Durée du CC : 1h30.

La Fiche-Triche JavaScript est le seul document autorisé.
Les deux parties sont indépendantes ; on pourra les traiter dans l'ordre de son choix. Pour le rendu, deux fichiers sont attendus : `partie1.js` pour la première partie, et `partie2.js` pour la seconde.

1 Apiculture

Le code pour cette section est à rendre dans un fichier nommé `partie1.js`.

Question 1 Écrire une classe `Ruche` possédant les deux attributs suivants :
- `abeilles`, qui représente le nombre d'abeilles dans la ruche, et
- `miel`, qui indiquera la quantité de miel, en grammes, que contient celle-ci.

On considérera qu'un ruche ne contient aucun (autrement dit, 0 grammes) miel à sa création. Le constructeur de cette classe aura donc un unique argument : le nombre d'abeilles.

Question 2 Implémenter la méthode `production()` de `Ruche`. Cette méthode n'attend aucun argument, et augmente la quantité de miel de 3 grammes pour chaque abeille présente dans la ruche.

Question 3 Écrire une classe `Apicole`, qui représentera une entreprise apicole (c'est-à-dire, une entreprise productrice de miel). Cette classe possédera deux attributs : un `nom` et un tableau `ruches` de ruches.
À sa création, une instance de la classe `Apicole` ne possède aucune ruche : le constructeur attendra donc uniquement une chaîne de caractères.
Doter cette classe d'une méthode `ajout_ruche()` qui attend un entier `n`, et ajoute à `ruches` une ruche nouvellement créée contenant `n` abeilles.

Question 4 Implémenter la méthode `production()` de la classe `Apicole` : un appel à cette méthode programmera, pour 5 secondes plus tard, l'appel de la méthode `production()` de chaque ruche de `ruches`.
On utilisera une boucle *for* pour répondre à cette question.

Les abeilles ayant besoin de miel pour vivre, les apiculteurs et apicultrices ne récoltent jamais l'intégralité du miel présent dans une ruche. Pour les besoins de la modélisation, on considérera qu'à chaque récolte, on laisse, si possible, 100g de miel dans chaque ruche. En particulier, si une ruche possède moins de 100g de miel, alors aucun miel n'y sera récolté.

Question 5 Implémenter la méthode `recolte()` de la classe `Apicole`. Cette méthode devra renvoyer la quantité, en grammes, du miel récolté, et devra mettre à jour la quantité de miel de chaque ruche.
Pour répondre à cette question, on n'utilisera ni boucle *for*, ni boucle *while*. On pourra, si besoin, ajouter une méthode à la classe `Ruche`.

2 King's Landing

Téléchargez le fichier `King's_Landing.html` sur Eprel.
Le code de cette section sera à rendre dans le fichier `partie2.js`. Le fichier `King's_Landing.html` n'est pas à rendre.

Question 6 Ajouter une balise au fichier `King's_Landing.html` permettant d'exécuter le code contenu dans le fichier `partie2.js`.

Pour les questions suivantes, on ne touchera plus au fichier html : les balises devront être créées ou modifiées dynamiquement depuis le script.

Question 7 Créer une balise `<p>` contenant le texte `Cette page concerne la ville de King's Landing.`, et la placer directement après le `<h1>` de la page.

Question 8 Faire en sorte que quand la souris passe au-dessus d'un lien `<a>` de la classe `mw-redirect`, ce lien soit remplacé par une balise `<b>` contenant le texte `lien mort`.
On utilisera une lambda-expression pour répondre à cette question.