
JavaScript - Travaux Pratiques 1

Pour JavaScript, il est fortement conseillé d'utiliser **emacs** sous **Linux** pour créer vos fichiers. Chaque fichier doit avoir l'extension **.js**. Pour associer du code JavaScript placé dans un fichier externe à une page HTML, il suffit d'utiliser l'attribut **type** avec l'attribut **src** dans un élément **script**. Pour exécuter le code après le chargement de la page, on utilisera :

```
<body>
  <!-- content here -->
  <script type="text/javascript" src="monCode.js"> </script> />
</body>
```

Pour identifier et corriger les erreurs (bugs), il faut absolument utiliser la console (fenêtre) d'erreurs. Pour Firefox, vous la trouverez sous

Tools - Error Console

Pour recharger une page XHTML, y compris le chargement du code JavaScript qui est associé à la page, vous devez :

Appuyez sur la touche CTRL tout en cliquant sur l'icône de rechargement

On utilisera DOM (level 0) pour accéder aux éléments d'une page HTML et interagir avec eux. Cela signifie que typiquement, on récupère une variable JavaScript référant un élément HTML à l'aide de la fonction `getElementById` et la valeur de son attribut `id`. Par exemple :

```
var but1 = document.getElementById("but1");
```

A l'aide de cette variable, on peut modifier dynamiquement le style comme par exemple :

```
but1.style.color="blue";
```

On peut lui associer des écouteurs comme par exemple :

```
function but1Listener() { alert("but1 cliqué"); }
but1.onclick=but1Listener;
```

Par ailleurs, les sites qui peuvent vous être utiles sont :

- Tout JavaScript sur <http://www.toutjavascript.com/main/index.php3>
 - Javascriptfr <http://www.javascriptfr.com>
 - w3school-Javascript sur <http://www.w3schools.com/js/default.asp>
-

1 Un premier contrôle de formulaire

1. Écrire une page XHTML comportant un mini-formulaire composé d'un champ de saisie et d'un simple bouton. Cette page doit passer la validation W3C.
2. Créer un fichier `exo1.js` comportant uniquement l'instruction `alert('coucou')`. Associer ce fichier à la page XHTML et rechargez à nouveau votre page XHTML. Que se passe-t-il ? Vérifier que votre page passe toujours la validation W3C et qu'il n'y a pas d'erreurs JavaScript dans la console d'erreurs.
3. On remplace l'instruction précédente par :

```
function controler() {  
    var zoneSaisie = document.getElementById("zoneSaisie");  
    alert("La Zone de saisie contient : " + zoneSaisie.value);  
}
```

Décrivez le code de cette fonction. Il faudrait maintenant associer ce code de contrôle à l'événement `click` sur le bouton du formulaire. Écrivez le code correspondant (pensez aux diapositives de cours pour vous aider).

4. Modifier la fonction de contrôle pour afficher un message d'erreur lorsque le champ est vide, et afficher le contenu du champ lorsqu'il ne l'est pas.
5. La fonction `eval`, prenant en paramètre une chaîne de caractères représentant une expression, permet d'évaluer cette dernière. Par exemple :

```
eval("3+4"); // retourne la valeur 7  
eval("bon"+"jour"); //retourne bonjour
```

Ajouter (concaténez) le résultat de l'évaluation de l'expression donnée par l'utilisateur dans le champ de saisie à votre affichage. Cela donne une page HTML contenant un formulaire permettant d'exécuter "on line" de petites opérations.

2 Valeur Absolue

1. Écrivez une page HTML contenant un formulaire permettant d'obtenir la valeur absolue d'un nombre donné par l'utilisateur. Le formulaire comprendra un premier champ de saisie, puis un bouton et enfin un deuxième champ de saisie.
2. Chaque fois que l'utilisateur clique sur le bouton, une vérification de la valeur placée dans le premier champ de saisie est effectuée. Si le champ est vide ou ne contient pas de valeur numérique (pensez à la fonction `isNaN`), alors un message d'alerte est affichée. Sinon, la valeur absolue de cette valeur est placés directement dans le second champ de saisie.
3. Placer aussi le code JavaScript permettant de rendre en lecture seule le second champ de saisie (attribut `readOnly`).