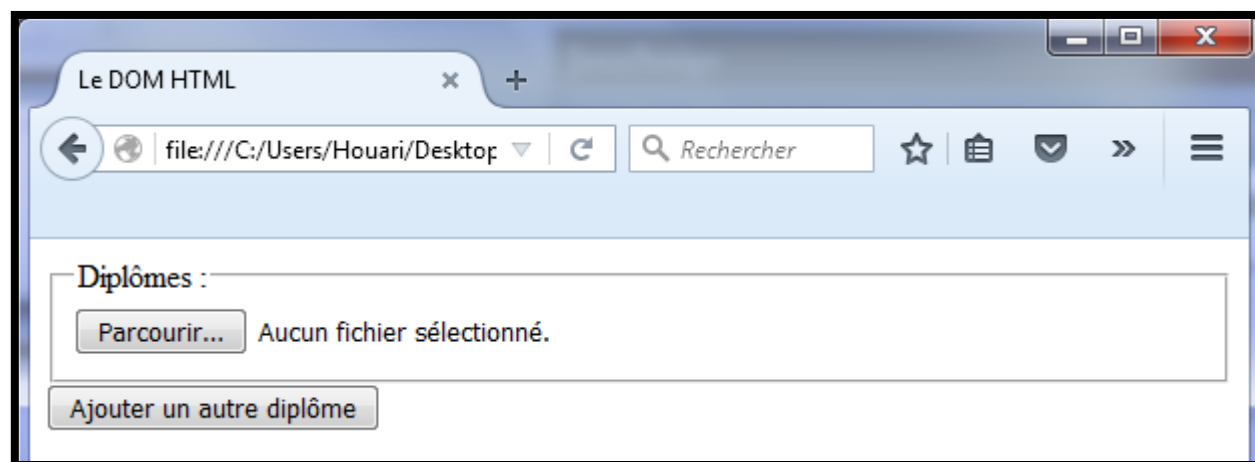
OK

Le DOM HTML

La propriété *innerHTML*:

Exemple 5:

```
<body>  
  <fieldset id="id_1">  
    <legend>Diplômes :</legend>  
    <input type="file" name="diplome_1">  
  </fieldset>  
  <input type="button" value="Ajouter un autre diplôme"  
    onclick="ajouter()">  
</body>
```



Le DOM HTML

La propriété *innerHTML*:

Exemple 5:

```
<body>
  <fieldset id="id_1">
    <legend>Diplômes :</legend>
    <input type="file" name="diplome_1">
  </fieldset>
  <input type="button" value="Ajouter un autre diplôme"
    onclick="ajouter()">
</body>
<script>
  function ajouter(){
    document.getElementById("id_1").innerHTML +=
      "<br><input type='file'>";
  }
</script>
```

Le DOM HTML

La propriété *innerHTML*:

Exemple 5:

```
<body>
  <fieldset id="id_1">
    <legend>Diplômes :</legend>
    <input type="file" name="diplome_1">
  </fieldset>
  <input type="button" value="Ajouter un autre diplôme"
    onclick="ajouter()">
</body>
<script>
  function ajouter(){
    document.getElementById("id_1").innerHTML +=
      "<br><input type='file'>";
  }
</script>
```



Mais comment gérer l'attribut *name* dynamiquement ?

Le DOM HTML

La propriété *innerHTML*:

Exemple 5:

```
<body>
  <fieldset id="id_1">
    <legend>Diplômes :</legend>
    <input type="file" name="diplome_1">
  </fieldset>
  <input type="button" value="Autre diplôme" onclick="ajouter()">
</body>
<script>
  var cpt = 1;
  function ajouter(){
    cpt ++;
    document.getElementById("id_1").innerHTML +=
      "<br><input type='file' name='diplome_' + cpt + '>";
  }
</script>
```

Le DOM HTML

La propriété *innerHTML*:

Exemple 6:

Avez-vous un compte ? ☒ *Oui* ☐ *Non*

Authentication:

Pseudo

Mot de passe

Accès au compte

Le DOM HTML

La propriété *innerHTML*:

Exemple 6:

Avez-vous un compte ? ☐ *Oui* ☒ *Non*

Inscription:

Pseudo

Mot de passe

Retapez-le

Inscription

Le DOM HTML

La propriété *innerHTML*:

Exemple 6:

```
<script>
  function choix(){
    var choix = document.getElementsByName("existe");
    if(choix[0].checked){
      authentication();
    }
    else {
      inscription();
    }
  }
</script>
```


Le DOM HTML

La propriété *innerHTML*:

Exemple 6:

```
<script>
function authentication(){
    document.getElementById("field_1").innerHTML =
        <form method= 'post' action = 'authentication.php'>" +
            "Pseudo: <input type='text'>" +
            "Mot de passe:<input type='password'>" +
            "<input type='button' value='Accès au compte'>" +
        "</form>";
}
</script>
```

Le DOM HTML

La propriété *innerHTML*:

Exemple 6:

```
<script>
function inscription(){
    document.getElementById("field_1").innerHTML =
        <form method= 'post' action = 'inscription.php'>" +
            "Pseudo:<input type='text'>" +
            "Mot de passe:<input type='password'>" +
            "Retapez-le mot de passe:<input type='password'>" +
            "<input type='button' value='Inscription'>" +
        "</form>";
}
</script>
```

Le DOM HTML

Autres propriétés:

- **firstElementChild**: Retourne le premier élément fils d'un élément donné.
- **lastElementChild**: Retourne le dernier élément fils d'un élément donné.
- **nextElementSibling**: Retourner le frère droit d'un élément donné.
- **previousElementSibling**: Retourne le frère gauche d'un élément donné.
- **parentNode**: Retourne l'élément père d'un élément donné.
- **childNodes**: Retourne un tableau contenant tous les éléments fils d'un élément donné.

Le DOM HTML

Quelques méthodes:

- `document.createElement(nom_balise)`: Permet de créer un nouvel élément HTML.

Exemple 7:

```
<script type="text/javascript">  
    var nouveau_parag = document.createElement("p") ;  
    nouveau_parag.textContent = "Nouveau paragraphe" ;  
</script>
```

Le DOM HTML

Quelques méthodes:

- **appendChild(élément_crée)**: Permet d'ajouter un nouvel élément comme dernier fils d'un élément existant. S'applique sur l'élément père.

Exemple 7 (suite):

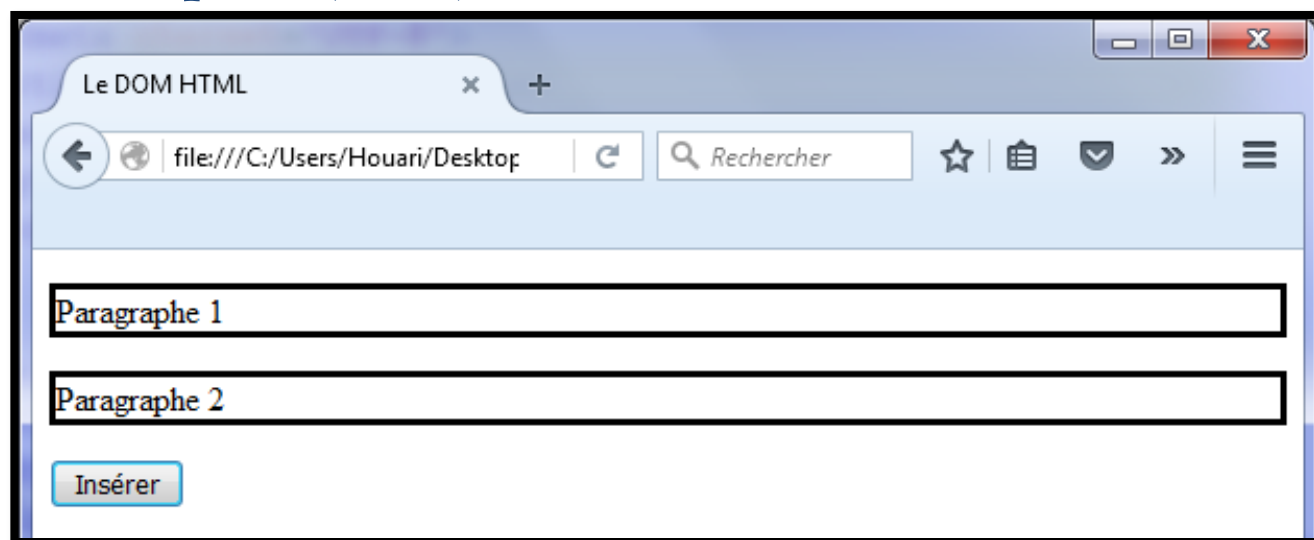
```
<body>
  <div id="div_1">
    <p style="border-style:solid">Paragraphe 1</p>
    <p style="border-style:solid">Paragraphe 2</p>
  </div>
  <input type="button" value="Insérer" onclick="Ajouter()">
</body>
```

Le DOM HTML

Quelques méthodes:

- **appendChild(élément_crée):** Permet d'ajouter un nouvel élément comme dernier fils d'un élément existant. S'applique sur l'élément père.

Exemple 7 (suite):



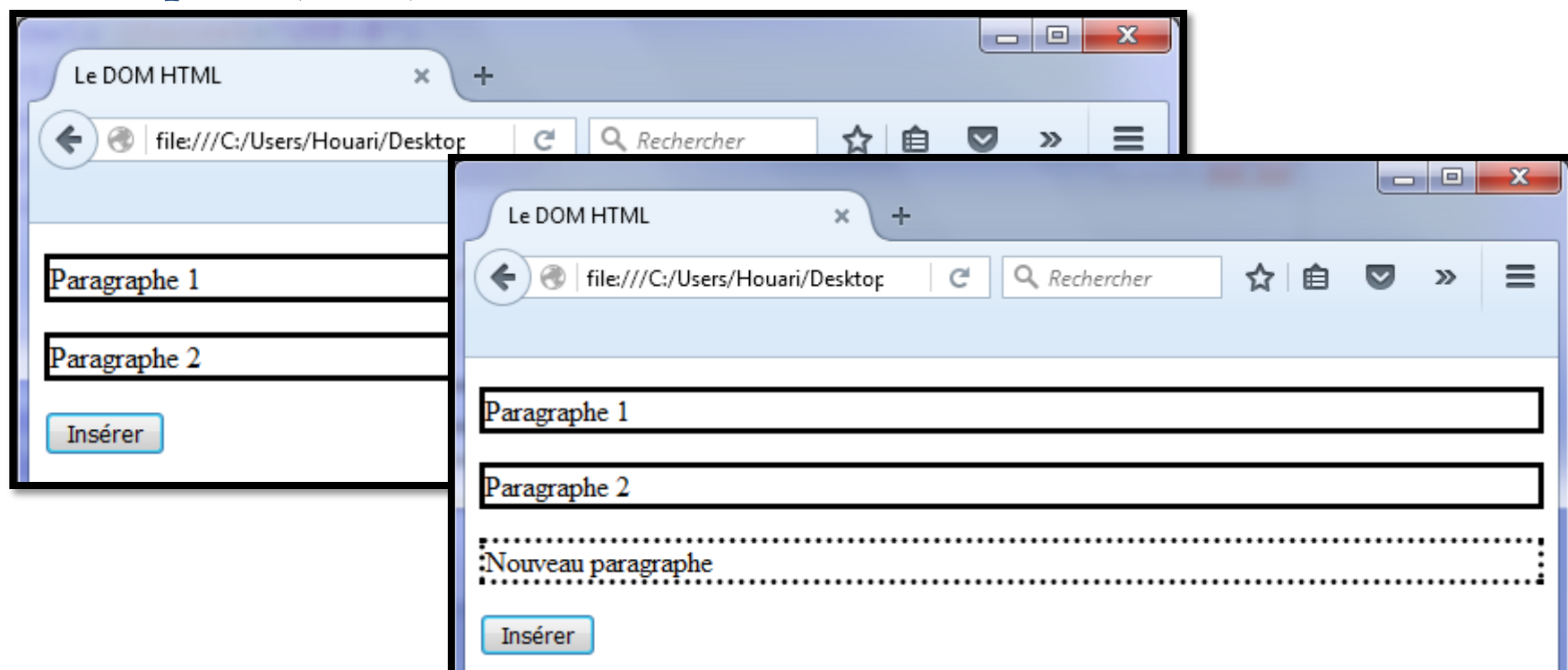
ter()">

Le DOM HTML

Quelques méthodes:

- **appendChild(élément_crée):** Permet d'ajouter un nouvel élément comme dernier fils d'un élément existant. S'applique sur l'élément père.

Exemple 7 (suite):



Le DOM HTML

Quelques méthodes:

- **appendChild(élément_crée)**: Permet d'ajouter un nouvel élément comme dernier fils d'un élément existant. S'applique sur l'élément père.

Exemple 7 (suite):

```
<script type="text/javascript">
```

```
function Ajouter(){
```

```
    var nouveau_parag = document.createElement("p");
```

```
    nouveau_parag.textContent = "Nouveau paragraphe";
```

```
    nouveau_parag.style.borderStyle = "dotted";
```

```
    document.getElementById("div_1").appendChild(nouveau_parag);
```

```
}
```

```
</script>
```

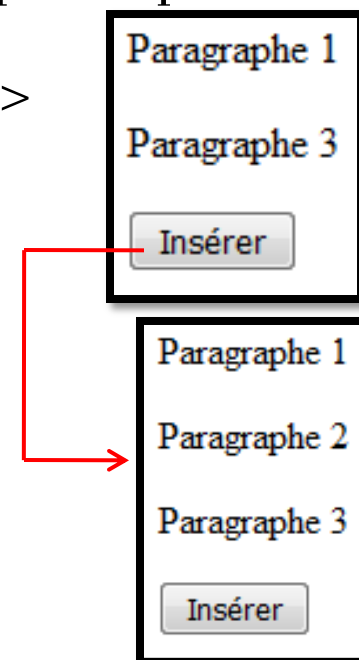

Le DOM HTML

Quelques méthodes:

- **insertBefore(nouvel_élément, fils)**: Ajoute le nouvel élément avant un certain élément fils. S'applique sur l'élément père.

Exemple 9 :

```
<body>  
  <div id="div_1">  
    <p id="p_1">Paragraphe 1</p><p id="p_3">Paragraphe 3</p>  
  </div>  
  <input type="button" value="Insérer" onclick="Ajouter()">  
</body>
```



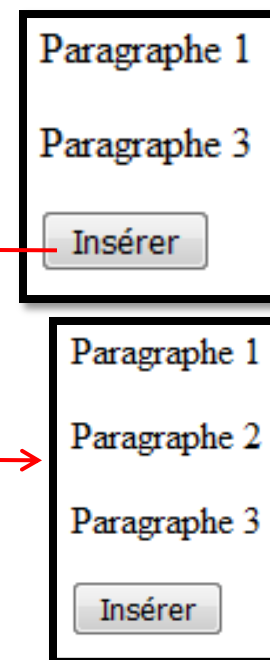
Le DOM HTML

Quelques méthodes:

- **insertBefore(nouvel_élément, fils)**: Ajoute le nouvel élément avant un certain élément fils. S'applique sur l'élément père.

Exemple 8 :

```
<script type="text/javascript">  
function Ajouter(){  
    var nouveau_parag = document.createElement("p") ;  
    nouveau_parag.textContent = "Paragraphe 2" ;  
    document.getElementById("div_1").  
    insertBefore(nouveau_parag, document.getElementById("p_3"));  
}  
</script>
```



Le DOM HTML

Quelques méthodes:

- **removeChild**(référence_fils_a_supprimer): S'applique sur l'élément père pour supprimer un élément fils.

Exemple 9 : Supprimer le 2^{ème} paragraphe

```
<body>
  <div id="div_1">
    <p id="p_1">Paragraphe 1</p><p id="p_2">Paragraphe 2</p>
  </div>
  <input type="button" value="Supprimer" onclick="Supprimer()">
</body>
<script type="text/javascript">
  function Supprimer(){
    document.getElementById("div_1").
    removeChild(document.getElementById("p_2"));
  }
</script>
```

Le DOM HTML

Remarque importante :

- Soit le tableau affiché identifié par *table_1*.
- Le tableau ne contient pas directement les 4 lignes, mais il a un conteneur et c'est ce conteneur qui contient les différentes lignes du tableau.
- Donc, `document.getElementById("table_1").firstElementChild` ne crée pas une référence vers la première ligne du tableau mais tout simplement ça crée une référence vers le conteneur du tableau.
- Donc, pour créer une référence vers la première ligne du tableau (ligne de titres), il faut appeler encore la méthode `firstElementChild` sur la référence du conteneur:

```
var ligne_de_titres = document.getElementById("table_1").  
    firstElementChild.firstElementChild;
```

```
var premiere_ligne_de_donnees = document.getElementById("table_1").  
    firstElementChild.firstElementChild.  
    nextElementSibling;
```

```
var deuxieme_ligne_de_donnees = premiere_ligne_de_donnees.  
    nextElementSibling;
```

Valeurs
13.5
15.75
16

Le DOM HTML

Exemple 11 : Supprimer la ligne d'un tableau contenant la valeur *val*

```
<body>
  <div id="div_1">
    <table border="1" id="table_1">
      <tr><th>Valeurs</th></tr>
      <tr><td>13.5</td></tr>
      <tr><td>15.75</td></tr>
      <tr><td>16</td></tr>
    </table>
    <br>
    <input type="button" value="Supprimer une ligne"
           onclick="Supprimer()">
  </div>
</body>
```

Valeurs
13.5
15.75
16

Supprimer une ligne

Le DOM HTML

Exemple 11 : Supprimer la ligne d'un tableau contenant la valeur *val*

```
<script>
function Supprimer(){
    var val = prompt("Donnez la valeur à supprimer");
    var ligne_courante = document.getElementById("table_1").
        firstElementChild.firstElementChild.nextElementSibling;

    //La ligne des titres a été ignorée et on part de la 2ème ligne
    while((ligne_courante != null) &&
        (ligne_courante.firstElementChild.textContent != val)){
        ligne_courante = ligne_courante.nextElementSibling ;
    }
    if(ligne_courante == null){
        alert("Aucune ligne ne contient la valeur "+val);
    } else {
        ligne_courante.parentNode.removeChild(ligne_courante);
    }
}
</script>
```