

Jour 1 : Hello JS

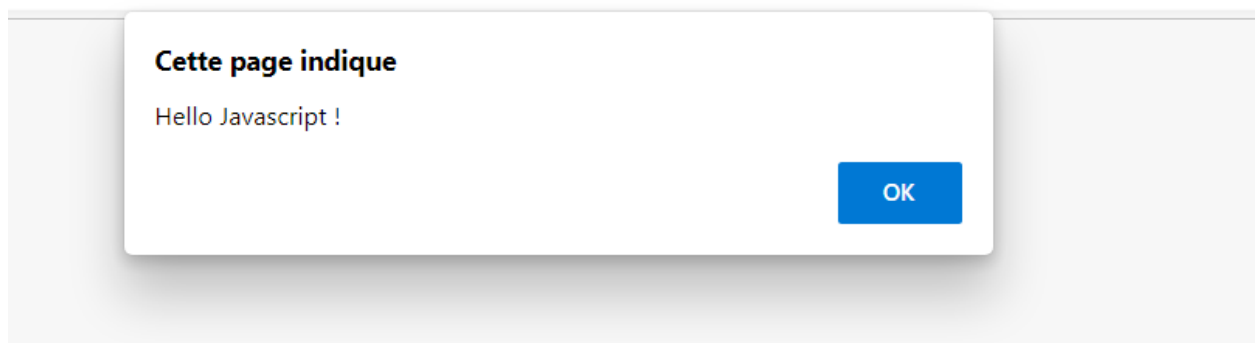
Que la force soit avec toi ...

Introduction du sujet

Créez sur GitHub un répertoire nommé **"runtrackJs"**. Dans ce répertoire, créez un dossier **"jour01"**. Pour chaque job, créez un dossier **"jobXX"** où **XX** est le numéro du job. N'oubliez pas d'envoyer vos modifications dès qu'un job est avancé ou terminé et mettez des commentaires explicites lors de vos commits.

Job 01

Créez un fichier **index.html** contenant les balises HTML de base (doctype, HTML, head, body). Dans la balise head, ajoutez une ligne de code JavaScript permettant d'afficher un **pop up** qui contient le texte **"Hello Javascript!"**.





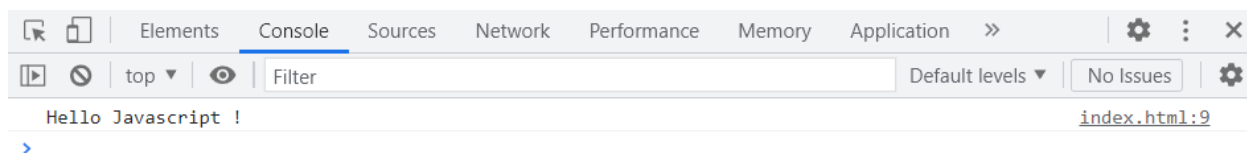
Job 02

Récupérez une copie de votre fichier **index.html**. Créez un fichier **script.js**. Modifiez vos fichiers de sorte que l'affichage du pop up se fasse maintenant dans le fichier `script.js`.

Job 03

Récupérez une copie de vos fichiers **index.html** et **script.js**. Modifier l'affichage de sorte que le message ne soit plus écrit dans un popup, mais dans la **console web**. Assurez-vous que tout est bien fonctionnel.

Résultat attendu :



Job 04

À présent que vous savez comment inclure du JavaScript et que vous maîtrisez la console web, vous allez pouvoir explorer davantage la syntaxe, la grammaire et le lexique du langage JavaScript.

Pour l'ensemble des exercices suivants, vous devez rendre un fichier **script.js** contenant le rendu de l'exercice et un fichier **index.html** qui l'inclut.

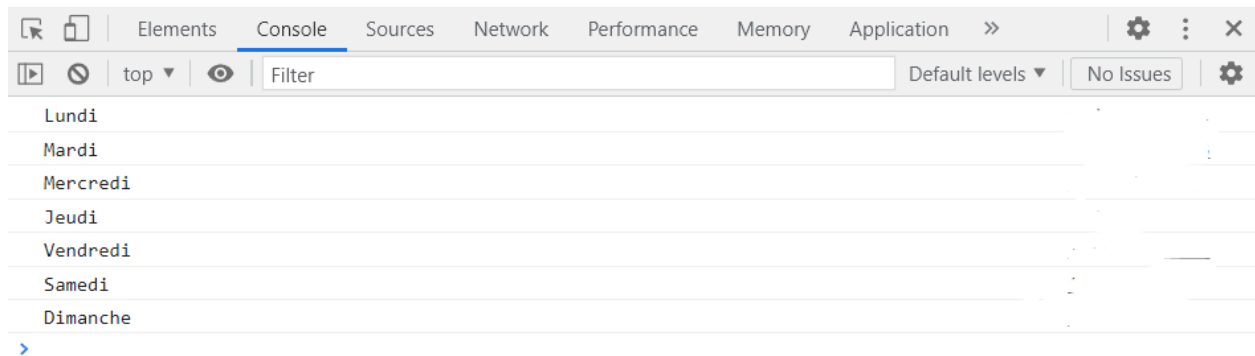
Déclarez une fonction "**bisextile**" qui prend en paramètre une variable "**année**". Si l'année est bissextile, la fonction retourne **true**, sinon elle retourne **false**.



Job 05

Créez une fonction “**afficherJoursSemaines**”. Cette fonction ne prend pas de paramètre. Créez un tableau de strings “**joursSemaines**” qui contient l’ensemble des jours de la semaine, du **lundi** au **dimanche**. Ensuite, à l’aide d’une boucle for (for!) affichez un par un ces jours.

Résultat attendu :



Job 06

Créez une fonction “**fizzbuzz**” qui ne prend pas de paramètre. Dans cette fonction, affichez dans la **console web** les nombres de **1** à **151**. Remplacez certains nombres par un mot selon les conditions suivantes :

- Si le nombre est un multiple de 3, affichez “**Fizz**”.
- Si le nombre est un multiple de 5, affichez “**Buzz**”.
- Si le nombre est un multiple de 3 et de 5, affichez “**FizzBuzz**”.

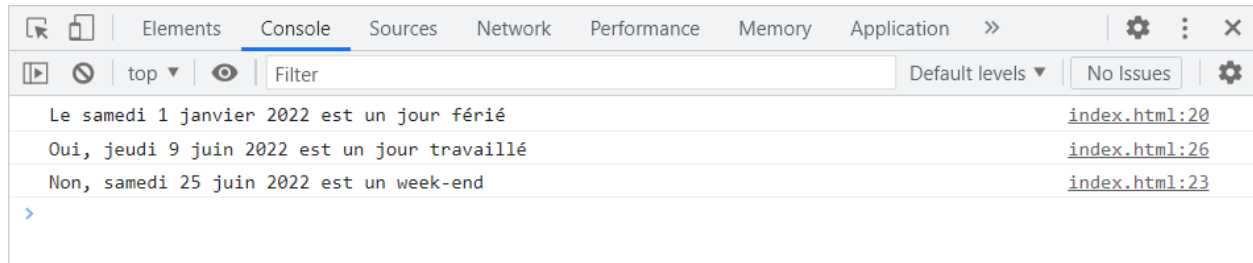
Job 07

Créez une fonction “**jourTravail**” qui prend en paramètre une **date** au **format Date**. Si la date correspond à un jour férié de l’année 2024, la fonction affiche “**Le jour mois année est un jour férié**”. Si elle correspond à un samedi



ou un dimanche, alors le message affiché est **“Non, jour mois année est un week-end”**, sinon afficher **“Oui, jour mois année est un jour travaillé”** ou jour, mois et année correspond aux paramètres passés à la fonction.

Résultat attendu :



Job 08

Créez une fonction **“sommeNombresPremiers”** qui prend en paramètres **deux variables**. Si ces deux variables sont des **nombres premiers**, alors la fonction retourne leur **somme**. Sinon, la fonction retourne **false**.

Job 09

Créez une fonction **“tri”** qui prend en paramètres **un tableau** de nombres nommé **“numbers”** et **une variable “order”** qui contient **“asc”** ou **“desc”**. Développer un algorithme qui doit trier le tableau dans l'ordre ascendant ou décroissant, selon le paramètre passé, puis retourner le **tableau trié**.



Compétences visées

- Utiliser la console
- Créer des fonctions
- Utiliser les boucles
- Algorithmie

Rendu

Le projet est à rendre sur <https://github.com/prenom-nom/runtrackJs>.
Pensez à mettre dans chaque sous-dossier le fichier HTML et JavaScript.

Base de connaissances

- [Documentation officielle JavaScript](#)
- [Tutoriel JavaScript](#)
- [Bien débiter en JavaScript](#)
- [Algorithme de tri](#)