Javascript
Partie 3

Javascript

DOM

- Le DOM (Document Object Model) est une interface de programmation (ou API, Application Programming Interface) pour les documents XML et HTML.
- Via le Javascript, le DOM permet d'accéder au code du document ; on va alors pouvoir modifier des éléments du code HTML.
- Contrairement à ce qui a été vu avant, alert() n'est pas vraiment une fonction, mais une méthode qui appartient à l'objet window, qui est implicite (il y a en fait très peu de variables globales.

Javascript

DOM

Les deux lignes suivantes signifient la même chose :

```
<script>
   alert('Hello world !');
   window.alert('Hello world !');
</script>
```

Javascript

DOM

- L'objet document est un sous-objet de window.
- L'objet document possède trois méthodes principales :
  - getElementById(),
  - getElementsByTagName()
  - getElementsByName().

Javascript

DOM

- L'objet document est un sous-objet de window.
- L'objet document possède trois méthodes principales :
  - getElementById(),
  - getElementsByTagName()
  - getElementsByName().

Javascript

DOM

getElementById() :

```
<div id="myDiv"><p>Un peu de texte <a>et un lien</a></p></div>
<script>
var div = document.getElementById('myDiv');
   alert(div); </script>
```

On dit alors que div est un objet de type HTMLDivElement

Javascript

DOM

getElementsByTagName():

(on récupère les éléments sous forme de tableau)

```
<script>
   var divs = document.getElementsByTagName('div');
   for (var i = 0, c = divs.length ; i < c ; i++) {
          alert('Element n° ' + (i + 1) + ' : ' + divs[i]);
   } </script>
```

On parcourt le tableau avec une boucle pour récupérer les éléments

Javascript

DOM

getElementsByName(),

(on récupère les éléments par name dans les formulaires)

On peut aussi utiliser querySelector(), qui renvoie le premier élément trouvé correspondant au sélecteur CSS spécifié, ou querySelectorAll(), qui renvoie tous les éléments (sous forme de tableau) correspondant au sélecteur CSS spécifié entre parenthèses :

```
<div id="menu"><div class="item"><span>Élément 1</span><span>Élément 2</span></div>
   <div class="publicite"><span>Élément 3</span><span>Élément 4</span></div></div>
<div id="contenu"><span>Introduction au contenu de la page...</span></div>
<script>
   var query = document.querySelector('#menu .item span'),
   queryAll = document.querySelectorAll('#menu .item span');
   alert(query.innerHTML); // Affiche : "Élément 1"
   alert(queryAll.length); // Affiche : "2"
   alert(queryAll[0].innerHTML + ' - ' + queryAll[1].innerHTML); </script>
```

Javascript

On peut jouer sur les attributs d'une balise HTML avec l'objet Element et getAttribute() et setAttribute(), permettant par exemple de modifier un lien :

```
    <a id="myLink" href="http://www.un_lien_quelconque.com">Un lien modifié
dynamiquement</a>
    <script>
       var link = document.getElementById('myLink');
       var href = link.getAttribute('href'); // On récupère l'attribut « href »
       alert(href);
       link.setAttribute('href', 'http://blog.crdp-versailles.fr/rimbaud/'); // on édite
    </script>
```

Javascript

On peut jouer sur les attributs d'une balise HTML avec l'objet Element et getAttribute() et setAttribute(), permettant par exemple de modifier un lien :

```
    <a id="myLink" href="http://www.un_lien_quelconque.com">Un lien modifié
dynamiquement</a>
    <script>
       var link = document.getElementById('myLink');
       var href = link.getAttribute('href'); // On récupère l'attribut « href »
       alert(href);
       link.setAttribute('href', 'http://blog.crdp-versailles.fr/rimbaud/'); // on édite
    </script>
```

Javascript

Autre exemple de modification de lien

```
    <a id="myLink" href="http://www.un_lien_quelconque.com">Un lien modifié
dynamiquement</a>
    <script>
       var link = document.getElementById('myLink');
       var href = link.href;
       alert(href);
       link.href = 'http://www.clg-rimbaud-aubergenville.ac-versailles.fr/';
    </script>
```

Javascript

innerHTML permet de récupérer le code HTML enfant d'un élément en texte :

```
<div id="myDiv">
   <p>Un peu de texte <a>et un lien</a></p>
</div>
<script>
   var div = document.getElementById('myDiv');
   alert(div.innerHTML);
</script>
```

Javascript

innerHTML permet de récupérer le code HTML enfant d'un élément en texte :

```
<div id="myDiv">
   <p>Un peu de texte <a>et un lien</a></p>
</div>
<script>
   var div = document.getElementById('myDiv');
   alert(div.innerHTML);
</script>
```

On peut alors définir un nouveau contenu :

```
   document.getElementById('myDiv').innerHTML = '<blockquote>Je mets une citation à la
place du paragraphe</blockquote>';
```

Javascript

innerHTML permet de récupérer le code HTML enfant d'un élément en texte :

```
<div id="myDiv">
   <p>Un peu de texte <a>et un lien</a></p>
</div>
<script>
   var div = document.getElementById('myDiv');
   alert(div.innerHTML);
</script>
```

On peut alors définir un nouveau contenu :

```
   document.getElementById('myDiv').innerHTML = '<blockquote>Je mets une citation à la
place du paragraphe</blockquote>';
```

Javascript

innerHTML permet de récupérer le code HTML enfant d'un élément en texte :

```
<div id="myDiv">
   <p>Un peu de texte <a>et un lien</a></p>
</div>
<script>
   var div = document.getElementById('myDiv');
   alert(div.innerHTML);
</script>
```

Ou encore ajouter un contenu à celui qui est en place (à éviter dans une boucle) :

```
   document.getElementById('myDiv').innerHTML += ' et <strong>une portion mise en emphase</strong>.';
```

Javascript

La propriété parentNode permet d'accéder à l'élément parent d'un élément :

```
<blockquote>
   <p id="myP">Ceci est un paragraphe !</p>
</blockquote>
<script>
   var paragraph = document.getElementById('myP');
   var blockquote = paragraph.parentNode;
</script>
```

Javascript

nodeType et nodeName permettent de vérifier le type et le nom d'un nœud :

```
var paragraph = document.getElementById('myP');
alert(paragraph.nodeType + '\n\n' +
paragraph.nodeName.toLowerCase());
```

Javascript

firstChild et lastChild permettent d'accéder au premier et au dernier élément d'un nœud :

```
<div>
   <p id="myP">Un peu de texte, <a>un lien</a> et <strong>une portion en
emphase</strong></p>
</div>
<script>
   var paragraph = document.getElementById('myP');
   var first = paragraph.firstChild;
   var last = paragraph.lastChild;
   alert(first.nodeName.toLowerCase());
   alert(last.nodeName.toLowerCase());
</script>
```

Javascript

childNodes retourne un tableau contenant la liste des enfants d'un élément.

```
<div>
  <p id="myP">Un peu de texte, <a>un lien</a> et <strong>une portion en
mphase</strong></p>
</div>
<script>
  var paragraph = document.getElementById('myP');
  var first = paragraph.firstChild;
  var next = first.nextSibling;
  alert(next.firstChild.data); // Affiche « un lien »
</script>
```

Javascript

.Créer et insérer des éléments

on crée <a>

```
var newLink = document.createElement('a');
```

On lui affecte des attributs :

```
newLink.id = 'sdz_link';
newLink.href = 'http://blog.crdp-versailles.fr/rimbaud/';
newLink.title = 'Découvrez le blog de la Classe Actu !';
newLink.setAttribute('tabindex', '10');
```

Javascript

.Créer et insérer des éléments

On l'insère dans le document :

```
<div><p id="myP">Un peu de texte <a>et un lien</a></p></div>
<script>
   var newLink = document.createElement('a');
   newLink.id = 'sdz_link';
   newLink.href = 'http://blog.crdp-versailles.fr/rimbaud/';
   newLink.title = 'Découvrez le blog de la Classe Actu !';
   newLink.setAttribute('tabindex', '10');
   document.getElementById('myP').appendChild(newLink); // le nouvel élément
est le dernier enfant dans le paragraphe avec id 'myP'
   var newLinkText = document.createTextNode("Le Tonnerre de Rimbaud");
   newLink.appendChild(newLinkText); // ces deux lignes pour ajouter le texte
</script>
```

Javascript

Cloner, remplacer, supprimer

Pour cloner un élément, on utilise cloneNode(), et on choisit avec (true) ou sans (false) ses enfants et ses attributs.

Pour remplacer un élément par un autre, on utilise replaceChild(), avec deux paramètres, le nouvel élément et l'élément qu'on veut remplacer :

```
<div><p id="myP">Un peu de texte <a>et un lien</a></p></div>
<script>
  var link = document.getElementsByTagName('a')[0];
  var newLabel= document.createTextNode('et un hyperlien');
  link.replaceChild(newLabel, link.firstChild);
</script>
```

Javascript

Pour supprimer un élément, on utilise removeChild(), avec le nœud enfant à retirer :

```
var link = document.getElementsByTagName('a')[0];
link.parentNode.removeChild(link);
```

Pour vérifier la présence d'éléments enfant, on utilise hasChildNodes() :

```
<div><p id="myP">Un peu de texte <a>et un lien</a></p></div>
<script>
   var paragraph = document.getElementsByTagName('p')[0];
   alert(paragraph.hasChildNodes()); // Affiche true
</script>
```

Javascript

Pour insérer un élément avant un autre, on utilise insertBefore() :

```
<p id="myP">Un peu de texte <a>et un lien</a></p>
<script>
   var paragraph = document.getElementsByTagName('p')[0];
   var emphasis = document.createElement('em'),
   emphasisText = document.createTextNode(' en emphase légère ');
   emphasis.appendChild(emphasisText);
   paragraph.insertBefore(emphasis, paragraph.lastChild);
</script>
```

Javascript

Exercice 1

Passer ce code HTML en script

```
    <div id="divTP1">
    Le <strong>World Wide Web Consortium</strong>, abrégé par le sigle <strong>W3C</strong>, est un
<a href="http://fr.wikipedia.org/wiki/Organisme_de_normalisation" title="Organisme de
normalisation">organisme de standardisation</a> à but non-lucratif chargé de promouvoir la
compatibilité des technologies du <a href="http://fr.wikipedia.org/wiki/World_Wide_Web" title="World
Wide Web">World Wide Web</a>.
    </div>
```

Javascript

Exercice 2

Passer ce code HTML en script en utilisant une boucle for

```
<div id="divTP2">
<p>Langages basés sur ECMAScript :</p>
<ul>       <li>JavaScript</li>
           <li>JScript</li>
           <li>ActionScript</li>
           <li>EX4</li>            </ul>          </div>
```

Javascript

Exercice 3

Passer ce code HTML en script

```
<div id="divTP4">
<form enctype="multipart/form-data" method="post" action="upload.php">
<fieldset>
<legend>Uploader une image</legend>
<div style="text-align: center">
<label for="inputUpload">Image à uploader :</label>
<input type="file" name="inputUpload" id="inputUpload" />
<br /><br />
<input type="submit" value="Envoyer" />
</div>
</fieldset></form></div>
```

Javascript

Fin de la présentation

Partie 3