

JS NATIF DOM

- 1 - Accès DOM et événements

Le but de l'exercice est de modifier les comportements de la page en fonction des interactions (événements). Récupérez le dossier « orange ».

Il est interdit de modifier le fichier HTML !

Suggestions :

- Utilisez au maximum « classList »
- Evitez de faire du CSS depuis JS
- Des fois, il n'y a pas de JS à écrire ;)

A faire :

- Modifiez la couleur du titre de la page
- Modifiez le texte du titre de la page
- Modifiez le style des toutes les span de classe « orange »
- Changez la couleur du fond (main) toutes les 2 secondes de manière aléatoire
- Changez la couleur des span de class « orange » toutes les 1 seconde
- Dans le tableau des couleurs, faites en sorte que la couleur soit en fond de la dernière case. Suggestion : récupération et réutilisation de la première case avec innerText
- Supprimez la ligne « purple » qui n'a rien à faire là !
- L'article ne contient qu'un image. Faites en sorte qu'elle soit remplacée toutes les 3 secondes par une nouvelle image, de manière cyclique (voir la liste des images dans le dossier « Images »)
- Modifiez le titre de l'article « image » → « Images »
- Le menu (en haut à droite) doit pouvoir être caché/montré quand on clique dessus. à vous de choisir ce que veulent dire « caché » et « montré » !
- Lâchez vous !! Il faut que ça pique les yeux !

- 2 - Bubbles

Créez 3 fichiers dans un nouveau dossier :

- index.html
- style.css
- main.js

En CSS, faites en sorte que le « body » fasse 100 % de la fenêtre en hauteur.

En JS, créez une « bulle » chaque seconde en position aléatoire (Astuces : `window.innerHeight`, `window.innerWidth`). Une bulle est un « div » avec un `border-radius` à 50 % placée en « absolute ».

Chaque bulle doit avoir une couleur de fond aléatoire.

Au survol (`mouseenter`, `mouseleave`), un bulle doit grossir, avec animation (`scale`, `transition`).

Au clic, la bulle doit disparaître.

En HTML, ajoutez une « div » pour le score. A chaque bulle éclatée, augmentez le score.

- 3 - Création DOM

Créez 3 fichiers dans un nouveau dossier :

- index.html
- style.css
- main.js

Dans le fichier html, la contrainte est d'avoir un body vide à l'exception du script JS. Le but est de tout construire en JS afin de maîtriser la manipulation du DOM.

```
<body>  
  <script src='main.js' defer></script>  
</body>
```

A réaliser :

- Créez une structure de données JSON contenant une « playlist » : groupe, titre, année
Exemples :
 - Deep Purple, Smoke on the water, 1972
 - Metallica, My friend of misery, 1991
 - Nirvana, Something in the way, 1991
- Affichez un titre (h1) au début de la page
- Parcourez la liste des titres et affichez-les dans un tableau html

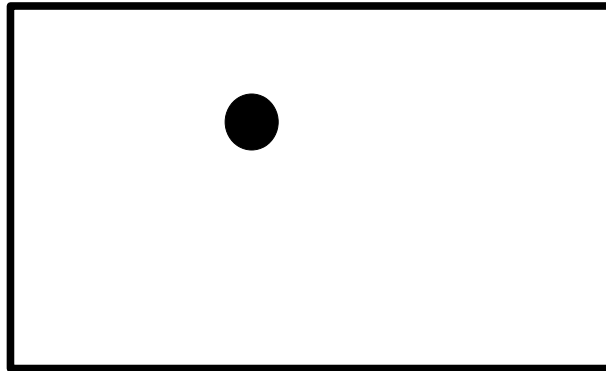
Fonction à utiliser pour récupérer le body : `querySelector`

Fonctions à utiliser pour créer des balises : `createElement`, `createTextNode`, `appendChild`

Une fois fonctionnel, factorisez votre code de façon à ne plus avoir avec utiliser les fonctions du DOM pour créer des éléments.

- 4 - Canvas

Réalisez un billard simpliste (une seule boule) :



- Créez en HTML un canvas de taille 800x600.
- En JS, récupérez ce canvas et créez la variable de dessin (context).
- Dessinez un rond noir dans le canvas.
- Créez un objet JSON représentant la boule : position, vitesse et rayon (x, y, vx, vy, r)
- Modifiez le dessin du rond pour qu'il utilise cette variable
- Créez une boucle infinie avec un setInterval réglé à 10ms
- La fonction associée doit :
 - Effacer le canvas (clearRect)
 - Dessiner le rond
 - Modifier la position en fonction de la vitesse
 - Modifier la vitesse en fonction de la collision avec le bord (calculée à partir des coordonnées). Multipliez la vitesse par un facteur inférieur à 1 pour simuler les frottements.
- Pour aller plus loin, s'il reste du temps :
 - Ajoutez des boules avec des couleurs et gérer les collisions
 - Gérez le clic sur la boule blanche pour lui procurer une vitesse