DUT-DST (Niveau L2)
TD/TP 2 : JS et jQuery

Exercice 2.1

Ecrire le code de l'examen QCM vu en cours.

Exercice 2.2

Vous avez une page avec un formulaire et un champ. Écrivez un script qui va valider le formulaire. Si le champ est vide, un message d'erreur est affiché à côté du champ et le formulaire n'est pas envoyé. Si le champ contient des données, le formulaire peut être envoyé.

Exercice 2.3

Vous avez une page avec un formulaire et un champ permettant de saisir une adresse mail. Écrivez un script qui va valider le formulaire. Si l'adresse mail saisie est incorrecte (pas de la forme xxx@yyy.zzz), un message d'erreur est affiché à côté du champ et le formulaire n'est pas envoyé. Si le champ contient une adresse correcte, le formulaire peut être envoyé.

Exercice 2.4

Soit le code ci-contre.

```
<html>
<head>
<title>Essai DHTML</title>
<script type="text/javascript">
<!--

    function ajout()
        {



        }

//-->
</script>

</head>
<body onClick="ajout()">

<ul id="listecommissions">
    <li>1kg de farine</li>
    <li>un pack de lait</li>
</ul>

</body>
</html>
```

1. Ecrire le code de la fonction ajout()

    a. Sélectionner la liste d'identifiant "listecommissions".
    b. Créer un nouvel item de liste et lui affecter un texte.
    c. Ajouter ce nouvel élément de liste à la liste

2. Modifier le code en ajoutant un champ de saisie de texte et un bouton de validation. Le texte ajouté à la liste devra être saisi dans le champ de saisie.
3. Ajouter un bouton permettant, par un appel de fonction, de supprimer le dernier élément de la liste.
4. Que se passe-t-il quand on essaie de supprimer un élément de la liste lorsque celle-ci est vide? Corriger le code pour éviter cette erreur.

Exercice 2.5

Le but de cet exercice est de réaliser une petite horloge tournante (certes pas en temps réel !).

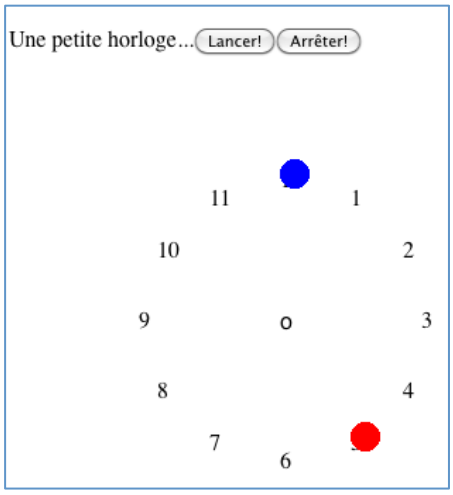
1. Récupérer le fichier **exo2.3.html**
2. Ajouter un bouton à la fin du paragraphe. Ce bouton lance l'horloge en appelant la fonction `tourne()`
3. Créer deux variables globales, `compteur_minutes` et `compteur_heures`, initialisées à 0.
4. Identifier les éléments correspondant à chaque image.
5. Selon la valeur de `compteur_minutes`, changer ses propriétés de position:
    - Lorsque le compteur vaut 0, attribuer à l'image *aiguilleminute* la position de l'élément chiffre1;
    - Lorsque le compteur vaut 1, attribuer à l'image *aiguilleminute* la position de l'élément chiffre2 ;
    - Lorsque le compteur vaut 2, attribuer à l'image *aiguilleminute* la position de l'élément chiffre3...
    - ... ainsi de suite jusqu'à la valeur de compteur 11, auquel cas on attribue à l'image *aiguilleminute* la position de l'élément chiffre12.
6. Incrémenter le `compteur_minutes`, et lui affecter le reste de sa division euclidienne par 12.
7. Temporiser un appel à la fonction toutes les secondes (par un appel à la méthode `window.setTimeout(nom_fonction, temps_en_millisecondes)`.

Il reste maintenant à faire tourner l'aiguille "des heures".

1. Juste après avoir changé la valeur du compteur, tester si l'aiguille des minutes est en 12 (cela se fait par un test sur la valeur du compteur). Si cela est le cas, alors reprendre la même procédure que précédemment pour le déplacement de l'aiguille des minutes, mais cette fois-ci appliquée à celle des heures, en n'oublient pas d'incrémenter le compteur des heures.

Ajouter un bouton permettant d'arrêter l'horloge (on fera appel à un booléen).

La figure ci-après est une capture du résultat à obtenir.



Exercice 2.6

Reprendre les exercices précédents avec du code jQuery à la place du simple JS.