

TD 01 : HTML – JavaScript

Nous travaillerons avec Firefox. Avant toute chose ajouter le plugin firebug à firefox. Firebug permet de déboguer le javascript, vous avez un console qui affiche les erreurs, une vue du fichier html, une vue des scripts et un explorateur du DOM.

Pour une référence rapide à la documentation vous pouvez utiliser le site Web <http://www.w3schools.com/>, contrairement à ce que son nom indique ce site n'a aucun lien officiel avec le W3C. Ils se contentent d'utiliser la ressemblance avec le nom pour attirer des visiteurs (et faire de l'argent avec de la pub). Mais le site est relativement bien fait. Les pages du W3C sont aussi très bien et beaucoup plus complètes.

Exercice 1. Alert

1.1 Écrivez une page HTML, affichant au chargement une boîte de message (fonction alert). Testez que cela fonctionne.

1.2 Modifiez cette page pour afficher la date du jour (objet Date), le nom du navigateur (objet navigator), sa version, sa plateforme d'exécution et la taille de l'écran. Pour cela vous écrirez dans une balise script dans le header, le message à afficher sera stocké dans une variable globale (qui sera donc le paramètre de alert). Testez.

1.3 Créez deux tableaux (objet Array) en variable globale du script pour faire la correspondance entre les jours (0,1,...6) de la semaine (attention les jours de la semaine commence le dimanche pour un anglo-saxon) et son nom ; pareil pour le numéro du mois (0,1,...,11) et le nom ('janvier', ...). Utilisez ces tableaux pour afficher la date en toute lettre plutôt qu'avec des numéros. À la fin, vous devez avoir un message du style

```
Ce navigateur est Netscape, version 5.0 (X11; fr).
```

```
Le système d'exploitation est Linux i686.
```

```
La date brute est Fri Oct 16 2009 11:04:31 GMT+0200 (CEST).
```

```
Nous sommes le Vendredi 16 octobre 2009 et il est 11H4
```

```
La largeur de l'écran est 1920.
```

```
La hauteur de l'écran est 1200.
```

1.4 Modifiez la page pour que le message s'affiche directement dans la page, (fonction document.write dans un balise script dans le body, de préférence).

Exercice 2. Manipuler le DOM

2.1 Faites un page contenant un formulaire avec deux boutons radio. Les deux choix seront visible et invisible. Un exemple de formulaire avec des boutons radio se trouve [ICI <http://www.w3.org/TR/html4/interact/forms.html#h-17.1>]

2.2 La page contiendra trois [paragraphes <http://www.w3.org/TR/html4/struct/text.html#h-9.3.1>] se suivant, chacun rempli de texte quelconque. Donnez un id à chaque paragraphe (voir un exemple [ICI <http://www.w3.org/TR/html4/struct/global.html#edef-id>]).

2.3 Faites un script dans le section head de la page, où vous définissez une fonction pour montrer() et une fonction cacher(). Dans ces fonctions, mettez une variable locale initialisé avec document.getElementById("id_du_deuxieme_paragraphe") pour initialiser la variable (changez la chaîne par votre bon id ...). Ensuite dans la fonction javascript passez l'attribut css visibility de ce paragraphe à la bonne valeur suivant la fonction (voir la norme de cet attribut [ICI <http://www.w3.org/TR/CSS/visufx.html#propdef-visibility>]). Il faut donc faire en sorte que le deuxième paragraphe disparaisse ou apparaisse. Pour chacun des deux radio bouton, interceptez le click de l'utilisateur par un évènement javascript et appelez une des deux fonctions (voir l'attribut [onclick <http://www.w3.org/TR/html4/interact/scripts.html#edef-onclick>]).

Exercice 3. Validation de formulaire

3.1 Faites un page HTML avec un formulaire demandant un non, un prénom et un courriel. Pour chaque champ, utilisez un [LABEL <http://www.w3.org/TR/html4/interact/forms.html#h-17.9.1>] pour marquez l'intitulé (Nom, Prénom, etc), ensuite placez le [INPUT <http://www.w3.org/TR/html4/interact/forms.html#h-17.4>] correspondant, et ensuite placez un autre label (avec un attribut id) sans contenu (on affichera les erreurs ici). N'oubliez pas de remplir l'attribut "for" des labels. N'oubliez pas de nommer votre formulaire et vos inputs avec l'attribut [NAME <http://www.w3.org/TR/html4/interact/forms.html#edef-name-INPUT>] (mais pas les labels, qui sont eux nommé avec un id). L'envoi du formulaire doit être fait par un input de type button (et pas un type submit). Il est possible de vérifier le formulaire avec l'envoi en utiliser l'évènement onclick du bouton.

3.2 Créez une fonction verifier dans l'entête de la page pour vérifier votre formulaire. Vous pouvez accéder à l'objet correspondant au formulaire avec une expression du type document.forms["name du formulaire"] (et ensuite pour les inputs avec document.forms["name du formulaire"].elements["name de l'input"]). L'attribut value de l'objet permet de connaître la valeur saisie.

3.3 Testez si l'utilisateur a effectivement rempli le champ nom avec l'attribut value.length. Si ce n'est pas le cas, affichez une erreur dans le label qui est initialement vide, vous pouvez retrouver le label avec un document.getElementById("ID du label") et ensuite changer le contenu avec l'attribut textContent. Testez aussi le champ courriel est rempli, si il ne l'est pas affichez une erreur, si il est testez si l'email contient le caractère @ et affichez une erreur si ce n'est pas le cas. Pour savoir si un caractère est contenu dans une chaîne il faut utiliser la méthode match qui prend en paramètre un objet RegExp (voir [ICI http://www.w3schools.com/jsref/jsref_match.asp] et [LÀ http://www.w3schools.com/jsref/jsref_obj_regexp.asp])

3.4 Si tout est ok, lancez la soumission du formulaire avec la méthode submit() de l'objet représentant le formulaire.

Exercice 4. Générer une palette de couleur

À chaque étape testez que le tout fonctionne, utilisez la console de firebug pour détecter les erreurs de syntaxe et visualiser le résultat du script. Attention le html généré par le script n'apparaît pas dans le code source de la page, utilisez l'icône d'inspection de firebug (c'est la deuxième icône) pour vérifier du html qui vous avez généré dans la page.

4.1 Faites une page qui contient une fonction qui génère un tableau (utilisez `document.write`). Le nombre de lignes et de colonnes du tableau généré sera réglé par deux paramètres de la fonction, les cases du tableau contiendront trois tirets. La fonction sera appelée dans une balise `script` dans la page.

4.2 Mettez ensuite dans la page un paragraphe avec du texte, donnez un `id` à votre paragraphe.

4.3 Faites une fonction (mettez toujours vos fonction dans le header) qui change la couleur de fond du paragraphe, avec une couleur passée en paramètre, lancez la fonction avec une couleur quelconque dans le `onLoad` du `body`. Normalement le paragraphe se colore tout seul au chargement de la page avec la couleur que vous avez choisi.

4.4 Transformez la fonction qui génère le tableau pour que la couleur de fond de chaque case ait une couleur aléatoire, pour cela utilisez `Math.random`, `Math.floor` et `.toString` sur le nombre pour obtenir de l'hexadécimal. Faites une fonction auxiliaire pour toujours obtenir un nombre à deux chiffres (utiliser `.length` pour connaître la taille de la chaîne). N'oubliez pas de tester à chaque étape sinon vous ne vous en sortirez pas.

4.4 Utilisez un évènement sur toutes les cases du tableau pour appeler la fonction changeant la couleur du paragraphe avec la couleur de fond de la case.

Normalement vous obtenez une page avec une palette de couleurs aléatoires et quand on clique sur une case, la couleur de fond du paragraphe prend cette couleur.

