

Mise en pratique

[Mon premier script](#)

[Ex. 1 :](#)

[Ex. 2 :](#)

[Le DOM](#)

[Changements d'attributs et méthode `getElementsByTagName`](#)

[Ajouts d'éléments](#)

[Le gestionnaire d'événements](#)

[Preloader d'image](#)

[Lightbox](#)

[Contrôle de formulaire](#)

[L'objet XMLHttpRequest](#)

[Contrôle de saisie de formulaire](#)

[FormDatas & JSON](#)

Mon premier script

Ex. 1 :

Pour cet exercice :

Soit un élève 'toto' qui a obtenu une série de notes : créer une fonction pour calculer la moyenne des notes et faire afficher cette moyenne dans la console en utilisant la méthode 'console.log()'.

Le nombre d'éléments du tableau 'toto' s'écrit : `toto.length` // Equivalent de la fonction `count($toto)` de PHP

Effectuez vos tests avec les jeux de données suivants :

- `toto = [15,14,16,12,18];`
- `toto = [150,14,16,12,18];`
- `toto = [15,14,160,12,18];`
- `toto = ['15',14,16,12,18];`
- `toto = [15,14,'16',12,18];`
- `toto = [null,14,16,12,18];`
- `toto = [15,14,null,12,18];`
- `toto = [false,14,16,12,18];`
- `toto = [15,14,false,12,18];`
- `toto = [undefined,14,16,12,18];`
- `toto = [15,14,undefined,12,18];`

Ex. 2 :

Pour cet exercice :

Soit ce même élève auquel on souhaite ajouter autant de nouvelles notes que voulu : pouvoir renseigner les nouvelles notes via une boîte de dialogue, stocker ces notes et actualiser la moyenne affichée.

Effectuez vos tests avec les jeux de données précédents.

Bonus : Faire une variante en récursivité

Le DOM

Changements d'attributs et méthode `getElementsByTagName`

Soit le code html suivant :

```
<ul onClick="change_liste()">
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
  <li>Item 4</li>
  <li>Item 5</li>
  <li>Item 6</li>
  <li>Item 7</li>
  <li>Item 8</li>
</ul>

<script type="text/javascript">
  //
  function change_liste() {
    // Code à écrire
  }
  //]]&gt;
&lt;/script&gt;</pre></div><div data-bbox="114 535 885 570" data-label="Text"><p>Complétez uniquement la fonction <code>change_liste()</code> avec votre code Javascript. Cette fonction sera appelé au clic sur la liste.</p></div><div data-bbox="144 570 843 659" data-label="List-Group"><ol><li>1. Récupérez l'objet de liste à l'aide d'une méthode <b><code>getElementsByTagName</code></b>.</li><li>2. Récupérez la collection d'items de liste à l'aide d'une méthode.</li><li>3. Changez la valeur de l'attribut de type de la liste en <i>"circle"</i>.</li><li>4. Pour chaque élément de liste, changez le texte en <i>"blabla"</i>, et affectez à l'attribut <i>onclick</i> un appel à une boîte d'alerte avec un petit texte.</li></ol></div><div data-bbox="114 677 886 729" data-label="Text"><p>Pour cet exercice, vous aurez besoin d'utiliser la méthode <b><code>setAttribute</code></b> qui s'applique à un élément sélectionné. Dans le cas du point 4, le premier paramètre de <code>setAttribute</code> est <i>"onClick"</i></p></div>
```

Ajouts d'éléments

Soit le code html suivant :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, user-scalable=no, initial-scale=1.0" />
    <title>Le Javascript - Mise en pratique</title>

    <script type="text/javascript">
      //
        function ajout() {
          // Code à écrire
        }

        function ajouter() {
          // Code à écrire
        }

        function supprimer() {
          // Code à écrire
        }
      //]]&gt;
    &lt;/script&gt;
  &lt;/head&gt;
  &lt;body onClick="ajout()"&gt;
    &lt;!-- &lt;input id="course" type="text" value="" /&gt;&lt;input onClick="ajouter()" type="button"
value="OK" /&gt;&lt;input onClick="supprimer()" type="button" value="Supprimer le dernier élément" /&gt; --&gt;
    &lt;ul id="listecommissions"&gt;
      &lt;li&gt;1kg de farine&lt;/li&gt;
      &lt;li&gt;un pack de lait&lt;/li&gt;
    &lt;/ul&gt;
  &lt;/body&gt;
&lt;/html&gt;</pre></div><div data-bbox="114 679 886 713" data-label="Text"><p>Dans un premier temps, complétez uniquement la fonction 'ajout()' avec votre code Javascript. Cette fonction sera appelé au clic sur la page.</p></div><div data-bbox="144 714 606 766" data-label="List-Group"><ol><li>1. Sélectionnez la liste d'identifiant 'listecommissions'</li><li>2. Créez un nouvel item de liste et affectez lui un texte</li><li>3. Ajoutez ce nouvel élément à la liste</li></ol></div>
```

Dans un second temps, complétez les fonctions 'ajouter()' et 'supprimer()' avec votre code Javascript. Ces fonctions seront appelées au clic sur les boutons.

1. Commentez l'attribut 'onClick' de la balise 'body'
2. Décommentez les balises 'input'
3. Copier le code de la fonction 'ajout()' dans la fonction 'ajouter()' et modifier le afin de faire en sorte que le texte ajouté à la liste soit saisi à partir du champ texte (nous n'autorisons pas d'ajout d'élément vide)
4. Complétez la fonction 'supprimer()' afin de faire en sorte que le dernier élément de la cible 'data-target' soit supprimé lorsque l'on clique sur le bouton
5. Modifier les fonctions 'ajouter()' et 'supprimer()' afin de pouvoir aussi supprimer un élément lorsque l'on clique sur ce dernier

Le gestionnaire d'événements

Preloader d'image

1. Récupérez le fichier "*03.01_-Le_gestionnaire_d_evenements_-_Preloader.html*" présent sur le serveur.
2. Commentez chaque action du script déjà écrit pour voir ce qu'il se passe avec le loader puis enlevez les commentaires
3. La fonction `setTimeout` nous permettra de simuler un chargement long via le web en temporisant notre action. Utilisez la pour ajouter votre code javascript.
4. Vous devrez faire en sorte que l'image de basse qualité soit remplacée par l'image de haute définition présente dans le dossier "*assets/uploads/photo.jpg*"
5. Modifier le script pour que toutes ces actions s'effectuent après le chargement complet de la page.

Lightbox

1. Récupérez le fichier "*03.02_-Le_gestionnaire_d_evenements_-_Lightbox*" présent sur le serveur.
2. Ajoutez un événement sur le clic de l'image préchargée. Cet événement doit créer un cadre :
 - qui occupe 100% de la largeur et 100% de la hauteur
 - qui a une opacité de 50% (en noir ou en blanc)
 - qui affiche en son centre, dans un cadre de 80% de la largeur, l'image sur laquelle on a cliqué mais dans sa version grand format "*assets/uploads/photo.jpg*"
3. Ajoutez un événement sur le cadre nouvellement créé pour qu'au clic, cela le supprime de l'affichage.

Contrôle de formulaire

1. Créez un formulaire HTML avec des champs de saisie *text*, *number*, *email*, *tel*, des liste déroulantes, des choix multiples.
2. Ajoutez pour chaque champs de saisie un événement de *focus* qui sélectionne le texte présent dans le champs.
3. Ajoutez pour chaque élément du formulaire un événement qui interviendra à la perte du *focus* et qui affichera dans une boîte de dialogue la valeur de l'élément.
4. Ajoutez une classe "*required*" sur certains champs du formulaire puis ajoutez un événement qui interviendra à la soumission du formulaire.
Cet événement devra vérifier que tous les champs marqués "*required*" aient bien une valeur de saisie. Si ce n'est pas le cas, l'événement devra afficher un message d'alerte et bloquer la soumission du formulaire.

L'objet XMLHttpRequest

Contrôle de saisie de formulaire

Reprenez l'exercice précédent et ajoutez les actions suivantes :

1. Interceptez l'événement de soumission du formulaire pour que la page ne se recharge pas.
A la place, utilisez l'objet XMLHttpRequest pour appeler un script PHP qui vérifiera l'intégrité des données saisies dans le formulaire.
Si cette saisie correspond aux valeurs attendues, la réponse renverra un message disant que le formulaire a bien été envoyé, sinon la réponse renverra un message indiquant les champs à modifier.
2. Récupérez la réponse de l'objet XMLHttpRequest pour afficher le message dans la page, au dessus du formulaire en format lightbox.

FormDatas & JSON

Reprenez l'exercice précédent et complétez le pour finaliser un script d'envoi d'emails en HTML5, CSS3, AJAX (Javascript, PHP et format JSON)

Cette application devra gérer les champs obligatoires, les types de données et les valeurs passés, avant d'envoyer l'e-mail.

Les contrôles devront être effectués en Javascript ET en PHP:

- à la soumission du formulaire pour tous les champs,
- à la perte de focus sur le champ en question

Si toutes les contraintes du formulaires sont respectées, votre script pourra envoyer l'e-mail et afficher un message de réussite dans la page. Sinon, l'application devra afficher le(s) message(s) d'erreur(s).