

Exercices – Cours JavaScript

Manipuler le DOM avec JavaScript

Bientôt vous serez des spécialistes du DOM. Mais il faut commencer à quelque part 😊

Prérequis : afin d'exécuter ces consignes vous aurez besoin des fichiers HTML dom1.html à dom8.html dans votre répertoire de travail.

Exercice 1 : Modifier le style

1. À l'aide du fichier dom1.html fourni, rédiger une méthode en JS, `modifierStyle()`, qui modifie le style du texte du paragraphe lorsqu'on appuie sur le bouton Modifier Style. La police de caractères doit être modifiée à 24 points, la police doit être "Comic Sans MS" et la couleur verte.

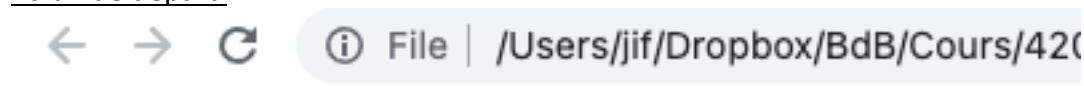
Le code JS doit être fait dans le fichier dom1.js externe qui est rattaché au bouton dans votre html.

Indice :

Comme appris en classe : vous devez dans un premier temps créer une constante qui contiendra un pointeur vers l'objet paragraphe que vous aurez localisé `document.getElementById ()` par son identifiant dans le code HTML (`id="text"`).

```
Nom_constante.style.fontSize = "24pt";  
fontFamily = "Comic Sans MS";  
color = "green";
```

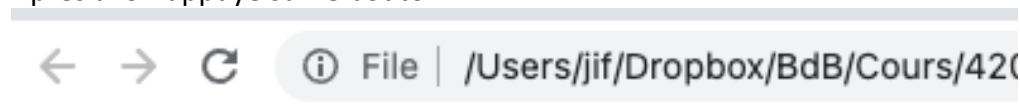
Écran de départ:



Exercice JavaScript - Changement de style

Modifier Style

Après avoir appuyé sur le bouton :



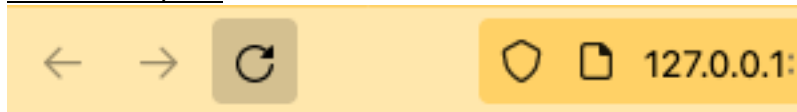
Exercice JavaScript - Changement de style

Modifier Style

Exercice 2 : Appel des fonctions des objets window et document

2. À l'aide du fichier html de départ dom2.html, écrivez un programme JS (dom2.js) qui contient une méthode (maintenant en JS on appellera cela une fonction car elle n'est pas rattachée à un objet) qui additionne deux paramètres (essai avec 5 et 6) et qui affiche le résultat dans une alerte (window.alert) et ensuite dans la page courante (document.write), attention ceci effacera le document HTML à l'écran, cette fonction est peu utilisée. Ensuite, modifier le bouton « Imprimer la page » afin qu'il appelle (window.print).

Écran de départ:



Utilisation de l'API window

Imprimer la page Additionner

Exercice 3 : Modifier le contenu d'une balise avec inner.HTML

3. À l'aide du fichier de départ dom3.html modifier le texte du paragraphe id="demo", afin que l'affichage soit en gras et souligné.

Résultat attendu :



Bonjour à vous tous!

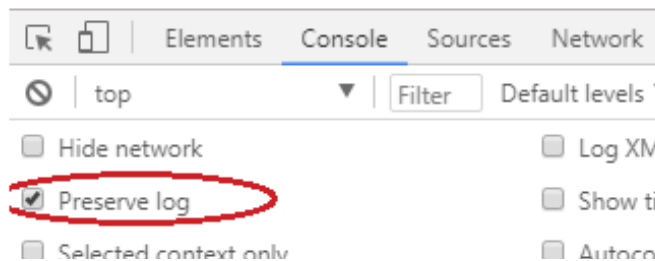
Exercice 4 : Lisez le contenu d'un formulaire et utilisez `console.log()`

4. À l'aide du fichier de départ `dom4.html`, écrivez une fonction JavaScript `getFormValue` pour obtenir les valeurs de nom et prénom du formulaire "form1" et qui les affiche à la console (`console.log`). Lorsque vous faites un `getElementById` d'un formulaire, vous recevrez un objet et vous pouvez accéder au contenu des valeurs en accédant la propriété `elements[i].value` du formulaire. Utilisez un boucle `for` indice de longueur `formulaire.length`.

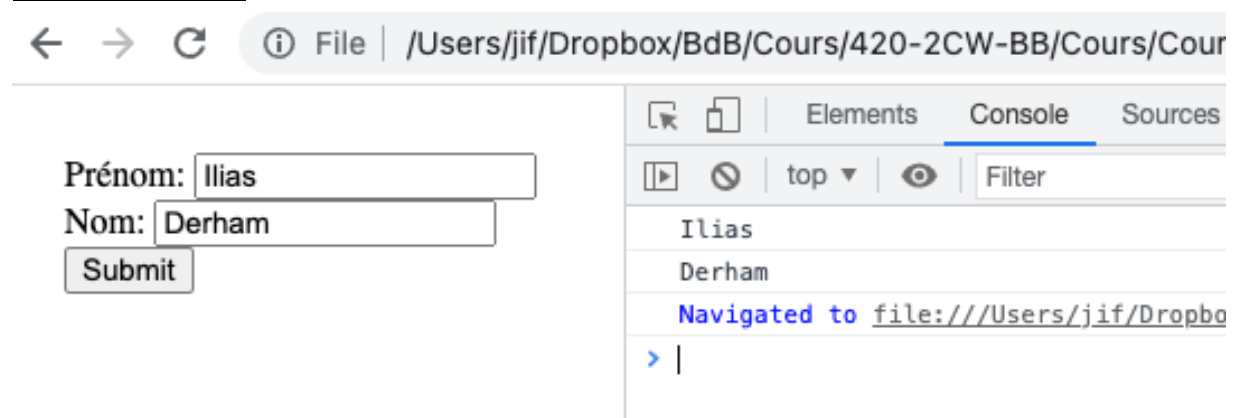
Si vous utilisez Chrome ou Edge gardez votre log de la console à l'aide des instructions suivantes

1. Open Chrome.
2. Hit **F12**
3. Click on 'Console Settings'.
4. Check 'Preserve log'.
5. And you are done !

Following pictorial view to make things easier for you.



Résultat attendu :



Exercice 5 : Insérez une rangée dans un élément table.

5. À l'aide du fichier de départ dom5.html, écrivez une fonction JavaScript insertRangee() pour ajouter des lignes à la table table1, les nouvelles cellules doivent contenir Cellule1 pour la première et Cellule2 pour la seconde. Indice : les fonctions à utiliser sont : insertRow() et insertCell().

rangée1 cellule1	rangée1 cellule2
rangée2 cellule1	rangée2 cellule2

Insérer rangée

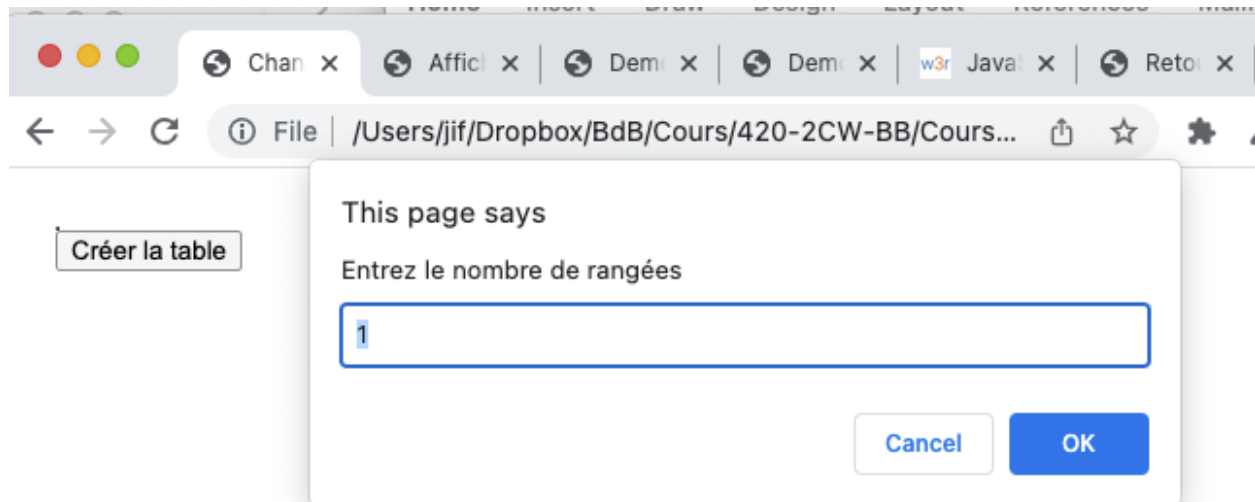
Résultat attendu : après avoir appuyé sur le bouton :

rangée1 cellule1	rangée1 cellule2
rangée2 cellule1	rangée2 cellule2
Cellule 1	Cellule 2

Insert rangée

Exercice 6 : Créer du HTML dynamique soit un élément table.

6. À l'aide du fichier de départ dom6.html, écrivez une fonction JavaScript qui demande de saisir un nombre de colonnes/rangées (window.prompt) et créer une table de cette dimension et mettre dans les cellules #rangée et #colonne une chaîne pour identifier la cellule.



Résultat attendu, après avoir appuyé sur le bouton et saisi les valeurs 6 et 4 :

Rangée: 0 Colonne: 0	Rangée: 0 Colonne: 1	Rangée: 0 Colonne: 2	Rangée: 0 Colonne: 3
Rangée: 1 Colonne: 0	Rangée: 1 Colonne: 1	Rangée: 1 Colonne: 2	Rangée: 1 Colonne: 3
Rangée: 2 Colonne: 0	Rangée: 2 Colonne: 1	Rangée: 2 Colonne: 2	Rangée: 2 Colonne: 3
Rangée: 3 Colonne: 0	Rangée: 3 Colonne: 1	Rangée: 3 Colonne: 2	Rangée: 3 Colonne: 3
Rangée: 4 Colonne: 0	Rangée: 4 Colonne: 1	Rangée: 4 Colonne: 2	Rangée: 4 Colonne: 3
Rangée: 5 Colonne: 0	Rangée: 5 Colonne: 1	Rangée: 5 Colonne: 2	Rangée: 5 Colonne: 3

Créer la table

Exercice 7 : Lire et écrire les valeurs d'un formulaire.

7. À l'aide du fichier de départ dom7.html, écrivez un programme JavaScript, volumeSphere(), pour calculer le volume d'une sphère et afficher le résultat.

Saisissez le rayon et obtenez le volume de la sphère.

Rayon

Volume

Exercice 8 : Lire et écrire les valeurs d'un formulaire.

8. À l'aide du fichier de départ dom8.html, écrivez un programme JS afficheImageAleatoire qui fera afficher une image aléatoire lorsque l'utilisateur clique sur le bouton et qui provient de l'internet, la liste des images est la suivante:

"http://farm4.staticflickr.com/3691/11268502654_f28f05966c_m.jpg", largeur : "240", hauteur : "160"

"http://farm1.staticflickr.com/33/45336904_1aef569b30_n.jpg", largeur : " 320", hauteur : "195"

"http://farm6.staticflickr.com/5211/5384592886_80a512e2c9.jpg", largeur : "500", hauteur : "343"

