



JavaScript & Node.js

Exercices

1 DOM & la gestion d'événements

Veillez réaliser une application web permettant d'afficher un compteur de clic sur votre page web.

Affichez le message suivant sur votre page web :

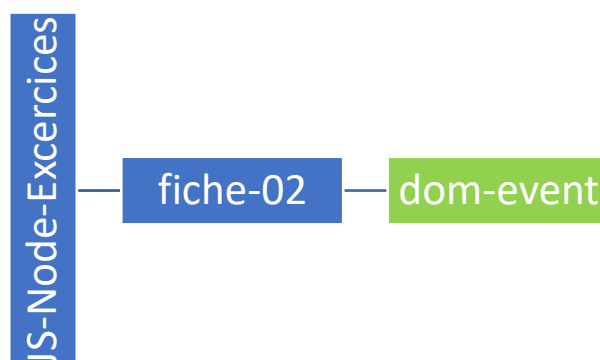
- Au 5^{ème} clic : « Bravo, bel échauffement ! »
- Au 10^{ème} clic : « Vous êtes passé maître en l'art du clic ! »

N'oubliez pas, bien sûr, d'afficher le compteur de clic sur votre page.



- Vous pouvez utiliser la propriété **.innerHTML** ou **innerText** d'un paragraphe **<p>** ou d'un **** pour afficher votre compteur.
- Idem pour votre message.
- Vous pouvez ajouter un gestionnaire d'événement sur l'objet **window** de votre navigateur.

Le code de votre application web doit se retrouver dans ce dossier (en vert) de votre repository local et de votre web repository (**JS-Node-Exercices**).



2 Formulaire & HTML dynamique

Vous allez réaliser une application web permettant de générer des tables sur base d'un formulaire.

Créez un formulaire permettant d'introduire un nombre de lignes, un nombre de colonne, et une chaîne de base. Utilisez Bootstrap pour formater votre application web.



JavaScript & Node.js

Exercices

Voilà à quoi pourrait ressembler votre formulaire :

Number of lines

Enter number of lines

Number of columns

Enter number of columns

Initial string

Init string

Create table

Veuillez valider chacun des champs du formulaire lors du clic sur le bouton.

Si tous les champs sont validés, veuillez générer et afficher une table HTML. Voici un exemple du résultat :

Number of lines

5

Number of columns

3

Initial string

CELL

Create table

CELL[0][0]	CELL[0][1]	CELL[0][2]
CELL[1][0]	CELL[1][1]	CELL[1][2]
CELL[2][0]	CELL[2][1]	CELL[2][2]
CELL[3][0]	CELL[3][1]	CELL[3][2]
CELL[4][0]	CELL[4][1]	CELL[4][2]

Afin de réaliser cet exercice nous vous proposons ces contraintes d'implémentation :

- Créez une 1^{ère} fonction qui retourne un **Array** à deux dimensions avec :
 - o comme valeur pour chaque élément : chaîne de base + [numéro de ligne] + [numéro de colonne],
 - o sur base de 3 arguments : le nombre de lignes, de colonnes, et la chaîne de base à afficher dans chaque élément du tableau.
- Créez une deuxième fonction qui renvoie, sous forme de string, une table HTML basée sur l'**Array** créé par la 1^{ère} fonction.



JavaScript & Node.js

Exercices

- Appelez ces deux fonctions afin d'afficher la table de manière dynamique au sein d'une div.



Vous pouvez utiliser la propriété `.innerHTML` de votre div pour afficher dynamiquement la table.

Le code de votre application web doit se retrouver dans ce dossier (en vert) de votre repository local et de votre web repository (**JS-Node-Exercices**).

