

Modifier la hiérarchisation

Le W3C DOM vous permet de créer vos propres nœuds élément ou nœuds texte et de les insérer dans le document, modifiant ainsi la hiérarchie de celui-ci.

1. La méthode createElement

La méthode `createElement` permet, comme son nom le suggère, de créer un nouvel élément.

Exemple

```
var x = document.createElement("h1");
var x = document.createElement("div");
```

Il faut noter que l'élément ainsi créé n'apparaît pas encore dans le document. Il faut utiliser les méthodes `appendChild` ou `insertBefore` pour l'ajouter à celui-ci.

2. La méthode createTextNode

La méthode `createTextNode` crée un nœud texte dans le document.

Exemple

```
var x = document.createElement("h1");
var y = document.createTextNode("Titre de niveau 1");
```

Il faut noter que le nœud ainsi créé n'apparaît pas encore dans le document. Il faut utiliser les méthodes `appendChild` ou `insertBefore` pour l'ajouter à celui-ci.

3. La méthode appendChild

La méthode `appendChild` ajoute un élément comme dernier nœud enfant d'un nœud spécifié en argument.

La méthode `appendChild` s'applique donc toujours à un nœud parent.

Exemple

```
var x = document.getElementsByTagName('p')[0];
x.parentNode.appendChild(x);
```

Prenons une application complète pour illustrer, de façon détaillée, le fonctionnement de la méthode `appendChild`.

Soit un fichier HTML :

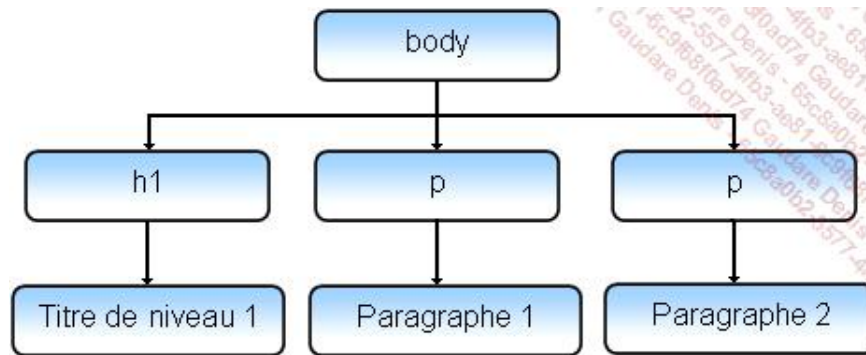
```
<!DOCTYPE html>
<html lang="fr">
<head>
<title>Le DOM</title>
```

```

<meta charset="UTF-8">
<body>
<h1>Titre de niveau 1</h1>
<p>Paragraphe 1</p>
<p>Paragraphe 2</p>
</body>
</html>

```

L'arbre des nœuds, limité à la balise body et à ses descendants, est le suivant :



Au clic dans la fenêtre du document, appliquons la méthode `appendChild` à la balise `<h1>`.

```

<!DOCTYPE html>
<html lang="fr">
<head>
<title>Le DOM</title>
<meta charset="UTF-8">
<script>
function actualiser(){
var x = document.getElementsByTagName('h1')[0];
x.parentNode.appendChild(x);
}
</script>
</head>
<body onclick=actualiser()>
<h1>Titre de niveau 1</h1>
<p>Paragraphe 1</p>
<p>Paragraphe 2</p>
</body>
</html>

```

Situation initiale :

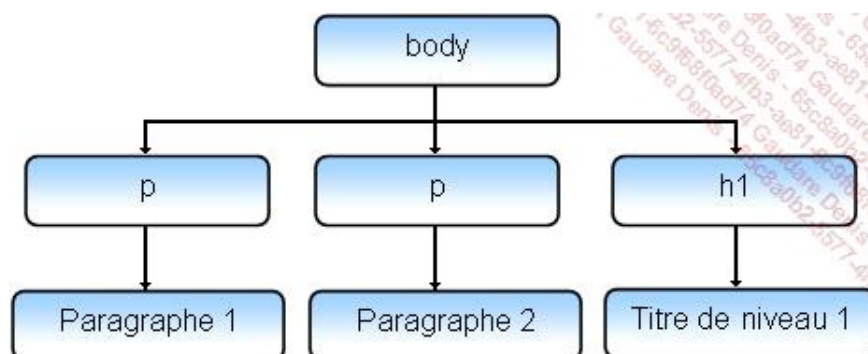


Après un clic sur le document :



La balise `<h1>` est obtenue par `getElementsByTagName('h1')[0]`, puis la balise parent `<body>` est atteinte par la méthode `parentNode`. Enfin, la balise `<h1>` est placée comme dernier enfant par `appendChild`, ainsi `<h1>` se retrouve en dernière position.

L'arbre du document a donc été modifié :



Ou encore, à partir du fichier HTML initial, ajoutons au document un élément créé par `createElement`.

```

<!DOCTYPE html>
<html lang="fr">
<head>

```

```

<title>Le DOM</title>
<meta charset="UTF-8">
<script>
function actualiser(){
var x = document.createElement("p");
var texte = document.createTextNode("Je suis le petit nouveau");
y = x.appendChild(texte);
document.body.appendChild(x);
}
</script>
</head>
<body onclick=actualiser()>
<h1>Titre de niveau 1</h1>
<p>Paragraphe 1</p>
<p>Paragraphe 2</p>
</body>
</html>

```

Par la variable `x`, la méthode `createElement("p")` crée un nouveau paragraphe. Un nouveau texte est défini par `createTextNode("Je suis le petit nouveau")`. Le texte est associé à la balise par `x.appendChild(texte)`. Le nouvel élément dans sa totalité est inséré dans le document comme dernier enfant de la balise parent `<body>` par la propriété `appendChild`.



4. La méthode insertBefore

La méthode `insertBefore` permet d'insérer un nœud avant un autre nœud.

Cette méthode est utilisée lorsque l'on souhaite ajouter un élément sans qu'il devienne le dernier élément enfant (voir `appendChild`).

Exemple

```
x.parentNode.insertBefore(x,y);
```

Ainsi, en reprenant le document XHTML du point précédent, insérons la balise `<h1>` après la première balise de paragraphe (`<p>`).

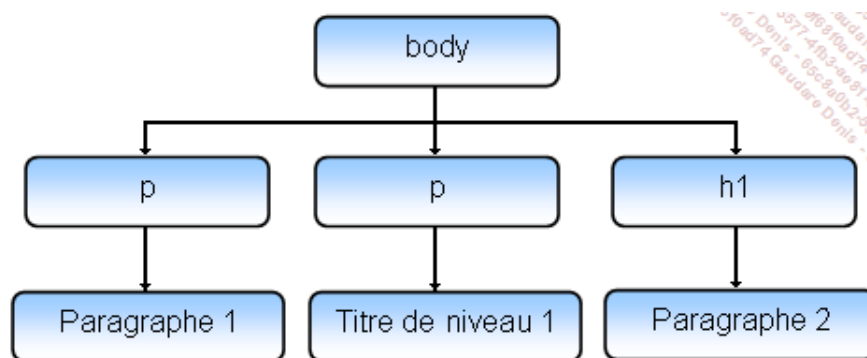
```
<!DOCTYPE html>
<html lang="fr">
<head>
<title>Le DOM</title>
<meta charset="UTF-8">
<script>
function actualiser(){
var x = document.getElementsByTagName('p')[0];
var y = document.getElementsByTagName('h1')[0];
x.parentNode.insertBefore(x,y);
}
//]]>
</script>
</head>
<body onclick=actualiser()>
<h1>Titre de niveau 1</h1>
<p>Paragraphe 1</p>
<p>Paragraphe 2</p>
</body>
</html>
```





La variable `x` repère, par `getElementsByTagName('p')[0]`, le premier paragraphe `<p>`. La variable `y` désigne avec `getElementsByTagName('h1')[0]` la balise `<h1>`. Ensuite, l'ordre des enfants de la balise `<body>` (`parentNode`) est interverti par `insertBefore(x,y)`.

L'arbre modifié se présente comme suit :



5. La méthode `replaceChild`

La méthode `replaceChild` remplace un nœud par un autre nœud.

Le nœud ainsi remplacé est supprimé, ainsi que tous ses descendants.

Exemple

```
x.parentNode.replaceChild(x,y);
```

Remplaçons le deuxième paragraphe de notre document XHTML par l'élément `<h1>`.

Le code devient :

```
<!DOCTYPE html>
<html lang="fr">
<head>
<title>Le DOM</title>
<meta charset="UTF-8">
<script>
```

```
function actualiser(){
var x = document.getElementsByTagName('h1')[0];
var y = document.getElementsByTagName('p')[1];
x.parentNode.replaceChild(x,y);
}
</script>
</head>
<body onclick=actualiser(>
<h1>Titre de niveau 1</h1>
<p>Paragraphe 1</p>
<p>Paragraphe 2</p>
</body>
</html>
```

Situation initiale :



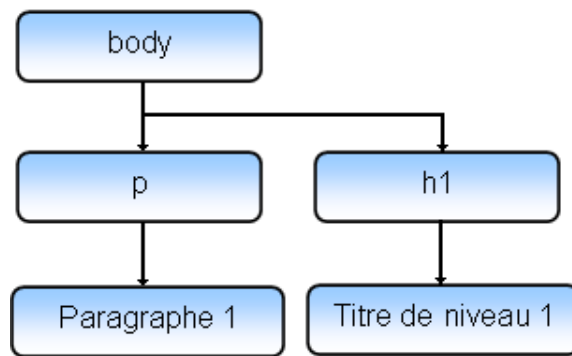
Après un clic sur le document :



La balise `<h1>` est sélectionnée par `getElementsByTagName('h1')[0]` et le second paragraphe `<p>` par `getElementsByTagName('p')[1]`. Par la propriété `replaceChild`, on remplace le second élément par le premier.

Il faut noter que l'élément `<h1>` a changé de position et que le second paragraphe `<p>` a disparu du document. Le node texte de `<h1>` a suivi le changement de position.

L'arbre modifié devient :



6. La méthode `removeChild`

La méthode `removeChild` supprime le nœud (et éventuellement ses descendants) fourni en argument.

```
var x = document.getElementsByTagName('p')[0];
x.parentNode.removeChild(x);
```

Supprimons le premier paragraphe `<p>` du document.

```
<!DOCTYPE html>
<html lang="fr">
<head>
<title>Le DOM</title>
<meta charset="UTF-8">
<script>
function actualiser(){
var x = document.getElementsByTagName('p')[0];
x.parentNode.removeChild(x);
}
</script>
</head>
<body onclick=actualiser()>
<h1>Titre de niveau 1</h1>
<p>Paragraphe 1</p>
<p>Paragraphe 2</p>
</body>
</html>
```

Situation initiale :

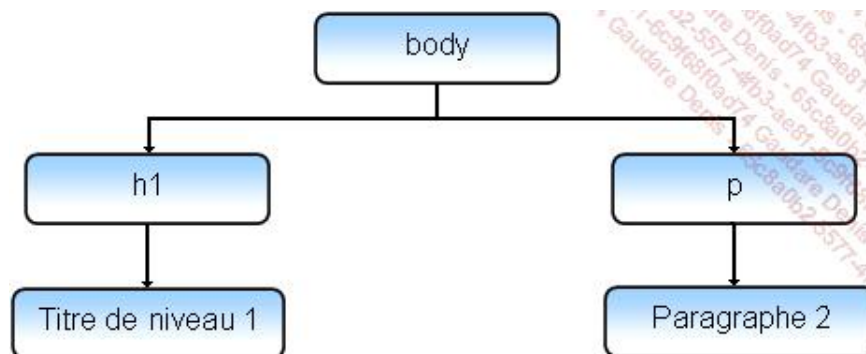


Au clic sur le document :



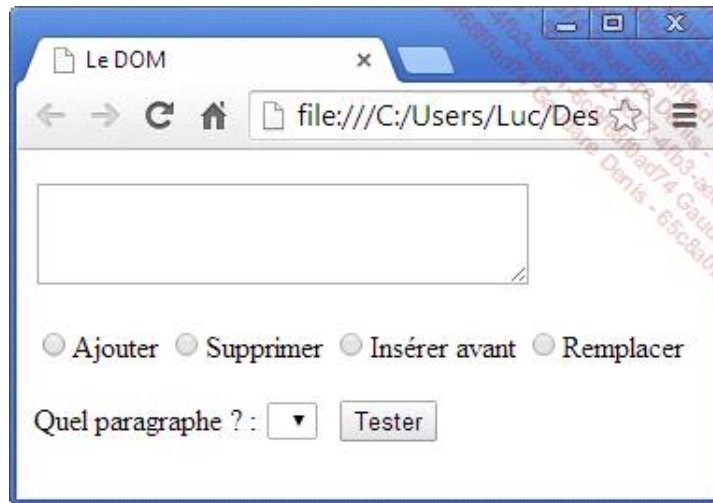
Après avoir repéré l'élément à supprimer, soit `getElementsByName('p')[0]`, il est supprimé de la balise `<body>` par la propriété `removeChild`.

L'arbre du document se présente comme suit :



7. Une application récapitulative

Ce script permet d'ajouter, de supprimer, d'insérer et de remplacer des paragraphes de texte (voir les quatre boutons radio pour les quatre actions possibles). Le paragraphe à insérer dans la page provient d'une zone de texte. Si l'action nécessite de connaître un numéro de paragraphe, celui-ci sera fourni par un menu déroulant, actualisé à chaque action antérieure.



Le code du fichier HTML initial est :

```
<!DOCTYPE html>
<html lang="fr">
<head>
<title>Le DOM</title>
<meta charset="UTF-8">
</head>
<body>
<form>
<p>
<textarea id="zonetexte" rows="3" cols="30"></textarea>
</p>
<p>
<input type="radio" name="bouton">Ajouter
<input type="radio" name="bouton">Supprimer
<input type="radio" name="bouton">Insérer avant
<input type="radio" name="bouton">Remplacer
</p>
<p>
Quel paragraphe ? : <select id="liste"></select>&nbsp;  
<input type="submit" value="Tester"></p>
</form>
<div id="modifiable"> </div>
</body>
</html>
```

Commençons le script par la fonction ajouter() :

```
function ajouter() {
var contenu = document.getElementById("zonetexte").value;
var newparag = document.createElement("p");
var newtexte = document.createTextNode(contenu);
newparag.appendChild(newtexte);
changement.appendChild(newparag);
document.getElementById("zonetexte").value="";
}
```

```
}
```

La zone de texte est atteinte par la méthode `getElementById` prenant en paramètre son identifiant (`zonetexte`). La propriété `value` nous fournit sa valeur. Cette valeur est stockée dans la variable `contenu`. La variable `newparag` créée, par la méthode `createElement`, un nouveau nœud élément (ici un paragraphe `p`). La variable `newtexte` créée, par `createTextNode`, un nouvel élément texte dont le contenu est fourni par la variable `contenu`. Le nœud texte est associé au nouveau nœud élément par `newparag.appendChild(newtexte)`. Le tout est inclus en dernière position dans le document `changement.appendChild(newparag)` où `changement` a été défini par ailleurs comme la zone de la balise `<div id="modifiable"> ... </div>`. La dernière ligne réinitialise la zone de texte.

Le script doit par ailleurs déclencher cette fonction `ajouter()` lorsque le bouton radio **Ajouter** est sélectionné. Il doit aussi modifier dynamiquement les options du menu déroulant `<select>`, lorsque le paragraphe est effectivement ajouté au document. Ce que fait la fonction `changer()`.

```
function changer() {
var typeaction = -1;
var nombreparag = changement.getElementsByTagName("p").length;
var boutonradio = document.getElementsByTagName("form")[0].bouton;
for (var i=0; i<boutonradio.length; i++) {
if (boutonradio[i].checked) {
typeaction = i;
}
}
if (typeaction== -1) {
alert("Pas de bouton radio sélectionné");
}
if (typeaction==0) {
ajouter();
}
document.getElementById("liste").options.length = 0;
for (i=0; i<changement.getElementsByTagName("p").length; i++) {
document.getElementById("liste").options[i] = new Option(i+1);
}
return false;
}
```

La variable `typeaction` est initialisée à -1 (`var typeaction = -1`). Le nombre de paragraphes (`var nombreparag`) est donné par la longueur (`length`) du tableau renvoyé par `getElementsByTagName("p")`.

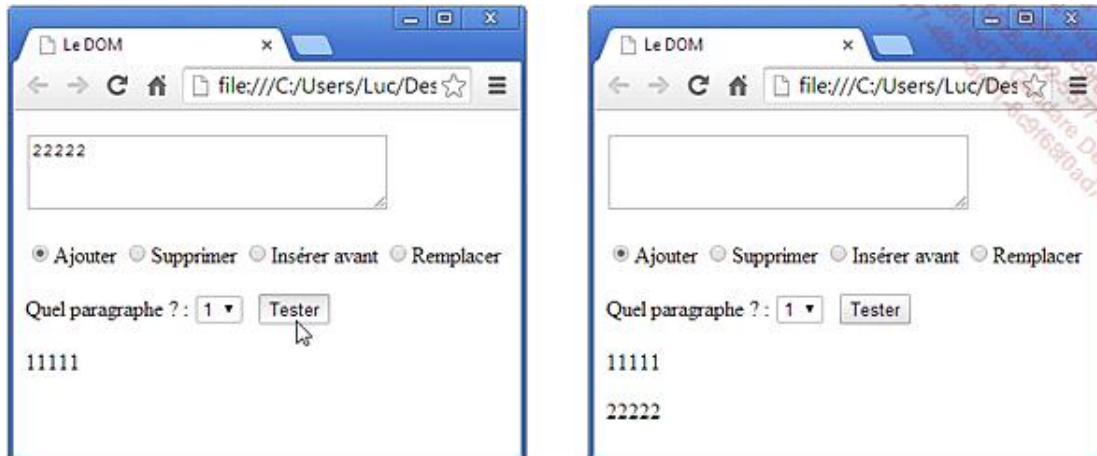
Le nombre de boutons radio est donné par le nombre de boutons (`bouton`) présents dans le formulaire (`getElementsByTagName("form")[0]`). Une boucle `for` passe ces boutons en revue pour trouver l'élément sélectionné (`checked`). Le numéro d'ordre du bouton (`boutonradio[i]`) est le numéro du type d'action (`typeaction = i`).

Si aucun bouton n'a été sélectionné par l'utilisateur, la variable `typeaction` a gardé sa valeur initiale. Un test sur cette valeur initiale (`if (typeaction== -1)`) permet de déclencher une boîte d'alerte.

Ici, le premier bouton étant sélectionné, le type d'action prend la valeur 0. Ainsi au moyen d'un simple test conditionnel (`if (typeaction==0)`), la fonction `ajouter()` peut être appelée.

Le nombre d'options (`options.length`) du menu déroulant (`liste`) est réinitialisé à 0. Une boucle `for` compte

tous les paragraphes (`getElementsByTagName("p")`) de la zone `<div id="modifiable">` (changement). À chaque passage de la boucle, la liste des options est alimentée (`getElementById("liste").options[i]`) par le nombre en cours plus 1 de la boucle (`new Option(i+1)`).



Pour la fonction `supprimer()` :

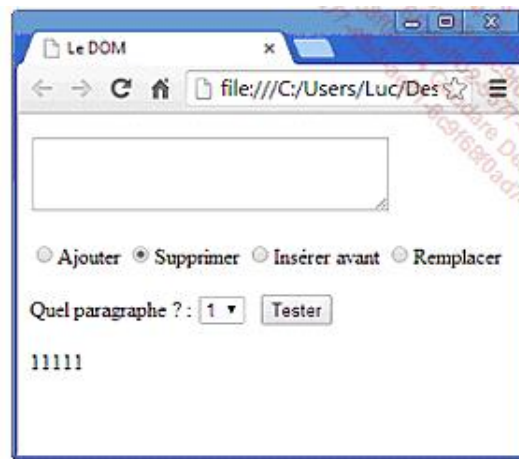
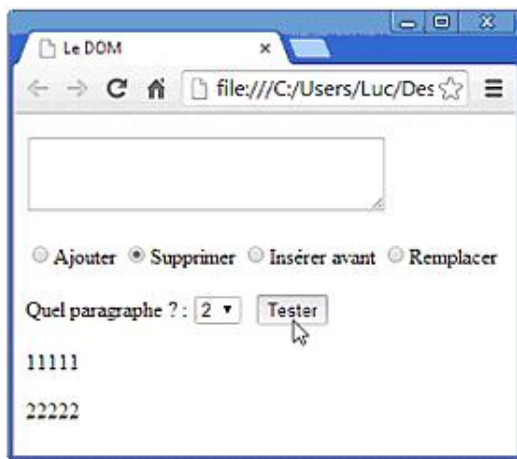
```
function supprimer() {  
    var choixout = document.getElementById("liste").selectedIndex;  
    var tous = changement.getElementsByTagName("p");  
    var asupprimer = tous[choixout];  
    changement.removeChild(asupprimer);  
}
```

La variable `choixout` mémorise l'option retenue par l'utilisateur dans le menu déroulant (`getElementById("liste").selectedIndex`). La variable `choixout` note ainsi le numéro d'ordre du paragraphe que l'on souhaite supprimer. La variable `tous` renvoie tous les paragraphes (`getElementsByTagName("p")`) sous forme d'un tableau. L'élément à supprimer (`var asupprimer`) est déterminé en appliquant la variable `choixout` au tableau (`tous[choixout]`). La suppression est obtenue par la méthode `removeChild(asupprimer)` appliquée à la zone de la balise `<div>`.

Il ne faut pas oublier d'ajouter à la fonction `changer()` :

```
if (typeaction==1 && nombreparag > 0) {  
    supprimer();  
}
```

La fonction `supprimer` n'a de sens que s'il y a des paragraphes dans la zone `<div>` (`nombreparag > 0`).



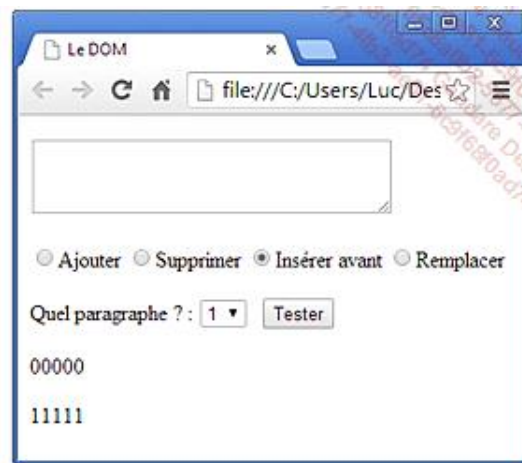
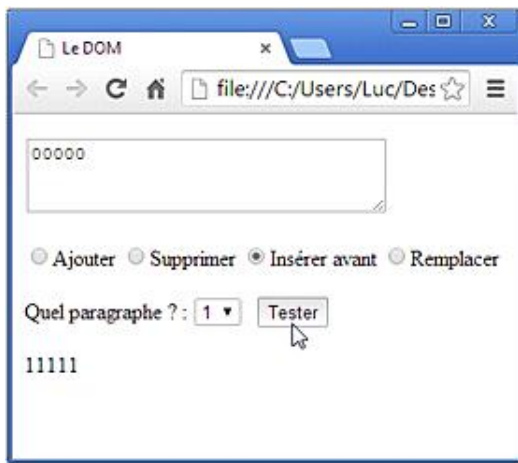
Passons maintenant à la fonction `inserer()` :

```
function inserer() {
var choixins = document.getElementById("liste").selectedIndex;
var intexte = document.getElementById("zonetexte").value;
var newparag = document.createElement("p");
var newtexte = document.createTextNode(intexte);
newparag.appendChild(newtexte);
var tous = changement.getElementsByTagName("p");
var oldparag = tous[choixins];
changement.insertBefore(newparag,oldparag);
document.getElementById("zonetexte").value="";
}
```

Le choix de l'utilisateur (`getElementById("liste").selectedIndex`) est stocké dans la variable `choixins`. La variable `intexte` récupère le contenu (`value`) de la zone de texte (`getElementById("zonetexte")`). Un nouveau paragraphe (`newparag`) est créé par la méthode `createElement("p")`. Un nouveau nœud de texte est créé par la méthode `createTextNode(intexte)`. Le nœud de texte est associé à l'élément (`newparag.appendChild(newtexte)`). La variable `tous` liste les paragraphes de la zone modifiable (`getElementsByTagName("p")`). Puis le paragraphe signalé par l'utilisateur (`var oldparag = tous[choixins]`) est identifié. Le nouveau paragraphe (`newparag`) est inséré avant celui-ci (`oldparag`) par la méthode `insertBefore(newparag,oldparag)`. Enfin, la zone de texte est réinitialisée.

Il faut encore ajouter à la fonction `changer()` :

```
if (typeaction==2 && nombreparag > 0) {
inserer();
}
```



Voici la fonction `remplacer()` :

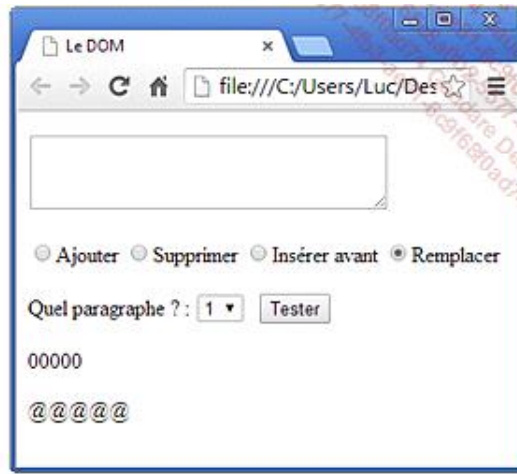
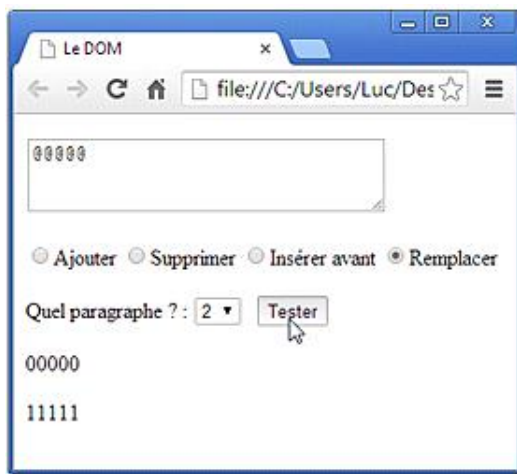
```
function remplacer() {
var inchoix = document.getElementById("liste").selectedIndex;
var intexte = document.getElementById("zonetexte").value;
var newparag = document.createElement("p");
var newtexte = document.createTextNode(intexte);
newparag.appendChild(newtexte);
var tous = changement.getElementsByTagName("p");
var oldparag = tous[inchoix];
changement.replaceChild(newparag,oldparag);
document.getElementById("zonetexte").value="";
}
```

La variable `inchoix` note le choix de l'utilisateur dans le menu déroulant. La variable `intexte` reprend le contenu de la zone de texte. La variable `newparag` crée le nouveau paragraphe par la méthode `createElement("p")`. La variable `newtexte` crée un nouveau nœud de texte `createTextNode(intexte)`. Le nœud de texte est associé au nœud élément (`newparag.appendChild(newtexte)`).

Le paragraphe à supprimer (`oldparag`) est fourni par `tous[inchoix]`. Le remplacement du paragraphe s'effectue par la méthode `replaceChild(newparag,oldparag)`. La zone de texte est ensuite réinitialisée.

Et il faut ajouter la fonction `changer()` :

```
if (typeaction==3 && nombreparag > 0) {
remplacer();
}
```



Le fichier avec le script complet devient :

```
<!DOCTYPE html>
<html lang="fr">
<head>
<title>Le DOM</title>
<meta charset="UTF-8">
<script>
window.onload = init;
var changement;
function init() {
document.getElementsByTagName("form")[0].onsubmit = function()
{return changer();}
changement = document.getElementById("modifiable");
}
function ajouter() {
var contenu = document.getElementById("zonetexte").value;
var newtexte = document.createTextNode(contenu);
var newparag = document.createElement("p");
newparag.appendChild(newtexte);
changement.appendChild(newparag);
document.getElementById("zonetexte").value="";
}
function supprimer() {
var choixout = document.getElementById("liste").selectedIndex;
var tous = changement.getElementsByTagName("p");
var asupprimer = tous[choixout];
changement.removeChild(asupprimer);
}
function inserer() {
var choixins = document.getElementById("liste").selectedIndex;
var intexte = document.getElementById("zonetexte").value;
var newtexte = document.createTextNode(intexte);
var newparag = document.createElement("p");
newparag.appendChild(newtexte);
var tous = changement.getElementsByTagName("p");
var oldparag = tous[choixins];
changement.insertBefore(newparag,oldparag);
document.getElementById("zonetexte").value="";
}
```

```

}
function remplacer() {
var inchoix = document.getElementById("liste").selectedIndex;
var intexte = document.getElementById("zonetexte").value;
var newtexte = document.createTextNode(intexte);
var newparag = document.createElement("p");
newparag.appendChild(newtexte);
var tous = changement.getElementsByTagName("p");
var oldparag = tous[inchoix];
changement.replaceChild(newparag,oldparag);
document.getElementById("zonetexte").value="";
}
function changer() {
var typeaction = -1;
var nombreparag = changement.getElementsByTagName("p").length;
var boutonradio = document.getElementsByTagName("form")[0].bouton;
for (var i=0; i<boutonradio.length; i++) {
if (boutonradio[i].checked) {
typeaction = i;
}
}
if (typeaction== -1) {
alert("Pas de bouton radio sélectionné");
}
if (typeaction==0) {
ajouter();
}
if (typeaction==1 && nombreparag > 0) {
supprimer();
}
if (typeaction==2 && nombreparag > 0) {
inserer()
}
if (typeaction==3 && nombreparag > 0) {
remplacer();
}
document.getElementById("liste").options.length = 0;
for (i=0; i<changement.getElementsByTagName("p").length; i++) {
document.getElementById("liste").options[i] = new Option(i+1);
}
return false;
}
</script>
</head>
<body>
<form>
<p>
<textarea id="zonetexte" rows="3" cols="30"></textarea>
</p>
<p>
<input type="radio" name="bouton">Ajouter
<input type="radio" name="bouton">Supprimer
<input type="radio" name="bouton">Insérer avant
<input type="radio" name="bouton">Remplacer
</p>

```



```
<p>
Quel paragraphe ? : <select id="liste"></select>&nbsp;
<input type="submit" value="Tester"></p>
</form>
<div id="modifiable"> </div>
</body>
</html>
```