

TP n°2 – Javascript

1 – Prise en main et mise en forme

Reprise du dernier exercice de la première séance.

Ouvrir le fichier **tp2.html** dans un éditeur de texte.

Visualiser le document à l'aide d'un navigateur. Ouvrir l'onglet « Elements » de la console Javascript.



Mise en forme

Créer un fichier CSS, le lier à l'HTML. Pour les sélecteurs :

body : mettre un fond de couleur gris clair (#eee)

h1 : centrer le texte

Pour les paragraphes de l'article de classe partie2 : mettre un fond de couleur gris #555, une largeur de 80px, une hauteur de 80px, des marges externes de 5px, un arrondi(border-radius) de 5px.

Pour les liens des paragraphes de l'article de classe partie2 : couleur du texte #555(de la même couleur que le fond, pour masquer l'écriture)

Création d'un fichier javascript

Créer un fichier exo3.js. Le lier au fichier HTML.

```
var noeudsElements = document.querySelector(".contenu").children;
var nombreDeNoeuds = noeudsElements.length;
for (var i=0; i < nombreDeNoeuds; i=i+1 ) {
    console.log(noeudsElements[i]);
}
```

2 - Parcours du DOM

NB : Consultez le guide de référence pour l'objet document :

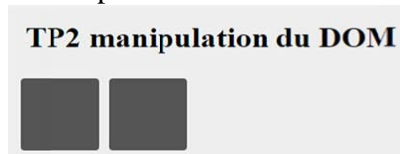
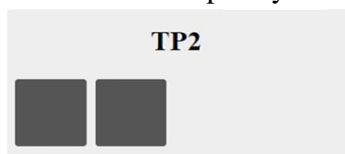
<https://developer.mozilla.org/fr/docs/DOM/document>

Ajoutez dans le script les instructions Javascript pour :

- afficher dans la console l'objet `body`
- afficher dans la console la propriété `innerHTML` de l'élément `h1`
- afficher dans la console l'élément dont l'identifiant est `'c1'`
- afficher la propriété `style` de l'élément dont l'identifiant est `'c1'`
- afficher dans la console toutes les propriétés de l'élément dont l'identifiant est `'c1'`

3 – Modification du DOM : des cartes...

- Utiliser la propriété `innerHTML` pour modifier le contenu de l'élément d'identifiant `'titre'` pour y mettre le texte "TP2 manipulation du DOM".

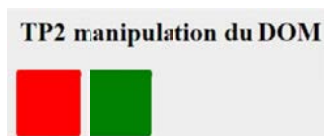


- Ajouter la date à l'emplacement (`<!-- pour la date -->`)



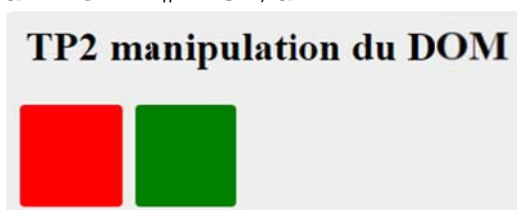
Utiliser la propriété `backgroundColor` de l'objet `style` d'un élément pour mettre l'élément d'identifiant `c1` en rouge ('red') et mettre l'élément d'identifiant `c2` en vert ('green').

- A l'aide de `childNodes`, modifier la couleur d'écriture du lien du paragraphe d'id `c1` (élément enfant de `c1`) en rouge et `c2` en vert

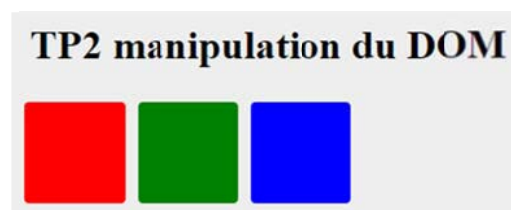


- Utiliser la méthode `createElement` de l'objet `document` pour stocker dans une variable un nouvel élément HTML `p` ; d'id « `c3` », de classe « `carte` », de couleur de fond et de police bleu ; ajouter ce nouvel élément comme fils de l'élément d'identifiant `p2` (utiliser pour cela la méthode `appendChild`).
- Utiliser la méthode `createElement`, pour ajouter un enfant au nouveau paragraphe créé : un lien tel que :

```
<a href= "# ">3</a>
```

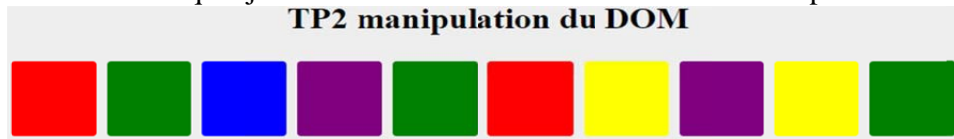


>



- Écrire une fonction qui ajoute une nouvelle carte connaissant dans le numero est passé en argument, comme fils de l'élément d'identifiant `p2`. Toutes les cartes auront les mêmes propriétés (celles de la question ci-dessus) sauf l'identifiant qui dépendra du numéro et idem pour le contenu `innerHTML`.
- Toutes les nouvelles cartes sont bleues ; pour pallier cela on va reprendre la solution de l'exercice 2.b du TP1 et utiliser cette fonction pour que la couleur de la carte soit choisie au hasard.

- Écrire une boucle qui ajoute les cartes 4 à 10 en utilisant la fonction précédente.



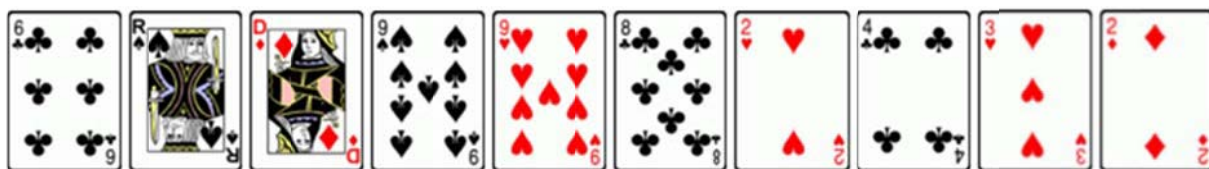
- Supprimer une carte : écrire une fonction qui supprime une carte dont on connaît le numéro.
- Déplacer une carte : écrire une fonction déplacer qui déplace la première carte en dernier. Utiliser pour cela l'attribut `firstElementChild` et la méthode `appendChild`.
- Défilement des cartes : utiliser la fonction javascript `setInterval` pour faire défiler automatiquement toutes les cartes au rythme d'une carte toutes les 2 secondes. Utiliser pour cela la fonction précédente déplacer.
- Retourner une carte : écrire une fonction permettant de « retourner » une carte (le fond devient blanc sur lequel est écrit le numéro).



- Zoomer sur une carte : écrire une fonction permettant de zoomer une carte (la carte doit voir sa taille x 2).

4 – Modification du DOM : des cartes encore...

- Récupérer sur Moodle les images correspondant aux 52 cartes d'un jeu standard.
- Récupérer sur Moodle le fichier `TP2CarteAJouer.html`
- Écrire une fonction javascript permettant d'afficher dans le navigateur une carte au hasard parmi les 52. Pour cela, la fonction doit créer un nouvel élément HTML de type `img` avec comme attribut `src` le nom du fichier image et comme attribut `id` le nom de la carte. Ce nouvel élément sera ensuite ajouté comme fils de l'élément d'identifiant `cartes`.
- Faire afficher 10 cartes dans le navigateur



Modifier l'affichage précédent pour faire en sorte que si deux cartes successives tirées au hasard ont la même couleur, alors on ne garde que la deuxième.