

# EXERCICE - ÉNIGMES

## 1 Travail à effectuer

Vous devez mettre en place un jeu de résolution d'énigmes destiné à des jeunes du primaire. 5 énigmes seront présentées à l'enfant (sur une possibilité de 10).

Vous ne devez en aucun cas modifier les fichiers **html** et **css**. Vous avez seulement le droit de modifier le fichier **javascript** mis à votre disposition.

### 1.1 Analyse du code fourni

Prenez quelques instants pour comprendre le code fourni, notamment le fichier **html** et le fichier **js** qui contient déjà un peu de contenu.

### 1.2 **initialiserEnigmes()**

Cette fonction est appelée au chargement (elle est appelée directement dans le code de la page à la ligne 12 du fichier de départ).

Elle doit tirer au hasard 5 énigmes parmi une banque d'énigmes déjà établies (**arrEnigmes** déclaré dans le fichier **enigmes.js**). Il va de soi que les réponses à ces énigmes doivent également être mémorisées dans le tableau **arrReponsesEnigmesPigees** dans **app.js**.



#### Indice

Faites 5 itérations dans une boucle **for** (Pensez à utiliser une constante pour le nombre d'énigmes!)

Dans chaque itération de la boucle, sélectionner aléatoirement une énigme du tableau **arrEnigmes**. Une fonction est fournie pour la génération d'un nombre aléatoire.

Les deux (2) tableaux existants devront être utilisés pour mémoriser les énigmes pigées et bonnes réponses énigmes pigées.

```
arrEnigmesPigees.push(arrEnigmes[indexAleatoire]);
```

```
arrReponsesEnigmesPigees.push(arrReponsesEnigmes[indexAleatoire]);
```



#### Attention!

N'oubliez pas de retirer à chaque fois du tableau **arrEnigmes** l'énigme pigée. Utilisez la fonction [\*\*splice\*\*](#) pour cela.

Une fois le tirage aléatoire terminé, la première énigme doit s'afficher dans le paragraphe **#enigme-courante**.

### 1.3 validerReponseEnigme ()

Cette méthode, appelée chaque fois qu'on tente une réponse, comparera ce qui a été entré dans le champ de saisie avec la réponse attendue pour l'énigme actuelle dont l'index est mémorisé dans `indexEnigmeCourante`.

Lorsque la réponse n'est pas celle attendue, un message d'erreur (de votre choix) est affiché dans l'espace `#commentaire-enigme` et la classe CSS `.erreur` est ajoutée sur le commentaire.



Lorsque la réponse est celle attendue, une suite d'actions doit se produire et varie dépendamment de si l'affichage d'une autre énigme est prévu ou si c'était la dernière:

- Si encore des énigmes à répondre
  - o Affichage de l'énigme suivante
  - o Réinitialiser/vider le champ de saisie
- Si toutes les énigmes sont répondues
  - o Disparition de la `#zone-enigme`
  - o Apparition de la `#message-succes`

#### Exercice de JavaScript



Bravo!



#### Disparition et apparition d'éléments dans la page

Notez que l'apparition et la disparition d'éléments de la page se font avec une classe CSS se nommant `cache` qui est appliquée ou retirée selon les besoins.

## 2 Modalités de remise

Remettez votre projet sur LÉA, dans la section travaux, à l'intérieur d'une archive *Zip au moment spécifié par ta ou ton professeur*.