

## 1. Modification d'un élément

### Avec `getElementsByName()`

Une première façon de faire est d'utiliser `getElementsByName()` au sein du `div #output` :

```
var output = document.getElementById('output');
var link   = output.getElementsByTagName('a')[0];

link.href = 'http://www.google.fr';
```

Ce code peut-être réduit de cette manière :

```
var output = document.getElementById('output');
var link   = output.getElementsByTagName('a')[0].href =
'http://www.openclassrooms.com';
```

Ou même comme ceci, mais ça devient illisible :

```
document.getElementById('output').getElementsByTagName('a')[0].href =
'http://www.google.fr';
```

### Avec `getElementsByName()` et gaffe aux erreurs

Les codes ci-dessus manquent de sécurisation, et il peut être intéressant de réaliser des tests avant de lancer des instructions. C'est nécessaire si votre script est destiné à un userscript par exemple :

```
var output = document.getElementById('output');

if (output) {
    var link = output.getElementsByTagName('a')[0];

    if (link) {
        link.href = 'http://www.google.fr';
    }
}
```

## 2. Utilisation du innerHTML

C'est assez simple en fait : il suffit de récupérer le contenu, avec `innerHTML`, puis de le réinjecter, encadré de balises `<blockquote>` :

```
var quote = document.getElementById('output').innerHTML;

document.getElementById('output').innerHTML = '<blockquote>' + quote
+ '</blockquote>';
```

## 3. Supprimer les enfants d'un élément

Pour ce genre de petit script, on se contente de faire une boucle tant qu'il reste des éléments à supprimer, et on supprime le premier élément enfant à chaque tour de boucle.

```
var output = document.getElementById('output');

var child = output.firstChild;
```

```
while (child) {

    output.removeChild(child);

    child = output.firstChild;

}
```

Dans un souci de concision, il est conseillé de placer l'affectation directement au sein de la condition de la boucle :

```
var output = document.getElementById('output'), child;

while (child = output.firstChild) {

    output.removeChild(child);

}
```

## Avec hasChildNodes()

Une alternative particulièrement séduisante est d'utiliser la méthode `hasChildNodes()` au sein de la condition de la boucle :

```
var output = document.getElementById('output');

while (output.hasChildNodes()) {

    output.removeChild(output.firstChild);

}
```

## Avec innerHTML, mais c'est un peu de la triche

Il est évidemment possible de le faire avec `innerHTML` (ou `innerText` et `textContent`). Mais si vous manipulez du XML (dans une extension pour Firefox par exemple), `innerHTML` n'est pas supporté !

```
document.getElementById('output').innerHTML = '';
```

Remarquez toutefois que l'élément n'est pas entièrement vide, puisqu'il contient un noeud `#text` vide !

## 4. Mini TP 1

### La manière longue et propre

```
// On crée l'élément conteneur

var mainDiv = document.createElement('div');

    mainDiv.id = 'divTP1';


// On crée tous les nœuds textuels, pour plus de facilité

var textNodes = [

    document.createTextNode('Le '),

    document.createTextNode('World Wide Web Consortium'),

    document.createTextNode(', abrégé par le sigle '),

    document.createTextNode('W3C'),

    document.createTextNode(', est un '),

    document.createTextNode('organisme de standardisation'),

    document.createTextNode(' à but non-lucratif chargé de promouvoir la compatibilité des technologies du '),

    document.createTextNode('World Wide Web'),

    document.createTextNode('.')

];


// On crée les deux <strong> et les deux <a>

var w3cStrong1 = document.createElement('strong');

var w3cStrong2 = document.createElement('strong');


w3cStrong1.appendChild(textNodes[1]);
```

```
w3cStrong2.appendChild(textNodes[3]);

var orgLink = document.createElement('a');
var wwwLink = document.createElement('a');

orgLink.href = 'http://fr.wikipedia.org/wiki/Organisme_de_normalisation';
orgLink.title = 'Organisme de normalisation';
orgLink.appendChild(textNodes[5]);

wwwLink.href = 'http://fr.wikipedia.org/wiki/World_Wide_Web';
wwwLink.title = 'World Wide Web';
wwwLink.appendChild(textNodes[7]);

// On insère le tout dans mainDiv
mainDiv.appendChild(textNodes[0]);
mainDiv.appendChild(w3cStrong1);
mainDiv.appendChild(textNodes[2]);
mainDiv.appendChild(w3cStrong2);
mainDiv.appendChild(textNodes[4]);
mainDiv.appendChild(orgLink);
mainDiv.appendChild(textNodes[6]);
mainDiv.appendChild(wwwLink);
mainDiv.appendChild(textNodes[8]);

// On insère mainDiv dans le #output
document.getElementById('output').appendChild(mainDiv);
```

Pour faire plus court, on aurait pu se passer de définir les noeuds textuels et les définir directement à l'insertion :

```
// On crée l'élément conteneur

var mainDiv = document.createElement('div');

    mainDiv.id = 'divTP1';


// On crée les deux <strong> et les deux <a>

var w3cStrong1 = document.createElement('strong');

var w3cStrong2 = document.createElement('strong');


w3cStrong1.appendChild(document.createTextNode('World Wide Web Consortium'));

w3cStrong2.appendChild(document.createTextNode('W3C'));


var orgLink = document.createElement('a');

var wwwLink = document.createElement('a');


orgLink.href = 'http://fr.wikipedia.org/wiki/Organisme_de_normalisation';

orgLink.title = 'Organisme de normalisation';

orgLink.appendChild(document.createTextNode('organisme de standardisation'));


wwwLink.href = 'http://fr.wikipedia.org/wiki/World_Wide_Web';

wwwLink.title = 'World Wide Web';

wwwLink.appendChild(document.createTextNode('World Wide Web'));


// On insère le tout dans mainDiv

mainDiv.appendChild(document.createTextNode('Le '));

mainDiv.appendChild(w3cStrong1);
```

```

mainDiv.appendChild(document.createTextNode(', abrégé par le sigle '));

mainDiv.appendChild(w3cStrong2);

mainDiv.appendChild(document.createTextNode(', est un '));

mainDiv.appendChild(orgLink);

mainDiv.appendChild(document.createTextNode(' à but non-lucratif chargé de
promouvoir la compatibilité des technologies du '));

mainDiv.appendChild(wwwLink);

mainDiv.appendChild(document.createTextNode('.'));


// On insère mainDiv dans le #output

document.getElementById('output').appendChild(mainDiv);

```

## 5. Mini TP2

```

// On crée l'élément conteneur

var mainDiv = document.createElement('div');

    mainDiv.id = 'divTP2';


// On crée tous les nœuds textuels, pour plus de facilité

var languages = [

    document.createTextNode('JavaScript'),

    document.createTextNode('JScript'),

    document.createTextNode('ActionScript'),

    document.createTextNode('EX4')

];


// On crée le paragraphe

var paragraph    = document.createElement('p');

```

```
var paragraphText = document.createTextNode('Langages basés sur ECMAScript :');

paragraph.appendChild(paragraphText);


// On crée la liste, et on boucle pour ajouter chaque item

var uList = document.createElement('ul'),

    uItem;

for (var i = 0, c=languages.length; i < c; i++) {

    uItem = document.createElement('li');

    uItem.appendChild(languages[i]);

    uList.appendChild(uItem);

}


// On insère le tout dans mainDiv

mainDiv.appendChild(paragraph);

mainDiv.appendChild(uList);


// On insère mainDiv dans le #output

document.getElementById('output').appendChild(mainDiv);
```

## 6. Insertion dans tableau

```
var output = document.getElementById('output'),
    trs;

var trNew = document.createElement('tr');

var thNavi = document.createElement('th');
var tdDevs = document.createElement('td');
var tdPart = document.createElement('td');

thNavi.setAttribute('scope', 'row');
thNavi.appendChild(document.createTextNode('Firefox'));
tdDevs.appendChild(document.createTextNode('Mozilla'));
tdPart.appendChild(document.createTextNode('27,8%'));

trNew.appendChild(thNavi);
trNew.appendChild(tdDevs);
trNew.appendChild(tdPart);

if (output) {
    trs = output.getElementsByTagName('tr');

    if (trs[2]) { // Le <tr> de Chrome
```



```
    trs[2].parentNode.insertBefore(trNew, trs[2]);  
  }  
}
```